# Coexistent Routing and Flooding Using WiFi Packets in Heterogeneous IoT Network

Wei Wang, Xin Liu, Yao Yao, Zicheng Chi, *Member, IEEE*, Yan Pan, and Ting Zhu

***Abstract*— Routing and flooding are important functions in wireless networks. However, until now routing and flooding protocols are investigated separately within the same network (i.e., a WiFi network or a ZigBee network). Moreover, further performance improvement has been hampered by the assumption of the harmful cross technology interference. In this paper, we present coexistent routing and flooding (CRF), which leverages the unique feature of physical layer cross-technology communication technique for concurrently conducting routing within the WiFi network and flooding among ZigBee nodes using a single stream of WiFi packets. We extensively evaluate our design under different network settings and scenarios. The evaluation results show that CRF i) improves the throughput of WiFi network by 1.12 times than the state-of-the-art routing protocols; and ii) significantly reduces the flooding delay in ZigBee network (i.e., 31 times faster than the state-of-the-art flooding protocol).**

***Index Terms*— Cross-technology communication, wireless communication, wireless network.**

## I. Introduction

ROUTING and flooding are important and fundamental functions in wireless networks. Routing is a protocol that forwards data from a source to a destination, while flooding delivers data from one node to all the other nodes inside the network. These two functions can be applied to support various applications such as disaster recovery, battlefield surveillance, smart homes, electric smart meters in smart cities, and internet access for communities. Routing is a fundamental function for data forwarding, while flooding is a fundamental operation for routing tree formation [1], data dissemination [2], node localization [3], and time synchronization [4].

Existing routing and flooding algorithms [5], [6] have demonstrated their effectiveness in achieving relatively high throughput, low latency, and high reliability in wireless networks. However, routing and flooding are normally treated as two different topics and investigated separately within the same network (i.e., a WiFi network or a ZigBee network). WiFi communications are treated as interference to the ZigBee network and vice versa. To mitigate the interference, researchers proposed various techniques [7], [8].

Manuscript received June 24, 2019; revised May 5, 2020; accepted July 22, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Andrews. Date of publication August 12, 2021; date of current version December 17, 2021. This work was supported by NSF under Grant CNS-1652669. *(Corresponding author: Ting Zhu.)*

Wei Wang, Xin Liu, Yao Yao, Yan Pan, and Ting Zhu are with the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250 USA (e-mail: ax29092@umbc.edu; xinliu1@umbc.edu; yaoyaoumbc@umbc.edu; yanpan@umbc.edu; zt@umbc.edu).

Zicheng Chi is with the Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115 USA (e-mail: z.chi@csuohio.edu).

Digital Object Identifier 10.1109/TNET.2021.3101949

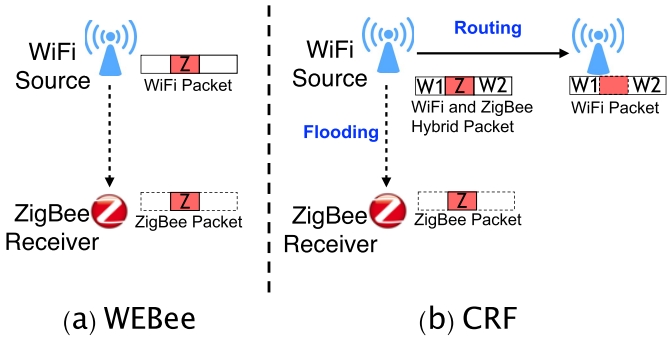



Fig. 1. **Difference between WEBee and our CRF.** (a) WEBee focuses on the physical layer design of WiFi to ZigBee communication. (b) Our CRF is the network layer design that enables coexistent routing in WiFi networks and flooding in ZigBee networks using the same stream of WiFi packets.

Instead of treating the communication in different networks as an interference, we explore how to leverage the unique features in cross-technology communication (CTC) for better performance. Our work is inspired by the recent advance in cross-technology communication (i.e., WEBee [9]), which uses WiFi to directly communicate with ZigBee devices without any modifications of hardware. To do this, the WiFi source carefully controls its payload so that the transmitted RF signal is similar to the ZigBee signal. At the receiver side, the WiFi preamble, the header and the trailer will be considered as noise and ignored while the payload will be recognized as the legitimate ZigBee packet. However, in WEBee, the WiFi to WiFi communication is destroyed due to the change of the WiFi payload. In addition, due to the limitation of the 802.11 physical layer, the emulated signal is not exactly the same as the desired ZigBee signal. As a result, the communication reliability from WiFi to ZigBee is not guaranteed.

Different from WEBee (see Figure 1), we propose a novel physical layer design that enables coexistent WiFi to WiFi and WiFi to ZigBee communications using the same string of WiFi packets. By leveraging the unique concurrent communication properties of our physical layer design, we introduce a new direction for routing and flooding algorithms – coexistent routing and flooding (CRF), which concurrently conducts routing within the WiFi network and flooding among ZigBee nodes using a single stream of WiFi packets.

With the exponentially increasing number of internet-of-things (IoT) devices [10], our approach has the following advantages: 1) our physical layer design does not need to change the hardware or firmware in commodity technologies – a feature that significantly reduces the deployment (and maintenance) costs and enables seamless operation with the existing infrastructure; 2) our network layer design enables the coexistent routing and flooding, which effectively

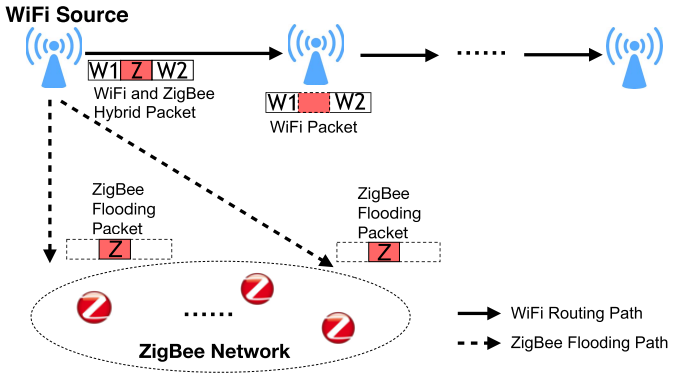



Fig. 2. **Network Architecture.** The WiFi source embeds a ZigBee flooding packet ($Z$) into its own packet for broadcasting. Other WiFi devices can receive the original WiFi data ($W_1$ and $W_2$) while the ZigBee nodes simultaneously receive the ZigBee flooding packet ($Z$).

avoids the cross-technology interference that happens when conducting the WiFi routing and ZigBee flooding separately in co-located WiFi and ZigBee networks. Therefore, the throughput of WiFi routing can be increased; 3) our approach can provide much higher reliability and lower latency when flooding in ZigBee networks. This is because i) the transmission power (TR) of WiFi is much larger than the TR of ZigBee; and ii) unlike ZigBee devices that wake up for a very short time duration, WiFi devices normally have a much longer wake up duration.

Specifically, our major contributions are as follows:

- This is the first work that seamlessly integrates the routing and flooding functions to create a win-win situation for both WiFi networks (i.e., improve the routing throughput) and ZigBee networks (i.e., significantly reduce the dissemination delay and increase flooding reliability). The features we provide and the challenges we address in this coexistent communication-based design are generic and have the potential to be applied in other heterogeneous networks.
- We extensively evaluated our design under different network settings and scenarios. The evaluation results show that CRF i) improves the throughput of WiFi network by 1.12 times than the state-of-the-art routing protocols; and ii) significantly reduces the flooding delay in ZigBee network (i.e., 31 times faster than the state-of-the-art flooding protocol).

## II. Network Architecture & Applications

In this section, we briefly introduce the network architecture and potential applications of CRF.

### A. Network Architecture

Figure 2 shows an example of the network architecture of CRF. The WiFi source uses a single stream of WiFi packets to conduct routing within the WiFi networks and flooding in ZigBee networks. Specifically, the WiFi broadcasts the packets with embedded ZigBee flooding information by leveraging ZigBee signal emulation techniques. The WiFi destination can receive the WiFi packets and get the original WiFi data. Meanwhile, the ZigBee nodes can sense the emulated signals and receive the flooding packets.

With the exponentially increasing number of IoT devices, WiFi devices and ZigBee nodes will be densely co-located. By using our CRF technique, WiFi devices can concurrently transmit i) WiFi packets to the WiFi destination; and ii) ZigBee flooding packets to ZigBee nodes. The ZigBee nodes use the channel that is overlapped with the current WiFi channel. Each ZigBee node follows its own working schedule to switch to the **the active state** or **the dormant state**. In the active state, it senses and receives packets from WiFi devices and its neighboring nodes while in the dormant state, it turns off all its function to save energy.

### B. Potential Applications

In modern society, more and more IoT devices have been widely deployed to support smart city and smart building applications, such as Building Automation, Real-time Energy Management, Environmental Monitoring and Management, Lighting System Management, Fast-reaction Disaster Management, etc [11], [12]. These IoT devices are required to provide reliable and low latency service in order to support the requirements of these smart city and smart building applications. For example, when a disaster happens in the smart city, the servers are required to manage and control each IoT device with extremely low latency so as to address and mitigate the hazard. To reduce the energy consumption in the smart building, the servers are required to control each device in real time while still meeting the dynamic requirements of the users. However, with the exponentially increasing number of IoT devices and the huge amount of generated data, these IoT devices suffer high interference and need to frequently back-off according to CSMA, which introduces huge delay to the IoT network. CRF has the potential to overcome this problem. Specifically, since WiFi coverage is now almost ubiquitous in smart city and smart building, it provides an obvious choice for IoT connectivity and management. By leveraging the most recent cross-technology communication technique, while WiFi is doing routing, the flooding information can be concurrently transmitted to ZigBee nodes. As more and more data are transmitted through WiFi [13], CRF has the potential to support real-time management and control requirements of IoT devices in smart city and smart building scenarios.

## III. Challenges and Solutions

To enable coexistent routing and flooding, we face the following challenges:

### A. How to Achieve Coexistent Communications From WiFi to WiFi and WiFi to ZigBee Using WiFi Packets?

Recent ZigBee signal emulation techniques enable communication from WiFi to ZigBee at the expense of sacrificing the WiFi packets [9], [14]. Specifically, the WiFi device controls the payload of its packet to emulate ZigBee signal. However, since the payload is changed, the received packet at the WiFi destination side is meaningless. In this paper, we develop a Data Extraction technique to address this challenge (detailed in Section V).

### B. How to Preserve the Throughput in the WiFi Network and Protect the Embedded Flooding Information?

Retransmission is one of the major problems that affects the network throughput. Different from traditional wireless

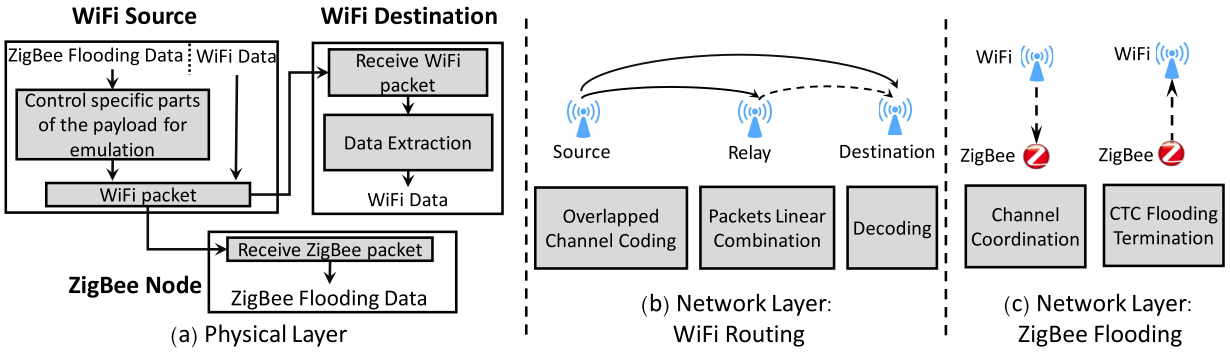



Fig. 3. Design overview.

networks, the retransmissions from the WiFi source to the destination not only decrease network throughput but also affect the flooding for ZigBee nodes. Specifically, due to the limited coverage range of a single WiFi source, the destination also needs to conduct flooding. Since the flooding packets are embedded in the WiFi packets, the retransmissions will therefore increase the flooding delay at the destination side. In our design, we introduce an Overlapped Channel Coding technique to overcome this problem (detailed in Section VI).

### *C. How to Improve the Flooding Reliability and Fully Leverage the WiFi to ZigBee Communication Capability?*

First, due to unreliable radio links from WiFi to ZigBee, the retransmissions of flooding packets introduce flooding delay. More importantly, the current physical layer emulation technique cannot perfectly emulate the ZigBee signal, which reduces the communication reliability from WiFi to ZigBee. Specifically, there are three types error during the signal emulation process: i) According to the ZigBee signal, the WiFi device should select the nearest QAM constellation point. The distance between the selected QAM point and the desired point will introduce signal distortion. ii) Since the WiFi uses Cyclic prefix (CP) to eliminate the Inter-Symbol Interference (ISI) and the Inter-Carrier Interference (ICI) while the ZigBee signal does not have cyclic prefix, the emulated ZigBee signal cannot be exactly the same as the desired ZigBee signal. iii) Since WiFi needs to use 4 symbols to emulate one ZigBee symbol ($16\mu$s), the discontinuity between each WiFi symbol also introduces distortions. Normally, since the ZigBee device uses a 32 Pseudo-random Noise Chip Sequence to improve the communication reliability, these distortions can be tolerated. However, the unreliable radio links also introduces distortions to the communication from WiFi to ZigBee. Therefore, in real-world scenarios, the WiFi to ZigBee communication is sensitive to changes of wireless environment and the communication reliability is relatively low (i.e., around 60% packet reception ratio [9], [14]). Second, since a ZigBee node has multiple channels that are overlapped with WiFi and it can only receive the packets on its current channel, it is wasting the communication capability of the WiFi devices. In Section VII-A, we introduce a flooding channel coordination mechanism to improve the flooding reliability and fully leverage the CTC capabilities to mitigate the imperfect emulations.

### *D. How to Terminate the Flooding?*

Current ZigBee to WiFi communication technique mainly use packet-level modulation [15], [16]. Transmitting a termination command (i.e., ACKs) back to the WiFi sender requires the ZigBee node to broadcast a large amount of ZigBee packets. As the number of ZigBee nodes increases, too many ACKs will introduce huge amount of interference on the ongoing WiFi traffic. Moreover, traditional silence-based feedback scheme cannot be used in such scenario due to the interference from the WiFi traffic [17], [18]. In Section VII-D, we introduce a CTC flooding termination scheme to overcome this challenge.

## IV. Design Overview

Our goal is to achieve coexistent routing and flooding in heterogeneous IoT networks (e.g., WiFi and ZigBee). Figure 3 shows the high-level design of CRF, which can be divided into three parts:

### *A. PHY Layer (See Figure 3 (a))*

The physical layer design enables the coexistent communications from WiFi to WiFi and WiFi to ZigBee using the same WiFi packets. Based on the ZigBee flooding data, the WiFi device controls the specific parts of its payload to emulate the corresponding ZigBee signal. The original WiFi data will be embedded in the remaining parts of the payload. For the ZigBee node, the emulated ZigBee signal is received as the flooding packet. For the WiFi destination, it applies the data extraction technique to get the original WiFi and ZigBee flooding data.

### *B. Network Layer: WiFi Routing (See Figure 3 (b))*

The objective of our routing scheme is to preserve the throughput and protect the embedded flooding information. To achieve this goal, the WiFi source applies an overlapped channel coding technique to encode the WiFi packets. The relay will linearly combine the received coded packets and forward them to the destination. At the destination side, even if the received packets are corrupted, the flooding information can still be decoded.

### *C. Network Layer: ZigBee Flooding (See Figure 3 (c))*

Our flooding scheme contains two parts: Channel Coordination and Flooding Termination. In CRF, the WiFi source

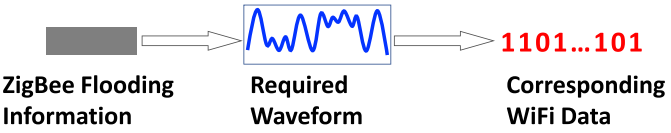



Fig. 4. Based on the ZigBee flooding information, the WiFi source calculates the required waveform and generates the corresponding WiFi data.

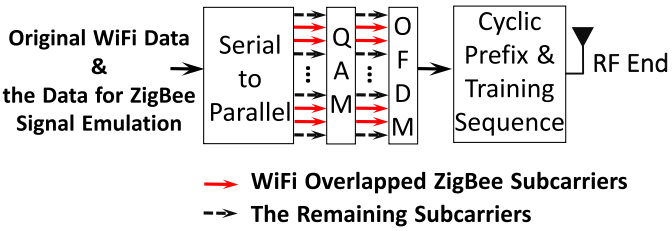



Fig. 5. The WiFi source can transmit the WiFi data using the remaining subcarriers.

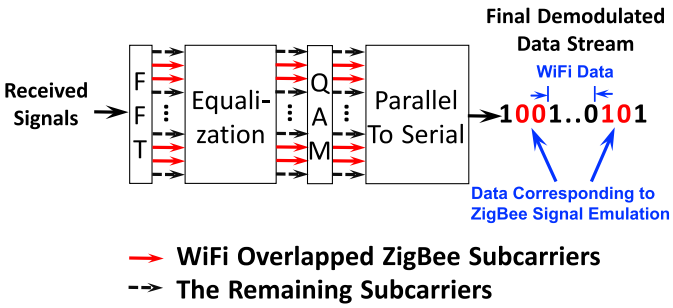



Fig. 6. Since the position of the data corresponding to ZigBee signal emulation is predictable, the WiFi destination can extract the WiFi data by ignoring those positions.

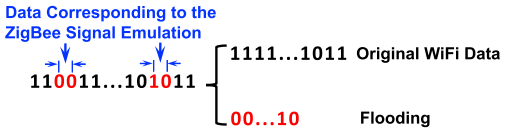



Fig. 7. WiFi destination extracts the data corresponding to the ZigBee signal emulation.

applies a channel coordination scheme to improve the flooding reliability and reduce the flooding delay. To terminate the flooding, only a limit number of ZigBee nodes are required to acknowledgments back to the WiFi source, which significantly reduces the interference on the WiFi network.

## V. Physical Layer of CRF

In this section, we introduce how to achieve coexistent WiFi-to-WiFi and WiFi-to-ZigBee communications.

### A. WiFi Data Transmission and Extraction

As shown in Figure 4, to transmit the flooding information to ZigBee nodes, the WiFi source controls the data in its payload based on the ZigBee flooding data. However, the original WiFi data cannot be directly recovered at the WiFi destination side since the payload is changed. To extract WiFi data from the received WiFi packets, we leverage the WiFi orthogonal frequency-division multiplexing (OFDM) feature that each subcarrier is parallel to the others. Specifically, since the ZigBee channel is overlapped with 7 WiFi subcarriers, the WiFi source can only transmits the corresponding flooding data in these subcarriers. The remaining subcarriers can still be used to transmit the original WiFi data.

Specifically, as shown in Figure 5, the WiFi source first divides the original WiFi data and the data for the ZigBee signal emulation into $N$ parallel pieces, which is modulated in $N$ subcarriers using the Quadrature Amplitude Modulation (QAM) scheme (e.g., $N = 48$). The data for ZigBee signal emulation is modulated in the overlapped ZigBee subcarriers while the remaining subcarriers are used to modulate the original WiFi data. Then, the WiFi source applies OFDM by utilizing Inverse Fast Fourier Transform (IFFT). Finally, a cyclic prefix and training sequence are applied to reduce Intersymbol Interference (ISI) and conduct synchronization between the source and destination.

At the WiFi destination side, the WiFi device applies the inverse process to get the WiFi data while the data for ZigBee signal emulation is still remain in the predictable positions, which is shown in Figure 6. To extract the WiFi data, the WiFi destination ignores the bits that are from the WiFi overlapped ZigBee channels. Similarly, the WiFi destination can extract the data corresponding to the ZigBee signal emulation by using the similar approach, which is shown in Figure 7

**Algorithm 1** WiFi Data Extraction Algorithm

**Input:** Demodulated WiFi data stream $H(N)$ and its length $N$; Data length from the start of load to the beginning of data for Zigbee emulation $a$; Data length corresponding to emulation $b$;Data interval between two data for Zigbee emulation $c$;Subcarrier number of one WiFi Channel T.

**Output:** Original WiFi data Stream *W(M).*

1: $X = N/T$;
2: $M = 0$;
3: **for** $x = 0$: $X - 1$ **do**
4: **for** $t = 1$: $T$ **do**
5: **if** $t \in (a,\ a+\ b] \cup (a\ +b+c,\ a+2b\ +\ c]$ **then**
6: break;
7: **else**
8: $M\ = M + 1$;
9: $W(M) = H(x{*}T + t)$;
10: **end if**
11: **end for**
12: **end for**

To show our approach clearly, we define the final demodulated data steam as $H(n)$. Then, the WiFi destination extracts the original WiFi data $W(m)$ based on Algorithm 1. According to the received WiFi data stream, the WiFi destination calculates the packet length and finds the positions for ZigBee data emulation (lines 1 to 5). Then, it ignores the data in those positions and the data in the remaining positions is the original WiFi data (lines 6 to 10). We are aware that in real world settings, the scrambler seed of a WiFi device affects the positions of the received data stream. However, the destination can easily apply the descrambling to get the original WiFi data.

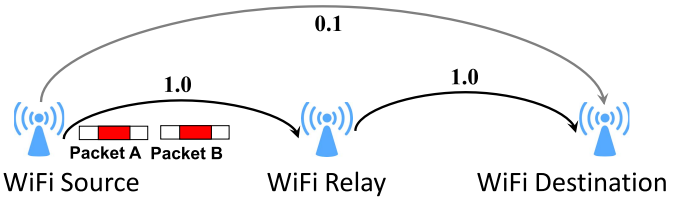



Fig. 8. The WiFi source can transmit the packets with embedded flooding information directly. It can also utilize the relay to conduct reliable transmissions.

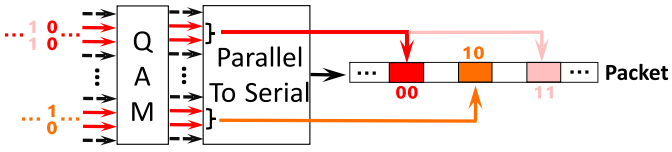



Fig. 9. The emulation data blocks are in the predictable positions.

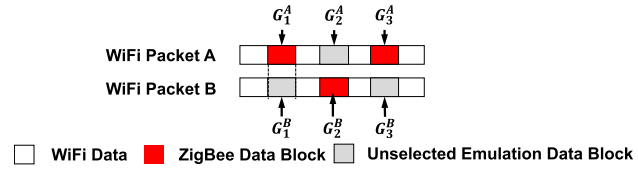



Fig. 10. An example of the coded packet.

## VI. Network Layer: WiFi Routing

In this section, we first discuss the limitations of existing routing algorithms and then describe our overlapped channel coding technique.

### A. Limitations of Existing Routing in Cross-Technology Network

Before introducing the overlapped channel coding, it is helpful to consider a simple WiFi network in Figure 8. The WiFi source transmits the packet $A$ and $B$ with embedded flooding information to the destination. To reduce the number of transmissions, it not only leverages the 2-hop route through the WiFi relay but also leverages the possibilities to directly deliver the packets to the destination. However, due to unreliable radio links, the directly received packets may be partially corrupted. Therefore, in most cases, the WiFi relay has to conduct forwarding, which reduces the throughput and ignores the fact that the directly received packets may have some correct parts. This solution may be suitable for traditional wireless networks (i.e., only contains one kind of device). However, in cross-technology networks, this solution will hamper the network performance. First, the WiFi packets contain the flooding information, which requires the WiFi to use parts of its subcarriers to conduct cross-technology communication. Therefore, if the packets directly received from the source already contain the correct flooding information, simply forwarding the original packets at the WiFi relay side will reduce the WiFi throughput. Second, if the packets directly received from the source are corrupted, the flooding performance is also hampered. To overcome this problem, we develop an Overlapped Channel Coding approach that can *i) preserve the throughput of the WiFi network;* and *ii) protect the embedded flooding packets.*

### B. Overlapped Channel Coding

The overlapped channel coding leverages the feature that the data for ZigBee signal emulation is in the predictable positions of a WiFi packet, which is shown in Figure 9. For the data in these positions, we give the following definition:

*Definition 1 (Emulation Data Block): The data in the position that can be used for ZigBee signal emulation is defined as the emulation data block.*

*1) For the WiFi Source:* It randomly selects the emulation data blocks in a WiFi packet for ZigBee flooding signal emulation. The selected emulation data blocks are defined as ***ZigBee Data Block.*** The unselected emulation data blocks can be used to transmit the original WiFi data. Due to the randomness of the selection, the ZigBee data blocks in each packet are likely to be in different positions. To conduct coding process, the unselected emulation data blocks are combined with the ZigBee data blocks in the same position from other packets by using a linear combination approach.

As shown in Figure 10, assume the WiFi source transmits packet $A$ and $B$. Each packet has 3 available emulation data blocks. For packet $A$, the source randomly selects block $G_1^A$ and $G_3^A$ as ZigBee data blocks for emulation while $G_2^A$ is the unselected emulation data block. For packet $B$, $G_2^B$ is selected as the ZigBee data block while $G_1^B$ and $G_3^B$ are the unselected emulation data blocks. Formally, we represent the data in $G_1^A$ and $G_1^B$ as $a_1$ and $b_1$, respectively. For the block $G_1^B$, the WiFi source picks two random numbers $\beta_1$ and $\beta_2$ and linearly combines the data in $G_1^B$ and the data in $G_1^A$ together, which can be represented as $G_1^B = \beta_1\ a_1 + \beta_2\ b_1$. $\beta_1$ and $\beta_2$ are the code vectors $\vec{v}_B = (\beta_1, \beta_2)$ for packet $B$. Assume the code vector for packet A is $\vec{v}_A = (\alpha_1, \alpha_2)$. By leveraging this approach, the final coded blocks in packets $A$ and $B$ can be represented as follow:

$$\begin{cases} G_1^A = a_1 & G_1^B = \beta_1 a_1 + \beta_2 b_1 \\ G_2^A = \alpha_1 a_2 + \alpha_2 b_2 & G_2^B = b_2 \\ G_3^A = a_3 & G_3^B = \beta_1 a_3 + \beta_2 b_3 \end{cases} \tag{1}$$

*2) For the WiFi Relay:* After receiving the coded packets, it decodes the original packets by solving linear equations in (1) and finds out the correct parts and the corrupted parts. Then, the relay selects random numbers as the code vectors to linearly combine the received packets. Meanwhile, the corrupted parts in each packet will be dropped. At last, the coded packets are transmitted to the destination.

Generally, we assume the number of received coded packets is $n_r^s$ and the number of the final coded packets for transmission is $n_t^d$ ($n_r^s > n_t^d$). For a decoded packet $I$ from $n_r^s$, the relay applies the code vector $\vec{v}_I = (\gamma_1, \ldots, \gamma_u)$ to combine this packet with other randomly selected decoded packets $U$. Then, the coded data blocks $j$ in packet $I$ can be represented as $G_j^I = \gamma_1\ G_j^A + \ldots + \gamma_u G_j^U$.

As shown in Figure 11, after receiving the coded packet $A$ and $B$, the relay solves the equations in 1. Then, by leveraging the code vector $\vec{v}_C = (\gamma_1, \gamma_2)$, the packet $A$ and $B$ are combined together to form a packet $C$. At last, the packet $C$ will be broadcast to the destination.

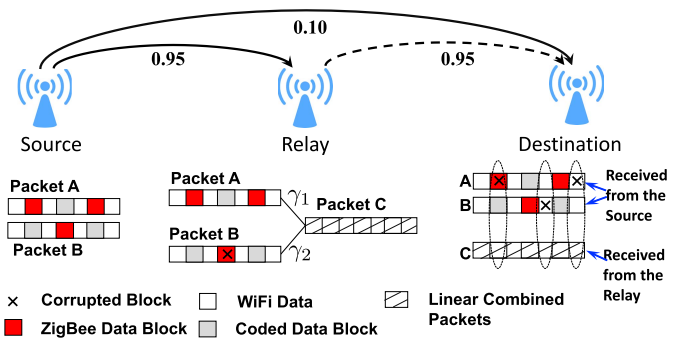



Fig. 11. The source broadcasts $A$ and $B$ to the relay and destination. The relay linearly combine $A$ and $B$ for transmission. The destination can recover the flooding information even if $A$ and $B$ are corrupted.

*3) For the WiFi Destination:* It receives the packets directly from the source and relays. If the packets received from the source are correct, the destination can decode the packets and get the original WiFi information and flooding information. If the directly received packets are partially corrupted, the WiFi destination should wait for the packets transmitted from the potential relays. Because of spatial diversity [19], even if the coded packets received from the relay are corrupted, the positions of the corruptions are likely to be in different positions. By ignoring the corrupted parts, the WiFi destination can decode the received packets.

As shown in Figure 11, the WiFi destination finds the corrupted parts from $A$ and $B$. Then, it waits for the transmission from the WiFi relay. After receiving the coded packet $C$, the destination can decode the packets. In the worst cases, the whole packets may not be decoded due to the extremely low link quality. Since the WiFi source randomly selects the ZigBee data block in each WiFi packet for signal emulation, it is still highly possible that the destination can at least recover the flooding information and conduct flooding thereafter.

## VII. Network Layer: ZigBee Flooding

In this section, we first describe the flooding channel coordination approach. Then, we introduce how to terminate the flooding.

### A. Flooding Channel Coordination

The WiFi devices in CRF leverage the physical layer emulation technique to emulate the ZigBee waveforms. However, due to the limitation of the 802.11 physical layer, the emulated ZigBee signal may not be exactly same as the desired ZigBee signal. *First*, based on the ZigBee signal, the WiFi select the nearest corresponding QAM points, which introduces imperfect emulation. *Second*, the WiFi devices use the Cyclic Prefix (CP) to eliminate the Inter-Symbol Interference and Inter-Carrier Interference. However, the ZigBee devices do not use this scheme. *Third*, since one WiFi symbol is only $4\mu s$, the WiFi devices should use $4$ symbols to emulate one ZigBee symbol ($16\mu s$). The discontinuity between the WiFi symbols also introduces imperfections. Fortunately, since ZigBee uses PN sequence for error tolerance, ideally the emulated signal can still be demodulated. However, in practice, due to the unreliable radio links, the packet reception ratio from WiFi to ZigBee is relatively low.

Traditionally, if the packets failed to be delivered to the ZigBee nodes, the WiFi source should conduct retransmissions. However, as mentioned in section III, it is hard for ZigBee nodes to inform the WiFi devices of the transmission status. Therefore, retransmissions of flooding packets will suffer a higher delay. In our design, we utilize Luby Transform codes (LT Codes) and develop a flooding channel coordination approach, which has following benefits: *i) It improves the flooding reliability and reduce the flooding delay;* and *ii) It can transmit additional control messages to mitigate the effects of imperfect emulations.*

*1) Preliminaries on LT Codes:* LT codes have been utilized to achieve reliable communication and reduce the network overhead [17]. The coding and decoding processes only involve XOR operations, which is very efficient and can be applied to the ZigBee nodes. Specifically, to transmit $X$ packets, the LT codes allow the sender to generate an infinite number of encoded packets for transmission. An encoded packet is generated by randomly selecting $D$ packets in the original $X$ packets and then combine them together by using the XOR operation. The receiver can decode the original message after receiving enough packets.

### B. Introduction of Perfect Emulation

To achieve reliable flooding, the WiFi source can simply apply LT codes to encode the ZigBee flooding packets and then broadcast to the ZigBee nodes during their active state. After receiving enough packets, the ZigBee nodes can decode the flooding information. However, due to the imperfect emulation and unreliable radio links, it is possible that some specific ZigBee symbols cannot be demodulated by several ZigBee nodes. In this case, *even though we leverage LT codes, the flooding reliability still cannot be guaranteed*.On the other hand, it is possible for some ZigBee nodes to decode the flooding packets before the end of its active period. Since they cannot immediately terminate the flooding, *both ZigBee and WiFi waste their communication resources*.To better show this concept, we have the following analysis:

Based on LT codes theory [20], a ZigBee node needs to receive $Y$ packets to have the probability $\epsilon$ to successfully decode $X$ packets. $Y$ can be represented as $Y = X + 2\ln(\frac{S}{1-\epsilon})S$, where $S$ is the expected number of degree-one checks packets and can be calculated as: $S = c\ln(\frac{X}{1-\epsilon})\sqrt{X}$. $c$ is a real number and $c \in (0, 1)$.

Formally, we define the Perfect Emulation Rate as follow:

*Definition 2 (Perfect Emulation Rate $R_{pe}$): Perfect Emulation Rate is defined as the percentage of the successfully demodulated ZigBee symbols that received from a WiFi device.*

In practice, due to imperfect emulation and unreliable radio links, the expected number of transmitted packets $E(Y)$ from the WiFi can be represented as $E(Y) = \frac{Y}{p_i}$, where $p_i$ is product between the link quality $P_w^z$ from WiFi to ZigBee node $i$ and the perfect emulation rate $R_{pe}$ ($0 \leq R_{pe} \leq 1$).

For the ZigBee nodes with large $p_i$, they can decode the flooding information before the end of its active period. For the ZigBee nodes with small $p_i$, they can never successfully decode the flooding information. Therefore, in both cases, the WiFi devices conducts lots of redundant transmissions.

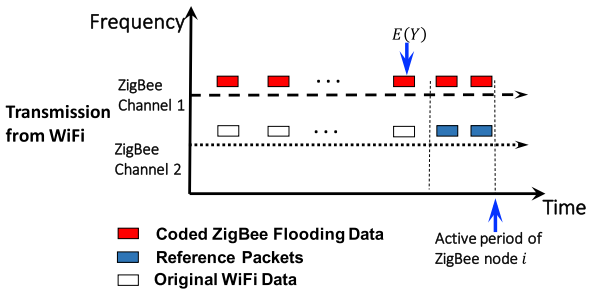



Fig. 12. After transmitting $E(Y)$ packets, the WiFi source simultaneously uses channel 2 to transmit additional packets.

Specifically, if we denote the number of packets that can be transmitted from WiFi to ZigBee during its active period $\tau_i$ as $N_{w2z}$, the number of redundant packets transmitted from WiFi can be calculated as follow:

$$\begin{cases} E(N_{rp}) = N_{w2z} - E(Y) \\ max(N_{rp}) = N_{w2z} - X \\ min(N_{rp}) = 0 \end{cases} \tag{2}$$

$E(N_{rp})$ and $max(N_{rp})$ are the expected number of redundant packets and the maximum number of redundant packets, respectively. $min(N_{rp})$ represents that the ZigBee node $i$ can decode the packets at the end of its active period and the corresponding probability can be represented as:

$$P_{min} = p_i\left(\sum_{n=Y}^{N_{w2z}} \rho(n) C_{N_{w2z}}^{n-1} p_i^{n-1} (1-p_i)^{N_{w2z}-n}\right) \tag{3}$$

$\rho(n)$ is the robust soliton distribution ($\rho \leq 1$). For the ZigBee node with high $p_i$, as the number of transmitted packets increases, the probability $P_{min}$ will be extremely low. Therefore, it is highly possible for the ZigBee node to successfully decode the packets while the WiFi is still in transmission. In other words, *this ZigBee node can receive additional packets during its active period.* For the ZigBee node with a relatively low $p_i$, it may not successfully decode the flooding information.

### *C. Flooding Channel Coordination and WiFi Emulation Selections*

As mentioned in section VII-A, the WiFi uses 4 symbols to emulate one ZigBee symbol. However, the selections of these 4 symbols are not unique. First, for a given ZigBee symbol, the nearest corresponding WiFi QAM points may introduce higher the discontinuity. Second, due to the uncertainty of the wireless environments, a fixed selection will not guarantee the communication reliability. In CRF, to improve the flooding reliability and reduce the number of redundant packets, the WiFi utilizes multiple selections of symbols to emulate the ZigBee symbols. In addition, CRF leverages the feature that the WiFi has two overlapped channels that can directly communicate with ZigBee nodes.

*1) For the WiFi Device:* It transmits the flooding packets on one of the overlapped ZigBee channels. The emulation on this channel is based on a single selection of WiFi symbols. On the other overlapped channel, the WiFi transmits the reference packets, which contains a predefined ZigBee symbol sequence. Each ZigBee symbol in this sequence is emulated by different

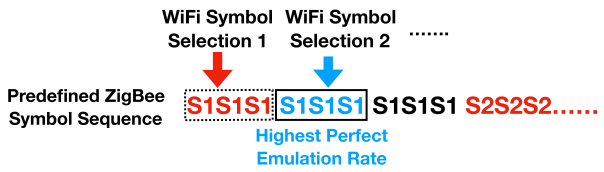



Fig. 13. The second WiFi symbol selection has the highest perfect emulation rate to emulate the ZigBee symbol S1.

| ZigBee ID | Receiving Status | Emulation Selections for the ZigBee with low $p_i$ |
|---|---|---|

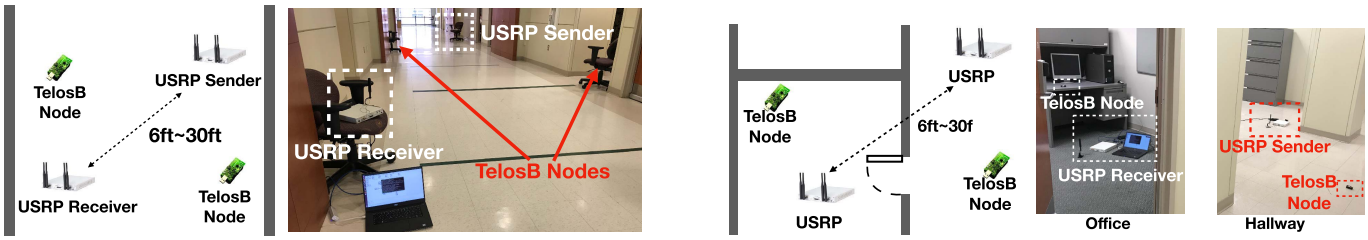
Fig. 14. An example of the acknowledgement.

selections of the WiFi symbols. To save subcarriers, the WiFi source calculates the expected number of transmitted packets $E(Y)$. After transmitting $E(Y)$ packets, the WiFi source uses the other overlapped channel to simultaneously transmit the reference signal to the ZigBee node, which is shown in Figure 12. The WiFi source will stop the transmission based on the ZigBee nodes' working schedules.

*2) For the ZigBee Node:* It senses and receives the coded flooding packets on its working channel. If it decodes the flooding information before $E(Y)$, it means its $p_i$ value is high. In this case, the ZigBee node will switch to the non-overlapped channel $C_n$ and go to dormant state. If the ZigBee node cannot decode the flooding information before $E(Y)$, that means its $p_i$ value is relatively low. In this case, it will switch to the overlapped ZigBee channel 2 ($C_2$) to receive the reference packets. It will check the Perfect Emulation Rate $R_{pe}$ in the predefined ZigBee symbol sequence and select the WiFi symbol selection with highest $R_{pe}$. After that, it will switch to the non-overlapped channel $C_n$ and go to dormant state. For example, as shown in Figure 13, the predefined ZigBee Symbol Sequence contains multiple ZigBee symbols. Each symbol appears 9 times and is emulated by 3 different WiFi symbol selections. In this example, for the symbol $S1$, the second WiFi symbol selection shows the highest perfect emulation rate. Therefore, the ZigBee node considers the second WiFi symbol selection as the preferred WiFi emulation option for ZigBee symbol $S1$.

### *D. CTC Flooding Termination*

In our design, to terminate the flooding with a limited number of acknowledgments, the ZigBee nodes coordinate together by utilizing the non-overlapping channel $C_n$. After switching to $C_n$, the ZigBee nodes will share their ID and ZigBee to WiFi (Z2W) link quality. The node with highest Z2W link quality is the selected ZigBee node. In addition, for the ZigBee nodes with low $p_i$ value, they will also transmit the preferred WiFi emulation option of each ZigBee symbol to the selected node. Then, this node will transmit a single ACK to the WiFi source. This ACK will indicate which ZigBee nodes have successfully received the packets and the preferred WiFi emulation selections, which is shown in Figure 14. As shown in this figure, the first block of the ACK indicates the ZigBee IDs that have switched to the channel $C_n$ while the second block will show the corresponding receiving status of these ZigBee node. The third block shows the suitable emulation




(a) LoS (b) NLoS

Fig. 15. The deployment of LoS and NLoS scenarios.

selections for the ZigBee nodes with low $p_i$. In practice, it is possible that multiple symbol selections have the same perfect emulation rates $R_{pe}$ for the ZigBee node. In this case, the ZigBee node will indicate all these possible suitable emulation selections in block three. For the WiFi device, based on the received ACK, it not only can decide whether to terminate the flooding but also can conduct better ZigBee signal emulation to improve the flooding reliability. If a ZigBee node shows multiple suitable symbol selections, the WiFi will choose the selection with lower Euclid Distance between the emulated symbol and the desired ZigBee symbol.

## VIII. Evaluation

We extensively evaluate our design under various settings and scenarios. Since this is the first work investigating concurrent routing and flooding in a heterogeneous IoT network (e.g., ZigBee and WiFi), the state-of-the-art is complimentary, however, it provides no appropriate baselines for comparison. To show the advantages of CRF, we use **PANDO** [17] as the baseline for ZigBee flooding. Specifically, in PANDO, ZigBee nodes also leverage LT codes to improve the flooding reliability and reduce the flooding delay. In addition, to reduce the number of acknowledgements, the sender in PANDO can passively listen to the channel to make sure the forwarded packets from the receiver are correct. For the WiFi routing evaluation, we use **Opportunistic Routing (OPPO)** [21] and **COPE** [22] as baselines for WiFi routing.

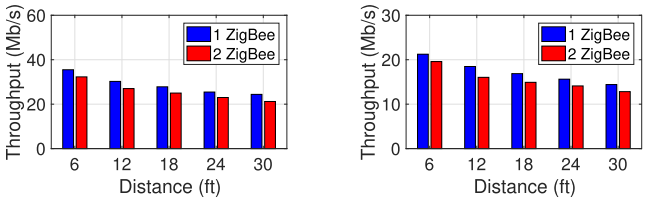
Moreover, to further show the benefits of our design, we also design a basic coexistent routing and flooding solution **BCRF** as our baseline. BCRF can concurrently conduct routing and flooding to ZigBee and WiFi, respectively. For the routing part, BCRF does not apply any coding techniques. For the flooding part, BCRF transmits the original flooding packets to the ZigBee nodes. After successfully receiving the flooding packets, the ZigBee nodes directly transmit ACKs back to the WiFi source by using a CTC packet-level modulation scheme.

### A. PHY Layer Evaluation

We use WiFi compliant USRP X300 with 802.11 b/g PHY as the WiFi devices and evaluate our design in the following scenarios:

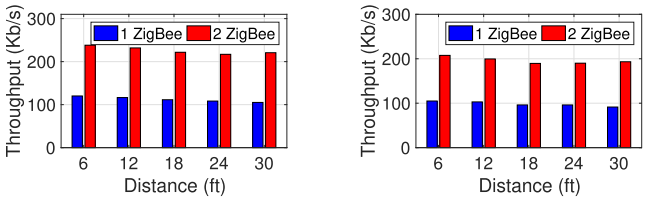
*1) Line-of-Sight (LoS):* The sender and receivers are within line-of-sight and the distance varies from 6ft to 30ft, which is shown in Figure 15(a).

*2) Non-Line-of-Sight (NLoS):* The sender and receivers do not have Line-of-Sight paths and the distance varies from 6ft to 30ft, which is shown in Figure 15(b).




(a) LoS (b) NLoS

Fig. 16. The throughput of WiFi to WiFi.




(a) LoS (b) NLoS

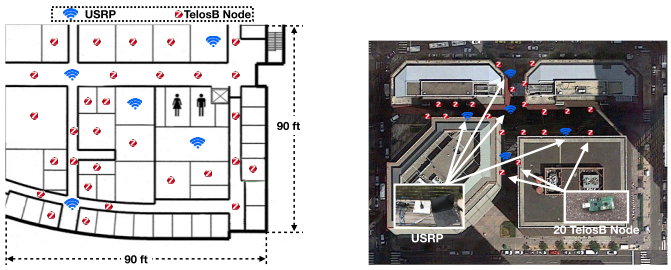
Fig. 17. The throughput of WiFi to ZigBee.

As shown in Figure 16(a) and (b), when communicating with one ZigBee node, the throughput of WiFi to WiFi in the Line-of-sight (LoS) scenario varies from 35.50Mbps to 24.41Mbps as the distance increases from 6ft to 30ft. When communicating with two ZigBee nodes, the throughput of WiFi to WiFi achieves 32.29Mbps at 6ft and 21.21Mbps at 30ft. In the Non-Line-of-sight (NLoS) scenario, the throughput achieves around 60% of the throughput in Line-of-sight scenario.

As shown in Figure 17(a) and (b), the throughput of WiFi to one ZigBee node in the Line-of-sight (LoS) scenario varies from 120.21Kbps to 105.17Kbps as the distance increases from 6ft to 30ft. When the WiFi is communicating with two ZigBee nodes, the throughput is 238.21Kbps at 6ft and 220.77Kbps at 30ft. For the Non-Line-of-sight (NLoS) scenario, the throughput is similar to the Line-of-sight scenario.

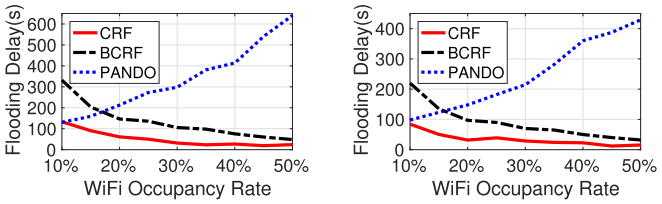
### B. Network Layer Evaluation

We evaluate the network performance of our system by deploying 30 ZigBee compliant TelosB nodes in both indoor and outdoor environments [shown in Figure 18(a) and (b)]. We carefully select these environments so that they represent the smart building (indoor inside a building) and smart city (outdoor on a street) applications. The duty cycle of the ZigBee node varies from 1% to 30%. The flooding packet size varies from 10 bytes to 100 bytes and the WiFi



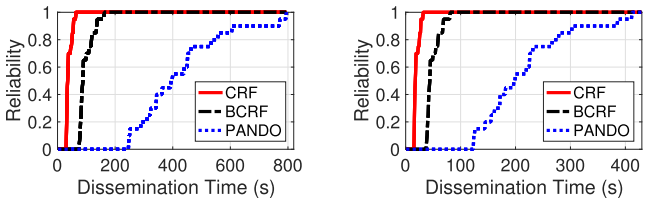

(a) Smart Building (b) Smart City

Fig. 18. Smart building and smart city scenarios.



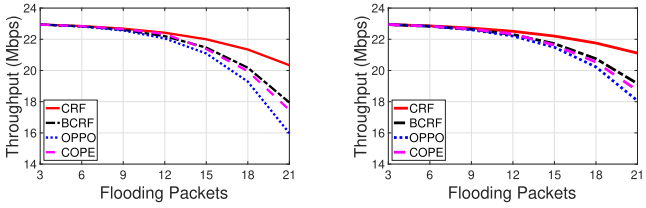

(a) Smart Building (b) Smart City

Fig. 19. Flooding Delay vs. WiFi Occupancy Rate.




(a) Smart Building (b) Smart City

Fig. 20. Reliability progress vs. Dissemination time.




(a) Smart Building (b) Smart City

Fig. 21. WiFi Network Throughput vs. Flooding Packets per Duty Cycle.

packet size varies from 100 bytes to 1,400 bytes. Each experiment is repeated multiple times with different node placements. We show the averaged value from the experiments.

*1) Flooding Delay vs. WiFi Occupancy Rate:* We evaluate the flooding delay under different WiFi traffic occupancy rate. As shown in Figure 19(a) and (b), CRF shows great advantages than the state-of-the-art solutions. When the WiFi traffic occupancy rate reaches 50%, the flooding delays of CRF in smart building and smart city scenarios are $23.62s$ and $15.45s$ respectively, which is around **27** times lower than the flooding delay of PANDO ($640.44s$ and $429.00s$). This is because as the WiFi traffic occupancy rate increases, there are more opportunities for CRF to route the WiFi packets and thus conduct flooding for ZigBee nodes. In contrast, due to the increased traffic from WiFi, the flooding delay of PANDO is increasing. Even in the best case (e.g., WiFi traffic is low), the flooding delay of PANDO is still higher than CRF. This is because PANDO is not designed to conduct flooding in cross-technology interference. Due to the Carrier Sense Multiple Access (CSMA) scheme of ZigBee nodes, the ZigBee nodes have to back off.

The comparison between CRF and BCRF shows the advantages of our design. As the WiFi traffic increases, the flooding delay of BCRF is much higher than that of CRF. This is because the flooding in BCRF is unreliable. As the WiFi traffic increases, BCRF conducts lots of retransmissions. Therefore, the flooding delay decreases. In contrast, CRF can conduct a more reliable flooding even if the WiFi traffic is low. **In summary,** *the flooding delay of CRF can be down to* ***27*** *times lower than that of PANDO.*

*2) Reliability Progress vs. Flooding Packets Dissemination Time:* Figure 20(a) and (b) depict the progress of the average flooding reliability for ZigBee nodes. In the experiment, the WiFi occupancy rate is set to 50%. CRF reaches 100% reliability with the lowest dissemination time while PANDO reaches 100% with the longest time (more than $800s$ for the smart building and $400s$ for the smart city scenario). We need to mention that PANDO also utilizes Fountain codes to conduct flooding. However, as shown in this evaluation, fountain codes cannot help PANDO survive in such scenario.

The dissemination time of BCRF is also higher than that of CRF. This result shows the advantages of our design. First, CRF utilizes fountain codes to ensure the flooding reliability, which reduces the number of retransmissions and decrease the dissemination time. Second, the routing design of CRF also contributes to the flooding. When conducting routing, CRF protects the embedded flooding packets by leveraging overlapped channel coding. **In summary,** *CRF improves the flooding reliability and significantly reduces the flooding delay.*

*3) WiFi Network Throughput vs. Number of Transmitted Flooding Packets Per Duty Cycle:* As shown in Figure 21(a) and (b), we evaluate the WiFi network throughput under different number of transmitted flooding packets. The throughput of OPPO decreases much faster than that of CRF, BCRF and COPE. This is because OPPO suffers the interference from the ZigBee network, which requires WiFi to frequently back off to avoid collisions. For COPE, it utilizes XOR coding to improve the WiFi throughput. When the number of flooding packets is lower than 12, the performance of COPE is better than OPPO and BCRF. This is because that the coding scheme leveraged in COPE is sufficient to maintain the WiFi throughput. However, as the ZigBee traffic increases, the throughput of COPE decreases rapidly. In contrast, CRF uses WiFi overlapped ZigBee subcarriers to conduct flooding, which has better performance. As the number of flooding packets increases to 21, the throughputs of OPPO decrease to $16.04Mbps$ and $18.11Mbps$ for smart building and smart city scenarios, respectively. While the throughputs of COPE decrease to $17.92Mbps$ and $19.10Mbps$ for smart building and smart city scenarios, respectively. In contrast, the throughputs of CRF are still as high as $20.46Mbps$ and $21.11Mbps$, which is around 1.12 times higher than those of OPPO.

We can also observe that the throughput of CRF is much higher than that of BCRF. This is because our overlapped channel coding protects the flooding information. In contrast,

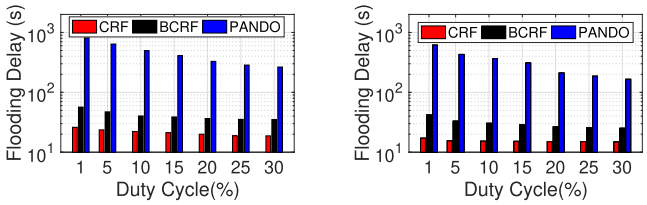



Fig. 22. Flooding Delay vs. Duty cycle.

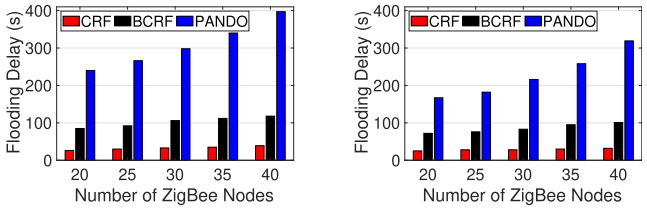



Fig. 23. Flooding delay vs. Network density.

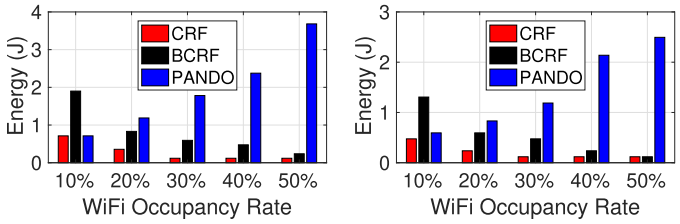



Fig. 24. Energy consumption vs. WiFi occupancy rate.

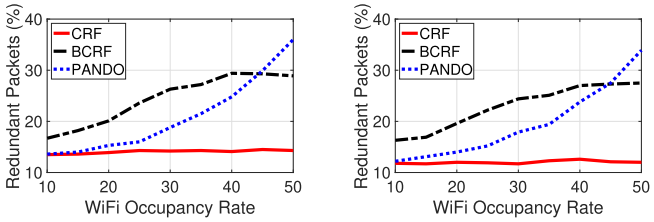



Fig. 25. Percentage of Redundant Packets.

as the number of transmitted flooding packets increases, BCRF has to frequently conduct retransmissions, which reduces the WiFi throughput. Moreover, the ACKs from the ZigBee nodes require packet-level modulation, which introduces huge interference to the WiFi network and significantly reduces the WiFi throughput. **In summary,** *CRF can reduce the interference and preserve the throughput of the WiFi network.*

### C. System Sensitivity Evaluation

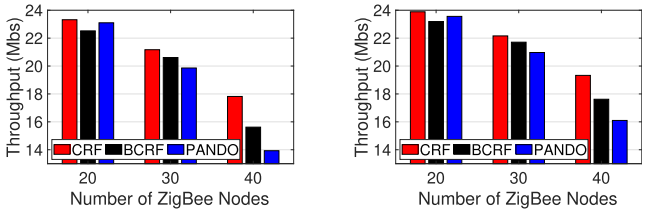
*1) Flooding Delay vs. ZigBee Duty Cycle:* Figure 22(a) and (b) show the flooding delay under different duty cycles. As the duty cycle increases, the flooding delay of BCRF and PANDO is decreasing. CRF shows a relatively stable flooding delay. Specifically, when the duty-cycle is 1%, the flooding delays of PANDO for the smart building and smart city scenarios are $811s$ and $612s$, respectively. In contrast, the corresponding delays of CRF are $26.08s$ and $17.10s$, which is much lower than PANDO. In addition, the flooding delays of BCRF are also as low as $56.71s$ and $41.64s$. This result shows the advantages of CRF.

We also observe that the performance of CRF is much better than that of BCRF. This is because CRF conducts reliable flooding during the duty-cycle of each ZigBee node. In contrast, BCRF shows higher flooding delay since the flooding of BCRF is unreliable. As the duty-cycle increases, BCRF has more opportunities to retransmit the flooding packets, which reduces the flooding delay. **In summary,** *the flooding delay of CRF is more than **31** times and **2** times lower than those of PANDO and BCRF, respectively.*

*2) Flooding Delay vs. Network Density:* We study the flooding delay under different densities in Figure 23(a) and (b). As shown in these figures, the flooding delays of CRF almost remain the same. This is because the flooding in CRF is conducted by the WiFi device. For a similar area, the increasing number of nodes will not affect the flooding delay. In contrast, the flooding delays of BCRF increase as the number of ZigBee nodes increases. This is because the flooding reliability in BCRF is relatively low. As the number of nodes increases, the WiFi needs to conduct frequent retransmissions, which reduce the network performance. PANDO shows the worst performance. This is because PANDO requires ZigBee nodes to conduct flooding. More ZigBee nodes introduces higher number of hops, which increases the flooding delay.

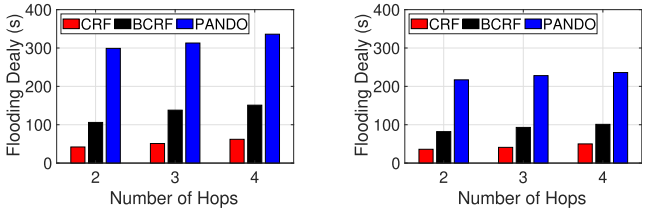
*3) Energy Consumption vs. WiFi Occupancy Rate:* By avoiding the interference from the WiFi traffic, CRF reduces the working time of the ZigBee radio and the corresponding energy consumption. As shown in Figure 24(a) and (b), the energy consumption of CRF is similar to that of PANDO when the WiFi occupancy rate is low. As the WiFi occupancy rate increases to 50%, the energy consumption of CRF reduces to $0.13J$ and $0.11J$ for smart building and smart city scenarios, which is **25** times and **22** times lower than PANDO ($3.28J$ and $2.49J$). The energy consumption of BCRF is much higher than PANDO and CRF when the WiFi occupancy rate is low and it reaches $0.23J$ for smart building and $0.13J$ for the smart city when the WiFi traffic increases to 50%. Even in this situation, CRF is still around 1.77 times and **1.20** times better than BCRF for smart building and smart city scenarios. **In summary,** *CRF can achieve lower energy consumption regardless of the WiFi traffic.*

*4) Message Overhead:* We show the message overhead under different WiFi occupancy rates in Figure 25(a) and (b). These two figures show similar trends. Since CRF utilizes the WiFi to conduct flooding and only requires few ZigBee nodes to transmit ACKs back to the WiFi, the message overhead is low. Since the communication of BCRF is unreliable, the WiFi has to conduct lots of retransmissions, which significantly increases the message overhead. In addition, too many packet level CTC ACKs also introduce huge message overhead to the network. The message overhead of PANDO stays low when the WiFi occupancy rate is lower than 40%. Surprisingly, as the WiFi occupancy rate reaches 50%, the message overhead of PANDO is higher than that of BCRF. This is because PANDO is not designed to work under high CTC interference and PANDO leverages a silence-feedback scheme to determine whether to conduct retransmissions. Therefore, as the WiFi




(a) Smart Building (b) Smart City

Fig. 26. Network Throughput vs. Number of ZigBee Nodes.




(a) Smart Building (b) Smart City

Fig. 27. Flooding Delay vs. Number of Hops.

occupancy rate reaches 50%, the sender in PANDO cannot know the transmission status. It has to frequently conduct retransmissions even though the receiver has already received the flooding messages.

*5) WiFi Throughput vs. Number of ZigBee Nodes:* Figure 26(a) and (b) show the WiFi throughput under different network sizes. In this experiment, the WiFi occupancy rate is 30%. As shown in the figures, CRF is able to support higher WiFi network throughput. In contrast, PANDO interfere with the ongoing WiFi traffic, which reduces the throughput. For the 20 ZigBee nodes scenario, the throughput of PANDO is even slightly higher than that of BCRF. This is because BCRF introduces huge amount of redundant packets when it conducts acknowledgements.

*6) Flooding Delay vs. Number of WiFi Hops:* We study the performance of CRF under different number of WiFi hops in Figure 27(a) and (b). In this experiment, the WiFi occupancy rate is set to 30%. As we can observe from these figures, the flooding delays of CRF are relatively stable with the increasing number of WiFi hops. However, for PANDO, the flooding delays increase as the number of WiFi hops increases. This is because the packet transmitted by the WiFi source is relayed by a higher number of WiFi devices, which introduces high interference to the ZigBee network. As a result, the flooding delay is increased. For BCRF, the flooding delay is significantly increased in the smart building scenario while the delay is relatively stable in the smart city scenario. This is because although BCRF can leverage WiFi traffic to conduct flooding for the ZigBee network, it is still relatively sensitive to the variations of wireless interference. Since the wireless interference in the smart building scenario is higher than that of smart city scenario, with the number of WiFi hops increases, the performances of WiFi routing and WiFi to ZigBee flooding are significantly affected, which introduces higher flooding delay.

### D. Application: Flash Flood Warning

We also conduct experiment near a river, which is shown in Figure 28. The WiFi devices receive the flood forecasting

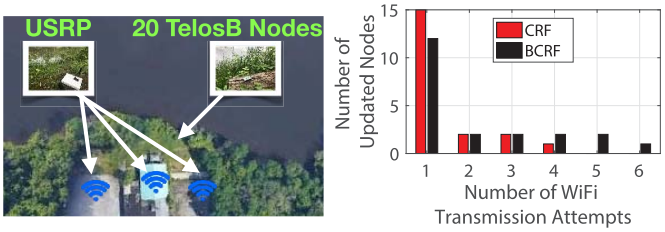



Fig. 28. Application: Flood warning scenario.

information from the internet and update the duty-cycle of each ZigBee node so as to transmit the water level data more frequently. CRF is able to update more than 75% ZigBee nodes' duty cycles as soon as they switch to the active state. In contrast, it takes 6 duty cycles for BCRF to finish the updating process, which is unacceptable for flood warning.

## IX. Simulation

This section shows the simulation results of CRF. In the simulation, we deploy 100 ZigBee nodes and 40 WiFi devices. Each simulation is repeated 1000 times with different random seeds. The duty-cycle of the ZigBee node is set to 10%. The ZigBee node is implemented according to the hardware specification of a ZigBee compliant TelosB node [23].

### A. Flooding Delay vs. Link Quality

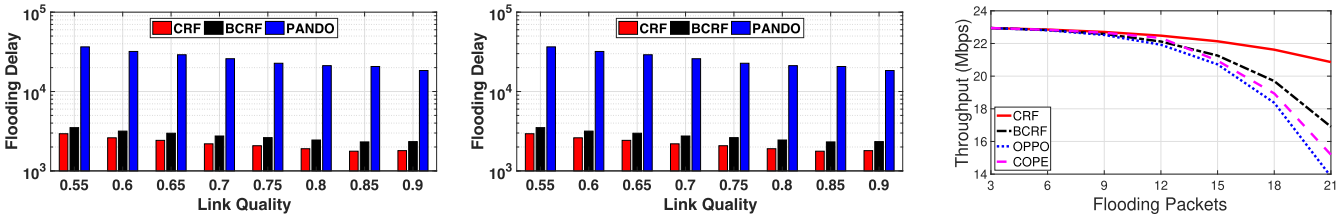
We first evaluate the flooding delay under different link qualities. Since PANDO does not have WiFi to ZigBee communication, the simulation results of PANDO show the flooding delay under different ZigBee to ZigBee link qualities. As shown in Figure 29(a), CRF shows great advantages in conducting reliable flooding. When the link quality is as low as 0.55, the flooding delay of CRF is 2939 time units, which is around **12.43** and **1.19** times lower than PANDO (36, 530 time units) and BCRF (3506 time units), respectively. In summary, these results show that our approach conducts reliable flooding under different link qualities with much lower latency. **In summary,** *the flooding delay of CRF is significantly reduced when the network size is larger.*

### B. Flooding Delay vs. Network Size

We evaluate the flooding delay under different network sizes. As we can see from Figure 29(b), the flooding delay of CRF is robust to the network size. Specifically, the flooding delays of CRF under 50 nodes and 200 nodes are 1,807 and 1,901 time units, respectively. Since PANDO is not designed for cross-technology interference and it has multiple layers to conduct flooding, the performance is not as good as that of CRF (21356 time units for 200 nodes). BCRF cannot ensure reliable flooding and routing, the performance is worse than CRF. **In summary,** *the flooding delay of CRF is almost stable, which is 11.75 times lower than that of PANDO.*

### C. WiFi Network Throughput vs. Number of Transmitted Flooding Packets Per Duty Cycle

We simulate the network throughput under different numbers of transmitted flooding packets in Figure 29(c).




(a) Flooding Delay vs. W2Z Link Quality (b) Flooding Delay vs. Network Size (c) WiFi Throughput vs. Flooding Packets

Fig. 29. Simulation Results.

When the packet number reaches 21, the throughput of CRF (20.94Mbps) is much higher than those of OPPO (14.01Mbps), COPE (15.36Mbps) and BCRF (17.05Mbps). This is because OPPO is suffering interference from the ZigBee network. The routing and flooding for BCRF is unreliable, which depresses the performance.

## X. Discussion

Although CRF mainly focuses on WiFi and ZigBee coexistence networks, we believe that our work can be applied to other heterogeneous IoT networks (i.e., WiFi and BLE coexistence network) without requiring significant modifications. In this section, we will discuss how to apply CRF to different heterogeneous IoT networks.

• For WiFi and ZigBee coexistence network, due to the limitation of 802.11 physical layer, the WiFi device cannot perfectly emulate the desired ZigBee signal. In other words, the emulated signal is distorted before being transmitted to the ZigBee device. Then, the distorted signal will pass the wireless channel, which introduces unpredictable changes. As a result, although ZigBee device utilize a 32-Pseudo-random Noise Chip Sequence to tolerate the chip error, the WiFi to ZigBee communication reliability still cannot be guaranteed. Therefore, for WiFi and ZigBee coexistence networks, as long as they are using physical layer emulation techniques for WiFi to ZigBee communication, CRF can be applied to these network without modifications.

• For WiFi and BLE coexistence network, instead of directly emulate the BLE signal, the WiFi device only needs to change its amplitude on the corresponding subcarriers that are overlapped with BLE. This is because the demodulation process of BLE only cares about the amplitude difference between space and mark frequencies. By slightly changing the amplitude of WiFi subcarriers, the BLE data can be embedded in the WiFi to WiFi communication with high reliability [24]. In this case, it is unnecessary to utilize LT codes for reliable WiFi to BLE communication. Therefore, for WiFi and BLE coexistence network, we only need to apply the overlapped channel coding to improve the WiFi to WiFi routing reliability.

## XI. Related Work

The related work can be divided into two categories:

### A. Routing & Flooding

**Routing** is one of the key topics in wireless networks. Researchers have proposed routing protocols for various types of wireless networks, such as the wireless mesh networks [25], [26], the intermittently connected sensor networks [27] and the wireless ad hoc networks [28]. The diversity of wireless networks gives the researchers various features that could help the design of the routing protocols. SocialCast [29] utilizes locations of acquaintances in the social network for routing. R3 [30] is a routing protocol that self-adapts replication for robust routing. Unnecessary forwarding [31] and network coding [32] can significantly improve the network performance in opportunistic routing.

**Flooding** protocols have been proposed in various wireless networks [33]. The performance of flooding can be improved from different angles. Chorus [34] improves the broadcast efficiency with a MAC layer that tolerates collisions among identical packets. The optimal transmission range for the flooding process to settle quickly [35] can be estimated. The reliability of flooding has also been improved when facing unreliable links in low-duty-cycle networks [36].

Unlike the above approaches that optimize performance of a single protocol (i.e., routing or flooding) within a single network (e.g., WiFi, or ZigBee), our approach treats the heterogeneous IoT networks as a whole and enables the coexistent routing and flooding for better performance improvements.

### B. Cross-Technology Communication

Based on the fact that multiple communication techniques may use overlapped frequency bands, researchers have proposed cross-technology communication (CTC) techniques. FreeBee [37] utilizes RSS for communication between WiFi and ZigBee devices. EMF [15] and $B^2W^2$ [16] embeds information in the existing traffic for concurrent communication among heterogeneous devices. WEBee [9] uses WiFi to emulate ZigBee signals for cross-technology communication. PMC [38] and Chiron [39] enable communication between WiFi and multiple ZigBee devices simultaneously. However, these two approaches require some specific hardware, which is not scalable and cannot be directly applied to current infrastructures.

Existing CTC techniques mainly focus on the physical layer. In the paper, we mainly focus on utilizing the CTC techniques for the network layer performance improvement.

## XII. Conclusion

In this paper, we present CRF, the first coexistent routing and flooding algorithm for concurrently conducting routing within the WiFi network and flooding among ZigBee nodes using a single stream of WiFi packets. With the exponentially increasing number of heterogeneous IoT devices deployed in smart communities, CRF can effectively leverage the heterogeneity of these IoT devices' communications to create a win-win situation for both WiFi networks (i.e., improve

the routing throughput) and ZigBee networks (i.e., significantly reduce the dissemination delay and increase flooding reliability). CRF opens a new direction for optimizing the network performance in heterogeneous IoT networks. The features we provide and the challenges we address in this coexistent communication-based design are generic and have the potential to be applied in other heterogeneous networks.

## REFERENCES


[1] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. SenSys*, 2009, pp. 1–14.

[2] K. Lin and P. Levis, "Data discovery and dissemination with DIP," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2008, pp. 433–444.

[3] K. Whitehouse and D. Culler, "A robustness analysis of multi-hop ranging-based localization approximations," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2006, pp. 317–325.

[4] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal clock synchronization in networks," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2009, pp. 225–238.

[5] W. Lou and J. Wu, "Double-covered broadcast (DCB): A simple reliable broadcast algorithm in MANETs," in *Proc. IEEE INFOCOM*, 2004, pp. 2084–2095.

[6] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza, "RBP: Robust broadcast propagation in wireless networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2006, pp. 85–98.

[7] S. Yun and L. Qiu, "Supporting WiFi and LTE co-existence," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 810–818.

[8] Y. He, J. Fang, J. Zhang, H. Shen, K. Tan, and Y. Zhang, "MPAP: Virtualization architecture for heterogenous wireless APs," in *Proc. ACM SIGCOMM Conf. (SIGCOMM)*, 2010, pp. 476–476.

[9] Z. Li and T. He, "WEBee: Physical-layer cross-technology communication via emulation," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 2–14.

[10] *Gartner Says 8.4 Billion Connected 'Things' Will be in Use in 2017, Up 31 Precent From 2016*. Accessed: Feb. 19, 2019. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[11] *Intel Smart Cities Solution*. Accessed: Jun. 20, 2019. [Online]. Available: https://www.intel.com/content/www/us/en/internet-of-things/smart-cities.html

[12] *Azure*. Accessed: May 16, 2019. [Online]. Available: https://azure.microsoft.com/en-us/blog/using-ai-and-iot-for-disaster-management/

[13] *Cisco White Paper*. Accessed: Apr. 12, 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html#_Toc953327

[14] Y. Chen, Z. Li, and T. He, "TwinBee: Reliable physical-layer cross-technology communication with symbol-level coding," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2018, pp. 153–161.

[15] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu, "EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, May 2017, pp. 1–9.

[16] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2W2: N-way concurrent communication for IoT devices," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. (CD-ROM)*, Nov. 2016, pp. 245–258.

[17] W. Du, J. C. Liando, H. Zhang, and M. Li, "When pipelines meet fountain: Fast data dissemination in wireless sensor networks," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2015, pp. 365–378.

[18] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *Proc. NSDI*, 2010, pp. 1–15.

[19] A. Miu, H. Balakrishnan, and C. E. Koksal, "Improving loss resilience with multi-radio diversity in wireless networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2005, pp. 16–30.

[20] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, Nov. 2002, p. 271.

[21] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," in *Proc. SIGCOMM*, 2004, pp. 69–74.

[22] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2006, pp. 243–254.

[23] *Telosb Datasheet*. Accessed: Jun. 2, 2019. [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf

[24] Z. Chi, Y. Li, X. Liu, Y. Yao, Y. Zhang, and T. Zhu, "Parallel inclusive communication for connecting heterogeneous IoT devices at the edge," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, Nov. 2019.

[25] S. Miskovic and E. W. Knightly, "Routing primitives for wireless mesh networks: Design, analysis and experiments," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[26] R. K. Sheshadri and D. Koutsonikolas, "Comparison of routing metrics in 802.11n wireless mesh networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1869–1877.

[27] L. Su, C. Liu, H. Song, and G. Cao, "Routing in intermittently connected sensor networks," in *Proc. IEEE Int. Conf. Netw. Protocols*, Oct. 2008, pp. 278–287.

[28] C. Sengul and R. H. Kravets, "Bypass routing: An on-demand local recovery protocol for ad hoc networks," *Ad Hoc Netw.*, vol. 4, no. 3, pp. 380–397, 2006.

[29] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 748–760, May 2008.

[30] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: Robust replication routing in wireless networks with diverse connectivity characteristics," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2011, pp. 181–192.

[31] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2007, pp. 169–180.

[32] M. K. Han, A. Bhartia, L. Qiu, and E. Rozner, "O3: Optimized overlay-based opportunistic routing," in *Proc. 12th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2011, pp. 1–11.

[33] Z. Li, M. Li, J. Liu, and S. Tang, "Understanding the flooding in low-duty-cycle wireless sensor networks," in *Proc. Int. Conf. Parallel Process.*, Sep. 2011, pp. 673–682.

[34] X. Zhang and K. G. Shin, "Chorus: Collision resolution for efficient wireless broadcast," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[35] M. Zúñiga and B. Krishnamachari, "Optimal transmission radius for flooding in large scale sensor networks," *Cluster Comput.*, vol. 8, nos. 2–3, pp. 167–178, 2005.

[36] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2009, pp. 2787–2802.

[37] S. M. Kim and T. He, "FreeBee: Cross-technology communication via free side-channel," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2015, pp. 317–330.

[38] Z. Chi, Y. Li, Y. Yao, and T. Zhu, "PMC: Parallel multi-protocol communication to heterogeneous IoT radios within a single WiFi channel," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–10.

[39] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for IoT devices," in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2018, pp. 204–216.