PRComm: Anti-Interference Cross-Technology Communication Based on Pseudo-random Sequence

Wei Wang
Dingsheng He
Wan Jia
Xiaojiang Chen
wwang@nwu.edu.cn
hedingsheng@stumail.nwu.edu.cn
jiawan@stumail.nwu.edu.cn
xjchen@nwu.edu.cn
Northwest University, Northwest University IoT Research Center, International Joint Research Centre for the Battery-Free IoT
Xi'an, China

Tao Gu
tao.gu@mq.edu.au
Macquarie University
Sydney, Australia

Haiyan Liu
H.Liu1@leeds.ac.uk
University of Leeds, Alan Turing Institute
Leeds, United Kingdom

Xiaoyang Sun
scxs@leeds.ac.uk
University of Leeds
Leeds, United Kingdom

Guannan Chen
503_Cguannan@stu.xupt.edu.cn
Xi'an University of Posts and Telecommunications
Xi'an, China

Fuping Wu
fpwu@xidian.edu.cn
Xidian University
Xi'an, China

ABSTRACT

With the rapid development of the Internet of Things (IoT), we have seen a larger number of devices deployed with different wireless communication protocols (i.e., WiFi, ZigBee, Bluetooth). Working in the same place opens a new opportunity for these devices to communicate directly with each other, leveraging on Cross-technology Communication (CTC). However, since these devices operate in the same frequency band which results in the competition against each other for network resources, severe interfere may arise. In this paper, we explore pseudo-random sequence (PR sequence) to design a novel CTC protocol that enables low-cost direct communication between WiFi and ZigBee in noisy indoor environments. Pseudo-random sequence offers a unique statistical feature to accomplish both information transmission and synchronization between heterogeneous devices. We design a dynamic synchronous decoding strategy to handle interference coexisted among different wireless protocols. Our system does not require any modification of communication protocol and underlying hardware and firmware. We implement our system on commercial devices (Intel 5300 WiFi NIC and MicaZ CC2420), and conduct extensive experiments to evaluate the system performance in three typical scenarios. The experimental results show that the synchronization time of our approach is lower than 0.5 ms, and the accuracy is greater than 84% while the channel occupancy is as high as 50%.

IPSN'2021, May 18-21, 2021, Nashville, Tennessee, USA

ACM ISBN 978-1-4503-8098-0/21/05…$15.00
DOI: 10.1145/3412382.3458264

CCS CONCEPTS

• **Networks** → Wireless Networks.

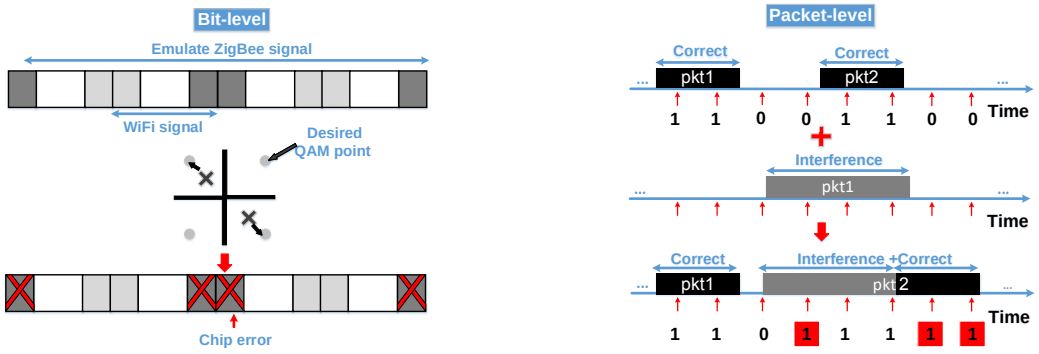
KEYWORDS

Cross-Technology Communication, coexistence interference, related-coding, pseudo-random sequence

ACM Reference Format:
Wei Wang, Dingsheng He, Wan Jia, Xiaojiang Chen, Tao Gu, Haiyan Liu, Xiaoyang Sun, Guannan Chen, and Fuping Wu. 2021. PRComm: Anti-Interference Cross-Technology Communication Based on Pseudo-random Sequence. In *IPSN'2021:International Conference on Information Processing in Sensor Networks,May 18-21, 2021,Nashville, Tennessee, USA.* ACM, New York, NY, USA, 13 pages.

1 INTRODUCTION

Recent years have witnessed the rapid growth of the Internet of Things (IoT)[15] in which a large number of devices are connected for a range of applications such as smart home, smart city, precision agriculture, etc. These devices are connected via different wireless protocols such as WiFi, Bluetooth, and Zigbee. Since they operate in the same ISM (Industrial Scientific Medical) band, severe interference may occur in the same channel, degrading their communication performance. On the other hand, operating in the same band also offers these devices an opportunity to communicate with each other directly, which could potentially open a door for new



(a) Bit-level (b) Packet-level

Figure 1: The influence of interference to CTC.

applications. For example, a WiFi-enabled smartphone can be used to control a ZigBee device for lighting control in smart home environments. To facilitate communication between wireless protocols, Cross-Technology Communication (CTC) has been emerging in recent years, providing a promising solution to the coexistence of different wireless technologies. CTC enables direct communication between different wireless protocols, and eventually improves the performance of the network.

Existing CTC methods mainly work at packet-level or bit-level. For packet-level methods [2, 4, 9, 10, 17, 19, 25, 26, 37, 38, 40–43], data transmission essentially depends on the external characteristics of packet such as length, quantity, transmission time, transmission rate, and packet law. As for bit-level methods [8, 11, 16, 24, 27–29], payload manipulation usually is required to embed packets from one wireless protocol to another. For example, ZigBee packets in the time domain can be simulated and incorporated by the payload of WiFi packets.

In general, existing CTC methods can solve the problem of interoperability between different protocols with high throughput. For example, WEBee [28] proposes a physical-layer CTC which can achieve a comparable data rate as that in ZigBee. However, CTC technology still facing many challenges. For example, how to cope with strong interference in real application scenarios and how to be able to work even when the underlying multi-access protocol is working properly. How to relax the binding between the communication process and the underlying protocols in certain application scenarios, which could make it more easily to be implemented and transplant to different hardware platforms. In addition, synchronization between devices remains a big challenge for applications.

In reality, it is essential to deploy CTC technology into different real scenarios and a large number of heterogeneous devices without any software or hardware modification. In this way, it will be easier to explore CTC technology for new applications. But real CTC applications can be complicated, i.e., many different devices coexist and operate on their unique protocol. Signals transmitted over the same band in the air will interfere [7][13] with CTC receivers, no matter they are working at packet-level or bit-level [6, 12, 20, 21].

In this paper, we propose PRComm base on pseudo-random sequence, a novel packet-level, interference-resistant method to enable cross-technology communication in coexisting environments. PRComm does not require any modification to the communication protocols and hardware of existing devices. PRComm essentially leverages pseudo-random sequence to combat severe interference in the same channel, and at the same time facilitate synchronization between devices. To resolve the time errors caused by CSMA (Carrier Sense Multiple Access) [22], we propose a dynamic synchronous decoding strategy based on identifiable coding features to improve immunity for time errors.

The major contributions of PRComm are following:

(1) We propose a novel CTC method to enable stable communication in coexisting environments. This method is both interference-resistant and low-cost.

(2) PRComm enables dynamic synchronization without the need for any preamble packets. We schedule packets based on pseudo-random sequences to present different information, and combine synchronization and communication so that the receiver can decode the sender's information successfully while keeping synchronized.

(3) We develop a prototype system using off-the-shelf WiFi and ZigBee devices, and conduct comprehensive evaluations in different scenarios–mild interference meeting room, moderate interference corridors, and severe interference labs. Results show that even the channel occupancy rate achieves as high as 50%, the reliability is still above 84%, and the synchronization time is less than 0.5ms.

2 MOTIVATION

A CTC receiver is usually more vulnerable to interference than a normal receiver. As shown in Fig. 1-(a), some bit-level CTC methods need to select the nearest QAM points to emulate ZigBee signals in the special part of a WiFi frame. It puts forward requirements for both accurate synchronization and signal demodulation, i.e., since the information only can be obtained with accurate extraction of the specific part of the frame, the anti-interference ability of the signal itself may degrade rapidly.

In packet-level CTC methods, receivers do not try to understand a packet bit by bit, instead obtain the information through the external characteristics of the signal, but these external characteristics are vulnerable. As shown in Fig. 1-(b), an interference packet preempts the position of a CTC packet, which will cause a delay and lead to changes in sequence characteristics.

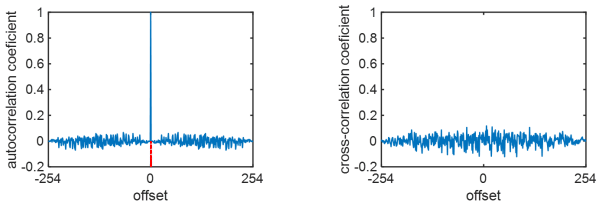
The interference is serious, but unfortunately existing CTC methods do not have a complete error control protocol to trace and fix the errors, and could not benefit from the error correction methods that work on the signal level. In a typical CTC application scenario, multiple terminals work together with different wireless network standards. In the case of multiple access, common data collision may not only result in the loss of the current CTC packet but also destroy the CTC sequence, which is essential for the correct decoding of the receiver. For a CTC terminal, it is difficult to retrieve these failures (i.e., no matter for one packet or the sequence of the packets) by protocols like Automatic Repeat Request (ARQ), hence affecting communication efficiency seriously. How to ensure the effective communication process in such a harsh situation without the coordination of multiple-layer protocols is a big challenge.

Besides, synchronization is also difficult to achieve in CTC solutions, which may suffer the same damage from interference like the data. Synchronization can be typically established by sending some special signals before transmitting data. Once destroyed by interference, the system can only restore synchronization by running the process from the beginning. It will greatly reduce the efficiency of CTC communication, and as compared with signal-level methods, it will cost much more channel resources.

From the above analysis, we can see CTC approaches, regardless of the level at which they work, are facing the challenge of interference from real application scenarios. Therefore, it is a great challenge for us to choose a suitable way to implement CTC and address these practical problems with minimal cost under the condition that the transmission capacity is limited and without the assistance of a complete error control system.

To ensure the generic of the system, our method should make no change to the hardware but can be easily applied to the existing commercial equipment directly. So the CTC method based on the packet level is a preferred option. However, this approach has inherent problems such as susceptibility to interference, low communication efficiency, and difficulty in synchronization, so it is a great challenge for us to effectively circumvent these shortcomings before taking full advantage of the zero-modification to the original facilities. This paper is strongly motivated by this vision, so we aim to improve the communication quality of heterogeneous devices in co-existing environments, not only the anti-interfere performance of data transmission but also that of synchronization.

To address the aforementioned challenges, we are inspired by pseudo-random sequence, for its nice feature of high auto-correlation and low cross-correlation. As illustrated in Fig. 2-(a), the sequence has a correlation of "1" (i.e., the maximum) when the offset is "0". Otherwise, the correlation will be much less than "1". This implies that even the interesting data has been flooded by noise, the receiver can still extract it by calculating the correlation function with the same PR sequence as the sender to see if it is "1". Therefore, base on a proper pseudo-random sequence n, we can construct the packet sequence for the sender and then retrieve the information



(a) Autocorrelation (b) Cross-correlation

Figure 2: When the offset is 0, the auto-correlation coefficient is 1, but cross-correlation does not have this property.

at the receiver. Since interference doesn't follow the rule of this sequence, its correlation function will be "0", which can be eliminated, as shown in Fig. 2-(b). In addition, the pseudo-random sequence brings the benefit of achieving synchronization at the same time. To enable CTC, the receiver needs not only to use the same random sequence but also to ensure that the offset is "0". This can be used as an indication that if the two sides are synchronized or not without any prior requirements such as preamble [33] and exclusive channel [39]. Therefore, we can achieve both CTC communication and synchronization by using a pseudo-random sequence.

Applying the pseudo-random sequence is challenging. What we want to achieve is packet-level CTC, so we need to form a packet sequence based on a pseudo-random sequence. However, we know that the longer the pseudo-random sequence, the stronger the anti-interference ability, but this also means a decrease in throughput. Therefore, we need to choose the best packet and sequence length to achieve a balance between anti-interference performance and throughput. In addition, how to achieve data recovery at the receiver with the underlying access protocol working is also a major challenge for us.

3 RELATED WORK

3.1 Packet-Level CTC methods

Packet-level CTC enables direct communication between heterogeneous devices, without modifications to the hardware. FreeBee [30] firstly folds a period of periodic beacon frames to find the reference time, synchronizes heterogeneous devices, and then combines the reference time to modulate the symbol message by offsetting the beacon sending time backward. GSense [41] uses multiple energy pulses to form a preamble to improve synchronization accuracy. Esense [4] builds up an "alphabet set" with different sizes of packets, and the receiver decodes information by perceiving the energy pulse duration. DCTC [25], EMF [9] and C-Mose [38] slightly disrupt transmission time point of existing traffic and recombine them to become a recognizable radio energy pattern, both DCTC and EMF are synchronized by folding periodic beacon frames. WiZig [19] encodes information by changing the transmission power with multiple amplitudes. B^2W^2 [10] and ZigFi [17] overlap the Bluetooth and ZigBee packet, respectively, with part of the subcarrier data packet of the WiFi sender, and then decode information by analyzing CSI features at WiFi receiver. Crocs [40] triggers WiFi and ZigBee synchronization based on the correlation of beacon

barker-code, and then sends a timestamp-aligned clock. The literature [2] implements the communication process from BLE to WiFi based on energy pattern, also the synchronization process based on barker-code, and encodes the payload based on the binary maximum-length sequence. This paper has similarities with our idea, but differs from us in the specific approach of synchronization and communication, and does not use dynamic adjustment means. StripComm [43] uses the Manchester code to eliminate interference in the time domain to smooth communication. AdaComm [34] combines existing CTC methods with machine learning, a lightweight model based on Text-CNN is established at receiver to track the channel state, and then the decoding strategy and parameters are used to resist environmental interference and improve the reliability of decoding.

We can see, in general, packet-level methods achieve CTC by scheduling packet transmission time, packet length, transmission rate, and transmission power at the sender and decoding correspondingly at the receiver. Obviously, the extrinsic characteristics of these signals are difficult to maintain due to the interference because of the presence of other ends and the lower level access control protocols. The packet-level CTCs methods also have no extra resources for error detection and correction. Accurate synchronization is another important issue. Beacon folding [30] maybe the most widely used method for synchronization, which folds the periodic beacon frame and applies correlation based on the coding rule. Before communicating, it firstly sends multiple beacon frames for synchronization and then repeats the process again if it loses synchronization due to the presence of other terminals. Obviously, sending a large number of beacon frames is time-consuming and communication-inefficient, as it wastes network throughput.

In summary, packet-level CTC methods have several drawbacks as follows.

1) Their ability of anti-interference is limited, as they may be easily interfered by other wireless signals in the same frequency band.

2) Their network throughput is limited, since one or more packets might be used to represent a bit, which leads to a much lower information transmitting rate than the original code rate.

3) Synchronization between heterogeneous protocols is a big challenge. Actually, to ensure strict synchronization, people need a considerable number of redundant packets, and the severe the interference is the more number of redundant packets is needed.

4) They may experience poor stability in the temporal property of packets. For example, WiFi adopts the CSMA protocol. So with coexistence of devices, WiFi sender could not transmit data packets according to the time pattern precisely, which can cause decoding to fail.

3.2 Bit-Level CTC methods

Bit-level CTC methods, a.k.a. physical-level, may operate based on signal simulation. WeBee [28] utilizes the payload of four WiFi frames to simulate a complete 802.15.4-compliant ZigBee frame in the time domain. PMC [11] uses channel overlap to support parallel CTC between heterogeneous devices. Bluetooth in XBee [24] interprets the ZigBee packet from the bit patterns obtained from the BLE receiver and implements cross-decoding. LEGO-Fi[18] uses ZigBee data packets to leave distinguishable features when passing through the WiFi module, and then uses downsampling technology to bridge the bandwidth gap between ZigBee and WiFi, and finally matches the filter to identify the phase shift sequence and complete the cross decoding. Passive-ZigBee [27] designs a low-power backscatter radio as a bridge between heterogeneous devices and converts WiFi signals directly to identifiable ZigBee packets. LongBee [29] concentrates WiFi TX power by down-clocking at the transmitter and utilizes an innovative transition coding at the receiver to improve its communication range. WIDE [16] proposes digital emulation based on ZigBee phase-shift decoding. It uses a square wave as a basic unit to generate a simulate waveform in order to overcome the limitation of the strict sine wave, thus improves PRR (Packet Reception Ratio). TwinBee [8] employs two symbols to form a byte, which improves the quality of analog signal simulation and PRR. TwinBee moves a big step forward to improve network throughput, but some constraints remain. First, the cyclic prefix must be repeated and the demodulation frame correlation threshold of the ZigBee receiver must be reduced to increase the acceptance rate of the frame. Second, to avoid collision with pilot/space subcarriers of WiFi OFDM, ZigBee needs to be jammed into a small band. This implies that in the environment with more WiFi APs, the performance of a simulated signal will be limited.

From these studies, we can see the bit-level methods are not very easy to be deployed on every kind of communication terminal. Coupled with a precise signal modulation or localization and demodulation process to achieve signal understanding across protocols. This is contrary to our original intention of using existing commercial equipments to implement the CTC communication process. Therefore, this paper adopts a packet-level CTC approach but will take full advantage of the pseudo-random sequences to compensate for the disadvantages of such methods in terms of anti-interference, and at the same time significantly improve the efficiency of synchronization.

4 PSEUDO-RANDOM SEQUENCES

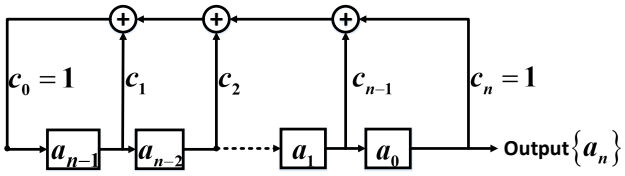


Figure 3: Nonlinear feedback shift register.

Pseudo-random sequences are generated by deterministic algorithms to simulate truly random sequences. They have many useful features including special auto-correlation function and cross-correlation function, they have been widely used in synchronizing, range-finding, and fault detection.

M-sequence is a typical pseudo-random sequence and has been widely used in many fields. For example, in the synchronization algorithm, we can apply the sharp correlation peak of an M-sequence

to identify synchronization points. Compared to traditional scrambling sequence-based algorithms, M-sequence could greatly improve synchronization performance in OFDM systems even with serious interference [36]. Modulating M-sequence with rotation factors in the time domain could promote the auto-correlation characteristic of Frequency-Shift, which could be used to achieve fast synchronization simply in conventional TDS-OFDM [44]. An iterative manner based on this property has been proposed to efficiently eliminate mutual interference among different antennas. An M-sequence based channel hopping algorithm was proposed to solve the problem of blind rendezvous in radio networks [31]. When nodes have a symmetrical channel set, this method can guarantee rendezvous without synchronization. A two-dimensional modified version of time-spreading/wavelength-group-hopping and embedded M-sequence code has been proposed to eliminate multiple-access interference to make codewords within the same group have a zero cross-correlation [3]. In this work, the cardinality and the BER of the synchronous system can be improved significantly. The correlation property of pseudo-random sequences does work well in improving synchronization and system robustness, which inspires us a lot.

The auto-correlation of M-sequence represents the dependence of the instantaneous value of a signal at two different time points. It provides a time-domain description of a random signal. Here we use the nonlinear feedback shift register as shown in Fig. 3 to generate M-sequence (i.e., a pseudo-random binary sequence), say $a_1, ..., a_N$. The sample auto-correlation function at offset j of the sequence is shown as follows:

$$R_a(j) = \frac{\sum_{i=1}^{N-j} a_i a_{i+j}}{\sum_{i=1}^{N} a_i^2} = \begin{cases} 1, j = 0 \\ -\frac{1}{N}, j \neq 0 \end{cases} \quad (1)$$

where j is the offset, right the displacement, and N is the sequence length. The sequence correlation we select is strictly binary. The auto-correlation value is "1" when there is no offset, otherwise, the value is small. The longer the sequence length, the better the value, even approximating zero. With this auto-correlation characteristic of the sequence, we can easily find out if the received data is the same as the local sequence and if there is an offset. We can thus efficiently combine the decoding and synchronization processes together.

Cross-correlation represents the degree of correlation between two different sequences, and it is a measurement of the similarity between them. Different from auto-correlation, the cross-correlation of two M-sequences, say $a_1, ..., a_N$ and $b_1, ..., b_N$, has no sharp binary property anymore but multi-valued. The cross-correlation calculation function of sequences a and b is shown as follows:

$$R_{a,b}(j) = \frac{\sum_{i=1}^{N-j} a_i b_{i+j}}{\sqrt{\sum_{i=1}^{N} a_i^2 \sum_{i=1}^{N} b_i^2}} \quad (2)$$

In summary, the M-sequence has two main properties: The auto-correlation has sharp binary characteristics as shown in Fig. 2-(a) and low cross-correlation with multi-valued, as shown in Fig. 2-(b). The low cross-correlation can be applied to resist interference well because badly interfered/contaminated signal does not have any similarity to the sender's sequence. Specifically, the auto-correlation at offset 0 will be "1" while the offset is "0", and it will be much smaller than 1 to present weakly correlation with non-zero offset. Such auto-correlation can be used to find out if the received data has an offset, i.e., whether the system has achieved synchronization, thus it can cope with both issues simultaneously.

5 OVERVIEW

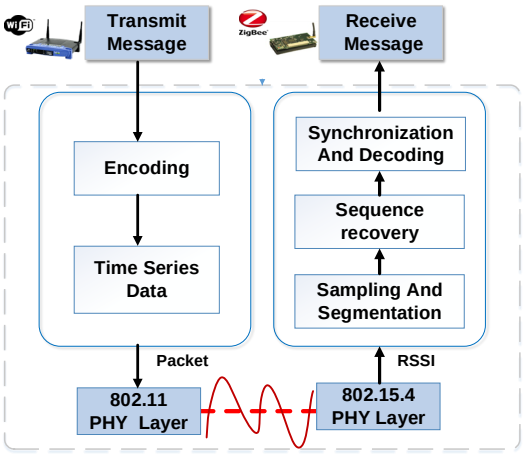


Figure 4: Overview of PRComm.

5.1 System Design

PRComm is built on the standards of 802.11.b and 802.15.4, so it could fully compatible with existing IoT devices without any hardware or firmware modifications. The system design of PRComm is shown in Fig. 4. The WiFi sender and the ZigBee receiver will be assigned an identical pseudo-random sequence in advance for CTC communication. The sender is a commercial WiFi device, and it creates the packet sequence according to the assigned sequence. In this sequence, "1" means there is a packet in the channel, and "0" means the channel stays idle for a while. Then the sequence of the packets will be sent out by the underlying hardware as normal.

The receiver is a commercial ZigBee terminal, which will continuously monitor the channel and detect the RSSI value of it to determine whether there are any packets or not. Judging the collected RSSI values by a threshold, we can recover them as a data sequence. With the pseudo-random sequence, the receiver could decode the original information by calculating the correlation between the received data sequence and the local sequence. However, restoring the information still facing several challenges including coexisting with other nodes in the same frequency band, the operation of the CSMA protocol, as well as the lower layer transmission process, which is not fully controllable in commercial devices. These issues may result in a great deviation in channel state judgment by the receiver. To address these issues, we firstly propose an ancillary step to segment the initial RSSI sequence and analyze the real starting point of the disturbing segment for recovering the correct sequence. The receiver then uses its local pseudo-random sequence to calculate the related functions of received data, execute

its synchronization process dynamically, and complete information decoding.

In CTC scenarios, coexistence of interference may be serious. The operation of the CSMA protocol will introduce inevitable random delays, especially when the channel is busy. Such delay may result in the fact that the sender is unable to transmit its packets strictly according to the time schedule. Since PRComm does not intend to modify the underlying hardware and the protocols of terminal devices, how to resist all kinds of external and internal interference factors in communication and how to restore the original data perfectly are challenges.

In existing CTC methods, synchronization and decoding have usually been separated into two independent steps: first achieving synchronization base on the transmission of some special signals, and then decoding the information. Therefore, how to complete the synchronization process without relying on special signals, to achieve lower cost and stronger adaptability is also a big challenge.

5.2 Design of Packets and Sequence

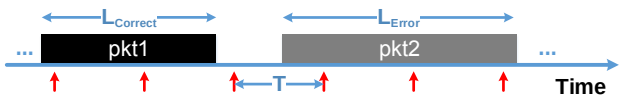


Figure 5: Discriminating interference using packet length, T is the sampling interval, $L_{Correct}$ and L_{Error} are correct packet length and interference packet length.

5.2.1 Control the characteristics of a packet . Our approach works at the packet-level, and it utilizes the special packet sequence to represent useful information. As mentioned before, the first issue is how long packets should be used to express information without consuming too much throughput. When the transmission rate of the sender and the sampling rate of the receiver are constant, the longer the data packet, the greater the probability that it will be detected correctly, but the lower the throughput. Therefore, it is crucial to adjust packet length to achieve the best trade-off between decoding rate and transmission rate.

We use a commercial ZigBee node CC2420 to evaluate various packet lengths in different scenarios. The maximum RSSI sampling rate of receiver CC2420 is 32KHz, in which the register is updated every 32μs, and the value is the average of 8 symbols (time-consuming 128μs). Obviously, if the receiver only detects (samples) once in a packet transmission interval, the packet may be easily corrupted by noise. We detect at least twice in a packet transmission interval, as shown in Fig. 5. In the worst condition (i.e., the receiver has just completed a sample, and the packet starts to be transmitted at the same time), it is still guaranteed to be sampled twice, so the packet length should not be less than twice the detection period. In PRComm, we choose two packet lengths, 150 (stable once) and 360 bytes (stable twice). We set up our experimental scenarios in a typical room with a channel occupation rate of more than 50%, as well as a laboratory with a channel occupation rate of less than 5%. Our experimental results show that, under severe CTI, the recognition rate of 360 bytes packets reaches 83%, while that of 150 bytes packets reaches 92%. Under mild CTI, the recognition rate of 360 bytes achieves 88%, however, the correct decoding rate of 150 bytes packets is only 50%. According to research [35][23][14], the packet size of the Internet usually follows bimodal mode as 40 bytes and 1500 bytes. Therefore, to obtain the best stability and distinguish from environmental interference, we set packet length as 360 bytes (i.e., sampled twice) for the best stability.

5.2.2 Construction of the packet sequence. With the right packet length, the next question is how to construct packets that can carry the sender's information while resisting interference. As mentioned earlier, we use a pseudo-random sequence to construct packets, containing both original and specific patterns. The coding enables the operations of both preamble and decoding tasks, and its ability of resistance to noise. Specifically, we intercept a pair of low cross-correlation segments from the sequence generated by the shift register, which is used to encode symbol "0" and "1", respectively.

5.3 Key Processes in PRComm

Synchronization: Achieving synchronization between the sender and receiver is always the premise of reliable communication. In particular, PRComm uses the pseudo-random sequence to realize a dynamic real-time synchronization process and has strong anti-interference capability at the same time. The system carries out shift convolution between the received RSSI sequence and the pre-allocated pseudo-random sequence. According to the autocorrelation and cross-correlation characteristics of the pseudo-random sequence, if the receiver receives with interference, then the cross-correlation will be 0; if the receiver receives without interference, then autocorrelation is small for non-zero offset; if the receiver receives without interference and with perfect synchronization, then received sequence is the same as the sending sequence and the autocorrelation is 1. So the peak position of autocorrelation indicates that the offset between the receiving sequence and the local sequence is 0and without interference, that is the perfect synchronization time.

Communication process: In CTC scenarios, interference is the biggest challenge for communication. So the key in PRComm is how to take full advantage of the relevant features of pseudo-random sequences to address this challenge. First, we need to eliminate the random delay introduced by the CSMA protocol and give full play to the anti-interference capability of the sequence. Second, we need to choose the sequence with the most suitable length and the best performance, to improve network throughput and eliminate interference.

Synchronization and communication operate simultaneously in our system, random delay and the selection of random sequence may affect their performance. We now analyze these problems in the following section.

6 DESIGN OF THE SEQUENCE

6.1 Choose Sequence Length

As mentioned above, the selection of packet length is crucial to trade off between recognition rate and efficiency. We first need to choose an appropriate sequence length for constructing multiple packets. A longer sequence can improve the ability of anti-interference, but

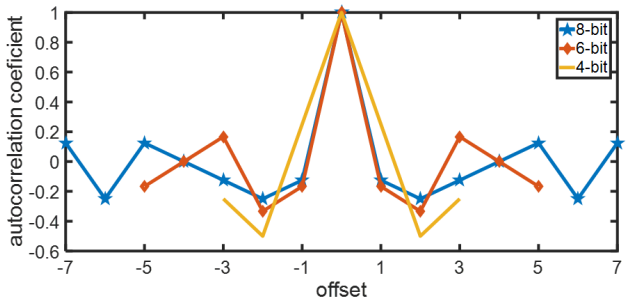


Figure 6: Correlation coefficients of different length symbols at different offsets.

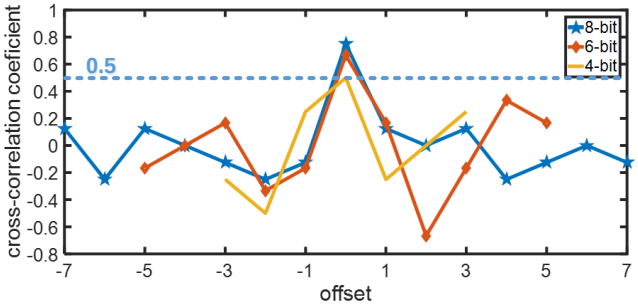


Figure 7: Correlation coefficients of symbols at different offsets with 1 bit error.

Table 1: Correlation Between x Bit Error Sequence and Ideal Sequence.

SYMBOL LENGTH	x Bit Error		
	0-Bit Error	*1-Bit Error*	*2-Bit Error*
8 bits	1	0.75	0.5
6 bits	1	0.66	0.33
4 bits	1	0.5	0

it will reduce communication efficiency. We seek a good balance between anti-interference ability and communication efficiency.

We perform multiple calculations on the correlation of sequences with different lengths. Fig. 6 shows that a longer sequence results in several desirable correlation properties. If the offset is not 0, the correlation function will return to zero more quickly. Fig. 7 shows the correlation between the received sequence and the local sequence with a bit error. It's obvious when the offset is 0, the correlation function of the long sequence will be larger, and it will returns to zero more quickly with the increase of the offset. This implies that the ability of long sequences to resist interference will also become stronger. Table 1 details how symbol correlation changes with different error rates. When the sequence length is 8, taking 0.5 as the threshold, it can resist two-bit errors. While the sequence length is 4, it even cannot defend against one-bit error.

The ability to resist interference depends on error rates in different application scenarios. We select proper sequence length based on the noise level in an application as we defined before. As we measured, the signal strength of every WiFi sender in three typical scenarios is shown in Fig. 9. In these scenarios, based on the Markov model, the system performs even in the presence of strongest interference, and a sequence of length 8 with the ability to resist two bits interference achieves good communication performance. Hence, we set 8 as the default sequence length in the remaining sections of this paper.

6.2 Optimize Sequence Decoding

On the receiving end of Zigbee, it continuously listens and records the RSSI value of the channel to determine if there is any data being transmitted, thus finally getting a sequence of packets. According to the CCA[1] mechanism, a channel is busy when the RSSI value is greater than -75 dBm and idle when the opposite is true.

However, WiFi and ZigBee both adopt CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). With this protocol, if the channel is busy, the CTC data cannot be sent on time; if the channel is idle, the packet schedule may still fail due to the random backoff procedure. In other words, the time interval between packets could be lengthened, and the sequence of packets becomes longer due to the addition of other terminal packets, thus breaking the operation of the sequence. As shown in Fig. 8, the black packet sequence in the top of the figure is the ideal sequence that was designed. While with the action of CSMA/CA, the packets with stripes from other nodes and some random delay will be inserted into the packet sequence. The real sequence has deviated far from its origin,as shown in the second line of Fig. 8. To facilitate the sequence with interference resistance in synchronization and communication processing, we design a preprocessing method as follows to optimize its performance.

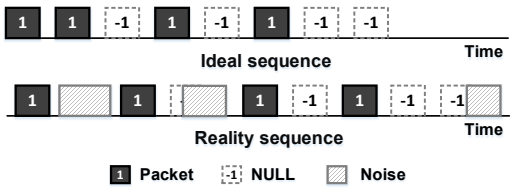


Figure 8: Effect of CSMA on the sequence.

The impact caused by the inserted data packet can be divided into two types, one is when it jumps in front of the original "1", and the other is when it appears at the position of "0". Out of the energy perception of CTC, "0" will be more fragile to interference than "1", and it will be falsely judged as "1". In this case, we could eliminate interference based on two characteristics. The first one is the energy level. When the transmitter power is settled, the received signal strength will not fluctuate too much. However, the interference signals come from other terminals with different transmitting power and different distances, so the probability of the signal strength being exactly the same as the transmitting end is not large. Second, in terms of packet length, only a few packets happen to be the same as ours. Base on these two rules, the wrong code can be recovered.

When the "1" error occur during channel competition, the channel is preempted by other users, then the other data will grab the

position of "1", results in a packet delay. In addition, because of the backoff delay, the interval between the CTC packet will become larger, which looks like a "0" is inserted. Dealing with this kind of errors, we could keep using the characteristics of the packet to discover the other packets and delete them. At the same time, we can adjust the position of the packet in the sequence according to the characteristics of the interval we designed. When the "0" error occur, we found that in many cases the position of "0" was not completely invaded, but only a part of it was occupied, and there are still some residues. Therefore, we adopted a compensation mechanism to save this incomplete "0" as a back code. If we detect a high degree of correlation in a suspicious sequence where an inserted data has been deleted, we will try to insert the back code into the sequence, and check the correlation again to find if it is the real sequence.

7 SYNCHRONIZATION AND COMMUNICATION

After choosing a suitable pseudo-random sequence, we will then implement the communication and synchronization process based on the computation of the correlation function. According to the previous introduction of pseudo-random sequences, the calculation of the correlation function having the ability of fault-tolerant. But if the interference data is inserted in the middle of the sequence, it will cause successive errors in the following sequence and lead to erroneous decoding. Unlike other communication proceedings, in CTC scenarios, maybe the competition for the channel from other nodes can cause such errors.

The lower layer of WiFi devices adopts the CSMA protocol to avoid collisions of data from different ends. For the CTC terminals, this method of listening to the channel and preempting or backoff may disrupt the pattern of the original packet sequence, resulting in error decoding. As mentioned before, the CTC sender needs to keep silent to represent the symbol "0", during this period, other terminals may detect the channel idle and send packets. When the length of this packet is shorter than the symbol "0", the packet may be misunderstood as symbol "1", and subsequent data will gain some listening delay, but the pattern remains the same. Conversely, when the length of this interfering packet is longer than symbol "0", not only makes the current data been wrong, but subsequent packets will incur a greater delay, resulting in a continuous misjudgment at the receiver.

To address these issues, the decoding process of PRComm should have a certain dynamic range that can accommodate the additional delays. Therefore we first investigate the magnitude of the possible delay under different levels of interference. Two commercial WiFi devices (equipped with Intel 5300 NICs) were used, one acting as the interference source, the other as the transmitter of the PRComm, and the ZigBee node as the receiver. The intensity of the interference packets was 100, 200, 400, and 500 per second, which were labeled as interference levels 1 to 4. In the absence of other interference signals, we repeated sending data at different interference levels 100 times to count the delay of PRComm packets.

The experimental results are shown in Fig. 10, we can see the presence of different levels of interference all bring additional delay to the packets. The average delay of the packets is 3.44T, 3.42T, 3.88T, 3.6T, and the total average value is about 3.5T. However, for level 1 and 2 interference, 62% of the packets are delayed within 2T. While for level 3 and 4 interference, only 28% of the packets are delayed within 2T, which means 72% of the packets have been delayed more than 256us. Based on the above experimental results, we increase the dynamic range of 2T when judging symbols in the receiver of PRComm, thus making the sequence have a certain fuzzy matching ability, which can further enhance the fault tolerance of the system and improve the throughput rate.

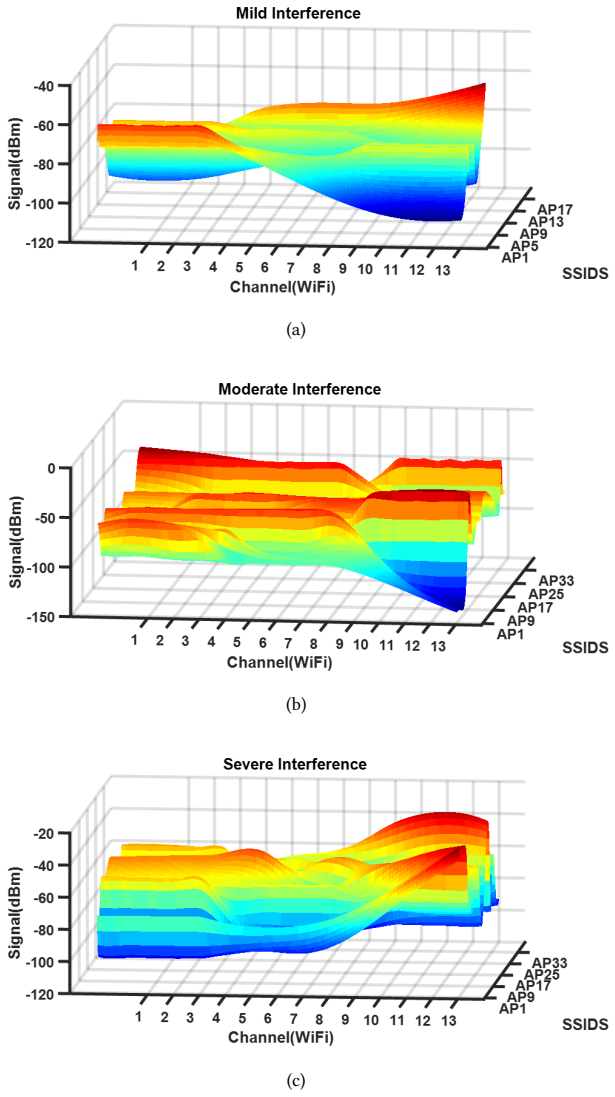


Figure 9: Energy Distribution of WiFi APs in mildly meeting room, moderately corridor and severely lab.

7.1 Process of Synchronization

To obtain synchronization, we first calculate the correlation of the received sequence. If the value is less than the threshold, keep

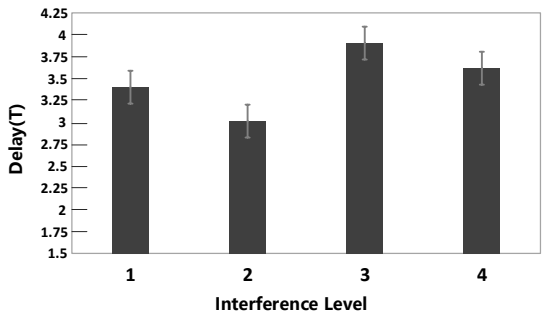


Figure 10: The packet delay caused by CSMA.

calculating after slide backward until the correlation value is greater than the threshold. However, we should note that the calculation is correct once does not represent real synchronization, it may just coincidence. Therefore, we call it the pre-synchronized state, and only when the next calculation result exceeds the threshold, we consider that the system has truly entered the synchronized state. So synchronization involves the following two processes, pre and real synchronization.

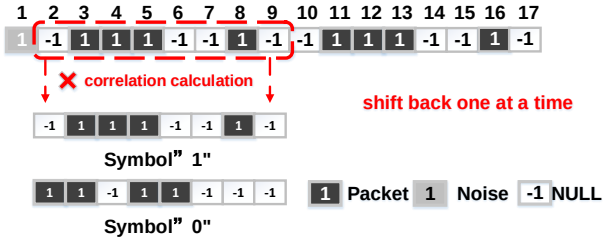


Figure 11: Shift correlation calculation.

Shift correlation calculation: As mentioned earlier, to achieve a compromise between anti-noise ability and efficiency, we chose a sequence of length 8. Therefore, at the receiving end, we set the decoding window length to 8, and bit by bit slides back on the received data stream. Then we calculate the correlation of the sequence falling in the window and the preset local sequence. As shown in Fig. 11. For example, let the 8-bit sequence in the window recorded as $a = \{a_1, ..., a_8\}$, and the symbols "0" and "1" recorded as $s_1 = \{s_{11}, ..., s_{18}\}$and $s_0 = \{s_{01}, ..., s_{08}\}$. Calculate the cross-correlation between a and, say, s_1 by using the formula $R_{a,s_1}(0) = \frac{\sum_{i=1}^{8} a_i s_{1i}}{\sqrt{\sum_{i=1}^{8} a_i^2 \sum_{i=1}^{8} s_{1i}^2}}$ if $R_{a,s_1}(0)$ is greater than the threshold, it is decoded into "1" at "offset" 0. Because only simple multiplication and addition calculations are required, our synchronization process is lightweight.

Pre-synchronization & Real-synchronization: Because of the uncertainty of coexistence interference, we cannot guarantee there is no interference sequence to break through the coding characteristics and exceed the correlation threshold. One high-level correlation is not enough to claim that synchronization has been achieved, which may be coincidental, we name this state pre-synchronization. To ensure the stability of PRComm, real-synchronization can only be achieved by exceeding the threshold twice in succession.

7.2 Ensuring the Synchronization

It is hard to complete synchronization between heterogeneous devices. The first challenge is that we don't want to modify the underlying layer and protocol. Moreover, it does not force the sender to transmit some special signals like a preamble to help the receiver recognize the arrival of the useful signal. The second challenge is that when we use the sequence of the packets to synchronize, the work level is very high. We cannot directly control the final sending process of the data packet, but only set the packet to be sent in the upper layer. Therefore, some unpredictable changes in the length and transmission time of the data packet may make it difficult to discover the parameter details in the process of synchronization.

To address the above problems, based on the related characteristics of the sequence itself and the sequence optimization process described in the previous section, most of the interference can be eliminated. However, errors that may be caused by coincidence cannot be eliminated. Therefore, we propose a state switching process from pre-synchronization to real-synchronization to deal with this problem. In the case of heavy interference, we send training sequences in advance to ensure faster and accurate synchronization.

Unlike other synchronization methods using preamble, PRComm features dynamic real-time synchronization. In the communication process, the receiving process of each information bit is both the decoding of the information and the re-confirmation of the synchronization. Therefore, whenever the decoding is incorrect, PRComm will immediately restart the synchronization process.

7.3 Process of Communication

The WiFi sender according to the content of the command customizes a string of symbols "1" and "0", then generates a series of packets according to the pseudo-random sequence, then sends them out in order. At the ZigBee receiver end, PRComm converts the data into an initial sequence according to the fluctuation of the energy. With optimizing the sequence dynamically, PRComm restores the initial sequence as much as possible, to reduce the impact of the interference and the random delay.

We only control the transmission order of the ideal sequence from the upper layer, but in the coexistence interference environment, the real transmission time of the packet has great randomness and uncertainty. Uncertain transmission delays result in differences between the received sequence and the true sequence sent by sender. To overcome the impact of these problems without changing the protocol is the main challenge we face.

After optimizing the sequence recognition, we slide the received sequence backward, according to the length of the coding window, and calculate the corresponding correlation between the received sequence and the sent sequence. When the correlation coefficient is higher than the threshold, a synchronization and decoding operation is completed. Specifically, in the light interference environment, we use 0.90 as the threshold. When the correlation coefficient is greater than 0.90, it is considered that the synchronization and decoding are achieved. The decoding threshold is appropriately

adjusted with the increase of interference to enhance the fault tolerance performance of the system. Moderately uses 0.6 as the threshold and severity as 0.4.

8 IMPLEMENTATION AND EVALUATION

We implement a prototype system on the commercial WiFi platform and the commercial ZigBee platform TinyOS. As the sender, the WiFi platform is a commercial development board equipped with an Intel-5300 AGN wireless network card, which has the advantage of being able to set the mode of operation compared to other commercial cards. So in our experiments, we simply set the card to the monitor mode, which allowed the card to send the packets according to the designed sequence to complete the communication process. At the receiver, the MicaZ cc2420 node is used to collect data at a sampling rate of 32KHz. We do not modify the hardware of it or the ZigBee protocol at any level. But simply invoke the channel monitoring function inherent in each node to identify the energy-level of the signal to obtain the information.

We use different levels of coexistence interference and select related coding lengths. The coding length under the mild interference condition is 4, 6 for the moderate condition and 8 under the severity scenario, and the corresponding symbol window is 2.37ms, 3.55ms, 4.74ms. Time series packets are transmitted according to symbols defined in Table 2.

Table 2: Coding in Different Interference Level.

SYMBOL	Interference Level		
	Mild	*Moderate*	*Severe*
1	-1 1 1 -1	-1 1 -1 -1 1 1	-1 1 1 1 -1 -1 1 -1
0	-1 -1 1 1	1 -1 -1 -1 -1 1	1 1 -1 1 1 -1 -1 -1

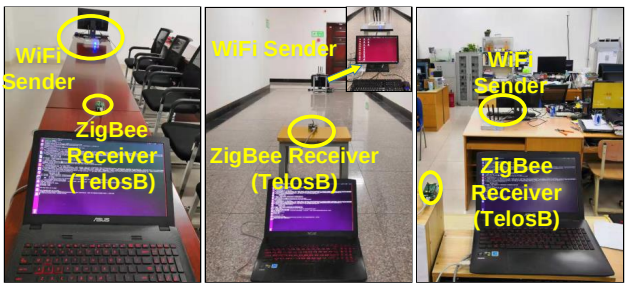


Figure 12: Experimental deployment from left to right, mildly interference meeting room, moderately interference corridor, severely interference lab.

8.1 Interference Analysis

Wireless signals are everywhere, providing convenience and challenges. Since the number and location of heterogeneous wireless devices might change, conflict avoidance can cause devices in the same physical space to interfere with different degrees. To evaluate PRComm, we deploy the prototype system in our campus building. We select three scenarios–the mild interference meeting room, the moderately disturbed indoor corridor, and the severe interference laboratory, shown in Fig. 12. We detect the WiFi AP in a lightly interfering meeting room (channel occupancy $< 5\%$), a moderately interfering corridor, and a severely interfering laboratory (channel occupancy $>50\%$). As shown in Fig. 9, the system detects 23, 38 and 40 APs, respectively. However, in the less sensitive room, only 8 APs with RSSI higher than -75dBm are detected, and there is almost no more than -65dBm, while 23 APs in the densely deployed laboratory exceed -75dBm. Since wireless LAN uses CSMA protocol to avoid collisions, energy detection (ED) is an important method for CSMA to determine whether the channel is busy. ZigBee node will indicate channel blocking while the channel energy exceeds -75dBm. Obviously, too many wireless devices make coexistence interference as a common and serious problem. Therefore, we chose to deploy intensive laboratories as an experimental site for severe interference, meeting room for mild interference, and indoor corridor for moderate interference. This experiment demonstrates that PRComm is fully adaptable to a variety of IoT scenarios.

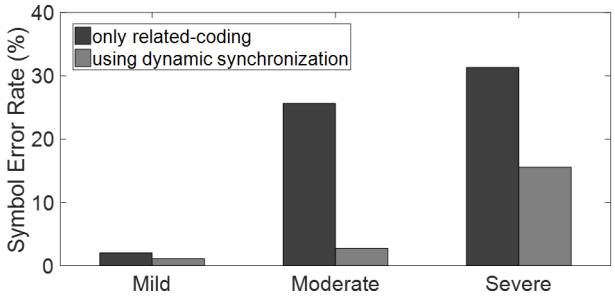
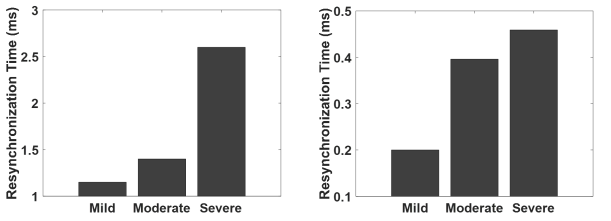


Figure 13: SER that relies on related-coding and with using dynamic synchronization.

8.2 Anti-Interference Performance

PR sequence has a certain anti-interference ability, hence PRComm has high reliability even without any other anti-interference measures. We conduct experiments in three scenarios with different levels of interference. We prefer to use commercial equipment that has been deployed in real scenarios as interference sources rather than special tools like JamLab-NG [5][32]. JamLab-NG can control the interference strength and turn off CSMA to leap off the uncontrollable delay. And it could send interference packets on time at the customized moment. However, in real applications, although some commercial equipment provides monitoring mode, CSMA cannot be manually turned off. Therefore, PRComm attempt to quantify the interference of real environments into different levels, and verify the stability of PRComm in this real world.

The results are shown in Fig. 13, PRComm can only rely on the PR sequence to resist an elementary degree of interference. In severe interference environments, even if the channel occupancy exceeds 50%, some other methods have almost completely stopped working, but our method has an SER of approximately 31.3%. In a moderately disturbed environment, the error rate is 25.64% and the



(a) only coding (b) coding and dynamic synchronization

Figure 14: Re-synchronization time when using only coding, and while accompanied with dynamic synchronization.

mildness is only 2.05%. This demonstrates that coding base on PR sequence performs well in an interference environment.

8.3 Dynamic Synchronous Decoding Strategy Performance

Although only using coding under heavy interference has good anti-interference performance, but it is still not enough to ensure the stability of the system in a real environment. In this experiment, we add a dynamic synchronous decoding strategy to PRComm. As mentioned before, we first restore the original sequence dynamically according to the correlation, and then synchronize and decode on the relatively reliable restored sequence.

The experimental results are shown in Fig. 13. We can see, the impact of the interference on PRComm has been significantly reduced. The SER under mild interference is as low as 1.1%, the SER with moderate interference is 2.75%, which is 89.27% lower than using the PR sequence alone. Similarly, the severe scene SER is reduced by 49.7%. The decoding performance under moderate interference has been significantly improved, but the SER is still only 15.56% under severe interference. This is because, severe interference means that the deployment of equipment in the environment is very dense, the number of interference sources and the energy of it is big. So the interference data packets may replace our original sequence greatly, make it impossible to recover through dynamic decoding strategy. In the moderate interference environment, the error is mainly due to the packet delay caused by the collision, so it can be recovered. When the channel occupancy rate is as high as 50%, the reliability still remains above 84%.

8.4 Synchronization Stability

As shown in Fig. 14 -(a), in mild, moderate, and severe interference scenarios, synchronization relies on related coding. Although the interference level is different, the re-synchronization time is still at the ms level, and the synchronization in the three scenarios is repeated. The synchronization time in the three scenarios is 1.15ms, 1.4ms, and 2.6ms, respectively. After using the dynamic recovery strategy, the synchronization time is successfully reduced by an order of magnitude, as Fig. 14 -(b). Because the dynamic synchronous decoding strategy restores the initial sequence without delay to some extent, the impact of interference on the packet level CTC is reduced. Therefore, the PRComm re-synchronization time after using the dynamic synchronous decoding strategy is kept on the order of 10^{-4} s. This means that our method not only improves SER, but also has an improvement in the response time of the cross-protocol system. This is an exciting development for some application scenarios that urgently require time accuracy.

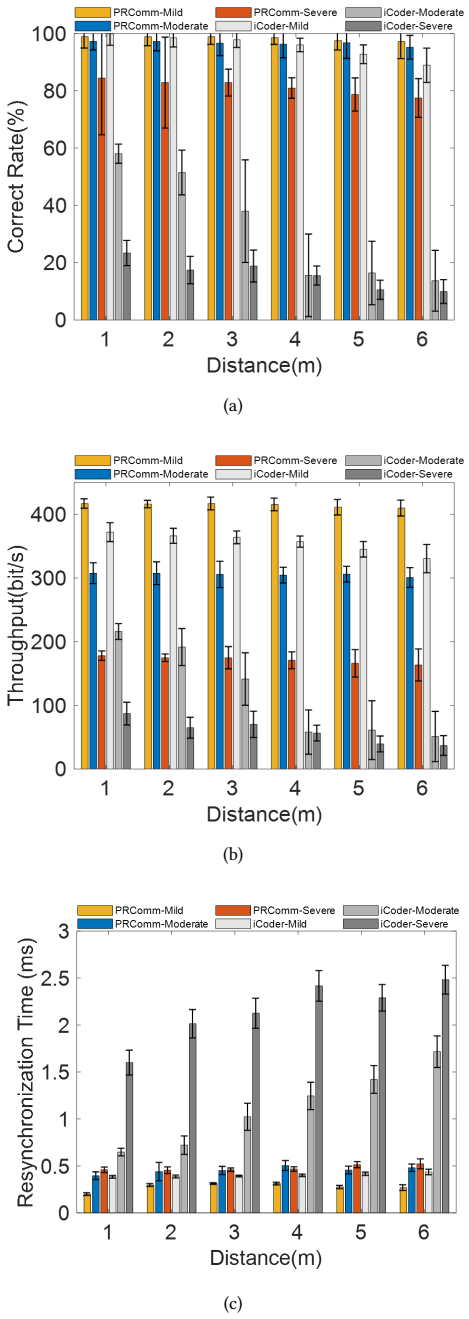


Figure 15: Decoding accuracy, throughput, re-synchronization time at different distances.

8.5 Comparison with State of the Art in Real-world Scenarios

To meet the requirements of real-world applications, we discuss different equipment deployment distances and analyze whether PRComm still have advantages even at longer distances. We change the distance between the facilities from 1m to 6m (the WiFi transmitter transmission power is 14dBm) and set a sampling point every 1m. When the communication distance exceeds 6m, the communication performance drops significantly under severe interference scenarios, and the results of the experiment fluctuate greatly. Therefore, to ensure the validity of the results and the consistency under different scenarios, we set the maximum distance of the experiment as 6m. To accommodate mild, moderate, and severe interference scenarios, the symbol window is set to 2.37ms, 3.55ms, and 4.74ms.

At the same time, we also implement the iCoder coding scheme in StripComm [43] and test them in the same natural interference environment as PRComm. iCoder's equipment of the sender, receiver and transmit power are all consistent with PRComm. In the experiment of each scene, for the iCoder, at each distance, we first measure the noise of the current position, and then continue to send the synchronization sequence and code in a loop for 1 minute, a total of about 2232 Symbols, and repeat the process 10 times. After changing to the next distance, repeat the above process. And the whole experiment will be repeated at different times on different days, the total running time exceeds 4 days. For PRComm, the number of experiments is the same as that of iCoder, but 25 different combinations of packet settings will be tried under each distance and environment, so the total experiment running time is about two weeks after multiple repetitions.

Figure 15 shows the average and standard deviation of the experimental results of PRComm and iCoder in three different scenarios. For PRComm, we can see that in different scenarios, there is no significant change in the throughput with different distances, but the decoding accuracy rate in Fig. 15-(a) slightly decreases as the distance becomes longer. The reason is that as the distance increase, the strength of the signal will decrease, and the ability to affect the RSSI value of the channel will be weakened. However, it is found in Fig. 15-(b) that the overall throughput in the range of 6m is still relatively stable. The time of re-synchronization takes in Fig. 15-(c) fluctuate no more than 0.1ms.

For iCoder, it can be seen from 15 that in different levels of interference environments, iCoder's correct rate, throughput, and synchronization will decay as the distance increases. Especially in the medium and severe interference environment, the correct rate is about 60% or lower of PRComm under the same conditions, especially when the distance is increased to 4m and above; throughput is 70% of PRComm, in the worst-case even less than 20%; the time required for re-synchronization is about three times or more than PRComm. By analyzing the decoding process, we find that in a real scenario, the uncontrollability operation process of the underlying protocol and the presence of interference change the interval between packets and the timing characteristics of the packets themselves. And these errors cannot be tolerated by the fixed decoding window at the receiving end of iCoder, resulting in data discard and decline of the communication efficiency.

Therefore, we conclude that PRComm can effectively achieve cross-protocol communication as long as the device deployment distance within the power coverage of the device itself.

9 CONCLUSION

In this paper, we propose PRComm, a novel CTC method with low power consumption and strong anti-interference capability. PRComm explores pseudo-random sequence, takes advantage of its correlation properties. PRComm has the following characteristics. First, PRComm has an outstanding advantage of anti-interference. PRComm could accomplish across protocol communication under strong interference. It can overcome interference as well as random time delay that brought by the underlying protocol, to guarantee the stability of the communication. Second, PRComm combines synchronization with communication, achieving a real-time and accurate synchronization process.

PRComm has been deployed on existing commercial WiFi and ZigBee equipment, without modifying any underlying hardware and firmware. In the common coexistence interference environment, the reliability is as high as 97%. When the channel occupancy rate is as high as 50%, the reliability is still above 84%, and the synchronization time is maintained within 1ms.

10 ACKNOWLEDGMENTS

This work was supported in part by the NSFC under Grant 61772422, in part by the International Cooperation Project of Shaanxi Province under Grant 2021KW-14, 2019KWZ-05, in part by the Innovation team Project of Shaanxi Province under Grant 202080001, in part by the Key Research and Development Project of Shaanxi Province under Grant 2019GY-146, in part by Science and Technology Innovation Guidance Project of Xi'an under Grant 201805029YD-7CG13(4).

REFERENCES

[1] Phil Beecher. 2008. Clear channel assessment. United States patent US 7,363,046. 2008 Apr 22.
[2] Alex Bereza, Ulf Wetzker, Carsten Herrmann, Carlo Alberto, and Marco Zimmerling. 2017. Demo: Cross-technology communication between BLE and Wi-Fi using commodity hardware. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks (EWSN)*.
[3] Yao-Tang Chang, Jen-Fa Huang, Li-Wei Chou, Chuen-Ching Wang, Chih-Ta Yen, and Hsu-Chih Cheng. 2011. Adaptive modified time-spreading and wavelength-group-hopping embedded M-sequence code for improved confidentiality over synchronous networks. *Optical Engineering* 50, 5 (2011), 055001.
[4] Kameswari Chebrolu and Ashutosh Dhekne. 2012. Esense: Energy sensing-based cross-technology communication. *IEEE Transactions on Mobile Computing* 12, 11 (2012), 2303–2316.
[5] Gonglong Chen and Wei Dong. 2018. Jamcloak: Reactive jamming attack over cross-technology communication links. In *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 34–43.
[6] Gonglong Chen, Wei Dong, Zhiwei Zhao, and Tao Gu. 2017. Towards accurate corruption estimation in zigbee under cross-technology interference. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 425–435.
[7] Ruirong Chen and Wei Gao. 2019. Enabling cross-technology coexistence for extremely weak wireless devices. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 253–261.
[8] Yongrui Chen, Zhijun Li, and Tian He. 2018. TwinBee: Reliable physical-layer cross-technology communication with symbol-level coding. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 153–161.
[9] Zicheng Chi, Zhichuan Huang, Yao Yao, Tiantian Xie, Hongyu Sun, and Ting Zhu. 2017. EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices. In *IEEE INFOCOM 2017-IEEE conference on computer communications*. IEEE, 1–9.

[10] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. 2016. B2w2: N-way concurrent communication for iot devices. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. 245–258.

[11] Zicheng Chi, Yan Li, Yao Yao, and Ting Zhu. 2017. PMC: Parallel multi-protocol communication to heterogeneous IoT radios within a single WiFi channel. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 1–10.

[12] Daniele Croce, Pierluigi Gallo, Domenico Garlisi, Fabrizio Giuliano, Stefano Mangione, and Ilenia Tinnirello. 2014. Errorsense: Characterizing wifi error patterns for detecting ZigBee interference. In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 447–452.

[13] Demin Gao, Shuo Zhang, Fuquan Zhang, Tian He, and Jinchi Zhang. 2019. RowBee: a routing protocol based on cross-technology communication for energy-harvesting wireless sensor networks. *IEEE Access* 7 (2019), 40663–40673.

[14] Eimantas Garsva, Nerijus Paulauskas, and Gediminas Grazulevicius. 2015. Packet size distribution tendencies in computer network flows. In *2015 Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE, 1–6.

[15] Inc Gartner. 2016. *Gartner Report.* http://cloudtimes.org/2013/12/20/gartner-theinternet-of-things-will-grow30-times-to-26-billion-by-2020/

[16] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. 2019. Wide: physical-level ctc via digital emulation. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 49–60.

[17] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu, and Omprakash Gnawali. 2020. Zigfi: Harnessing channel state information for cross-technology communication. *IEEE/ACM Transactions on Networking* 28, 1 (2020), 301–311.

[18] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Zihao Yu, and Yunhao Liu. 2019. Lego-fi: Transmitter-transparent ctc with cross-demapping. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2125–2133.

[19] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. Wizig: Cross-technology energy communication over a noisy channel. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[20] Anwar Hithnawi, Su Li, Hossein Shafagh, James Gross, and Simon Duquennoy. 2016. Crosszig: combating cross-technology interference in low-power wireless networks. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Ieee, 1–12.

[21] Anwar Hithnawi, Hossein Shafagh, and Simon Duquennoy. 2014. Understanding the impact of cross technology interference on IEEE 802.15. 4. In *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. 49–56.

[22] IEEE. 2016. *CSMA IEEE Std 802.11-2016[EB/OL].* https://standards.ieee.org/findstds/standard/802.11-2016.html

[23] Takashi Ikegawa. 2016. Data-unit-size distribution model with retransmitted packet size preservation property and its application to goodput analysis for Stop-and-Wait protocol: case of independent packet losses. *arXiv preprint arXiv:1610.00149* (2016).

[24] Wenchao Jiang, Song Min Kim, Zhijun Li, and Tian He. 2018. Achieving receiver-side cross-technology communication with cross-decoding. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MOBICOM)*. 639–652.

[25] Wenchao Jiang, Zhimeng Yin, Song Mim Kim, and Tian He. 2017. Transparent cross-technology communication over data traffic. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[26] Song Min Kim and Tian He. 2015. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MOBICOM)*. 317–330.

[27] Yan Li, Zicheng Chi, Xin Liu, and Ting Zhu. 2018. Passive-zigbee: Enabling zigbee communication in iot networks with 1000x+ less power consumption. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems (SENSYS)*. 159–171.

[28] Zhijun Li and Tian He. 2017. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MOBICOM*. 2–14.

[29] Zhijun Li and Tian He. 2018. Longbee: Enabling long-range cross-technology communication. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 162–170.

[30] Radius Networks. 2016. *Altbeacon.* http://altbeacon.org/

[31] Rajib Paul, Young-June Choi, Jiwoong Jang, and Young-Sik Kim. 2019. Channel hopping using p-ary m-Sequence for rendezvous in cognitive radio networks. *IEEE Wireless Communications Letters* 8, 6 (2019), 1516–1519.

[32] Markus Schuß, Carlo Alberto Boano, Manuel Weber, Matthias Schulz, Matthias Hollick, and Kay Römer. 2019. JamLab-NG: Benchmarking low-power wireless protocols under controllable and repeatable Wi-Fi interference.. In *Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks (EWSN)*. 83–94.

[33] Shuai Wang, Song Min Kim, and Tian He. 2018. Symbol-level cross-technology communication via payload encoding. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 500–510.

[34] Weiguo Wang, Xiaolong Zheng, Yuan He, and Xiuzhen Guo. 2019. Adacomm: Tracing channel dynamics for reliable cross-technology communication. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[35] Xiao-Long Wu, Wei-Min Li, Fang Liu, and Hua Yu. 2012. Packet size distribution of typical Internet applications. In *2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)*. IEEE, 276–281.

[36] Min Yang, Fanglin Gu, Jun Liu, and Ling Wang. 2019. An anti-interference synchronization for OFDM systems based on scrambling sequence. *IEEE Access* 7 (2019), 51121–51128.

[37] Shengrong Yin, Qiang Li, and Omprakash Gnawali. 2015. Interconnecting wifi devices with ieee 802.15. 4 devices without using a gateway. In *2015 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 127–136.

[38] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. 2017. C-morse: Cross-technology communication with transparent morse coding. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[39] Zhimeng Yin, Zhijun Li, Song Min Kim, and Tian He. 2018. Explicit channel coordination via cross-technology communication. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MOBISYS)*. 178–190.

[40] Zihao Yu, Chengkun Jiang, Yuan He, Xiaolong Zheng, and Xiuzhen Guo. 2018. Crocs: Cross-technology clock synchronization for WiFi and ZigBee. In *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks (EWSN)*. 135–144.

[41] Xinyu Zhang and Kang G Shin. 2013. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *IEEE INFOCOM 2013-IEEE Conference on Computer Communications*. IEEE, 3094–3101.

[42] Xiaolong Zheng, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. 2014. Zisense: towards interference resilient duty cycling in wireless sensor networks. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SENSYS)*. 119–133.

[43] Xiaolong Zheng, Yuan He, and Xiuzhen Guo. 2018. Stripcomm: Interference-resilient cross-technology communication in coexisting environments. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 171–179.

[44] Xiao Zhou, Fang Yang, and Jian Song. 2012. Novel transmit diversity scheme for TDS-OFDM system with frequency-shift m-sequence padding. *IEEE transactions on broadcasting* 58, 2 (2012), 317–324.