Enabling Cross-technology Communication from LoRa to ZigBee in the 2.4 GHz Band

JUNYANG SHI and XINGJIAN CHEN, State University of New York at Binghamton
MO SHA, Florida International University

IEEE 802.15.4-based wireless sensor-actuator networks have been widely adopted by process industries in recent years because of their significant role in improving industrial efficiency and reducing operating costs. Today, industrial wireless sensor-actuator networks are becoming tremendously larger and more complex than before. However, a large, complex mesh network is hard to manage and inelastic to change once the network is deployed. In addition, flooding-based time synchronization and information dissemination introduce significant communication overhead to the network. More importantly, the deliveries of urgent and critical information such as emergency alarms suffer long delays, because those messages must go through the hop-by-hop transport. A promising solution to overcome those limitations is to enable the direct messaging from a long-range radio to an IEEE 802.15.4 radio. Then messages can be delivered to all field devices in a single-hop fashion. This article presents our study on enabling the cross-technology communication from LoRa to ZigBee using the energy emission of the LoRa radio as the carrier to deliver information. Experimental results show that our cross-technology communication approach provides reliable communication from LoRa to ZigBee with the throughput of up to 576.80 bps and the bit error rate of up to 5.23% in the 2.4 GHz band.

CCS Concepts: • **Networks** → **Network protocol design**; **Sensor networks**; *Network management;*

Additional Key Words and Phrases: Cross-technology communication, LoRa, ZigBee, wireless sensor-actuator networks

ACM Reference format:
Junyang Shi, Xingjian Chen, and Mo Sha. 2021. Enabling Cross-technology Communication from LoRa to ZigBee in the 2.4 GHz Band. *ACM Trans. Sen. Netw.* 18, 2, Article 21 (December 2021), 23 pages.
https://doi.org/10.1145/3491222

1 INTRODUCTION

Industrial networks have developed alongside the Internet. Whereas the Internet is built to interconnect billions of heterogeneous devices communicating globally large amounts of data, industrial networks typically connect hundreds or thousands of sensors and actuators in industrial

21

Part of this article was published in *Proceedings of the ICII* [34]. J. Shi and X. Chen contributed to this work when they were with State University of New York at Binghamton, advised by M. Sha.
This work was supported in part by the National Science Foundation under grants CNS-1657275, CNS-2046538, and CNS-2150010.
Authors' addresses: J. Shi and X. Chen, State University of New York at Binghamton, 4400 Vestal Parkway East, Binghamton, NY, 13902; emails: {jshi28, xchen218}@binghamton.edu; M. Sha (corresponding author), Florida International University, 11200 SW 8th Street, Miami, FL, 33199; email: msha@fiu.edu.

1550-4859/2021/12-ART21 $15.00
https://doi.org/10.1145/3491222

facilities, such as steel mills, oil refineries, chemical plants, and infrastructures implementing complex monitoring and control processes. Although the typical process applications have low data rates, they pose unique challenges because of their critical demands for *reliable* and *real-time* communication in harsh industrial environments. Failing to achieve such performance can lead to production inefficiency, safety threats, and financial loss. Those demands have been traditionally met by specifically chosen wired solutions, such as the Highway Addressable Remote Transducer (HART) communication protocol [18], where cables connect sensors and forward sensor readings to a control room where a controller sends commands to actuators. However, wired networks are often costly to deploy and maintain in industrial environments and difficult to reconfigure to accommodate new production requirements.

Wireless sensor-actuator network (WSAN) technology is appealing for use in industrial applications because it does not require wired infrastructure. Battery-powered wireless modules easily and inexpensively retrofit existing sensors and actuators in industrial facilities without running cabling for communication and power. IEEE 802.15.4-based WSANs operate at low power and can be manufactured inexpensively, which make them ideal where battery lifetime and costs are important. The leading industrial WSAN standards (WirelessHART [13] and ISA100 [20]) have adopted the IEEE 802.15.4-based WSANs.

The current approach to implementing industrial WSANs relies on a multi-hop mesh network to deliver sensing data and control commands. Today, industrial WSANs are becoming tremendously larger and more complex than before. A large and complex mesh network is hard to manage and inelastic to change once the network is deployed. In addition, flooding-based time synchronization and information dissemination introduce significant communication overhead to the network. More importantly, the deliveries of urgent and critical information such as emergency alarms suffer long delays, because those messages must go through the hop-by-hop transport.

Low-power wide-area networks (LPWANs) are emerging as a promising technology, which provides long-distance connections to a large number of devices [4]. Recent years have witnessed rapid real-world adoption of LPWAN for various Internet of Things applications. The limitations of multi-hop mesh networks can be overcome by enabling the direct messaging from a long-range LPWAN radio to an IEEE 802.15.4 (ZigBee [49]) radio. Leveraging the large coverage, a LPWAN-enabled base station can disseminate the network management messages, time synchronization beacons, and urgent information to WSAN devices in a single-hop fashion. Semtech's recently announced LoRa SX1280/SX1281 wireless RF chips [33], operating in the 2.4 GHz **industrial, scientific, and medical (ISM)** band, open new opportunities for the direct messaging from LoRa to ZigBee. This article presents a direct messaging solution from LoRa to ZigBee, leveraging the recent advancements on the **cross-technology communication (CTC)** technologies. The CTC from LoRa to ZigBee is achieved by putting specific bytes in the payload of legitimate LoRa packets. The bytes are selected such that the corresponding information can be understood by the ZigBee devices through sampling the **received signal strength (RSS)**. Our LoRa to ZigBee CTC solution does not require any hardware modification to the existing WSAN field devices. Specifically, this work makes the following contributions:

- To the best of our knowledge, this is the first work to investigate the CTC from LoRa to ZigBee in the 2.4 GHz band, distinguished from previous work pertaining to the CTC among WiFi, ZigBee, and Bluetooth devices.
- We perform an empirical study that investigates the characteristics of LoRa in the 2.4 GHz band from a CTC's point of view and provides a set of new observations.
- We introduce a novel LoRa to ZigBee CTC approach. By elaborately tuning LoRa's packet payload, a ZigBee device is capable of decoding the information carried by the LoRa packet by purely sampling the RSS.

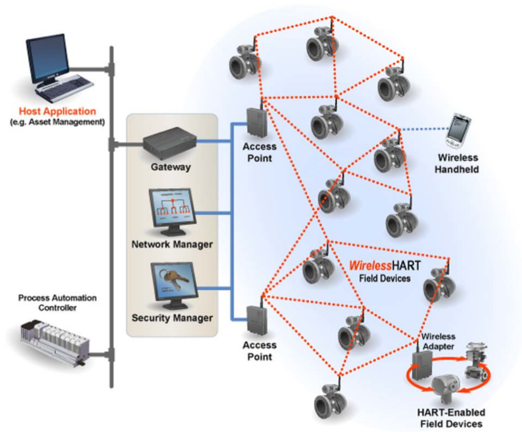


Fig. 1. Architecture of a WirelessHART network (Credit: HART Communication Foundation [13]).

- We present an approach to support reliable CTC from LoRa to ZigBee in the 2.4 GHz band and a new solution to integrate the CTC to the IPv6 **Routing Protocol for Low-Power and Lossy Networks (RPL)**.
- Our proposed CTC approach has been implemented and tested on real hardware. Experimental results show that our approach provides reliable communication from LoRa to ZigBee with the throughput of up to 576.80 bps.

The rest of this article is organized as follows. Section 2 discusses the background of IEEE 802.15.4-based industrial WSANs and LoRa technology. Section 3 introduces our empirical study, and Section 4 presents the design of our CTC approach. Section 5 describes our approach that provides reliable CTC. Section 6 shows our evaluation. Section 7 reviews the related work, and Section 8 concludes the article.

2 BACKGROUND

In this section, we provide a brief introduction to the IEEE 802.15.4-based industrial WSANs and LoRa technology.

2.1 IEEE 802.15.4-Based Industrial WSANs

To meet the stringent reliability and real-time requirements, industrial WSAN standards make a set of unique network design choices that distinguish industrial WSANs from traditional wireless sensor networks designed for best effort services [30]. For instance, WirelessHART [13] and ISA100 [20], the leading industrial WSAN standards, specify a centralized network management architecture that enhances the timing predictability of packet deliveries and visibility of network operations. Figure 1 shows the architecture of a WirelessHART network. A WirelessHART network consists of a gateway, multiple access points, and a set of field devices (sensors and actuators). The access points and field devices are equipped with half-duplex omnidirectional radio transceivers, which are compatible with the IEEE 802.15.4 physical layer, and form a multi-hop wireless mesh network. The access points are connected with the gateway device through wired links and serve as bridges between the gateway and field devices. The network manager, a software module running on the gateway, is responsible for managing the entire wireless network. The network manager collects the link traces and network topology information from the field devices, and determines the routes between itself and all devices.

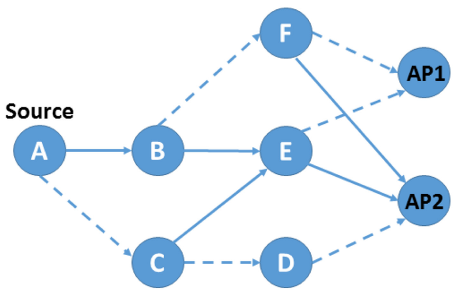


Fig. 2. An example of graph routing. The solid lines represent the primary routing paths, and the dashed lines denote the backup routes.

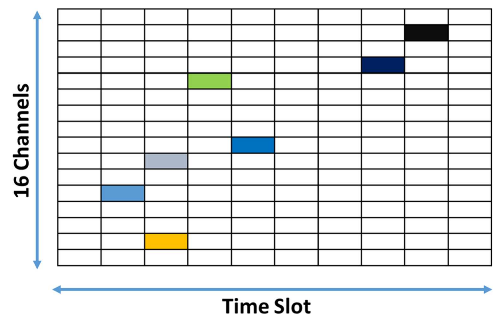


Fig. 3. TSCH technology.

To enhance the reliability of packet deliveries, WirelessHART supports source routing and graph routing. Source routing provides a single routing path for each dataflow (from sensors to actuators), whereas graph routing first generates a reliable graph in which each device should have at least two neighbors to which they may send packets and then provides multiple redundant routes based on the graph. Figure 2 shows a graph routing example. To send a packet to access points, Device A may transmit the packet to Device B by using the main routing path or Device C through the backup route. From those devices, the packet may take several alternate routes to reach the access points. Graph routing is designed to enhance the network reliability through route diversity and redundancy.

To enhance the timing predictability of packet deliveries, WirelessHART adopts the **time-slotted channel hopping (TSCH)** technology in the MAC (medium access control) layer. As Figure 3 shows, all devices clocks are synchronized, and time is divided into time slots with a fixed length. To combat narrow-band interference and multi-path fading, TSCH uses up to 16 channels operating in the 2.4 GHz band, and each device switches its channel in every slot. Channel black-listing is an optional feature that allows the network operator to restrict the channel hopping of field devices network-wide to selected channels in the wireless band.

2.2 LoRa Overview

LPWAN is emerging as a promising wireless technology to provide long-distance connections with a greater than 1-km range, covering a large number of Internet of Things devices [4]. LoRa, which is short for "Long Range," is an industry LPWAN technology, initiated by Semtech [9] and promoted by the LoRa Alliance [2] to build scalable wireless networks. LoRa leverages the chirp spread spectrum to modulate data in the physical layer and operates in the unlicensed 915-MHz (in the United States, Canada, and South America) and 2.4 GHz bands (globe). In this work, we focus on the LoRa technology, which operates in the 2.4 GHz band, specifically using the Semtech's new SX1280/SX1281 wireless RF chips [33].

Physical-layer characteristics. LoRa employs chirp spread spectrum modulation, which leverages frequency chirps with a constantly increasing or decreasing frequency sweeping through a predefined bandwidth. Figure 4 shows an example of LoRa transmission with upchirps, downchirps, and data chirps in the frequency variation over time. The first several upchirps, which are configurable from 2 to 65, 535, are preambles. Each chirp's frequency sweeps from the minimum frequency (f_{min}) to the maximum frequency (f_{max}). The following 2.25 downchirps are the start frame delimiter, whose frequency goes from f_{max} to f_{min}. The rest chirps are data chirps. The position of

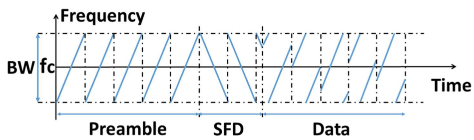


Fig. 4. An example of LoRa transmission with upchirps, downchirps, and data chirps.

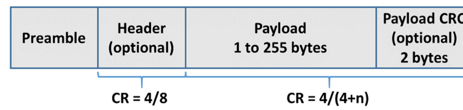


Fig. 5. LoRa variable-length packet format ($n \in [1, 4]$).

Table 1. Key LoRa Physical-Layer Parameters in the 2.4 GHz Band

Parameter	Options
f_c	Between 2,400 and 2,482 MHz
SF	5, 6, 7, 8, 9, 10, 11, 12
BW(kHz)	203, 406, 812, 1,625
CR	4/5, 4/6, 4/7, 4/8

frequency discontinuity (a sudden change from f_{max} to f_{min}) of data chirps represents different encoded data bits.

The key LoRa physical-layer parameters, which are configurable by the user, include the frequency bandwidth (BW), central carrier frequency (f_c), spreading factor (SF), and coding rate (CR). Table 1 lists the possible values for each parameter. The time duration of transmitting a single LoRa chirp (T_s) is

$$T_s = \frac{2^{SF}}{BW}, \tag{1}$$

and each LoRa chirp can convey SF bits of information. Thus, the physical-layer data transmission bit rate of LoRa (R_b) is

$$R_b = \frac{SF * CR}{T_s} = SF * \frac{BW}{2^{SF}} * CR. \tag{2}$$

The selection of those parameters makes significant impacts on the LoRa decoding sensitivity and transmission range. For instance, either an increase in SF or a decrease in BW enlarges the transmission range.

Physical frame format. Semtech specifies the physical frame format of LoRa packets. As Figure 5 shows, a LoRa frame starts with a preamble followed by an optional header using a coding rate of 4/8. The payload size (PL) of each LoRa packet ranges from 1 to 255 bytes. LoRa uses one byte to store the payload size. CRC check is optional and uses a configurable coding rate.

The number of LoRa data chirps (N_{chirp}) for transmitting a packet with PL bytes payload can be calculated by Equation (3), where PL is the LoRa payload size in bytes, CRC is 16 if the CRC check is enabled or 0 otherwise, H is the size of LoRa packet header, and DE is either 2 if the low data rate optimization is enabled or 0 otherwise.

$$N_{chirp} = 8 + max\left(\left\lceil \frac{8PL - 4SF + 8 + CRC + H}{4(SF - DE)} \right\rceil * \frac{4}{CR}, 0\right) \tag{3}$$

With Equations (1) and (3), the on-air time of a LoRa packet can be calculated as

$$T_o = (N_{chirp} + N_{preamble}) * \frac{2^{SF}}{BW}, \tag{4}$$

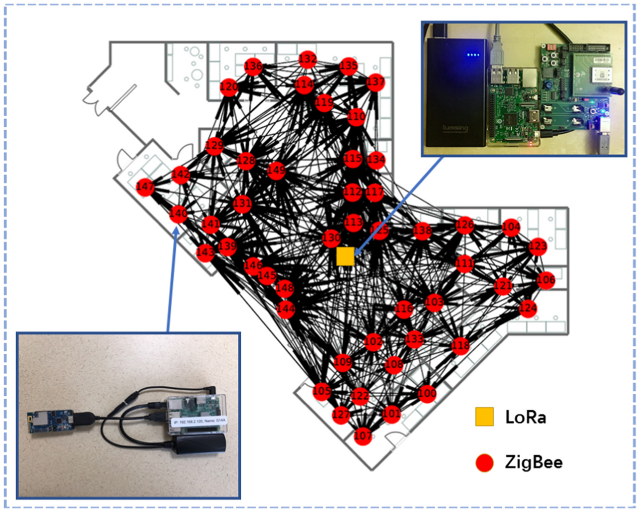


Fig. 6. Testbed deployment: red circles are 50 TelosB motes, the yellow square is a WiMOD iM282A LoRa device, and black lines are wireless links when ZigBee devices transmit at 0 dBm.

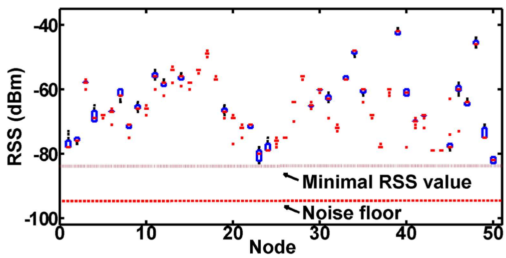
where $N_{preamble}$ denotes the number of preamble chirps and $N_{chirp} + N_{preamble}$ represents the total number of chirps used to carry the LoRa packet.

3 EMPIRICAL STUDY

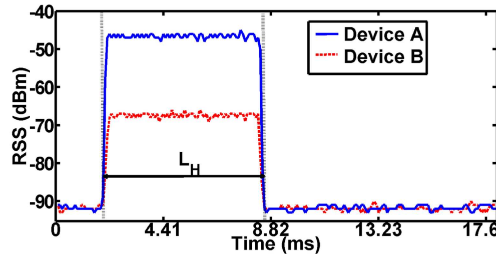
In our empirical study, we first examine the detectability of LoRa signals on ZigBee devices and then explore the RSS features, which can be used for the CTC from LoRa to ZigBee. The empirical study is performed on our testbed, which consists of 50 TelosB motes [31] (ZigBee devices) placed throughout 22 student offices, lounge, labs, and conference rooms [36]. Figure 6 shows the device placement on our testbed. The wireless network has up to four hops when the testbed devices transmit at 0 dBm. A Raspberry Pi Model B [11] integrated with a WiMOD iM282A LoRa transceiver [19] (with a Semtech SX1280 LoRa chip [33]) is used as the LoRa transmitter, as Figure 6 shows. We configure the LoRa transceiver to transmit at 15 dBm.

3.1 Detectability of LoRa Signals on ZigBee

We first perform experiments to examine the detectability of LoRa signals on ZigBee devices in the 2.4 GHz band. We configure the LoRa transmitter placed in the center of our testbed to broadcast packets and control the 50 ZigBee devices on our testbed to sample the RSS. The ZigBee and LoRa channels are configured to overlap with each other. Figure 7(a) shows the boxplot of RSS measurements. All ZigBee devices on our testbed can detect the ongoing LoRa transmissions if they set the RSS threshold between the minimal RSS value (−83 dBm) and the noise floor (−92 dBm). As a comparison, the transmissions generated by any ZigBee device can reach up to 66.0% of devices on the testbed. We also measure the number of hops, which can be covered by the LoRa signals, on our testbed. We configure all ZigBee devices to run the IPv6 RPL [22], which creates a routing tree for the network. We repeat the experiment three times. Figure 8 plots the **cumulative distribution function (CDF)** of the number of hops, which can be covered by the LoRa signals. The median number of hops is 3 and the maximum number of hops is 6. The results demonstrate the feasibility of delivering messages using direct LoRa to ZigBee CTC instead of a hop-by-hop transport.



(a) Boxplot of RSS measurements.



(b) Example RSS signatures measured by two ZigBee devices located at different places when the LoRa device transmits a packet.

Fig. 7. Detectability of LoRa signals on ZigBee devices.

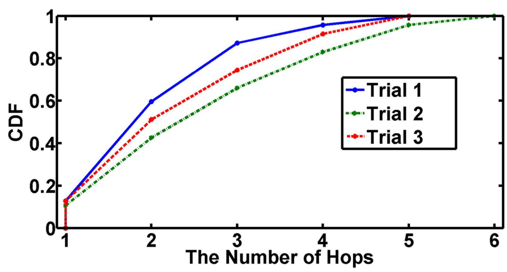


Fig. 8. The number of hops, which can be covered by the LoRa signals. All ZigBee devices run RPL.

Observation 1. *ZigBee devices can detect ongoing LoRa transmissions through sampling the RSS when the ZigBee and LoRa channels overlap.*

We perform numerical analysis to illustrate the potential benefit of enabling CTC in a multi-hop network. When the ZigBee devices on our testbed run TSCH and Orchestra [10], it takes 750 ms (50 time slots) and consumes 35.47 mJ of energy on all ZigBee devices to disseminate a message in a single slotframe. By enabling the CTC from LoRa to ZigBee, the LoRa device can deliver the message to all ZigBee devices using a single time slot, which can shorten the latency to 15 ms and reduce the energy consumption of ZigBee devices to 18.61 mJ.

Figure 7(b) shows the example RSS signatures measured by two ZigBee devices located at different places when LoRa transmits a packet. We define the sequence of RSS values measured by ZigBee when LoRa transmits a packet as a RSS signature. The naive approach would be to use different RSS values to encode different information, but this would require each device to generate its own mapping between the RSS values and encoded information since the RSS values depend on the link distance. An alternative approach is to use the number of consecutive RSS values higher than the threshold (L_H) to encode information. As Figure 7(b) shows, both Device A and B get $L_H = 73$ when the RSS threshold is −85 dBm, and L_H is independent of link distance. Using L_H instead of the absolute RSS values, the network only needs one device to generate the mapping between RSS values and encoded information and share it with the rest. This significantly reduces the device setup and calibration overhead.

Observation 2. *The number of consecutive RSS values higher than the threshold (L_H) can be used to encode information.*

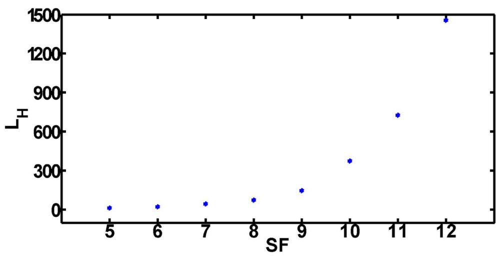


Fig. 9. L_H values captured by ZigBee when LoRa transmits the same packet using different *SF*.

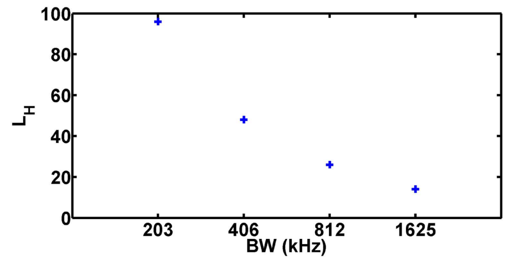


Fig. 10. L_H values captured by ZigBee when LoRa transmits the same packet using different *BW*.

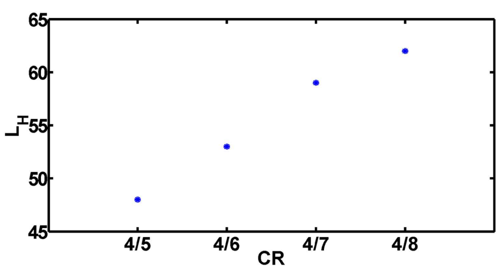


Fig. 11. L_H values captured by ZigBee when LoRa transmits the same packet using different *CR*. *CR* represents coding rate.

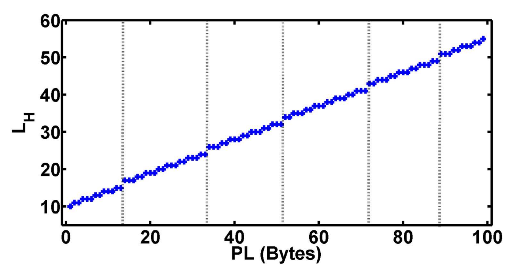


Fig. 12. L_H values captured by ZigBee when LoRa transmits packets carrying different payload. Payload size is *PL*.

3.2 Creation of RSS Signatures with Different Features

As discussed in Section 2.2, using different physical-layer parameters (i.e., *SF*, *BW*, *CR*, and *PL*) can create the RSS signatures with different L_H. Our goal is to maximize the number of distinguishable RSS signatures (with different L_H). We next run experiments to study the impact of tuning those physical-layer parameters on the number of distinguishable RSS signatures.

We first set *BW* to 1,625 kHz, *CR* to 4/5, and *PL* to 5 bytes; vary *SF* from 5 to 12; and then measure the L_H captured by the ZigBee device. Figure 9 plots L_H under different *SF*. The L_H is 13, 24, 46, 74, 147, 374, 728, and 1,457 for *SF* from 5 to 12. Tuning *SF* can generate eight distinguishable RSS signatures. We then fix *SF* to 5, *CR* to 4/5, and *PL* to 5; vary *BW* from 203 to 1,625 kHz [32]; and measure L_H. As Figure 10 shows, every time *BW* doubles, L_H roughly reduces to a half. Tuning *BW* can generate four distinguishable RSS signatures. Please note that using neither *SF* larger than 7 nor *BW* lower than 406 kHz is infeasible because their corresponding L_H values are too large to fit into a single TDMA time slot. We also repeat the experiments under different *CR* when $SF = 5$, $BW = 406$, and $PL = 5$. Tuning *CR* can generate four distinguishable RSS signatures, as Figure 11 shows.

Finally, we run the experiments when LoRa transmits packets with different payload sizes *PL*. Figure 12 shows L_H when the LoRa payload size increases from 1 to 99 bytes when $SF = 5$, $BW = 1,625$, and $CR = 4/5$. Due to the ceiling function in Equation (3), L_H exhibits the step changes pattern. Please note that L_H exhibits the step changes pattern because of the ceiling function in Equation (3). From the results, we can see that through changing *PL*, the ZigBee device obtains a large number of distinguishable RSS signatures with different L_H. As Table 2 lists, changing *PL* while using $SF = 5$ and $BW = 1{,}625$ kHz can create all L_H values from 10 to 59, and therefore there is no need to change *SF* or *BW*.

Table 2. PL, T_o, T_r, T_t, and L_H of Each Distinguishable RSS Signature

Index	PL (bytes)	T_o (ms)	T_r (μs)	T_t (ms)	L_H
1	1	0.905	2,800	3.705	10, 11, 12
2	9	1.305	3,100	4.405	13, 14, 15, 16
3	17	1.605	3,400	5.005	17, 18, 19
4	23	1.805	3,700	5.505	20, 21, 22
5	32	2.205	4,000	6.205	23, 24, 25, 26
6	39	2.505	4,300	6.805	27, 28, 29
7	47	2.805	4,600	7.405	30, 31, 32
8	54	3.105	4,900	8.005	33, 34, 35, 36
9	62	3.405	5,200	8.605	37, 38, 39
10	69	3.705	5,500	9.205	40, 41, 42
11	77	4.005	6,800	10.805	43, 44, 45, 46
12	84	4.305	7,100	11.405	47, 48, 49
13	92	4.605	8,000	12.605	50, 51, 52
14	99	4.905	8,800	13.705	53, 54, 55, 56
15	107	5.205	9,200	14.405	57, 58, 59

OBSERVATION 3. *Tuning the payload size PL is the most effective way to generate a large number of distinguishable RSS signatures.*

4 CTC DESIGN

In this section, we present the design of our CTC approach from LoRa to ZigBee in the 2.4 GHz band based on the observations presented in Section 3.

Following Orchestra [10] and Alice [23], we set the time slot length to 15 ms and define the total time (T_t) for the LoRa device to transmit a packet as

$$T_t = T_o + T_r, \tag{5}$$

where T_o denotes the on-air time of a LoRa packet and T_r denotes the software delay of packet transmission. The software delay limits the selection of the maximum L_H. Our ZigBee device has an RSS sampling rate of 11.33 kHz, providing 170 samples in every time slot.

Because of measurement inaccuracy, the ZigBee device may produce multiple L_H values when the LoRa device transmits the same payload. Thus, we must identify a set of payload sizes, which can be used to reliably generate RSS signatures with distinct L_H. There are three requirements for the payload size selection: (i) different LoRa payload sizes must provide distinct L_H, which can be captured by the ZigBee device; (ii) T_t must not exceed 15 ms; and (iii) the other three physical-layer parameters (SF, BW, and CR) must be determined before selecting the payload sizes. When we set $SF = 5$, $BW = 1,625$, and $CR = 4/5$, we get 15 payload sizes, which meet the preceding requirements. Table 2 lists the payload size (PL), LoRa packet on-air time (T_o), software delay (T_r), total time (T_t), and possible L_H values of each distinguishable RSS signature. Figure 13 shows the percentage histogram of each distinguishable RSS signature when we configure LoRa to transmit 5,000 packets using each PL and control ZigBee to measure L_H. As Figure 13 shows, the L_H values belonging to any two distinguishable RSS signatures are completely different. For example, when PL is 1 byte, 4.18% of the L_H values captured by ZigBee is 10, 42.26% is 11, and 53.56% is 12. When PL is 9 bytes, 0.83% of the L_H values captured by ZigBee is 13, 25.21% is 14, 72.95% is 15, and 1.01% is 16.

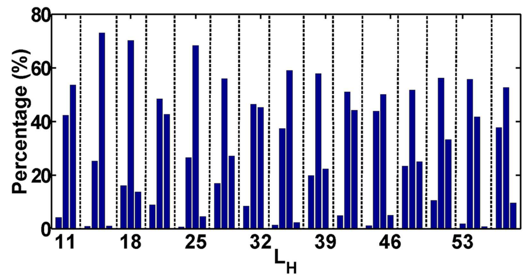


Fig. 13. Percentage histogram of each distinguishable RSS signature (L_H).

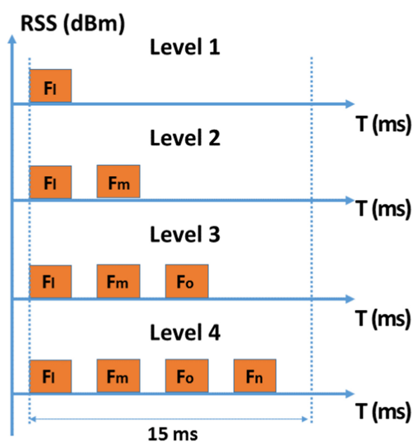


Fig. 14. RSS patterns for different levels.

After identifying the set of payload sizes, the next step is to determine how to use those distinguishable RSS signatures to encode information. As Table 2 shows, each RSS signature $F_k = (k, PL_k, T_{o_k}, T_{r_k}, T_{t_k}, L_{H_k})$ $\{1 \le k \le 15\}$ is denoted by an index k, a payload size PL_k, a packet on-air time T_{o_k}, a software delay T_{r_k}, a total time T_{t_k}, and a set of L_{H_k}. We can put multiple distinguishable RSS signatures in a single time slot to get a set of distinguishable RSS patterns to encode information. Multiple RSS signatures $\{F_l, F_m, \ldots, F_n\}$ are combined to form the RSS pattern P_z in a single time slot. We follow the four steps presented next to get the total number of distinguishable RSS patterns.

Step I: Identify the maximum distinguishable RSS pattern level n_{level}. n_{level} determines the maximum number of distinguishable RSS signatures, which can be put in a single time slot. n_{level} is computed as

$$n_{level} = \left\lfloor \frac{15}{T_{t_0}} \right\rfloor, \tag{6}$$

where 15 is the time slot length and T_{t_0} is the smallest time duration of our distinguishable RSS signatures. According to Table 2, T_{t_0} is 3.705 ms. Then $n_{level} = 4$. In each time slot, we can put (i) one RSS signature $\{F_l\}$, (ii) two RSS signatures $\{F_l, F_m\}$, (iii) three RSS signatures $\{F_l, F_m, F_o\}$, or (iv) four RSS signatures $\{F_l, F_m, F_o, F_n\}$ to form the distinguishable RSS pattern P_z $(1 \le l, m, o, n \le 15)$. Figure 14 shows the example RSS distinguishable patterns from level 1 to level 4.

Step II: Combine the distinguishable RSS signatures. Multiple distinguishable RSS signatures (F_k $\{1 \le k \le 15\}$) can be combined to form different distinguishable RSS patterns. Algorithm 1 shows the algorithm, which computes the number of distinguishable RSS patterns. $count_1$, $count_2$, $count_3$, and $count_4$ store the number of distinguishable RSS patterns in level 1 to level 4, respectively. Algorithm 1 first initializes all $count_j$ to zero (line 1). There are four nested loops (lines 2–19), and each loop iterates over the 15 distinguishable RSS signatures. Algorithm 1 uses four nested loops because at most four distinguishable RSS signatures can be put in a single time slot. The counter $count_1$ increases by 1 in line 3 because it only considers the distinguishable RSS patterns in level 1 ($\{F_l\}$), which uses one RSS signature to form the distinguishable RSS pattern. Because T_{t_k} is not longer than 15 ms, a single distinguishable RSS signature can always be directly put into the distinguishable RSS pattern set. Similarly, lines 6, 10, and 14 increase the counters by 1 for level 2, 3, and 4 distinguishable RSS patterns, respectively. Please note that the sum of T_{t_k} should be not longer than 15 ms for levels 2 (line 5), 3 (line 9), and 4 (line 13). The output *count* denotes the total number of distinguishable RSS patterns. By running Algorithm 1, we get 15, 81, 80, and 1 for $count_1$, $count_2$, $count_3$, and $count_4$, respectively. The total number of distinguishable RSS patterns ($count$) is $15 + 81 + 80 + 1 = 177$.

ALGORITHM 1: Algorithm to compute the number of distinguishable RSS patterns

```
   Input  : T_{t_k}
   Output : count
1  count_1 = 0, count_2 = 0, count_3 = 0, count_4 = 0, count = 0;
2  for l = 1; l ≤ 15; l++ do
3      count_1 ++;
4      for m = 1; m ≤ 15; m++ do
5          if T_{t_l} + T_{t_m} ≤ 15 then
6              count_2 ++;
7          end
8          for o = 1; o ≤ 15; o++ do
9              if T_{t_l} + T_{t_m} + T_{t_o} ≤ 15 then
10                 count_3 ++;
11             end
12             for n = 1; n ≤ 15; n++ do
13                 if T_{t_l} + T_{t_m} + T_{t_o} + T_{t_n} ≤ 15 then
14                     count_4 ++;
15                 end
16             end
17         end
18     end
19 end
20 count = count_1 + count_2 + count_3 + count_4;
```

Step III: Add the empty signature. More distinguishable RSS patterns can be created by adding the empty RSS signature (F_{kX}) into the time slot. F_{kX} represents the RSS signature captured by ZigBee when LoRa does not transmit any packet for the time duration T_{t_k}. Figure 15 shows two example distinguishable RSS patterns ($\{F_1, F_2, F_1\}$ and $\{F_1, F_{2X}, F_1\}$). Please note that we do not add the distinguishable RSS pattern that only has empty signatures into the distinguishable RSS pattern set, because using only empty RSS signatures to deliver a message can easily be interfered by external interference. For each level j, Algorithm 1 computes the number of combined signatures

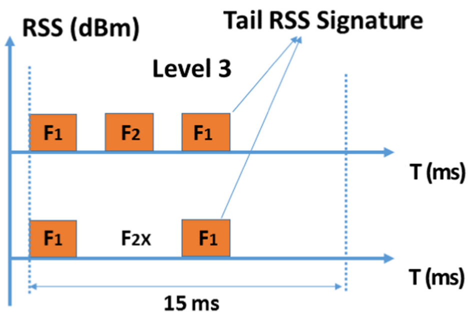


Fig. 15. Two example distinguishable RSS patterns in level 3.

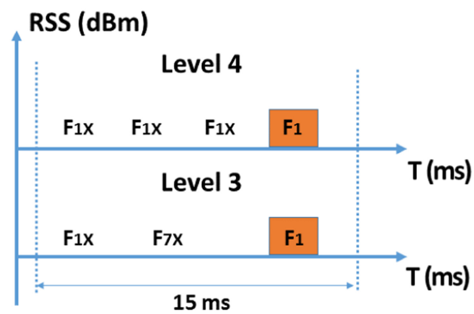


Fig. 16. An example cross-level duplication.

($count_j$) without considering the empty signature. To prevent repetition, we fix the tail RSS signature in each RSS pattern. Then each RSS signature (F_k) except the tail one can be substituted by an empty RSS signature (F_{kX}). The number of distinguishable RSS patterns in each level after adding the empty signature is

$$e_j = count_j * 2^{(j-1)}. \tag{7}$$

Then the total number of distinguishable RSS patterns after adding the empty signature is

$$sum = \sum_{i=1}^{n_{level}} e_j. \tag{8}$$

With Equations (7) and (8), we get $sum = 1 * 2^3 + 80 * 2^2 + 81 * 2^1 + 15 * 2^0 = 505$.

Step IV: Remove the duplication. Because the distinguishable RSS patterns have the empty signature, there may exist multiple RSS patterns that cannot be distinguished by the ZigBee device. We first remove the duplication at the same level. For instance, if there exist two distinguishable RSS patterns $\{F_1, F_2, F_1\}$ and $\{F_2, F_1, F_1\}$ in level 3 and we substitute F_1 and F_2 with the empty signatures, then $\{F_{1X}, F_{2X}, F_1\}$ and $\{F_{2X}, F_{1X}, F_1\}$ are same. We use F_{kX} to denote the empty signature, which locates at the kth position in the distinguishable RSS pattern. The time duration of $F_{1X} + F_{2X}$ is equal to the one of $F_{2X} + F_{1X}$. The ZigBee device cannot distinguish them by sampling the RSS. Therefore, we must remove the duplication from *sum*, which is computed in Step III. Duplication also happens between different levels. Thus, the cross-level time equivalent test is required. The ZigBee device can generate 170 RSS samples in each time slot. The sampling interval is 0.088 ms. For two different distinguishable RSS patterns in different levels, if the time difference between two consecutive empty RSS signatures is less than 0.088 ms, the ZigBee device cannot distinguish them. Figure 16 shows an example duplication. Because $|(T_{t_1} + T_{t_1} + T_{t_1}) - (T_{t_1} + T_{t_7})| = 0.005$ ms < 0.088 ms, the ZigBee device cannot distinguish $\{F_{1X}, F_{1X}, F_{1X}, F_1\}$ and $\{F_{1X}, F_{7X}, F_1\}$ by sampling the RSS. We use the brute-force method to compare any two distinguishable RSS patterns with each other in the same level and identify 33 duplicated distinguishable RSS patterns. We also find 18 duplicated distinguishable RSS patterns in the cross-level time equivalent test. After removing the duplication, the total number of distinguishable RSS patterns, which can be used to encode information, is $505 - 33 - 18 = 454$.

To maximize the throughput, the LoRa device can convert its binary data $i_{(2)}$ to 454-ary data $i_{(454)}$, whereas the ZigBee device can reverse the process to get the original binary data.

5 RELIABLE CTC

In this section, we present our approach that provides reliable CTC.

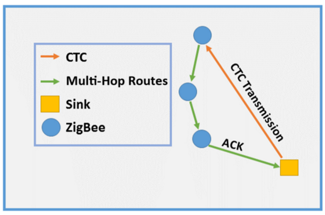


Fig. 17. An example of CTC transmission and its acknowledgment forwarded through the ZigBee network.

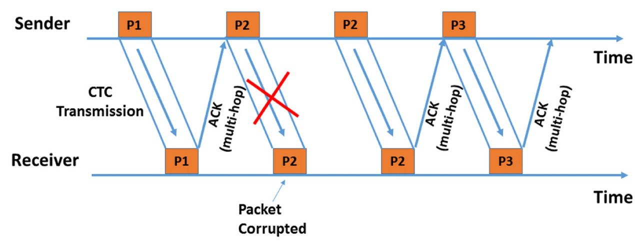


Fig. 18. An example of our ARQ approach.

5.1 ARQ Design

Automatic repeat request (ARQ) [28] is widely used to provide reliable communication in various wireless technologies. Various ARQ designs have been proposed in the literature including stop-and-wait ARQ [26], go-back-N ARQ [37], and selective-repeat ARQ [3]. Unfortunately, none of them can be directly applied to our CTC scenario because the LoRa device cannot detect the signals transmitted by the ZigBee device. To address the challenge, we propose to physically connect the LoRa device to the sink of the ZigBee networks. Please note that the communication range of our CTC from LoRa to ZigBee is much larger than the one between ZigBee devices (see Section 3). Therefore, the acknowledgments have to be forwarded to the LoRa device through a hop-by-hop transport. As Figure 17 shows, after the LoRa transmitter sends a CTC packet, the ZigBee receiver has to send back the acknowledgment through a series of relay devices. We follow the stop-and-wait ARQ design and develop the following rules. The LoRa transmitter sends next CTC packet only after receiving the acknowledgment of the current packet; otherwise, it retransmits the current packet after a predefined timeout; and the ZigBee receiver sends an acknowledgment toward the sink after receiving a CTC packet from the LoRa transmitter. Figure 18 shows an example of our ARQ approach, where the LoRa transmitter sends three CTC packets to the ZigBee receiver. The first CTC packet P_1 is successfully delivered to the ZigBee receiver, and the corresponding acknowledgment is received by the LoRa sender. Assuming the second CTC packet P_2 is lost and the LoRa transmitter fails to receive an acknowledgment before the timeout. Then the CTC packet P_2 is retransmitted until its acknowledgment is received by the LoRa transmitter. The third transmission P_3 is followed and consumed by the ZigBee receiver to conclude the transmissions. The CTC transmission from LoRa to ZigBee is delivered in a single-hop fashion, whereas the acknowledgment takes several hops to reach the sink.

5.2 Integration in RPL

We develop a solution to integrate our ARQ approach to RPL, which is widely used in ZigBee networks. RPL is a gradient routing technique that organizes the ZigBee network as a directed acyclic graph (DAG) rooted at the sink [1]. Each ZigBee device in the network tries to minimize the cost to reach the sink using an objective function. The upward routes can be used by the Zigbee device to send the identified CTC parameters such as L_H to the base station and the application traffic for data collection. Our solution embeds 1 bit in the application packets to deliver CTC acknowledgments to the sink and schedules the downward transmissions (CTC traffic) and upward transmissions (acknowledgment/application traffic) in each TSCH slotframe.

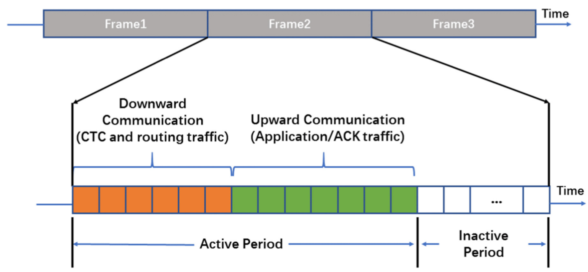


Fig. 19. TDMA frame structure.

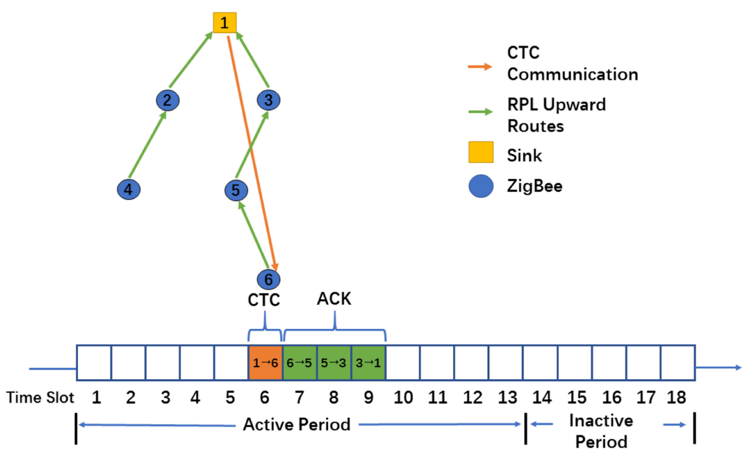


Fig. 20. An example transmission schedule for data collection from five devices. The sink's ID is 1. The IDs of five ZigBee devices range from 2 to 6.

As Figure 19 shows, the LoRa device can use the first several time slots in each slotframe for CTC, and the rest time slots in the active period can be used for the communication between ZigBee devices including the deliveries of application traffic and CTC acknowledgments. The number of time slots for downward and upward traffic can be configured based on the application traffic demand. We assign fixed and unique time slots for downward CTC transmissions to different ZigBee devices and use the rate monotonic [29] scheduling algorithm to schedule the upward traffic containing acknowledgments. The LoRa transmitter can deliver the CTC packet and receive its corresponding acknowledgment from the ZigBee networks in each slotframe, and thus the predefined timeout ($T_{timeout}$) for CTC retransmission can be calculated using the following equation:

$$T_{timeout} = (N_{frame} - 1) * T_{timeslot}, \tag{9}$$

where N_{frame} is the number of time slots in each slotframe and $T_{timeslot}$ is the time duration of a single TDMA time slot. If the LoRa device fails to receive the acknowledgment in the current slotframe, it will automatically retransmit the CTC packet in the next slotframe.

Figure 20 shows an example schedule for data collection from five ZigBee devices (IDs 2 through 6). We configure the LoRa device at sink to transmit CTC to the ZigBee device with $ID = 6$ in the sixth time slot of the slotframe. Based on the rate monotonic scheduling algorithm, the ZigBee device with $ID = 6$ sends an acknowledgment back hop by hop using the seventh, eighth, and ninth time slots.

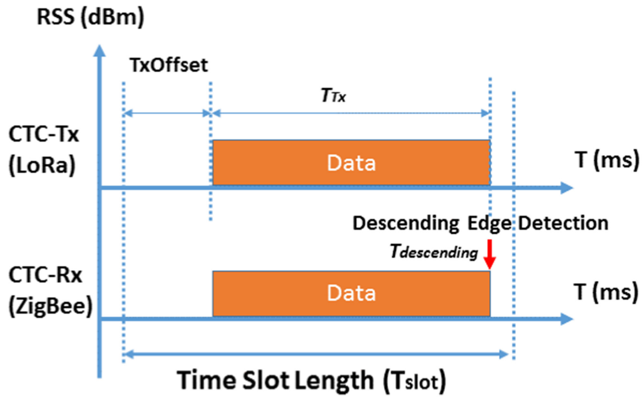


Fig. 21. Time synchronization between LoRa and ZigBee devices. T_{slot} denotes the length of a time slot.

5.3 Time Synchronization

Time synchronization is important for our design because the ZigBee devices in industrial WSANs run TSCH in the MAC layer and work in a duty-cycled manner. It is important to synchronize the clocks of LoRa and ZigBee devices so that the ZigBee receiver can receive the CTC packet sent by the LoRa transmitter within a single time slot. The LoRa device broadcasts beacons for time synchronization periodically. The time interval between beacons is shared between LoRa and ZigBee devices. To reduce the time synchronization overhead, we develop a method that leverages the CTC signals generated by the LoRa device for time synchronization. Specifically, the ZigBee devices detect the descending edge of the LoRa signals and use that to adjust their clocks. Figure 21 illustrates the time synchronization process. The LoRa device transmits a packet after the time offset $TxOffset$ in each time synchronization slot. The ZigBee device records the time after it captures the descending edge of the LoRa packet ($T_{descending}$) and uses it as a reference to synchronize its clock. The ZigBee device can identify the LoRa device's time when the current time slot starts (T_{LoRa}) by calculating the following equation:

$$T_{LoRa} = T_{descending} - T_{TxOffset} - T_{Tx}, \tag{10}$$

where T_{Tx} is the time duration of a LoRa packet. The time drift (T_{drift}) can be calculated as

$$T_{drift} = T_{LoRa} - T_{ZigBee}, \tag{11}$$

where T_{ZigBee} denotes the time when the current time slot starts based on the ZigBee device's own clock. The ZigBee device can use Equations (10) and (11) to adjust its clock.

6 EVALUATION

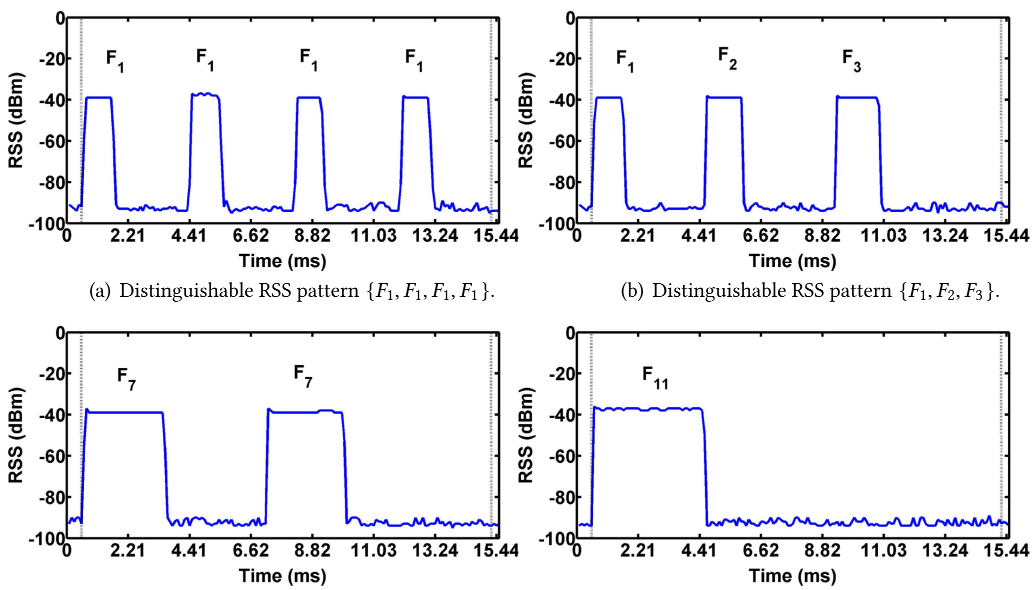
In the evaluation, we first run microbenchmark experiments to examine whether the ZigBee device can correctly capture every distinguishable RSS pattern in a single time slot and then measure the throughput and **bit error rate (BER)** of our CTC approach.

6.1 Microbenchmark Experiments

As presented in Section 4, we combine 15 distinguishable RSS signatures in different ways to generate 454 distinguishable RSS patterns, which are used to encode information. Table 3 lists 30 examples for RSS patterns in different levels. In this set of experiments, we control the LoRa device to generate all patterns in a round-robin fashion by putting different payloads in the LoRa packets and observe that the RSS measurements captured by the ZigBee device always match the

Table 3. Thirty Distinguishable RSS Patterns

	RSS Patterns		RSS Patterns		RSS Patterns		RSS Patterns
1	$\{F_1, F_1, F_1, F_1\}$	2	$\{F_1, F_1, F_{1X}, F_1\}$	3	$\{F_1, F_{1X}, F_1, F_1\}$	4	$\{F_{1X}, F_1, F_1, F_1\}$
5	$\{F_1, F_{1X}, F_{1X}, F_1\}$	6	$\{F_{1X}, F_{1X}, F_1, F_1\}$	7	$\{F_{1X}, F_1, F_{1X}, F_1\}$	8	$\{F_{1X}, F_{1X}, F_{1X}, F_1\}$
9	$\{F_1, F_1, F_1\}$	10	$\{F_1, F_1, F_2\}$	11	$\{F_1, F_1, F_3\}$	12	$\{F_1, F_1, F_4\}$
13	$\{F_1, F_1, F_5\}$	14	$\{F_1, F_1, F_6\}$	15	$\{F_1, F_1, F_7\}$	16	$\{F_1, F_2, F_1\}$
17	$\{F_1, F_2, F_2\}$	18	$\{F_1, F_2, F_3\}$	19	$\{F_1, F_2, F_4\}$	20	$\{F_1, F_2, F_5\}$
21	$\{F_1, F_1\}$	22	$\{F_1, F_2\}$	23	$\{F_1, F_3\}$	24	$\{F_1, F_4\}$
25	$\{F_1, F_5\}$	26	$\{F_1\}$	27	$\{F_2\}$	28	$\{F_3\}$
29	$\{F_4\}$	30	$\{F_5\}$				



(a) Distinguishable RSS pattern $\{F_1, F_1, F_1, F_1\}$.

(b) Distinguishable RSS pattern $\{F_1, F_2, F_3\}$.

(c) Distinguishable RSS pattern $\{F_7, F_7\}$.

(d) Distinguishable RSS pattern $\{F_{11}\}$.

Fig. 22. Four example RSS traces. Each trace lasts 15 ms.

design. Figure 22 shows four example RSS traces following the distinguishable RSS, $\{F_1, F_1, F_1, F_1\}$, $\{F_1, F_2, F_3\}$, $\{F_7, F_7\}$, and $\{F_{11}\}$, respectively. By comparing the RSS measurements and pattern design, we confirm that all distinguishable RSS patterns generated by the LoRa device can be effectively identified by the ZigBee device.

6.2 Throughput and BER

In this set of experiments, we measure the throughput and BER of our CTC approach. We randomly generate 550 bytes, control the LoRa device to encode and transmit them, and measure the throughput and BER on the ZigBee device after it decodes them. We repeat the experiments 20 times. Figure 23 plots the CDF of the measured throughput. The measured throughput ranges from 555.71 to 576.80 bps with a mean value of 571.52 bps. The measured throughput values are very close to our theoretical maximum CTC throughput of $log_2^{454} * \frac{1000}{15} = 588.44$ bps. The averaged throughput is 2.87% less than the theoretical value. The results show the efficiency of the encoding and decoding processes of our CTC approach. Figure 24 shows the CDF of BER. The BER ranges

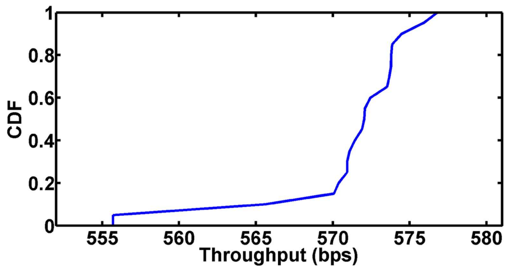


Fig. 23. CDF of CTC throughput.

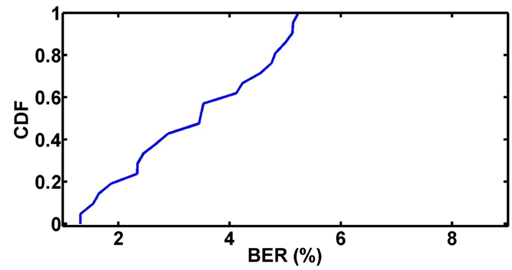
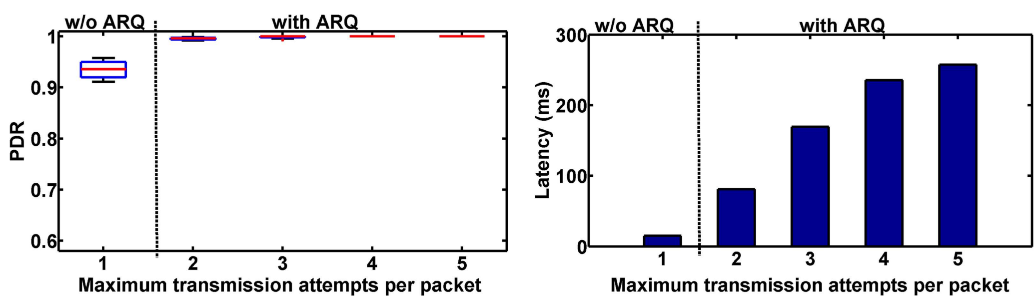


Fig. 24. CDF of CTC BER.



(a) Boxplot of PDR.

(b) Average latency.

Fig. 25. Boxplot of PDR and average latency with and without our ARQ approach.

from 1.32% to 5.23%, and the average value is 3.45%. The low BER values demonstrate the high reliability of our CTC approach.

6.3 Reliable CTC

We run experiments to compare the reliability of our CTC solution with and without our ARQ approach. We configure the LoRa device to transmit 450 CTC packets to the Zigbee devices that are two hops away from the sink and repeat the experiments eight times. Each slotframe has 50 time slots. We use the first time slot in each slotframe for CTC transmission and vary the maximum retransmission attempts from 0 to 4. Figures 25(a) and 25(b) plot the boxplot of **packet delivery ratio (PDR)** and average latency with and without our ARQ approach, respectively. As Figure 25(a) shows, the median PDR increases from 93.78% to 99.56% when our ARQ approach allows two transmission attempts for each CTC packet. The median PDR increases to 100% when more transmission attempts are allowed for each CTC packet. All PDRs become 100% when at most four transmission attempts are allowed for each CTC packet. The latency of all missing packets are not counted. As Figure 25(b) shows, the average latency is about 15.00 ms when the retransmission is disabled. The average latency increases to 81.18 ms when two transmission attempts are allowed for each CTC packet. It further increases to 169.41 ms, to 235.59 ms, and then to 257.65 ms when three, four, and five transmission attempts are allowed for each CTC packet, respectively. The results show that it is beneficial to enable ARQ to enhance the CTC reliability at the cost of slightly increased latency.

To investigate the impact of external interference on our CTC solution, we repeat the experiments under controlled WiFi interference, generated by a jammer running JamLab [5]. Figure 26 plots the boxplot of PDR with and without our ARQ approach in a noisy environment. The median PDR increases from 85.78% to 97.77% when two transmission attempts are allowed for each

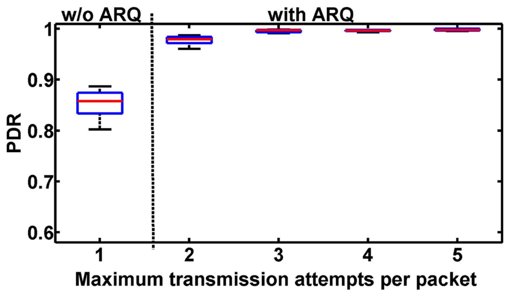


Fig. 26. Boxplot of PDR with and without our ARQ approach under controlled interference.

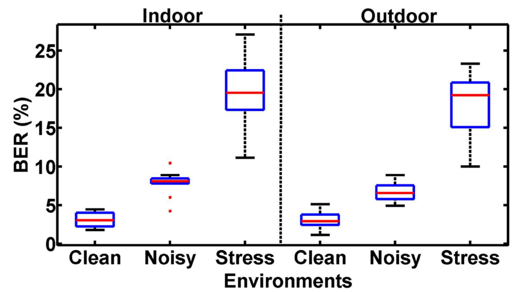


Fig. 27. Boxplot of BER under the clean, noisy, and stress testing conditions.

CTC packet. It further increases to 99.56%, to 99.66%, and then to 99.78% when three, four, and five transmission attempts are allowed for each CTC packet, respectively. The results show that our CTC solution with ARQ can consistently provide reliable communication from LoRa to ZigBee.

6.4 Impact of Interference

In this set of experiments, we measure the BER of our CTC approach under different wireless conditions with indoor non-line-of-sight and outdoor line-of-sight settings. We configure a jammer that runs JamLab [5] to generate controlled interference and vary the distance between the jammer and our LoRa and ZigBee devices to create three wireless conditions (clean, noisy, and stress test). We let the LoRa device transmit 450 CTC packets in each experiment and repeat the experiments 10 times under each wireless condition. As Figure 27 shows, the median BER values of our CTC approach are 3.00%, 8.11%, and 19.56% under the clean, noisy, and stress test conditions in indoor environments, respectively. In outdoor environments, the median BER values are 2.89%, 6.56%, and 19.22% under the clean, noisy, and stress conditions, respectively. The results show that our CTC solution performs well with low BER when facing moderate interference. The results also indicate that the transmission scheduling algorithm should schedule the CTC and regular transmissions between ZigBee devices to use different channels because of the high BER when facing strong interference.

6.5 Impact of Different RSS Patterns

In this set of experiments, we evaluate the impact of different RSS patterns on the performance of our CTC approach. We select six RSS patterns, three of which consist of the RSS signatures with adjacent indexes ($\{F_1, F_1\}$, $\{F_1, F_2, F_3\}$, and $\{F_5, F_6\}$). The rest consist of the RSS signatures with non-adjacent indexes ($\{F_1, F_7\}$, $\{F_2, F_8\}$, and $\{F_1, F_9\}$). We let the LoRa device transmit 450 CTC packets using each RSS pattern and repeat the experiments 10 times. Figures 28 and 29 plot the boxplot of BER when using different RSS patterns in the clean and noisy wireless conditions. The median BER values range from 2.89% to 4.22% in the clean wireless condition, whereas they increase to [6.78%, 7.78%] with the presence of the controlled interference. We did not observe significant improvements by using the RSS signatures with non-adjacent indexes. The results indicate that the measured RSS lengths used by our CTC approach are accurate enough.

6.6 Impact of LoRa Signals on ZigBee Communication

In this set of experiments, we evaluate the impact of the LoRa signals on the communication between ZigBee devices. We configure the LoRa device to use $SF = 9$, vary its BW from 203 to 1,625 kHz, and measure the **packet reception ratio (PRR)** of a ZigBee link where the transmitter

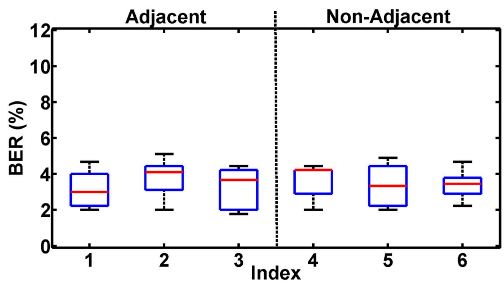


Fig. 28. Boxplot of BER under six RSS patterns in the clean condition.

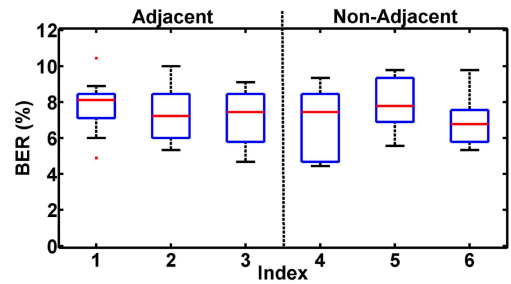


Fig. 29. Boxplot of BER under six RSS patterns in the noisy condition.

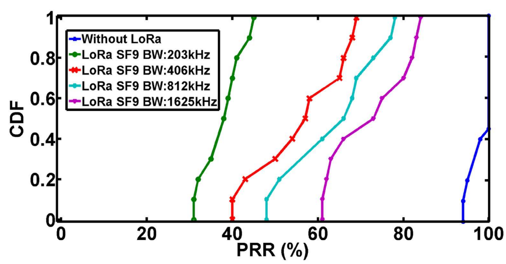


Fig. 30. CDF of the ZigBee link's PRR when the LoRa device uses different *BW*.

sends 100 packets. We repeat the experiments 10 times. Figure 30 plots the CDF of the ZigBee link's PRR when the LoRa device uses different *BW*. The median PRR of the ZigBee link is 100% without the LoRa transmissions. The median PRR are 74.00%, 67.00%, 57.50%, and 38.50% when the LoRa device transmits with the *BW* of 1,625, 812, 406, and 203 kHz, respectively. The results show that the LoRa signals significantly interfere the communication between ZigBee devices and the PRR decreases sharply when the LoRa channel becomes wider with increased *BW*, which emphasizes the importance of scheduling the CTC and regular transmissions between ZigBee devices to use different channels.

7 RELATED WORKS

With the unprecedented proliferation of heterogeneous wireless technologies and wireless devices, there exist severe wireless coexistence and management problems with the devices sharing the same unlicensed ISM bands. Early studies show that enabling CTC among heterogeneous devices can effectively address those problems and significantly improve the network performance. For instance, Zhou et al. [48] developed ZiFi, which allows an embedded device to reduce its power consumption by using a low-power ZigBee radio to detect nearby WiFi APs. Hao et al. [17] and Yu et al. [43] used WiFi signals to achieve time synchronization among ZigBee devices. Gawlowicz et al. [12] leveraged CTC to coordinate the coexistence between LTE-U and WiFi devices in the 5-GHz band. CTC has seen appreciable advancement in recent years. Significant efforts have been made to enable the CTC among ZigBee, WiFi, and Bluetooth devices in the 2.4 GHz band [6–8, 14–16, 21, 24, 27, 38–47]. Among those solutions, the wireless devices' capability of sensing the RSS in the air has been used to enable CTC between heterogeneous devices, and the most widely used CTC scheme is to encode information on the temporal or amplitude dimension. For instance, Kim and He [24] and Kim et al. [25] enabled the CTC from WiFi to ZigBee by shifting the appearance of WiFi beacons in a temporal dimension to embed different symbols. Chebrolu and

Dhekne [6] achieved the same goal by building an alphabet set and modulating the WiFi energy profile lengths to convey messages. In addition, a CTC message (sequence of bits) is mapped to an alphabet and a packet corresponding to that size is transmitted at a predetermined rate. Yin et al. [42] designed C-Morse, which modulates the timing of WiFi packets to construct special energy patterns. Kim et al. [24, 25] proposed a method to optimize the CTC throughput over a noisy channel. More recently, Guo et al. [15] developed the cross-demapping technique, which achieves the physical-level CTC from ZigBee to WiFi and leaves the computation overhead to the receiver. Chen and Gao [7] proposed to reserve part of the spectrum for narrow-band devices to perform concurrent transmissions and allowed a WiFi device to detect ZigBee signals without introducing extra traffic. Li and He [27] developed WEBee, which emulates the ZigBee signals in the physical layer on commercial off-the-shelf (COTS) WiFi devices, and Wang et al. [38] proposed SymBee, which achieves symbol-level CTC from ZigBee to WiFi. Jiang et al. [21] developed XBee, which interprets a ZigBee frame by observing the bit patterns obtained at the Bluetooth receiver. Chi et al. [8] proposed a communication framework that enables multiple concurrent communication among WiFi and Bluetooth devices. Shi et al. [35] developed an approach that enables the CTC from LoRa to ZigBee in the sub-1-GHz bands by detecting individual LoRa chirps. Unfortunately, those solutions are not directly applicable to send messages from a long-range LoRa radio to a ZigBee device because of the unique characteristics of LoRa radios operating in the 2.4 GHz ISM band. In contrast to previous studies among ZigBee, WiFi, and Bluetooth, this article investigates the CTC from LoRa to ZigBee; to the best of our knowledge, it represents the first systematic study of the characteristics of LoRa in the 2.4 GHz ISM band from a CTC's point of view. Our work is therefore orthogonal and complementary.

We developed an approach that enables the CTC from LoRa to ZigBee in the sub-1-GHz bands by encoding the LoRa packet payload with specific bytes [35]. The corresponding LoRa chirps (different features) can be detected by the ZigBee device through sampling the RSS. Specifically, the ZigBee device detects the sudden RSS value drop caused by each LoRa chirp and uses the time intervals of all RSS value drops to encode and decode information. Unfortunately, this approach is not applicable in the 2.4 GHz band because the LoRa device transmits much faster and the ZigBee device is not capable of sampling RSS frequent enough to detect individual LoRa chirps. For example, we have measured the RSS sampling rate of our ZigBee devices when they operate in the 2.4 Ghz band. The maximum RSS sampling rate is 11.33 kHz, and the time duration of transmitting a single LoRa chirp is 0.019 ms. The RSS sampling interval, 0.088 ms = 1/11.33 kHz, is longer than the time duration of transmitting a single LoRa chirp, and thus the ZigBee device is incapable of detecting the changes of each individual LoRa chirp in the 2.4 GHz band. To enable the CTC from LoRa to ZigBee in the 2.4 GHz band, we have developed this new approach, which encodes information using the sizes of the LoRa payloads. The ZigBee device uses the number of the consecutive RSS values higher than a threshold to decode information. Such an approach encodes less information in each time unit, which results in lower throughput if applied in the sub-1-GHz bands compared to LoRaBee [35]. Therefore, our two CTC approaches reported in this work and our earlier work [35] complement each other.

8 CONCLUSION

IEEE 802.15.4-based WSANs operate at low power, can be manufactured inexpensively, and have been adopted by the leading industrial WSAN standards (WirelessHART and ISA100). The current approach to implementing industrial WSANs relies on a multi-hop mesh network to deliver sensing data and control commands. However, a large and complex mesh network is hard to manage and inelastic to change once the network is deployed. In addition, flooding-based time synchronization and information dissemination introduce significant communication overhead to the network.

More importantly, the deliveries of urgent and critical information such as emergency alarms suffer long delay, because those messages must go through the hop-by-hop transport. A promising solution to overcome the limitations of using multi-hop mesh networks for industrial WSANs is to enable the direct messaging from a long-range radio to an IEEE 802.15.4 radio. Then messages can be delivered to field devices in a single-hop fashion. This article presents our study on enabling the CTC from LoRa to ZigBee using the energy emission of the LoRa radio in the 2.4 GHz band as the carrier to deliver information. Our CTC approach puts specific bytes in the payload of legitimate LoRa packets. The bytes are selected such that the corresponding information can be understood by the ZigBee devices through sampling the RSS. Experimental results show that our CTC approach provides reliable communication from LoRa to ZigBee with the throughput of up to 576.80 bps and the BER of up to 5.23% in the 2.4 GHz band.

REFERENCES

[1] Nicola Accettura, Luigi Alfredo Grieco, Gennaro Boggia, and Pietro Camarda. 2011. Performance analysis of the RPL routing protocol. In *Proceedings of the 2011 IEEE International Conference on Mechatronics*. IEEE, Los Alamitos, CA, 767–772.

[2] LoRa Alliance. 2015. LoRa. Retrieved November 24, 2021 from https://lora-alliance.org/.

[3] Miltiades E. Anagnostou and Emmanuel N. Protonotarios. 1986. Performance analysis of the selective repeat ARQ protocol. *IEEE Transactions on Communications* 34, 2 (1986), 127–135.

[4] Aloÿs Augustin, Jiazi Yi, Thomas Heide Clausen, and William Townsley. 2016. A study of LoRa: Long range and low power networks for the Internet of Things. *Sensors* 16 (10 2016), 1466.

[5] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Romer, and Marco Zuniga. 2011. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Proceedings of the ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN'11)*. ACM, New York, NY, 175–186.

[6] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: Communication through energy sensing. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom'09)*. ACM, New York, NY, 85–96.

[7] Ruirong Chen and Wei Gao. 2019. Enabling cross-technology coexistence for extremely weak wireless devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'19)*. IEEE, Los Alamitos, CA, 253–261.

[8] 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 ACM Conference on Embedded Networked Sensor Systems (SenSys'16)*. ACM, New York, NY, 245–258.

[9] Semtech Corporation. 1960. Semtech. Retrieved November 24, 2021 from https://www.semtech.com/.

[10] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. 2015. Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15)*. ACM, New York, NY, 337–350.

[11] Raspberry Pi Foundation. 2012. Raspberry Pi 3 Model B. Retrieved November 24, 2021 from https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[12] Piotr Gawlowicz, Anatolij Zubow, and Adam Wolisz. 2018. Enabling cross-technology communication between LTE unlicensed and WiFi. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'18)*. IEEE, Los Alamitos, CA, 144–152.

[13] Fieldcomm Group. 2007. WirelessHART. Retrieved November 24, 2021 from https://fieldcommgroup.org/technologies/hart/hart-technology.

[14] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu, and Omprakash Gnawali. 2018. ZIGFI: Harnessing channel state information for cross-technology communication. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'18)*. IEEE, Los Alamitos, CA, 360–368.

[15] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Zihao Yu, and Yunhao Liu. 2019. LEGO-Fi: Transmitter-transparent CTC with cross-demapping. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'19)*. IEEE, Los Alamitos, CA, 2125–2133.

[16] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. WiZig: Cross-technology energy communication over a noisy channel. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'17)*. IEEE, Los Alamitos, CA, 1–9.

[17] Tian Hao, Ruogu Zhou, Guoliang Xing, Matt W. Mutka, and Jiming Chen. 2014. WizSync: Exploiting Wi-Fi infrastructure for clock synchronization in wireless sensor networks. *IEEE Transactions on Mobile Computing* 13, 6 (June 2014), 1379–1392.

[18] HART. 1986. HART Communication Protocol and Foundation (Now the FieldComm Group). Retrieved September 28, 2018 from https://fieldcommgroup.org/.

[19] IMST. 2019. SK-iM282A. Retrieved November 24, 2021 from https://wireless-solutions.de/products/starterkits/sk-im282a.html.

[20] ISA100 Wireless. 2005. ISA 100. Retrieved November 24, 2021 from http://www.isa100wci.org/.

[21] Wenchao Jiang, Song Min Kim, Zhijun Li, and Tian He. 2018. Achieving receiver-side cross-technology communication with cross-decoding. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom'18)*. ACM, New York, NY, 639–652.

[22] Hyung-Sin Kim, Jeonggil Ko, David E. Culler, and Jeongyeup Paek. 2017. Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey. *IEEE Communications Surveys Tutorials* 19, 4 (2017), 2502–2525.

[23] Seohyang Kim, Hyung-Sin Kim, and Chongkwon Kim. 2019. ALICE: Autonomous link-based cell scheduling for TSCH. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. ACM, New York, NY, 121–132. https://doi.org/10.1145/3302506.3310394

[24] Song Min Kim and Tian He. 2015. FreeBee: Cross-technology communication via free side-channel. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom'15)*. ACM, New York, NY, 317–330.

[25] Song Min Kim, Shigemi Ishida, Shuai Wang, and Tian He. 2017. Free side-channel cross-technology communication in wireless networks. *IEEE/ACM Transactions on Networking* 25, 5 (2017), 2974–2987.

[26] Jun Li and Yiqiang Q. Zhao. 2009. Resequencing analysis of stop-and-wait ARQ for parallel multichannel communications. *IEEE/ACM Transactions on Networking* 17, 3 (2009), 817–830.

[27] Zhijun Li and Tian He. 2017. WEBee: Physical-layer cross-technology communication via emulation. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom'17)*. ACM, New York, NY, 2–14.

[28] Shu Lin, Daniel J. Costello, and Michael J. Miller. 1984. Automatic-repeat-request error-control schemes. *IEEE Communications Magazine* 22, 12 (1984), 5–17.

[29] Jane Liu. 2000. *Real-Time Systems*. Prentice Hall, Upper Saddle River, NJ. http://www.cse.hcmut.edu.vn/~thai/books/200020_20Liu-20Real20Time20Systems.pdf.

[30] Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. 2016. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proceedings of the IEEE: Special Issue on Industrial Cyber Physical Systems* 104, 5 (2016), 1013–1024.

[31] Memsic. 2004. TelosB: TelosB Mote Platform, Datasheet Provided by MEMSIC Inc. Retrieved November 24, 2021 from http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb5fdatasheet.pdf.

[32] Semtech. 2017. Long Range, Low Power, 2.4 GHz Transceiver with Ranging Capability. Retrieved November 24, 2021 from https://www.mouser.com/datasheet/2/761/DS_SX1280-1_V2.2-1511144.pdf.

[33] Semtech. 2017. Semtech's New SX1280/SX1281 Wireless RF Chips. Retrieved November 24, 2021 from https://www.digikey.com/catalog/en/partgroup/sx1280-and-sx1281/68471.

[34] Junyang Shi, Xingjian Chen, and Mo Sha. 2019. Enabling direct messaging from LoRa to ZigBee in the 2.4 GHz band for industrial wireless networks. In *Proceedings of the International Conference on Industrial Internet (ICII'19)*. IEEE, Los Alamitos, CA.

[35] Junyang Shi, Di Mu, and Mo Sha. 2019. LoRaBee: Cross-technology communication from LoRa to ZigBee via payload encoding. In *Proceedings of the 2019 IEEE 27th International Conference on Network Protocols (ICNP'19)*.

[36] BU Testbed. 2017. Testbed at the State University of New York at Binghamton. Retrieved November 24, 2021 from http://www.cs.binghamton.edu/7emsha/testbed.

[37] Don Towsley. 1979. The Stutter Go Back-N ARQ protocol. *IEEE Transactions on Communications* 27, 6 (1979), 869–875.

[38] Shuai Wang, Song Min Kim, and Tian He. 2018. Symbol-level cross-technology communication via payload encoding. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS'18)*. IEEE, Los Alamitos, CA, 500–510.

[39] Shuai Wang, Zhimeng Yin, Song Min Kim, and Tian He. 2017. Achieving spectrum efficient communication under cross-technology interference. In *Proceedings of the International Conference on Computer Communication and Networks (ICCCN'17)*. IEEE, Los Alamitos, CA, 1–8.

[40] Wei Wang, Xin Liu, Yao Yao, Yan Pan, Zicheng Chi, and Ting Zhu. 2019. CRF: Coexistent routing and flooding using WiFi packets in heterogeneous IoT networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'19)*. IEEE, Los Alamitos, CA, 19–27.

[41] Shengrong Yin, Qiang Li, and Omprakash Gnawali. 2015. Interconnecting WiFi devices with IEEE 802.15.4 devices without using a gateway. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS'15)*. IEEE, Los Alamitos, CA, 127–136.

[42] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. 2017. C-Morse: Cross-technology communication with transparent Morse coding. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'17)*. IEEE, Los Alamitos, CA, 1–9.

[43] 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 International Conference on Embedded Wireless Systems and Networks (EWSN'18)*. 135–144.

[44] Xinyu Zhang and Kamg G. Shin. 2013. Cooperative carrier signaling: Harmonizing coexisting WPAN and WLAN devices. *IEEE/ACM Transactions on Networking* 21, 2 (2013), 426–439.

[45] Xinyu Zhang and Kamg G. Shin. 2013. Gap Sense: Lightweight coordination of heterogeneous wireless devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13)*. IEEE, Los Alamitos, CA, 3094–3101.

[46] Yifan Zhang and Qun Li. 2013. HoWiES: A holistic approach to ZigBee assisted WiFi energy savings in mobile devices. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13)*. IEEE, Los Alamitos, CA, 1366–1374.

[47] Xiaolong Zheng, Yuan He, and Xiuzhen Guo. 2018. StripComm: Interference-resilient cross-technology communication in coexisting environments. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'18)*. IEEE, Los Alamitos, CA, 171–179.

[48] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. 2010. Zifi: Wireless LAN discovery via ZigBee interference signatures. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom'10)*. ACM, New York, NY, 49–60.

[49] ZigBee Alliance. 2002. ZigBee. Retrieved November 24, 2021 from https://zigbeealliance.org/.

Received June 2020; revised October 2021; accepted October 2021