COF: Exploiting Concurrency for Low Power Opportunistic Forwarding

Daibo Liu[1], Mengshu Hou[1], Zhichao Cao[2], Yuan He[2], Xiaoyu Ji[3], Xiaolong Zheng[2]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[2]School of Software, TNLIST, Tsinghua University
[3]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
dbliu.sky@gmail.com, mshou@uestc.edu.cn, {caozc, he, xiaolong}@greenorbs.com, xji@ust.hk

***Abstract*—Due to the constraint of energy resource, the radio of sensor nodes usually works in a duty-cycled mode. Since the sleep schedules of nodes are unsynchronized, a sender has to send preambles to coordinate with its receiver(s). In such contexts, opportunistic forwarding, which takes the earliest forwarding opportunity instead of a deterministic forwarder, shows great advantage in utilizing channel resource. The multiple forwarding choices with temporal and spatial diversity increase the chance of collision tolerance in concurrent transmissions, potentially enhancing end-to-end network performance. However, the current channel contention mechanism based on collision avoidance is too conservative to exploit concurrency. To address this problem, we propose *COF*, a practical protocol to exploit the potential *C*oncurrency for low power *O*pportunistic *F*orwarding. *COF* determines whether a node should concurrently transmit or not, by incorporating: (1) a distributed and light-weight link quality measurement scheme for concurrent transmission and (2) a synthetic method to estimate the benefit of potential concurrency opportunity. *COF* can be easily integrated into the conventional unsynchronized sender-initiated protocols. We evaluate *COF* on a 40-node testbed. The results show that *COF* can reduce the end-to-end delay by up to 41% and energy consumption by 18.9%, compared with the state-of-the-art opportunistic forwarding protocol.**

I. Introduction

Energy efficiency is a fundamental issue in the design of forwarding protocols for low power wireless sensor networks (WSNs). A primary low power mechanism is duty cycling [1] [2]. In the duty-cycled mode, a node periodically switches its radio state between on (awake) and off (sleep). A widely adopted protocol of duty-cycled media access control (MAC) is low power listening (LPL) [1]. Taking X-MAC [20] as an example, as shown in Fig. 1, a node periodically turns on its radio to detect the ongoing traffic by checking the received signal strength. If the channel is clear, it will turn off the radio. Because the sleep schedules of different nodes are unsynchronized, the sender has to wait until the receiver turns on the radio. During the waiting period, the sender has to continuously transmit the same data packet (called *data preamble* or *preamble* in the rest of this paper) until the receiver's acknowledgement is received or a pre-configured timer on the sender expires.

Blind waiting in the above-mentioned duty cycles is generally energy-inefficient and limits network throughput. To shorten the waiting time, a practical approach is opportunistic routing [21], which takes the earliest forwarding opportunity

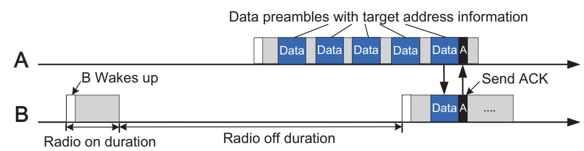


Fig. 1. An example of low power listening with X-MAC.

instead of a deterministic forwarder. The forwarding opportunities include all the neighbors that are awake and offer sufficient routing progress. The state-of-the-art opportunistic routing protocols, such as ORW [22] and DOF [23], have shown promising improvement in terms of end-to-end delay and network throughput.

To what extent can we seize the forwarding opportunities is the key of opportunistic routing. Despite that the waiting period is shortened, the practical performance of opportunistic routing is still far from being satisfactory. Our key observation is that the current collision avoidance based MAC is too conservative for duty-cycled opportunistic routing. Specifically, the multiple forwarding choices with temporal and spatial diversity increase the chance to tolerate collision in concurrent transmissions. The interference from a specific neighbor is likely to have different influence on different candidate forwarders. If any one of the potential forwarders can successfully decode the sender's packet under interference, opportunistic forwarding should promote rather than arbitrarily suppress such a transmission opportunity. We use the term *opportunistic exposed terminal* to denote such a phenomenon. Moreover, the chance of overhearing data preambles is fairly high, according to our observation. The overheard data preambles may provide sufficient information to estimate the potential benefit of concurrent transmissions. Hence, it is feasible and profitable to exploit concurrency for low power opportunistic forwarding. Many existing works, such as capture effect [4][5][6][7][8][9], conflict graph [14][15][16][17][24], and parameter adjustment [10][11][12][13], have been proposed to achieve concurrent transmission in the scenario of deterministic forwarding. Concurrency for low power opportunistic forwarding, however, has not been well studied so far.

In this paper, we propose *COF*, a practical protocol to exploit the potential *C*oncurrency for low power *O*pportunistic *F*orwarding. To achieve this goal, there are several challenges.

1092-1648/15 $31.00 © 2015 IEEE
DOI 10.1109/ICNP.2015.13

IEEE computer society

First, link quality is essential input for making a forwarding decision, but it is resource-exhausting to measure a full graph of all links in real time and a distributed manner. Second, the benefit of a concurrent transmission is difficult to quantify in an unsynchronized duty-cycled context. Third, in a large-scale and ad-hoc WSN, the feasible concurrency patterns (which pair of links can concurrently transmit) are unpredictable and changeable. An adaptive scheme for exploiting concurrency is needed. Last but not least, the scheme should be light-weight and easy to be integrated into the existing duty-cycled mechanisms.

To address the above challenges, *COF* uses a light-weight and distributed approach to obtain conditional link quality of the links, denoted by conditional packet delivery ratio (*cpdr*), when two senders are concurrently transmitting. Then *COF* estimated the benefit of the concurrent opportunity. Based on the estimated benefit, *COF* recommends to the MAC layer one of the following three options: making a concurrent transmission without carrier sense, pausing the ongoing transmission, or trying to transmit using the original carrier sense. According to the recommendation, MAC takes full advantage of the concurrent opportunity. The contributions of this work are summarized as follows:

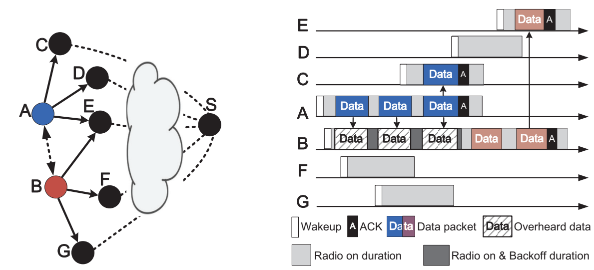
- Based on the observation of the *opportunistic exposed terminal* phenomenon, we propose a light-weight protocol, *COF*, to fully exploit concurrency for low power opportunistic forwarding.
- We propose a distributed scheme to measure links' conditional packet delivery ratio in real time, when multiple neighboring nodes are concurrently transmitting, and to estimate the potential benefit of the concurrent opportunity.
- We implement *COF*, integrate it with ORW [22] (the state-of-the-art opportunistic forwarding protocol) and LPL [1], and evaluate it on a 40-node testbed. Experimental results show that *COF* can significantly improve network throughput and reduce energy consumption.

The rest of the paper is organized as follows. Section II presents the empirical studies and shows our key observations. Section III introduces the detailed design of *COF*. We implement *COF* and evaluate its performance in Sections IV. Section V discusses the related work. We conclude this paper in Section VI.

II. Empirical Study

In this section, we conduct empirical studies to show that:

- Contention avoidance based MAC is too conservative for low power opportunistic forwarding. Many forwarding opportunities are overly suppressed.
- It is feasible and profitable to exploit concurrency for low power opportunistic forwarding. There is great room for improvement.
- The chance for a suppressed sender to successfully overhear a neighboring sender's preambles is fairly high. Those overheard preambles provide sufficient information to estimate the benefit of potential concurrent opportunity.



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

Fig. 2. Example of existing 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 within the carrier sense range of each other, and (b) is a general process of low power opportunistic forwarding.

A. Low Power Opportunistic Forwarding

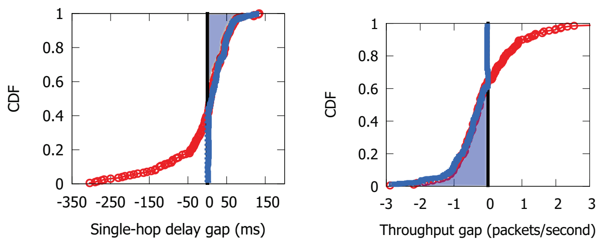
In LPL, as Figure 1 shows, each node periodically turns its radio on to check the channel condition. A sender will not stop sending repeated preambles until it receives an acknowledge (ACK) from the receiver or the pre-configured timer on the sender expires.

In low power opportunistic forwarding, a preamble may be overheard and acknowledged by an earlier wake-up neighbor (called forwarder), as long as the forwarder provides sufficient routing progress. The period of repeating preambles is therefore shortened. As Fig. 2(a) shows, *A* maintains a set of candidate forwarders $\mathbf{F}_A$ ($\mathbf{F}_A$={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 intended destination *S*, as shown in Fig. 2(b), *A* sends preambles until *C* wakes up and acknowledges the reception of a preamble.

Now we consider a typical scenario. When *A* is transmitting, *B* is also holding a packet to transmit. In current low power opportunistic forwarding mechanisms, e.g., ORW and DOF, *A* or *B* can only exclusively access the channel. Thus *B* keeps its radio on and takes backoff, so as to wait the channel to be free. In fact, during the waiting period, *B* misses the early opportunities to send a 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 limited network throughput.

B. Opportunistic Exposed Terminal

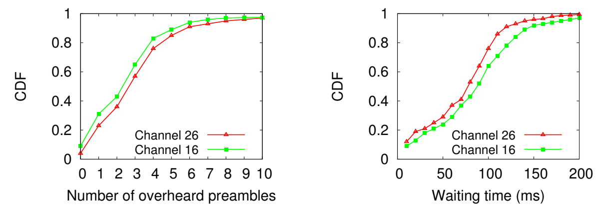
The current designs of WSNs generally adopt contention avoidance based MAC, e.g., LPL based on CSMA/CA. The existing low power opportunistic forwarding protocols are directly built upon such MAC, ignoring a fundamental characteristic of opportunistic forwarding. In opportunistic forwarding, each node maintains a set of candidate forwarders with temporal and spatial diversity, which means the impact of the interferer on different candidate forwarders (e.g., the impart of *A* on *E*, *F*, and *G* in Fig. 2(a)) is likely to be different. When two senders within the carrier sense range of each other concurrently transmit, even though the received packets are corrupted due to collision at most of



(a) Single-hop delay gap

(b) Throughput gap

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



(a) Number of overheard preambles

(b) Waiting time

Fig. 4. The CDF of the number of overheard preambles (a), and the CDF of the length of waiting time (b).

the candidate forwarders, there may exist some forwarders that can successfully decode the packets. We use the term *opportunistic exposed terminal* to denote such a phenomenon. By tolerating avoidance at opportunistic exposed terminals, one can realize concurrent transmissions with opportunistic forwarding and significantly enhance network performance. Using a contention avoidance based MAC, however, existing low power opportunistic forwarding protocols overly suppress the opportunities of concurrent transmissions.

C. The Two Sides of Concurrent Transmissions

In this subsection, we present experiments to show the great space for concurrent transmissions and the potential performance improvement in low power opportunistic forwarding. On the other hand, we show that unrestrained concurrent transmissions will seriously degrade network performance.

The experiments are conducted on an indoor testbed with 22 TelosB nodes. Each node runs the open source version of ORW and opportunistically sends data packets to one of its forwarders. 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 nodes as senders, which are within the carrier sense range of each other and continuously generate data packets. The other nodes generate data packets at an inter-packet interval (IPI) of 5 minutes. We repeat the experiments with CSMA mechanism enabled and disabled at the two senders. The CSMA of all the other nodes keeps enabled. The experiments are conducted for more than 100 times. We record the single hop delivery time of each packet, the total number of received packets at the sink, and the duration of each experiment. We compute the average single hop transmission delay of CSMA-enabled (T_{csma}) and CSMA-disabled (T_{nocsma}), and the average network throughput (received packets per second) of CSMA-enabled (TP_{csma}) and CSMA-disabled (TP_{nocsma}). Then we compute the single hop delay gap (T_{gap}) and throughput gap (TP_{gap}) according to

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

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

and plot the cumulative distribution function (CDF) of those 100+ experiments in Fig. 3(a) and Fig. 3(b), respectively.

As shown in Fig. 3, when CSMA is disabled in low power opportunistic forwarding, a great portion of transmissions can achieve shorter single hop delay (the plus zone of Fig. 3(a)) and finally achieve higher throughput (the minus zone of Fig. 3(b)) than the CSMA-enabled case. This observation indicates that there are many data transmissions (about 63.7% in our experiments) suffering from the *opportunistic exposed terminal* problem. Exploiting concurrency of transmissions can alleviate this problem.

On the other hand, low power opportunistic forwarding with CSMA disabled may induce serious data collisions and longer transmission delay (minus zone of Fig. 3(a)), if concurrent transmission is out of control. In this case, the network throughput will be sharply degraded (plus zone of Fig. 3(b)).

This group of experiments imply that: we should allow senders to concurrently transmit in the presence of *opportunistic exposed terminal*, and suppress concurrent transmissions when it is likely to hurt network performance. If we can achieve this goal, we can get ideal single hop delay gap and throughput gap as shown by the bold blue lines plotted in Fig. 3. Visually, the dash areas of both Fig. 3(a)) and 3(b) are the potential improvement space for low power opportunistic forwarding.

To achieve the above goal, the sender should first know who is the ongoing transmitter. Next, we give an experiment to show the chance of overhearing in low power opportunistic forwarding to obtain such information.

D. Chance of Overhearing

Overhearing is a fundamental characteristic of wireless communication. When a sender transmits a packet, each of its neighbors has a chance to overhear the packet. In low power opportunistic forwarding, to transmit a packet, the sender repeatedly sends preamble until it is acknowledged. If a neighbor of the sender overhears the packet and it can provide sufficient routing progress towards the destination, it acknowledges the sender; Otherwise it does nothing.

It is worth noticing that the consecutively transmitted preambles impel an awake neighbor to successfully decode the preamble. A preamble carries sufficient information, e.g., who is the ongoing transmitter, to estimate the potential benefit of concurrent transmission. To show the chance of overhearing, we conduct experiments in a scenario illustrated by Fig. 2(a), with little external interference (channel 26) and unpredictable interference (channel 16, overlapped with WiFi),

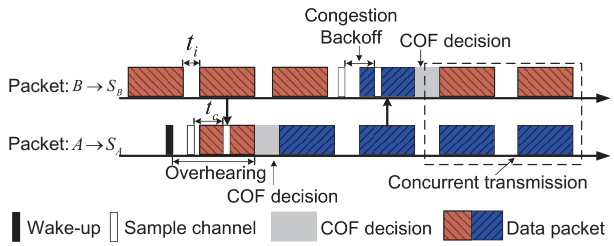


Fig. 5. Overview of *COF* for exploiting concurrency in low power opportunistic forwarding.

respectively. Both A and B periodically generate data packets by setting IPI to 1 second. When receiving a preamble, each forwarder only replies an ACK and does not further forward the packet. The CSMA mechanism is enabled for all nodes in this experiment. We just pay attention to the following case: when a sender intends to transmit, the other sender happens to be transmitting. During the waiting period, both A and B will record the number of preambles it overhears and the length of waiting time, respectively. The experiment lasts for two hours.

We plot the CDFs of the number of overheard preambles and the waiting time in Fig. 4(a) and 4(b), respectively. As Fig. 4(b) shows, a sender will possibly wait for a long period of time (with more than 60% probability to wait for more than 60 microseconds) if another neighbor is transmitting. During the waiting time, the sender will possibly overhear multiple preambles, i.e., with more than 90% probability to overhear at least one in Fig. 4(a). Under the external interference (WiFi, channel 16), the probability of successfully decoding a preamble will degrade. Hence, it needs more time to transmit one packet, resulting in other senders' prolonged waiting time. Overall, the experimental results demonstrate the high chance of overhearing and motivate us to utilize the overheard preambles to estimate the benefit of potential concurrent opportunity.

The empirical studies shed light on the potential of exploiting concurrency for low power opportunistic forwarding. Motivated by the above-mentioned results, we propose the design of *COF*.

III. Design of COF

In this section, we present the detailed design of *COF* that exploits concurrent opportunity in low power opportunistic forwarding. Basically, according to the information of data overhearing, *COF* develops light-weight method to make each node to accumulatively measure the success probability of concurrent transmission. First, we describe the meaning of all symbols in Table I. Then we present the overview of *COF*, conditional probability model, detailed data structure, and communication scheme.

A. Overview of COF

We now take Fig. 5 as an example to show how *COF* works. A and B are *opportunistic exposed terminal* of each other. When A intends to transmit a data packet to S_A, it first conducts two channel assessments, called CCA (Clear Channel Assessment), with a fixed interval t_c to judge whether the channel is busy. t_c is larger than the inter preamble interval (t_i) [25] to guarantee that a silent sender will not preempt the channel during the inter preamble interval of another ongoing sender. Since channel is busy (B is transmitting), A defers its data transmission.

TABLE I
Symbols description.

Symbol	Description
$P_{i,j}^k$	the probability that forwarder j will receive the packet sent by node i when neighbor k 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;
$epdr(A\|\o)$	the expected packet delivery ratio of A's transmission when there is no other ongoing transmitter;
$EGain(A\|B)$	the expected benefit of concurrent transmission than the individual transmission of B;
$T(A\|B)$	the overall gain of concurrent transmission;
BTable	a table maintaining for the expected benefit of concurrent transmission;
CPDR	a table maintaining links' *cpdrs*;
$\mathbf{S}_{pkt}$	the set of all transmitted packet;
T_k	the number of transmissions of packet k;
A_k	indicating whether the packet k is acknowledged;
π_k	a correction parameter for accurately computing *cpdr*;
δ_m	indicating whether a forwarder received the m^{th} transmission;
$\mathbf{AT}_k$	the set of all transmissions corresponding to packet k;
θ, α	parameters of moving average for updating link's *cpdr*;

Nevertheless, A keeps overhearing the data preamble transmitted by B. According to *COF* protocol, A extracts attached information in the overheard data preamble, and then it knows B is transmitting. Then A queries the expected benefit of concurrent transmission with B and confirms that the concurrent transmission can gain more benefit than the individual transmission of B. Hence, A disables carrier sensing temporarily and begins to transmit data preambles repeatedly. After overhearing the data preamble of A, B will also disable its carrier sensing and concurrently transmit.

The criteria of *COF* decision is based on the expected benefit of concurrent transmission. We define the expected benefit as the expected gain of temporal throughput in Section III-C. The expected benefit of concurrent transmission is computed by conditional packet delivery ratio (*cpdr*), which denotes the conditional link quality under the influence of another ongoing transmission.

B. Conditional Packet Delivery Ratio

To compute links' *cpdr*, several statuses of local packet transmission are indispensable. For each packet that is concurrently transmitted with another ongoing transmission, the sender needs to know which forwarder has successfully received at least one data preamble. Meanwhile, the sender needs to know what causes the failed transmission, data collision at forwarders' receiving end or ACK collision at its own transmitting end. We first discuss the data structure and communication scheme to collect these statuses. Then we discuss the definition and computation of *cpdr*.

1) Exploiting bitmap to record information: In multi-hop WSNs, each node acts as both a sender to transmit data packets in its sending queue and a forwarder to acknowledge the data packets transmitted by its potential children nodes. As a sender, it records each transmission with three-tuple statuses, which consist of a unique DSN[1] (Data Sequence Number) of the concurrent transmission, the address of the neighbor node which is concurrently transmitting with itself, and whether the transmission is acknowledged. As a forwarder, it records the number of totally received preambles with the identical DSN from each children node.

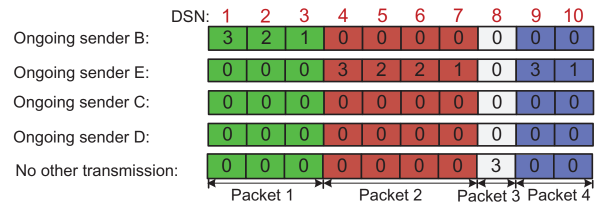
We first discuss the case where a node is a sender. As a sender, each node maintains a set of bitmaps for its neighbor nodes, by assigning a unique bitmap to each node. In addition, it also maintains a bitmap for the case that there is no other transmission. These bitmaps are allocated with the same size (10 bytes) and equally divided into many units. A unit of the bitmap corresponds to the statuses of a unique packet DSN. In sensor network, a sender assigns different data transmissions with different DSNs in ascending (or descending) order. The repeated preambles of the same packet transmission shares an identical DSN. Whereas a retransmission or new packet transmission will add the DSN by (DSN+1) mod 256. After each transmission, the sender will update these bitmaps simultaneously. If the sender transmits concurrently with one neighbor node, according to its own packet DSN, it first updates the concurrent transmission state of the unit in the bitmap of neighbor's ID. The unit is set to a non-zero state. Then, according to the DSN, it sets the states of the related units in the rest of bitmaps to zero. If there is no concurrent transmitter, the sender only sets the related unit in the *no other transmission* bitmap to non-zero state, and sets the related units in all the other bitmaps to zero.

Note that only one non-zero state cannot distinguish acknowledged transmissions from unacknowledged transmissions. Hence, we use three non-zero values to denote different transmission states. Necessarily, each unit of the bitmaps is two bits. State 1 denotes that an ACK is received after the transmission, state 2 denotes an unacknowledged transmission, and state 3 denotes the beginning transmission of another data packet. The states of each unit of all bitmaps are initialized to 0, which indicates the sender does not transmit concurrently with these neighbor nodes.

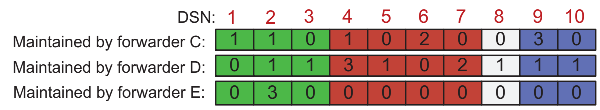
Taking node A in Fig. 2(a) as an example, it maintains five bitmaps for neighbor B, C, D, E, and *no other transmission* as shown in Fig. 6(a). A successfully transmits 4 packets (the same color square boxes indicate the same packet in Fig. 6) over total 10 transmissions, separately assigned with DSN from 1 to 10. For each transmission, A updates a related unit of the bitmap corresponding to the concurrent neighbor. The rest of bitmaps are also updated by A.

The bitmap of each neighbor is organized circularly and orderly according to DSN. By recording each transmission state in related bitmaps, the sender will get a serial of assembled non-zero transmission states, e.g., 3213221331 in Fig. 6(a).

[1]Data preambles of the same packet transmission share the same DSN.



(a) Bitmaps recorded by sender A



(b) Bitmaps recorded by A's candidate forwarders

Fig. 6. (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.

Considering the packet retransmission, to classify the subset of DSN sequences corresponding to the same data packet, the following four principles should be obeyed. First, if a sub serial of states satisfy *subserial* = **3**3, then we can conclude the previous packet is acknowledged without retransmission; second, if *subserial* = **31**3, the previous packet is acknowledged with only one retransmission; third, if *subserial* = $\mathbf{3\overline{2}1}3$, where an overline denotes repeated number (repeat count ranges from 0 to the system maximum retransmission count threshold), then we can conclude that the previous packet is acknowledged after multiple retransmissions; and fourth, if *subserial* = $\mathbf{3\overline{2}}3$, we can know the previous packet is dropped because the maximum retransmission limitation is reached.

We then discuss the case where a node is a forwarder. As a forwarder, each node also maintains a set of bitmaps for its candidate children nodes. A bitmap unit is used to record the number of received copies of data preamble corresponding to one transmission of a neighbor. These copies share an identical DSN. Due to retransmission, the transmission of the same packet maybe use several continuous DSNs. Just for one transmission, if the forwarder receives k ($k > 1$) copies of the same data packet, it can conclude that at least the previous $k-1$ ACKs were lost. Furthermore, if it receives the retransmission of the same data packet with an increased (or decreased) DSN, it can further conclude all previous k ACKs were lost. Actually, by feeding back these bitmaps to candidate children nodes, combining with the maintained bitmaps mentioned above, each candidate children node could exactly infer the number of lost ACKs. For example, in Fig. 6(b), E consecutively received 3 packets sharing DSN 2 sent by A, however, its acknowledgements all collide at A's transceiver because A retransmitted the packet with DSN 3 (A can infer it after collecting E's bitmap). We also use two bits to record the number of received copies of each packet corresponding to the same DSN. In low power WSNs, the duration for keeping in active state after each wake-up is very short, and receiving a duplicate packet will not trigger any extended wake-up time. Hence, two bits are enough to record the number of received copies for the vast majority of cases.

2) Computation of cpdr: We define the *cpdr* as the probability, $P^k_{i,j}$, that forwarder *j* can receive the data packet transmitted by sender *i* when neighbor *k* is concurrently transmitting. As a sender, by collecting the relative receiving information from its candidate forwarders, it will know which candidate forwarder receives or loses its transmission under the influence of a neighbor node. It also can determine the cause of a failed transmission as the packet collision at the receiving end or ACK collision at its own transmitting end. Hence, a sender can compute the links' bidirectional *cpdr*s between itself and the candidate forwarders. First, sender *i* can compute the up-to-date *cpdr* of the link from it to a specific forwarder *j* under the influence of an ongoing neighbor N according to

$$P^N_{i,j} = \frac{\sum_{k \in \mathbf{S}_{pkt}} (\sum_{m \in \mathbf{AT}_k} \delta_m)}{\sum_{k \in \mathbf{S}_{pkt}} (T_k - (A_k - \pi_k))}. \quad (1)$$

where j is one of i's candidate forwarders. k denotes an individual packet in the packet set ($\mathbf{S}_{pkt}$) recorded in both j's forwarder bitmap and i's sender bitmap. Taking Fig. 6 as an example, there are four packets in the $\mathbf{S}_{pkt}$ of A. $\mathbf{AT}_k$ denotes the set of all transmissions corresponding to packet k, m denotes the m^{th} transmission of packet k, and δ_m denotes whether the forwarder j received the m^{th} transmission. *COF* should refer to forwarder bitmaps to determine the value of δ_m.

$$\delta_m = \begin{cases} 0 & \textit{if } j \textit{ didn't receive the } m^{th} \textit{ transmission} \\ 1 & \textit{if } j \textit{ received the } m^{th} \textit{ transmission} \end{cases}$$

For example, in Fig. 6, $\mathbf{AT}_1$ consists of three transmissions corresponding to different DSNs (1, 2 and 3, respectively). Overall, $\sum_{k \in \mathbf{S}_{pkt}} (\sum_{m \in \mathbf{AT}_k} \delta_m)$ denotes the number of transmissions which are successfully received by j.

On the other hand, $\sum_{k \in \mathbf{S}_{pkt}} (T_k - (A_k - \pi_k))$ calculates the accurate number of total transmissions which were sent by i, and j did have opportunities to receive them. Specifically, T_k is the number of transmissions of packet k. A_k denotes whether packet k was acknowledged. If packet k is dropped without receiving an ACK, A_k is 0. Otherwise A_k is 1. In Fig. 6, A_1, A_2, A_3 and A_4 are 1, because these packets are definitely acknowledged (either there is state 1 or the number of retransmissions does not exceed the pre-defined threshold). Before considering the accuracy of Eqn. 1, we should first note that the acknowledged transmission can only indicate some forwarder has successfully received the transmitted packet. However, for the other forwarders which did not receive the packet, we cannot infer whether the packet was lost due to the influence of N or the packet was missed due to forwarders' sleeping in asynchronous low power opportunistic forwarding. Hence, we add a correction parameter π_k in Eqn. 1. If a forwarder replies an ACK and the sender happens to receive an ACK for a transmission, the value of π_k is 1, no matter whether the received ACK does come from the forwarder or not. Otherwise π_k is 0. This assignment makes sense, because if the forwarder does not reply an ACK (may be in sleep state) for a transmission but the sender receives an ACK, the last transmission will not be used to compute the unidirectional *cpdr* from sender to the forwarder.

The objective of Eqn. 1 is to compute the probability that a data packet will be successfully delivered to j. In the same way, the *cpdr* of an ACK transmitted from j to sender i can be computed according to

$$P^N_{j,i} = 1 - \frac{\sum_{k \in \mathbf{S}_{pkt}} (ACKT_k - \pi_k)}{\sum_{k \in \mathbf{S}_{pkt}} ACKT_k}, \quad (2)$$

where $ACKT_k$ is the number of ACK transmissions which is equal to the number of total received copies of a packet k, and π_k is the same to that of Eqn. 1. $\sum_{k \in \mathbf{S}_{pkt}} (ACKT_k - \pi_k)$ is the number of the total failed transmissions of ACKs, and $\sum_{k \in \mathbf{S}_{pkt}} ACKT_k$ is the number of all ACK transmissions. Hence, $P^N_{j,i}$ denotes the probability that an ACK can be successfully delivered from j to i under the influenced of N's transmission.

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 moving average to update both $P^N_{i,j}$ and $P^N_{j,i}$ by

$$P^N_{i,j} = (1 - \theta) \times P^{N,old}_{i,j} + \theta \times P^{N,new}_{i,j}, \quad (3)$$

$$P^N_{j,i} = (1 - \alpha) \times P^{N,old}_{j,i} + \alpha \times P^{N,new}_{j,i}. \quad (4)$$

Both θ and α will be detailedly discussed in the implementation of *COF*.

C. Expected Benefit of Concurrent Transmission

We define the expected benefit of concurrent transmission as the expected gain ($EGain$) of throughput for a period of one duty cycle. 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 neighbor node N which could concurrently transmit with itself. The entry forms as (N, $< P^N_{i,F_1}, P^N_{F_1,i} >$, $\cdots$, $< P^N_{i,F_n}, P^N_{F_n,i} >$, $epdr(i|N)$), where $\{F_1, \cdots F_n\}$ is i's forwarder set, marked as $\mathbf{F}_i$. According to each entry, *COF* computes the expected packet delivery ratio (*epdr*) of the transmission influenced by N. Under the interference of N, i computes the *epdr* namely $epdr(i|N)$ by

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

where $P^N_{i,j} \times P^N_{j,i}$ denotes the probability that both a data packet from i to j and the replied ACK from j to i succeed within a period of duty cycle, and $\prod_{j \in \mathbf{F}_i} (1 - P^N_{i,j} \times P^N_{j,i})$ denotes i can not receive an ACK from all candidate forwarders. Note that $epdr(i|N)$ only indicates the influence of N's transmission on i's data forwarding.

By completing *epdr* and recording it in the last column ($epdr(i|N)$) of each entry of **CPDR** table, each node will attach the first column and the last column of each entry in probe together with bitmaps and broadcast it to neighbor nodes presented in Section III-F. Once overhearing the probe, *COF* extracts and maintains the entry relevant to itself in benefit table ($\mathbf{BTable}$). If i overhears N's probe, it only extracts the entries of (i, $epdr(N|i)$) and (i, $epdr(N|\o)$) and maintains them in $\mathbf{BTable}$ table corresponding to $epdr(i|N)$. For node i, the entry of $\mathbf{BTable}$ is (N, $epdr(i|N)$, $epdr(N|i)$, $epdr(N|\o)$, permission of concurrency).

According to the $\mathbf{BTable}$ table, each node will compute the expected benefit of concurrent transmissions ($EGain$) than transmission in sequence. If N is transmitting now and i intends to transmit, i should compute the overall gain of concurrent transmission namely as $T(i|N)$ by

$$T(i|N) = epdr(i|N) + epdr(N|i). \quad (6)$$

Consider that it has recorded $epdr(N|\o)$, the expected packet delivery ratio of N's transmission without being influenced by i, in $\mathbf{BTable}$ table. Then, the overall benefit of concurrent transmissions is

$$EGain(i|N) = T(i|N) - epdr(N|\o). \quad (7)$$

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

$$EGain(i|N) > \omega, \quad (8)$$

we believe that concurrent transmission is better than sequential transmission. ω is a compensation value for the extra consumption (e.g., energy) of concurrent transmission and will be discussed in the implementation. To ensure the consistency of the decision made by both i and N, i also checks N's $EGain(N|i)$ by

$$EGain(N|i) = T(N|i) - epdr(i|\o) > \omega. \quad (9)$$

According to the double check, *COF* can guarantee the consistency of decisions made by concurrent senders.

If both Eqn. 8 and 9 are satisfied, *COF* permits the transmission of i even though N is transmitting, and adds the permission marker (yes) in the last column (permission of concurrent) of $\mathbf{BTable}$ table. Otherwise, it adds the denied marker (no) in the table.

D. Decision Maker

To transmit a data packet, the *transmission decision module* of MAC layer not only uses carrier sense to know whether a channel is busy, but also sends a transmission notification event to the *decision maker* module of *COF*. *Decision maker* module first queries the data preamble logs recorded by *overhear module* to confirm whether a data preamble was overheard during the last several milliseconds. If nothing was overheard, *Decision maker* module returns a value denoting *no recommendation*. Otherwise, if a data preamble sent by a neighbor was overheard, *Decision maker* module first queries the $\mathbf{BTable}$ table to verify whether concurrent transmission is permitted. Then it returns a value denoting the *permission of concurrent* or *denial of concurrent* to the *transmission decision module*.

If the returned value from *COF decision maker* module is the *permission of concurrent*, the *transmission decision module* of MAC immediately transmits the preamble of the pending data packet by disabling the carrier sensing for a period of wake-up interval. However, if the returned value is

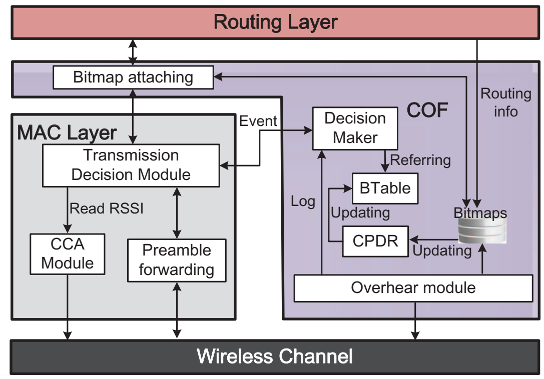


Fig. 7. Integration of *COF* into the protocol stack of low power opportunistic forwarding.

the *denial of concurrent*, no matter whether the MAC layer is transmitting or not, it pauses MAC layer's transmission and makes a random backoff. After each backoff, if the node intends to transmit, the *transmission decision module* will also use carrier sense and send a transmission notification event to *COF* as mentioned above. If *COF decision maker* module returns *no recommendation*, the *transmission decision module* of MAC decides to transmit or not by only referring to the carrier sense result.

E. Initial Stage

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

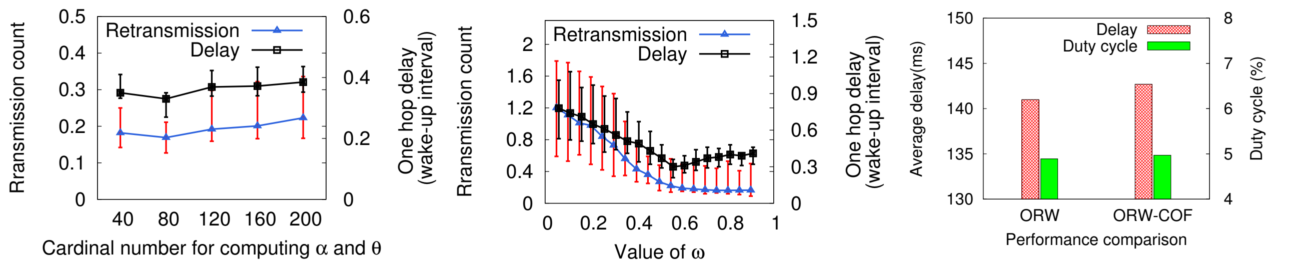
Note that the excessive indulgence of concurrent transmission in initial stage may result in consecutive transmission failures. To address this problem, *COF* uses the enforcement of denial of concurrent. Once *COF* gets aware of the consecutive failures in the routing layer (exceeding 6 retransmissions in our implementation), it actively issues a *denial of concurrent* event to the *transmission decision module* of MAC layer to enable the carrier sense for the next transmission.

F. 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.

Without changing the original mechanism of probe transmission, *COF* only broadcasts a probe carrying the forwarder bitmaps and recorded information in $\mathbf{BTable}$ table with a long time interval T_{max} periodically. In *COF*, T_{max} is set to 5 minutes for information exchange. A *COF* probe will not be concurrently transmitted with another ongoing sender, 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 the probe footer.

Additionally, *COF* also exploits the possible opportunity of small data packets, which have free space to carry at



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

Fig. 8. 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.

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 data packet is independent of probe transmission, and *COF* does not guarantee the attached bitmaps can be overheard by all children nodes. As time goes, each node will collect sufficient information to increase the accuracy of *cpdr*. By exchanging the recorded information in $\mathbf{BTable}$ table between each pair of neighbor nodes, they will immediately update the expected benefit of concurrent transmission, and keep the consistency of permission of concurrency between them.

G. Integration into Protocol Stack

The integration of *COF* into the protocol stack of low power opportunistic forwarding is shown as Fig. 7. To transmit a data packet, routing layer first delivers it down to the *Bitmap attaching* module of *COF* to attach the frequently updated bitmaps to probe or small data packet footer, and then, *COF* continues to deliver it to low power MAC layer. In MAC layer, the data packet is transmitted in the form of repeated preambles, and will be forwarded to the earliest wake-up forwarder. When receiving a data preamble, the low power MAC delivers it upwards to *Bitmap attaching* module. *COF* extracts the relevant bitmaps recorded by candidate forwarders for updating $\mathbf{CPDR}$ table and $\mathbf{BTable}$ table.

Additionally, by exploiting the routing information from routing layer, *COF* maintains a set of bitmaps for both potential children nodes and neighbors, respectively. Once overhearing a data preamble, the *overhear module* updates the related bitmap (if needed) and records a log for *decision maker* module. By using the updated bitmaps maintained by itself and the overheard bitmaps from candidate forwarders, *COF* recomputes links' *cpdr*s and updates $\mathbf{CPDR}$ table and $\mathbf{BTable}$ table.

IV. Implementation and Evaluation

A. Implementation

We implemented *COF* in TinyOS 2.1.1. The RAM and ROM consumptions of *COF* are 947 bytes and 3186 bytes, respectively. Next, several implementation issues are carefully discussed.

1) Link cpdr Update: In Eqn. 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, hence 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 Eqn. 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 (two bits denote a DSN and 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 single hop delay. We think that the optimal θ and α can result in good performance. In each experiment, we set each node's inter-packet interval to 4 seconds, and each experiment lasts 2 hours in the indoor testbed with 22 Telosb nodes. We plot the average retransmission count and average one hop delay by changing the value of CN in Fig. 8(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.

2) Compensation Value ω : In Eqn. 8 and 9, the weight ω is a compensation value for the expected benefit. Generally, a larger ω will reduce the opportunity for concurrent transmission, but it could reduce the retransmission rate. On the other hand, assigning a very low value to ω could increase retransmission rate caused by data collision, resulting in 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 evaluation using an indoor testbed with 22 Telosb nodes by calibrating ω. We plot the average retransmission count and average one hop delay by changing the setting of ω in Fig. 8(b). From the experimental results plotted in the figure, 0.55 is a reasonable value. Hence, we set ω to 0.55 in our implementation.

3) Network Overhead: *COF* adopts both network probe and data packet footer to share the recorded receiving information. The overhead is very limited. Here, we conduct two experiments in the indoor testbed to test the extra overhead

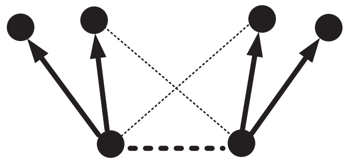


Fig. 9. Constraints on topologies for testing the expected benefit (*EGain*) of concurrent transmission.

introduced by *COF*: the first one runs the original version of ORW by setting nodes' IPI to 4 minutes; and the second one runs the version of ORW combined with *COF* also setting IPI to 4 minutes, but the *decision maker* module always returns a *no recommendation* to disable concurrent transmission. We compute the average single hop delay and average radio duty cycle of all nodes in Fig. 8(c). As the figure shows, the extra overhead of *COF* brings 0.9% extra delay and 0.75% extra energy consumption. Considering the benefit of exploiting concurrent opportunity, we are confident that the performance of ORW combined with *COF* is superior to that of ORW, as demonstrated by the evaluation results.

B. Experimental Results

In this section, we evaluate *COF* by conducting various testbed experiments. We first test the expected benefit of concurrent transmission in the presence of *opportunistic exposed terminal* shown in Fig. 9, and then we conduct experiments in indoor testbed to test the performance improvement of low power opportunistic forwarding (ORW) by combining with *COF*.

1) Experimental Testbed and Method: Our experiments are conducted in an indoor testbed with 40 Telosb nodes (22 nodes fixed in office wall and 18 nodes are scattered on the floor). By setting data transmission power level to 3, these nodes automatically form a four-hop network. All experiments are conducted in the 26th channel. Unless mentioned otherwise, all senders transmit 100-byte data packets by setting wake-up interval to 512ms.

2) COF Exploiting Concurrent Opportunity: In this experiment we seek to quantify the throughput gain by adopting *COF* in presence of *opportunistic exposed terminal*, which is shown as Fig. 9. In this situation, (1) two senders are within the carrier sense range of each other, (2) each sender has two candidate forwarders and one of the two is influenced by the other sender, (3) the bold line denotes a link with high (above 0.9) packet delivery ratio and dotted lines denote the interference between two nodes. Before this experiment, we first let all nodes take turns to broadcast beacons (100 beacons per node) to identify the signal strength and link quality between each pair of nodes. Then, we carefully select nodes from the testbed to construct this target topology. The other nodes turn off their radios during this experiment. To quantify the performance of different network configurations (e.g., with carrier sense disabled or enabled, with *ORW* combining with *COF* or not), we measure the total throughput at the designated forwarders by eliminating duplicate packets. For example, nodes B and C are assigned to be A's forwarders. If both B and C receive a packet with the identical DSN, we regard one of the two as a duplicate packet. The selected two senders generate data packets by setting IPI to 512ms, the same to wake-up interval.

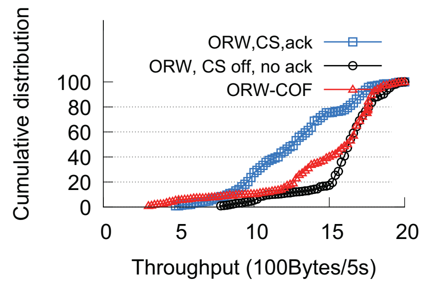
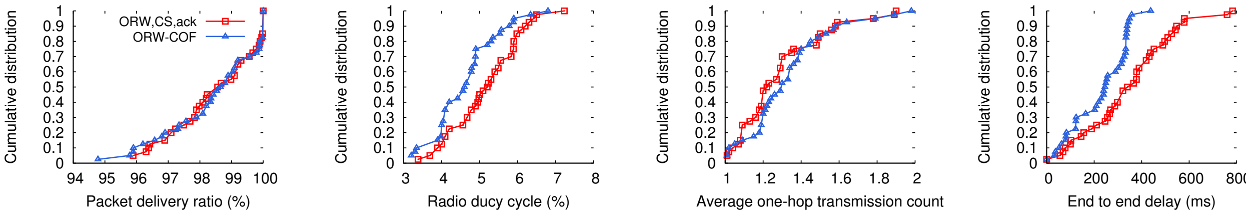


Fig. 10. *COF* exploits concurrent opportunity to achieve a 1.64× gain.

We select 10 topologies from the testbed meeting the mentioned three conditions. Fig. 10 presents the distribution of throughput across 10 exposed terminal topologies. We quantify the throughput by counting the total packets received by four candidate forwarders during a period of 5 seconds windows by excluding all duplicates. The theoretical maximum throughput in this set of experiments is 20. Each evaluation is run for 10 minutes and repeated 5 times, hence, the plotted cumulative distribution is computed from a set of 6000 samples (5×10×120, there are 120 time windows in the 10 minutes experiment). With the same *opportunistic exposed terminal* topology, we evaluate the performance of ORW with carrier sense enabled requesting ACK (marked as *ORW,CS,ack* in figures), ORW with carrier sense disables without requesting ACK (*ORW,CS off,no ack*), and ORW combining with *COF* (*ORW-COF*), respectively. As shown in Fig. 10, the throughput of *ORW,CS,ack* is far less than that of *ORW,CS off,no ack*, because in this case *ORW,CS,ack* suppresses the opportunities of concurrent transmissions. However *ORW,CS off,no ack* concurrently transmits data packets periodically (512ms) without requesting forwarders' acknowledgements. It can maximally exploit the spatial diversity of forwarders while remove the influence of ACK collision at transmitting end. The candidate forwarders, which locates outside the interference range of the other sender, can successfully receive the packets. Hence the *ORW,CS off,no ack* can attain the largest throughput of the three. Compared with *ORW,CS,ack*, by combining *COF* with ORW, the throughput is increased by 64%, because *COF* can quickly confirm the feasibility of exploiting concurrent opportunity by considering the packet receipt rate at both the forwarder ending and transmitting end.

3) Network performance: To evaluate the performance of *COF* in a network, we conduct experiments in the indoor testbed with 40 Telosb nodes. Every node generates a packet randomly with an average interval of 4 minutes, and the network forwards it to sink. We run *ORW,CS,ack* and *ORW-COF* in the testbed, respectively, for at least 24 hours. By computing the mean packet delivery ratio (PDR), radio duty cycle, mean one hop transmission count, and each node's end-to-end delay, we plot the cumulative distributions in Fig. 11.



(a) Packet delivery ratio (b) Duty cycle (c) One hop transmission count (d) End-to-end delay

Fig. 11. Network performances, consisting of network reliability (packet delivery ratio), network efficiency (duty cycle and one hop transmission count), and end-to-end delay, of ORW and that of ORW combining with *COF*.

Overall, the cumulative distribution of nodes' PDR of *ORW-COF* is almost the same as that of *ORW,CS,ack* shown as Fig. 11(a), except a node's PDR (94.7%) which may cost a long period of time to construct $\mathbf{BT}$able resulting in data loss. However, once completing the construction of $\mathbf{BT}$able, the exploiting of concurrent transmission significantly reduces the expected waiting time of each transmission, so as to reduce the radio duty cycle by 18.9% compared with that of *ORW,CS,ack* shown in Fig. 11(b). In addition, compared with *ORW,CS,ack*, the opportunistically exploiting concurrent transmission of *ORW-COF* also reduces the end-to-end delay by about 41% shown as Fig. 11(d). Although concurrent transmission is adopted in *COF*, indicating network interference will increase in some degree, the average one hop transmission count increases slightly compared with that of *ORW,CS,ack*, shown in Fig. 11(c). Overall, *COF* can significantly improve the performance of low power opportunistic forwarding.

V. Related Work

Spatial reuse is a well-known concept in wireless communications networks of different types. MACA [3] makes the observation that carrier sense cannot make correct transmission decisions. Researchers have proposed a lot of methods and mechanisms to address the problem. Here we have a brief discussion on the existing proposals in three aspects respectively.

In wireless communication community, capture effect is a well know phenomenon [4][5]. Various capture models have been proposed and evaluated mostly for ALOHA networks [6] and recently for some 802.11 [7] and 802.15.4 [8][9] networks. The most common model uses a constant threshold for each modulation and coding scheme with the ratio of the signal strength and summation of interference strength. However, these are primarily theoretical study and analysis.

Furthermore, there also have been a lot of works on adjusting protocol parameters to achieve a better spatial reuse in wireless ad hoc networks. On one hand, several works [11][10] have been designed to mitigate data collision by adjusting transmission power. On the other hand, based on CSMA-based MAC, previous studies on adjusting the CCA threshold to better avoid collisions in both 802.11 [12] and 802.15.4 [13] networks. However, all of them focus on deterministic forwarding protocols, ignoring the spatial reuse of low power opportunistic forwarding.

In addition, conflict graph is a good tool for exploiting exposed terminals. Existing works can be divided into two categories based on the type of conflict graphs they use. The first category exploits physical modes [14][15]. However, these theoretical models are difficult to be adopted to real-deployed networks. The second category uses per-link signal measurements [24] to capture interference conditions among individual links, using either active measurements [18][19] or passive measurements [16][17]. However, all of them are not suitable for low power opportunistic forwarding with new features.

Some other related works, such as partial packet recovery [31] and interference cancellations [32], have also been proposed to achieve concurrent transmission. However, these techniques are heavily dependent on high precise time synchronization (microsecond level). In resource-restricted wireless sensor networks, considering network dynamics, it is difficult to achieve microsecond level time synchronization. Hence, these techniques are difficult to be directly applicable to low power WSNs.

Different from these proposed methods, although some theoretical studies and analysis [29][30] have been proposed to improve the performance of opportunistic forwarding in Ad Hoc network, *COF* is a proposed lightweight and effective approach to exploit the opportunity of concurrent transmission for low power opportunistic forwarding.

VI. Conclusion

In this paper, we propose *COF* to exploit concurrent opportunity for low power opportunistic forwarding. *COF* uses an efficient and distributed way to estimate link's conditional packet delivery ratio in real time when different neighbors are transmitting and computes the benefit of concurrent transmission. We implement *COF* and integrate it with ORW [22] (the state-of-the-art opportunistic forwarding protocol) and LPL [1], and evaluate it on an indoor testbed. Experimental results show that *COF* can significantly improve network throughput and reduce energy consumption of the stat-of-the-art low power opportunistic forwarding protocol.

Acknowledgment

The authors would like to thank the shepherd, Honggang Zhang, for his constructive feedback and valuable input.

Thanks also to anonymous reviewers for reading this paper and giving valuable comments.

This study is supported in part by NSFC No. 61472067, National Basic Research Program (973 program) under Grant of 2014CB347800, National Key Technology R&D Program No. 2013BAH33F02, Key Technologies Research and Development Program of Sichuan Province No. 2013GZ0006, NSFC No. 61472217, NSFC No. 61170213, and National Science Fund for Excellent Young Scientist No. 61422207.

REFERENCES

[1] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks", *In Proceedings of ACM SenSys*, 2004.

[2] D. Liu, X. Wu, Z. Cao, M. Liu, Y. Li, M. Hou, "CD-MAC: A Contention Detectable MAC for Low Duty-Cycled Wireless Sensor Networks", *In Proceedings of IEEE SECON*, 2015.

[3] P. Karn, "MACA-A New Channel Access Method for Packet Radio", *In ARRL Computer Networking Conference*, 1990.

[4] L. Roberts, "Aloha packet system with and without slots and capture", *ACM SIGCOMM Computer Communication Review, vol.24(4), p.28-42*, 1975.

[5] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, D. Culler, "Exploiting the capture effect for collision detection and recovery", *In Proceedings of ACM EmNetS-II*, 2005.

[6] I. Habbab, M. Kavehrad, C. Sundberg, "Aloha with capture over slow and fast fading radio channels with coding and diversity", *IEEE Journal on Selected Areas in Communications, vol.40. no.3, p.79-88*, 1989.

[7] J. Kim, J.K. Lee, "Capture effects of wireless csma/ca protocols in rayleigh and shadow fading channels", *IEEE Transactions on Vehicular Technology, vol.4. no.4, p.1277-1286*, 1999.

[8] F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, "Efficient network flooding and time synchronization with glossy", *In Proceedings of ACM IPSN*, 2011.

[9] J. Lu, K. Whitehouse, "Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks", *In Proceedings of IEEE INFOCOM* 2009

[10] M. Sha, G. Xing, G. Zhou, S. Liu, "C-MAC: Model-driven concurrent medium access control for wireless sensor networks", *In Proceedings of IEEE INFOCOM*, 2009.

[11] S. Lin, S. Lin , J. Zhang, G. Zhou, L. Gu, T. He , J.A. Stankovic, "ATPC: Adaptive transmission power control for wireless sensor networks", *In Proceedings of ACM SenSys*, 2006.

[12] M.Z. Brodsky, R.T Morris, "In defense of wireless carrier sense", *In Proceedings of ACM SIGCOMM*, 2009.

[13] C.A. Boano, T. Voigt, N. Tsiftes, L. Mottola, K. Romer, M.A. Zuniga, "Making sensornet mac protocols robust against interference", *In Proceedings of EWSN*, 2010.

[14] P. Gupta, P.R. Kumar, "The capacity of wireless networks", *IEEE Transactions on Information Theory, vol.46, no.2, p.388-404*, 2000.

[15] I. Rhee, A. Warrier, J. Min, L.S. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks", *In Proceedings of ACM MobiHoc*, 2006.

[16] S.C. Liu, G.L. Xing, H.W. Zhang, J.P. Wang, J. Huang, M. Sha, L.S. Huang, "Passive interference measurement in wireless sensor networks", *In Proceedings of IEEE ICNP*, 2010.

[17] V. Shrivastava, S. Rayanchu, S. Banerjee, K. Papagiannaki, "PIE in the sky: Online passive interference estimation for enterprise WLANs", *In Proceedings of USENIX NSDI*, 2011.

[18] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, P. Steenkiste, "DIRC: Increasing indoor wireless capacity using directional antennas", *In Proceedings of ACM SIGCOMM*, 2009.

[19] M. Vutukuru, K. Jamieson, H. Balakrishnan, "Harnessing exposed terminals in wireless networks", *In Proceedings of USENIX NSDI*, 2008.

[20] M. Buettner, G.V. Yee, E. Anderson, R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks", *In Proceedings of ACM SenSys*, 2006.

[21] S. Biswas, R. Morris, "ExOR: opportunistic multi-hop routing for wireless networks", *In Proceedings of ACM SIGCOMM*, 2005.

[22] O. Landsiedel, E. Ghadimi, S. Duquennoy, M. Johansson, "Low power, low delay: opportunistic routing meets duty cycling", *In Proceedings of ACM IPSN*, 2012.

[23] D.B. Liu, Z.C. Cao, J.L. Wang, Y. He, M.S. Hou, Y.H. Liu, "DOF: duplicate detectable opportunistic forwarding in duty-cycled wireless sensor networks", *In Proceedings of IEEE ICNP*, 2013.

[24] X. Zhou, Z.B. Zhang, G. Wang, X.X. Yu, B.Y. Zhao, H.T. Zheng, "Practical Conflict Graphs for Dynamic Spectrum Distribution", *In Proceedings of ACM SIGMETRICS*, 2013.

[25] A. Dunkels, "The ContikiMAC radio duty cycling protocol", *In SICS Report*, 2012.

[26] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, "Collection tree protocol", *In Proceedings of ACM SenSys*, 2009.

[27] K. Srinivasan, M.A. Kaz, S. Agarwal, P. Levis, "The β-factor: measuring wireless link burstiness", *In Proceedings of ACM SenSys*, 2008.

[28] D.B. Liu, Z.C. Cao, J.L. Wang, M.S. Hou, Y.J. Li, "RxLayer: adaptive retransmission layer for low power wireless", *In Proceedings of ACM MobiHoc*, 2014.

[29] A. Bhorkar, M. Naghshvar, T. Javidi, B. Rao, "An adaptive opportunistic routing scheme for wireless Ad-hoc networks", *In IEEE/ACM Transactions on Networking, vol.20, no.1, p.2838-2842*, 2012.

[30] A. Bhorkar, T. Javidi, A.C. Snoeren, "Achieving congestion diversity in wireless ad-hoc networks", *In Proceedings of IEEE INFOCOM*, 2011.

[31] K. Jamieson, H. Balakrishnan, "PPR: Partial Packet Recovery for Wireless Networks", *In Proceedings of ACM SIGCOMM*, 2007.

[32] S. Gollakota, S.D. Perli, D. Katabi, "Interference Alignment and Cancellation", *In Proceedings of ACM SIGCOMM*, 2009.