PPM: Preamble and Postamble Based Multi-Packet Reception for Green ZigBee Communication

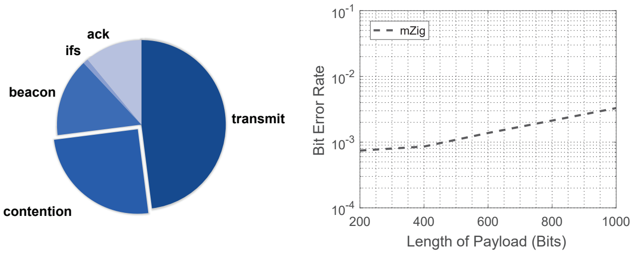
Zhe Wang[1], Linghe Kong[1], Guihai Chen[1], Liang He[2]
[1]Shanghai Jiao Tong University, China
[2]University of Colorado Denver, USA
Email: {wang-zhe, linghe.kong, chen-gh}@sjtu.edu.cn, liang.he@ucdenver.edu

***Abstract*—ZigBee, a low-power wireless communication technology, has been used in various applications such as smart health/home/buildings. The proliferation of ZigBee-based applications (and thus devices), however, makes the concurrent transmissions — i.e., multiple transmitters send packets to the same receiver at the same time — common in practice, leading to inevitable collisions. To facilitate the concurrent transmissions of ZigBee, we design Pre/Post-amble based Multi-packet reception (PPM), a method that recovers the collided ZigBee messages by exploiting their collision-free chips and the overlapped chips in their pre/post-ambles. Such a collision recovery of PPM reduces the retransmissions caused due to collisions, facilitating the realization green ZigBee. We have prototyped and evaluated PPM with USRP, showing PPM recovers the collided messages with bit-error-rates in the order of 10^{-6}, which is magnitudes lower than state-of-the-art methods.**

I. Introduction

ZigBee [1], a low-power wireless protocol based on the IEEE 802.15.4 standard, has been widely used in applications such as smart home [2] and transportation systems [3], thanks to its short-distance, low-energy, and low-cost features. It is envisioned that ZigBee will draw even more attentions with the further development of Internet of Things (IoT) [4] [14] [18] and cross technology communication (CTC) [5] [15]. The proliferation of ZigBee devices, however, makes the concurrent transmission, i.e., multiple transmitters send packets to the same receiver at the same time in sending the ZigBee messages, common in practice. Standard ZigBee uses Carrier-sense Multiple Access with Collision Avoidance (CSMA/CA) [6] in its media access control (MAC) layer to avoid collisions, at the cost of additional energy and time consumption for synchronization and channel contention [12] [17].

Collision recovery, instead of avoidance, has been demonstrated to be an effective alternative to facilitate ZigBee's concurrent transmission [7] [8]. Collision recovery methods achieve lower power and time consumption when compared to standard ZigBee, so long as certain packet reception ratio (PRR) can be achieved. According to the energy consumption of different protocol phases in ZigBee's transmission [12], less than 50% of the total energy is used for transmission and 25% of the energy is spent during CSMA/CA contention as shown in Fig. 1(a). Thus, the energy consumption for retransmissions when collision recovery fails will be less than the persistent overhead of channel contention in standard ZigBee as long as the PRR is larger than 67% for concurrent transmissions without CSMA/CA.



(a) (b)

Fig. 1. (a) Breakdown of overhead. (b) Accumulative error of mZig.

These existing collision recovery methods, however, suffer from accumulated error in packet reception. For example, Fig. 1(b) plots the bit error rate (BER) of mZig [8], a state-of-the-art collision recovery method that decomposes collided packets based on the features of ZigBee's physical layer, showing mZig's BER increases significantly with longer packets due to accumulated errors.

To mitigate this deficiency, we design a novel physical layer of ZigBee, called *Pre/Post-amble Based Multi-Packet Reception* (PPM), which achieves ppm-level (i.e., 10^{-6}) BER in recovering the collided messages. PPM divides the preamble of standard ZigBee packet into a pre- and a post-amble of the same size whose bits sequences are known already, and then extracts received signals/samples of single chips representing bit 0 or 1 and the overlap of chips from multiple transmitters (TX) based on the known pre/post-ambles, called *reference chips*, to recover the collided packet by comparison.

Let us use a walk-through example shown in Fig. 2 to explain PPM's core idea: Alice (A) and Bob (B) concurrently send packets to a receiver (RX) and a collision occurs at RX. Because Alice and Bob are not synchronized, time offset usually exists between their packets' arrivals at RX, as shown in Fig. 3. Experimental results in [8] demonstrate over 96% collisions have such time offsets. These time offsets may include chip offset, sample offset, or both of them, where the duration of chip offset is of several chip cycles and sample offset consists of a number of sample intervals as depicted in Fig. 3(a).

PPM's construction of the reference chips can be divided into the following 3 cases.

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

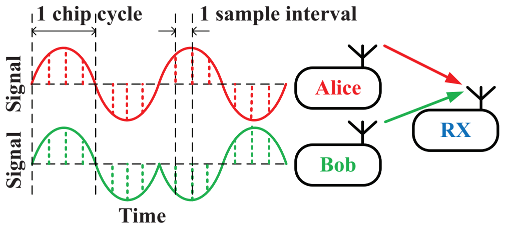
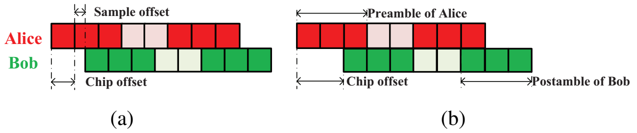


Fig. 2. An example of concurrent transmission with two TXs and one RX.



(a) (b)

Fig. 3. Two categories of time offsets: chip offset C_o and sample offset S_o.

1) If the collision has only chip offset, we can obtain two kinds of information. First, there are several collision-free chips in pre/post-ambles, e.g., the first two chips of A and the last two chips of B in Fig. 4, which can be directly identified. Using these collision-free chips, we can generate the *single chips' reference chips* (**R**) representing 0 or 1. Second, some chips in pre/post-amble are overlapped. As there are two transmitters in this example, the overlapped chips have a total number of four possible combinations: 00, 01, 10 and 11. The overlapped chips of these four combinations can be found in the overlapped pre/post-amble as shown in Fig. 4. We can extract *2-overlapped chips' reference chips* ($\mathbf{R}^2$) by averaging these overlapped chips.
2) If the collision has sample offset, we observe that one overlap unit (duration of one chip cycle) involves exactly three chips. For example, a chip of A is overlapped with two partial chips of B as shown in Fig. 5. Denoting the sequence of these three chips as 'BAB' or 'ABA'. In Fig. 5, we mark the chip sequence representing the overlap combination as '000' and '111'. Hence, the overlapped chips have totally 8 combinations for A and B, respectively. We can extract these combinations from pre/post-amble as depicted in Fig. 5, named *3-overlapped chips' reference chips* ($\mathbf{R}^3$).
3) If the collision has both chip offset and sample offset, which is the most case in concurrent transmission, we have both collision-free chips and 3-chip overlaps. Especially, the abstracted $\mathbf{R}^3$ from the pre/post-amble may be incomplete, i.e., some overlap combinations are not appeared. In this case, we use linear transformation of collision-free chips and abstracted overlaps to generate the missing one.

With the thus-constructed reference chips, PPM compares the overlapped payload and the reference chips to identify the collision chip-by-chip.

This paper makes the following major contributions.

- Design of PPM, a novel physical layer technique to decode the collided ZigBee packets without suffering from the accumulative error.

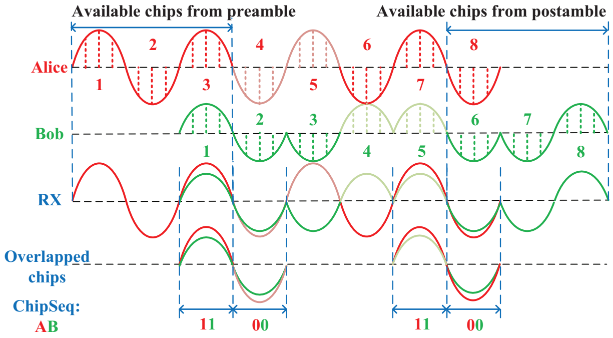


Fig. 4. Decode two-packet collision with only chip offset in PPM.

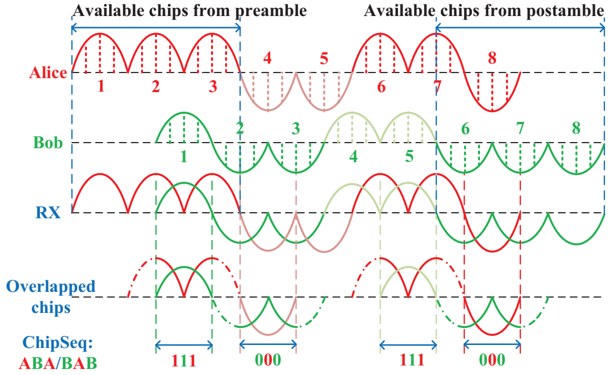


Fig. 5. Decode two-packet collision with chip and sample offsets in PPM.

- Implement and evaluation of PPM with USRPs, showing PPM achieves (i) BER of 10^{-6}, which is 2 magnitudes lower than state-of-the-art solutions, (ii) a *packet reception ratio* (PRR) of 90%, and (iii) less than 40 retransmissions in a concurrent transmission experiment with 400 packets.

II. Background

We first review the physical layer of standard ZigBee and introduce ZigBee's unique features related to our design.

A. Physcial layer of standard ZigBee

The standard ZigBee [1] has three ISM bands: 868.3MHz in Europe, 902-928MHz in America, and 2.4GHz worldwide. Among them, 2.4 GHz ISM band is the most widely used band and its corresponding bit rate is 250kbps. The data frame of ZigBee consists of 3 parts: Synchronization Header (SHR), Physcial Layer Header (PHR) and Protocol Service Data Unit (PSDU). The 32-bit preamble is a part of SHR which consists of 32 zeros.

Fig. 6 shows the flow graph of ZigBee's transmission: the transmitter (TX) sends bitstream using five procedures and the reciever (RX) receives data with corresponding inverse procedures. Here we mainly focus on the first three processes including spread, modulation and pulse shaping. Firstly, ZigBee adopts Direct Sequence Spread Spectrum (DSSS) as its spreading. In this step, each symbol made up of 4 bits is transformed into 32 chips according to a decided mapping table. Next, ZigBee modulates chips exploiting O-QPSK. As

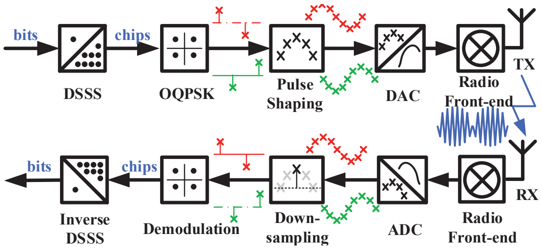


Fig. 6. Physical layer of standard ZigBee.

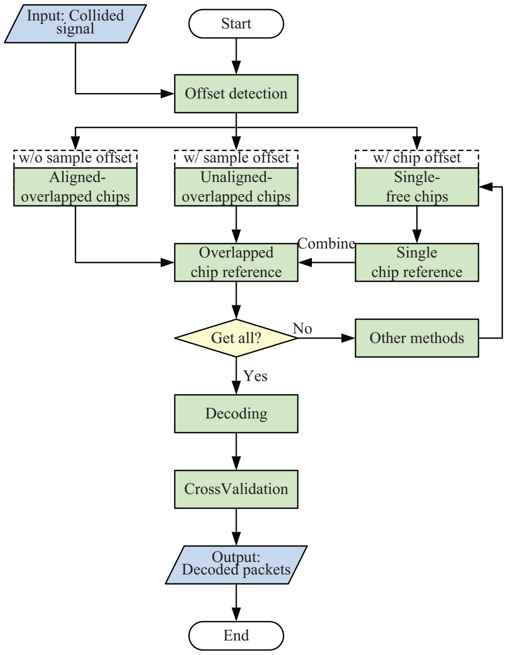


Fig. 7. Flow chart of PPM.

a result, chips in one packet have nearly uniform amplitude. Then pulse shaping is performed and all the chips in one packet are shaped as half-sine wave with positive or negative amplitude of the same absolute value.

B. Feature of ZigBee

The standard ZigBee has the following two features that closely relate to our design.

- Preamble: Every ZigBee's frame has a preamble of 32 bits, i.e., 256 decided chips.
- Pulse shaping and modulation: These two components present that chips 0 or 1 in one packet have identical waveforms.

Next we will show more details of our design based on these features.

III. Core Design of PPM

In this part, we present PPM design in details.

A. Overview of PPM

1) Overview: Based on the above features of ZigBee, we propose a novel Preamble and Postamble based Multi-packet reception (PPM). The main idea of PPM is abstracting reference chips from known data to recover unknown overlapped chips. However, the reference chips cannot be synthesized by the standard chips from TXs because the received packet affected by the dynamic channel is usually different from the standard ones. We can only get the known information from the received and collided packet at RX and the known information is limited by short-length preamble. In order to obtain more known information, we divide the preamble of a ZigBee packet into a preamble and a postamble of the same size. By collecting more collision-free and overlapped chips in pre- and postamble, reference chips can be estimated more accurately.

Leveraging the known pre-/postambles and the detectable time offsets, all combinations of reference chips can be synthesized. Since chips in the same packet are transmitted in short terms, i.e., they meet the same channel state, the overlapped chips can be recognized by comparing reference chips. As a result, we can achieve accurate decoding of collided packets, which reduces retransmission times when collisions happen and corresponding energy consumption.

The flow chart of PPM is illustrated in Fig. 7. PPM abstracts available known chips from pre- and postamble. After that, reference chips are calculated by averaging known data of the same values. By combining existing overlapped chips' reference chips and single chips' reference chips, sufficient overlapped chips' reference chips can be synthesized. Otherwise, we can seek to other methods for help. After all combinations of reference chips are generated, the collided packets can be decoded precisely by mapping unknown overlapped chips with reference chips since the chip arrays of reference chips are known.

2) Two-packet collision decoding: To show how PPM decodes collided packets in details, we give two examples of two-packet collision in Fig. 5 and Fig. 4. Time offset between packets from different TXs can be detected and calculated by self-correlation as mentioned in [6] [7]. Time offset is the sum of two parts which are chip offset and sample offset. The number of chips cycles in *chip offset* and the number of sample intervals in *sample offset* is denoted by C_o and S_o respectively. The number of sample intervals in one chip cycle is denoted by λ, which is determined by bandwidth and sampling rate.

Considering two packets from A and B of the same length L in chips, the number of known chips are denoted by $2t$ where t is the length of pre- and postamble. The received chips from A or B in the RX are denoted as $A[i]$ or $B[i]$ respectively, where i is the i-th chip in the packet. Meanwhile, a chip of A or B consists of λ sample points, which can be represented as $A[i][j]$ or $B[i][j]$ where j is the j-th sample in one chip. Next, we will show details about how to extract reference chips utilizing A as an example.

Collision with chip offset. After time offset is detected, available known chips are abstracted from pre- and postamble as shown in Fig. 5 and Fig. 4. When two-packet arrives at RX with chip offset, there are several collision-free chips, e.g., $A[1]$ and $B[8]$. Since waveforms of chips representing the same binary value are identical in one packet, we can

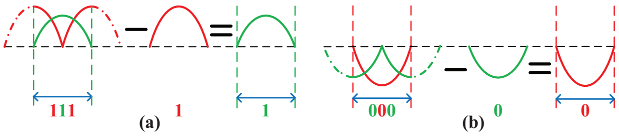


Fig. 8. Obtain the reference waveform of single chip by minus operation between the known overlapped waveform and another single chip.

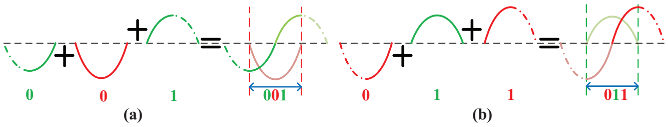


Fig. 9. Generate the reference waveform of overlapped chips by add operation of two known single chips.

calculate the *single chips' reference chips* (**R**) presenting 0 or 1 by averaging collision-free chips. Denoting the chip value of $A[i]$ as V_i, $\mathbf{R}_A[k]$ can be calculated as follows:

$$\mathbf{R}_A[k][j] = \frac{\sum_{V_i=k} A[i][j]}{\sum_{V_i=k}}, (1 \leqslant j \leqslant \lambda). \quad (1)$$

where $k = 0/1$ and $1 \leqslant i \leqslant \min(C_o, t)$. More precise estimate of **R** can be achieved by averaging the impact of noise, multi-path and other channel factors. Besides, by appending postamble, collision-free chips of B also exist in the packet tail. In Fig. 5, $\mathbf{R}_A[1]$ for A's chip 1 and $\mathbf{R}_B[0]$ for B's chip 0 can be obtained.

Collision with sample offset. When there exists sample offset between packets from two-TX, we observe that in a collision, one chip of A is overlapped with two parts of sequential B's chips as shown in Fig. 3. It is the same for B's overlapped chips. Denoting the result of overlap by a sequence of 'BAB' or 'ABA', each overlapped chip of A or B can be labeled with a specific chip array. In Fig. 5, we can abstract known overlapped chips from pre- and postamble of A and B, e.g., $A[8]$ with label $'000'$ and $B[1]$ with label $'111'$. After that, *3-overlapped chips' reference chips* ($\mathbf{R^3}$) can be calculated by averaging known overlapped chips of the same chip array. Denoting V_i as the value of the chip array of $A[i]$, $\mathbf{R}_A^3$ can be computed as follows:

$$\mathbf{R}_A^3[k][j] = \frac{\sum_{V_i=k} A[i][j]}{\sum_{V_i=k}}, (0 \leqslant k \leqslant 7, 1 \leqslant j \leqslant \lambda). \quad (2)$$

where $C_o + 2 \leqslant i \leqslant t$ or $L - t + C_o + 2 \leqslant i \leqslant L$. In Fig. 5, $\mathbf{R}_A^3[0]$ and $\mathbf{R}_A^3[6]$ can be estimated by A's overlapped chip $A[8]$, $A[3]$ which are labeled $'000'$, $'110'$.

For a two-packet collision with sample offset, there are 8 combinations of $\mathbf{R^3}$ for A and B respectively as shown in Table. I. When $\mathbf{R^3}$ or **R** are incomplete, we can combine existing $\mathbf{R^3}$ and **R** together based on the sample offset. $\mathbf{R}_A$ of chip 0 or 1 can be abstracted by subtracting two partial $\mathbf{R}_B$ from existing $\mathbf{R}_A^3$ as shown in Fig. 8. This procedure can be formulated as follows:

TABLE I
BIT SEQUENCE OF OVERLAP COMBINATION WITH SAMPLE OFFSET.

Bob	0	0	0	0	1	1	1	1
Alice	0	0	1	1	0	0	1	1
Bob	0	1	0	1	0	1	0	1
BitSeq	000	001	010	011	100	101	110	111

TABLE II
BIT SEQUENCE OF OVERLAP COMBINATION WITH ONLY CHIP OFFSET.

Alice	0	0	1	1
Bob	0	1	0	1
BitSeq	00	01	10	11

$$\mathbf{R}_A[k[1]][j] = \begin{cases} \mathbf{R}_A^3[k][j] - \mathbf{R}_B[k[0]][l], \\ \quad (1 \leqslant j \leqslant S_o, \lambda - S_o + 1 \leqslant l \leqslant \lambda); \\ \mathbf{R}_A^3[k][j] - \mathbf{R}_B[k[2]][l], \\ \quad (S_o + 1 \leqslant j \leqslant \lambda, 1 \leqslant l \leqslant \lambda - S_o). \end{cases} \quad (3)$$

where $'k[0]k[1]k[2]'$ denotes the chip array of $\mathbf{R}_A^3[k]$.

Leveraging existing $\mathbf{R}_A^3[0]$ and $\mathbf{R}_B[0]$, $\mathbf{R}_A[0]$ can be obtained as depicted in Fig. 8. And the average value of abstracted $\mathbf{R}_A$ representing the same chip will be calculated as the final result. After that, any combinations of $\mathbf{R}_A^3$ can be generated by the overlay of $\mathbf{R}_A$ and $\mathbf{R}_B$ as follows:

$$\mathbf{R}_A^3[k][j] = \begin{cases} \mathbf{R}_A[k[1]][j] + \mathbf{R}_B[k[0]][l]), \\ \quad (1 \leqslant j \leqslant S_o, \lambda - S_o + 1 \leqslant l \leqslant \lambda); \\ \mathbf{R}_A[k[1]][j] + \mathbf{R}_B[k[2]][l]), \\ \quad (S_o + 1 \leqslant j \leqslant \lambda, 1 \leqslant l \leqslant \lambda - S_o). \end{cases} \quad (4)$$

As shown in Fig. 9, unknown $\mathbf{R}_A^3[1]$ can be generated by the overlay of $\mathbf{R}_B[0]$, $\mathbf{R}_A[0]$ and $\mathbf{R}_B[1]$.

Since $\mathbf{R^3}$ having the same chip sequence is the most similar one with the real overlapped chip as shown in Fig. 5, we can match chip sequence of unknown overlapped chips by known $\mathbf{R^3}$. After all combinations of $\mathbf{R^3}$ are obtained from existing data, we can calculate the root mean square error (RMSE) between unknown overlapped chips $O[i]$ and $\mathbf{R^3}$ as follows:

$$RMSE(O[i], \mathbf{R^3}[k]) = \sqrt{\frac{\sum_{j=1}^{\lambda}(O[i][j] - \mathbf{R^3}[k][j])^2}{\lambda}}. \quad (5)$$

Based on equation (5), we can get the most similar $\mathbf{R^3}$ for a given overlapped chip, whose RMSE is the smallest.

Collision without sample offset. When there is no sample offset, we also abstract known collision-free chips according to chip offset and pre- or postamble length. In Fig. 4, all kinds of **R** of A and B can be estimated using collision-free chips $A[1], A[2]$ and $B[7], B[8]$. Besides, chips from A and B are aligned since there is no sample offset. We can get *2-overlapped chips' reference chips* ($\mathbf{R^2}$). Each $\mathbf{R^2}$ can be labeled with a chip array $'AB'$. For a two-packet collision without sample offset, there are 4 kinds of $\mathbf{R^2}$ totally as shown in Table. II. However, not all combinations of $\mathbf{R^2}$ can be obtained directly and we can only acquire $\mathbf{R^2}[0]$ and $\mathbf{R^2}[3]$ in Fig. 4. Leveraging existing **R** and $\mathbf{R^2}$, unknown **R** can be abstracted as follows:

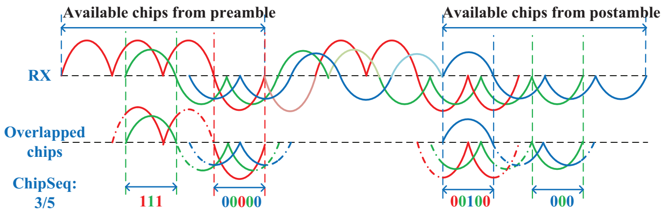


Fig. 10. Decode three-packet collision with chip and sample offsets in PPM.

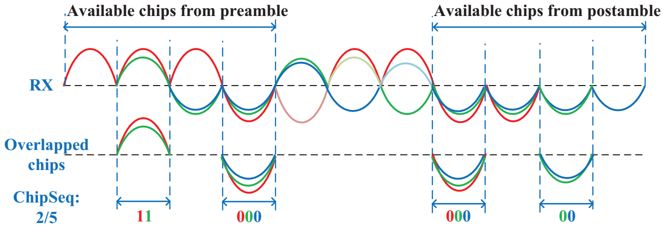


Fig. 11. Decode three-packet collision with only chip offset in PPM.

$$\mathbf{R}_A[k[0]][j] = \mathbf{R}^2[k][j] - \mathbf{R}_B[k[1]][j], (0 \leqslant j \leqslant \lambda). \tag{6}$$

where k is the value of $\mathbf{R}^2[k]$'s chip array and $'k[0]k[1]'$ denotes the corresponding chip array. The rest combinations of $\mathbf{R}^2$ can be generated by the overlay of $\mathbf{R}$, which can be formulated as follows:

$$\mathbf{R}^2[k][j] = \mathbf{R}_A[k[0]][j] + \mathbf{R}_B[k[1]][j]). \tag{7}$$

where $0 \leqslant k \leqslant 3$ and $1 \leqslant j \leqslant \lambda$. On the basis of all combinations of $\mathbf{R}^2$, we can match collided packets chip-by-chip. Choosing the $\mathbf{R}^2$ which has the smallest RMSE for a specific overlapped chip, we can acquire a high accuracy estimation for the whole collision.

Further Improvement of PPM. In a standard ZigBee packet, there are 256 known chips in the preamble. We divide preamble into preamble and postamble of 128 chips in our design. Thus, it is of high probability that we can obtain all combinations of $\mathbf{R}^2$/$\mathbf{R}^3$ and complete $\mathbf{R}$ according to known chips in pre- and postamble, e.g., 4 combinations of $\mathbf{R}^2$ and 16 combinations of $\mathbf{R}^3$ for two-packet collision with or without sample offset. Otherwise, by combining available $\mathbf{R}^2$/$\mathbf{R}^3$ and $\mathbf{R}$, other combinations of $\mathbf{R}^2$/$\mathbf{R}^3$ can be generated as mentioned before. However, it is possible that there is no chip offset between multi-packet. Under such circumstance, we can decompose chips of pre- and postamble utilizing other methods, e.g., ZigZag [7] and mZig [8]. Leveraging these known decomposed chips from pre- and postamble, all kinds of $\mathbf{R}$ can be estimated. As a result, any combinations of $\mathbf{R}^2$/$\mathbf{R}^3$ can be generated by the overlay of $\mathbf{R}$ and the whole packet can be decoded via reference chip comparison.

B. Three-packet collision decoding

In this section, we show that how PPM decodes a three-packet collision.

We give two examples of three-packet collision as shown in Fig. 10 and Fig. 11. Leveraging detected time offset and pre- or postamble length, known collision-free and overlapped chips can be abstracted from pre- and postamble. Based on $A[1]$ and $C[9]$, $\mathbf{R}_A[1]$ for A's chip 1 and $\mathbf{R}_C[0]$ for C's chip 0 can be obtained. It is worth noting that five-chip overlaps also exist in pre- or postamble besides three-chip overlaps.

For a three-packet collision with sample offset, we observe that a overlapped chip of A consists of four parts from four different chips of B, C as shown in Fig. 10. Each five-chip overlap can be represented by a 5-bit chip array. By averaging known five-chip overlaps of the same chip array, $\mathbf{R}^5$ of five-chip overlaps can be acquired and there are 32 combinations of $\mathbf{R}^5$ for A, B and C respectively. Similarly, unknown $\mathbf{R}$ can be generated by subtracting four partial $\mathbf{R}$ of other chips from existing $\mathbf{R}^5$. Besides, $\mathbf{R}^5$ of five-chip overlaps can be not only formed by the overlay of $\mathbf{R}$, but also can be generated by overlaying $\mathbf{R}^3$ of three-chip overlaps with $\mathbf{R}$. On the basis of all combinations of reference chips, RMSE between the unknown overlapped chips and reference chips can be calculated. By choosing the reference chip which has the smallest RMSE for a specific overlapped chip, the whole collided packet can be decoded accurately.

When there exists no sample offset for a three-packet collision, we also abstract collision-free and overlapped chips. Different from collision with sample offset, each overlapped chip is aligned whose chip sequence can be denoted by a 3-bit array. Partial $\mathbf{R}^3$ and $\mathbf{R}$ can be calculated based on known chips. Similarly, by combining existing $\mathbf{R}$ and $\mathbf{R}^3$, all kinds of reference chips can be generated. After that, the whole collision packet can be decoded by comparing with reference chips.

C. Extend to multi-packet collision

In this subsection, we demonstrate that how to extend PPM to multi-packet collision scenarios.

The flow chart of multi-packet decoding is the same as two-packet scenarios as shown in Fig. 7. After abstracting available chips from pre- and postamble, we can estimate partial reference chips based on known collision-free and overlapped chips. For an n-packet collision, we have to deal with 2, 3, $\cdots$, n overlapped chips. After that, we try to abstract all kinds of reference chips by combing known reference chips. If sufficient reference chips cannot be obtained from existing data, we can seek to other methods, e.g., ZigZag [7], mZig [8], for help. We can abstract chip sequence of unknown overlapped chips by comparing known reference chips.

Cross-validation. For an n-packet collision with sample offset, we can decode the collision packet referring to overlapped chips' reference chips of different packets. For example, in three-packet collision with sample offset, we can decode three times referring to reference chips of A, B and C respectively. Thus, each decoded result can be used for cross-validation. As a result, we can achieve a more precise decoding result and the number of retransmissions for multi-packet collision is effectively reduced by our design.

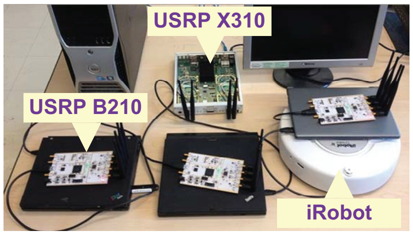
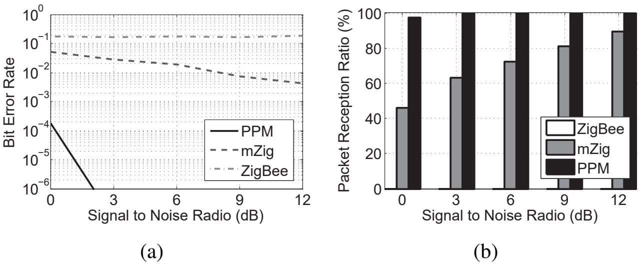


Fig. 12. Our USRP testbed includes three TXs and one RX.



(a)

(b)

Fig. 13. BER and PRR under different SNRs in two-packet collision.

IV. Evaluation

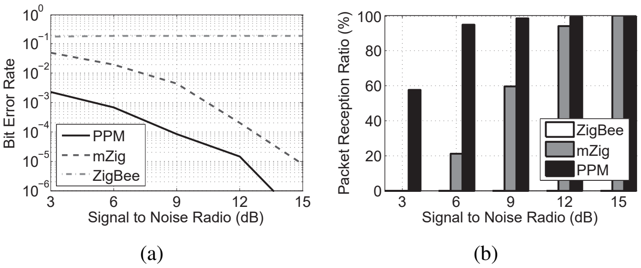
We implement PPM on USRPs and build a 4-node testbed as shown in Fig. 12. We conduct extensive experiments based on this testbed to compare the performance of PPM with existing methods.

A. Experiment Setting

We implement PPM RX in USRP X310, which is connected to a desktop to log the results. We develop PPM TXs in USRP B210s, which are portable and powered by a USB 3.0 port. These TXs are connected to laptops and carried by mobile nodes (i.e., iRobot Roomba). Our experiments are conduct in an office environment with both static and mobile tests.

We compare PPM's performance with the standard ZigBee and mZig. The standard ZigBee is used for unicast, which adopts CSMA/CA to avoid collision and cannot resolve multi-packet collision. mZig [8] resolves the collision using physical layer features but its error will be accumulated with the length of packet. The main purpose of our experiment is to test whether PPM can improve the accuracy of multi-packet collision and reduce retransmission overhead for green communication. We evaluate PPM with three metrics: bit error rate (BER), packet reception ratio (PRR), and packet retransmission times (PRT).

In our experiment, the number of samples in one chip λ is 32, based on the ADC in USRP. Time offset in a collision is random according to packet arrivals at TX. We randomly generate equal length packets with the same preamble and postamble of 16-bit 0s. The payload is 1000 bits. To estimate the overall performance and impact of different Signal to Noise Radio (SNR), we inject noises and conduct 200 runs for every experiment. In addition, we set a threshold of BER as 10^{-3}, below which we consider that a packet is decoded correctly.



(a)

(b)

Fig. 14. BER and PRR under different SNRs in three-packet collisions.

B. Experiment Result

Two-packet collision BER and PRR. Fig. 13 shows BER and PRR comparison of PPM with mZig and ZigBee under different SNRs in two-packet collision decoding. PPM achieves a PPM-level (i.e., 10^{-6}) BER and more than 90% PRR even under very low SNR. As we can see in Fig. 13(a), 13(b), PPM outperforms mZig and ZigBee from both aspects of BER and PRR.

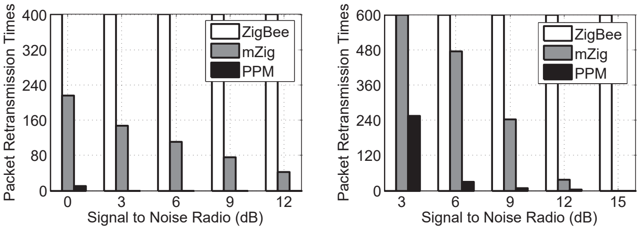
Three-packet collision BER and PRR. Fig. 14 shows performance of PPM and mZig under different SNRs in three-packet collision decoding. We find that the performance of PPM degrades, e.g., BER increases 1 or 2 order of magnitude and PRR decreases 10% on average, compared with two-packet collision scenarios. However, BER of PPM still has 1 or 2 order of magnitude better than mZig. Besides, PRR of PPM can maintain more than 90% when SNR$\geqslant$ 6dB. This experiment presents that PPM is a general solution for two- three- and multi-packet collision.

PRT of two- and three-packet collision. Fig. 15 shows the PRT needed for 200 transmissions in two and three packets collision scenarios. For two-packet collision, the PRT of PPM is less that 20 as shown in Fig. 15(a). In three-packet collision, the PRT of PPM still has a huge gap with that of mZig and standard ZigBee as shown in Fig. 15(b).

Evaluation Summary. According to the experimental result, we see that PPM accurately decodes multi-packet collision and reduces the energy consumption for retransmissions. Different from existing methods, e.g., mZig [8] and ZigZag [7], PPM decouples current chip or chunk from previous ones and decoding each chip or chunk independently without cumulative error. From the collision resolution side, PPM outperforms the advanced mZig in terms of BER and PRR, usually one or two order of magnitude lower than mZig in BER. From the green communication side, PPM can maintain high PRR and low PRT in most cases. By reducing PRT, PPM reduces the energy consumption due to retransmissions significantly.

V. Related Work

Collision is inevitable in concurrent transmissions and there are a lot of studies aiming at solving this problem. Existing research wants to deal with collision problem from different views of aspects and can be roughly divided into two types.



(a) Two-packet collision.

(b) Three-packet collision.

Fig. 15. PRT under different SNRs in two- and three-packet collision to indicate the energy saving.

CSMA/CA [6] is a typical example of collision avoidance. However, CSMA/CA is unavailable in hidden terminal scenarios. Besides, CSMA/CA will lead to delay because of exponential backoff algorithm and retransmission when collision happens [10] [17]. Apart from retransmission, channel contentions in CSMA/CA are another major source of extra energy overhead [12]. As a complementary strategy of CSMA/CA, RTS-CTS [9] is proposed to deal with hidden station problem based on handshake. However, RTS/CTS CSMA/CA takes additional channel and power overhead for which it it not recommended in wireless networks [11]. Existing methods try to reduce power consumption by optimizing the CSMA/CA mechanism [13] [16].

Different from collision avoidance, the state-of-the-art method, e.g., mZig [8], ZigZag [7] try to address collision by decomposing collided packet. ZigZag [7] utilizes the different time offsets between packet retransmissions. Thus, the ZigZag's throughput is limited. mZig [8] makes use of ZigBee's waveform and amplitude to decompose collision from header to tail. However, mZig falls into high accumulative error and has poor performance under inferior channel conditions. Even though there exist some limitations, collision resolution strategy can reduce energy consumption by eliminating the overhead of CSMA/CA and decreasing retransmissions.

Compared with mZig, PPM can achieve higher decoding accuracy which further reduces the number of retransmissions. As long as there exist known chips in preamble or postamble and the base band signal is almost identical, PPM can be used for collision decoding and green communication in ZigBee and other wireless technologies. Furthermore, PPM can take advantage of mZig or ZigZag as a complementary method for collision resolution.

VI. Conclusion

We present PPM, a physical layer design for collision recovery and green transmission in ZigBee. Leveraging known preamble and attached postamble, PPM can resolve multi-packet collisions without accumulative error by comparing reference chips generated from collided packets. Thus, PPM can decode collided packets with a relatively high accuracy which reduces energy consumption of CSMA/CA contention and packet retransmissions simultaneously. Experiment results based on USRP testbed show that PPM can achieve ppm-level BER (i.e., 10^{-6}), which is superior compared with the-state-of-art method, e.g., mZig, and standard ZigBee.

PPM has been implemented in ZigBee but not limited to ZigBee. As long as the reference chips can be obtained from overlapped packets, we can extend PPM to other wireless technologies' accurate and green collision recovery, e.g., Bluetooth Low Energy (BLE).

Acknowledgment

This work is partly supported by National Key Research and Development Program (2016YFE0100600), NSFC (61672349, 61672353, 61472252), and China 973 project (2014CB340303).

References

[1] ZigBee Introduction. http://www.zigbee.org

[2] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay. Towards the Implementation of IoT for Environmental Condition Monitoring in Homes. IEEE Sensors Journal, 2013.

[3] R. Sundar, S. Hebbar and V. Golla. Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection. IEEE Sensors Journal, 2015.

[4] E. Ronen, A. Shamir, A. O. Weingarten, and C. OFlynn. IoT Goes Nuclear: Creating a ZigBee Chain Reaction. IEEE Symposium on Security and Privacy, 2017.

[5] Z. Li and T. He. WEBee: Physical-Layer Cross-Technology Communication via Emulation. ACM MOBICOM, 2017.

[6] L. Kleinrock and F. Tobagi. Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. IEEE Transactions on Communications, 1975.

[7] S. Gollakota, D. Katabi. ZigZag decoding: combating hidden terminals in wireless networks. ACM SIGCOMM, 2008.

[8] L. Kong, X. Liu. mZig: Enabling Multi-Packet Reception in ZigBee. ACM MOBICOM, 2015.

[9] F. Tobagi and L. Kleinrock. Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution. IEEE Transactions on Communications, 1975.

[10] E. Ziouva and T. Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. Computer Communications, 2002.

[11] J. L. Sobrinho, R. de Haan, and J. M. Brazio. Why RTS-CTS is not your ideal wireless LAN multiple access protocol. IEEE WCNC, 2005.

[12] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan and W. Dehaene. Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives. IEEE DATE, 2005.

[13] Y. Zhang and F. Shu. Packet Size Optimization for Goodput and Energy Efficiency Enhancement in Slotted IEEE 802.15.4 Networks. IEEE WCNC, 2009.

[14] B. Nour, K. Sharif, F. Li and H. Moungla. A Distributed ICN-Based IoT Network Architecture: An Ambient Assisted Living Application Case Study. IEEE GLOBECOM, 2017.

[15] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. ACM MOBICOM, 2015.

[16] J. Kwak, C. Lee and D. Y. Eun. A High-Order Markov-Chain-Based Scheduling Algorithm for Low Delay in CSMA Networks. IEEE/ACM Transactions on Networking, 2016.

[17] W. Du, J. C. Liando, H. Zhang, and M. Li. When Pipelines Meet Fountain: Fast Data Dissemination in Wireless Sensor Networks. ACM SenSys, 2015.

[18] Z. Zhou, F. Xiong, C. Xu, Y. He and S. Mumtaz. Energy-Efficient Vehicular Heterogeneous Networks for Green Cities. IEEE Transactions on Industrial Informatics, 2018.

[19] Z. Zhou, C. Gao, C. Xu, T. Chen, D. Zhang and S. Mumtaz. Energy-Efficient Stable Matching for Resource Allocation in Energy Harvesting-Based Device-to-Device Communications. IEEE Access, 2017.

[20] C. Xu, J. Feng, Z. Zhou, Z. Chang, Z. Han and S. Mumtaz. Two-Stage Matching for Energy-Efficient Resource Management in D2D Cooperative Relay Communications. IEEE GLOBECOM, 2017.