CRF: Coexistent Routing and Flooding using WiFi Packets in Heterogeneous IoT Networks

Wei Wang, Xin Liu, Yao Yao, Yan Pan, Zicheng Chi and Ting Zhu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Email: {ax29092, xinliu1, yaoyaoumbc, yanpan, zicheng1, zt}@umbc.edu

***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 networks by 1.2 times than the state-of-the-art routing protocols; and ii) significantly reduces the flooding delay in ZigBee networks (i.e., 31 times faster than the state-of-the-art flooding protocol).**

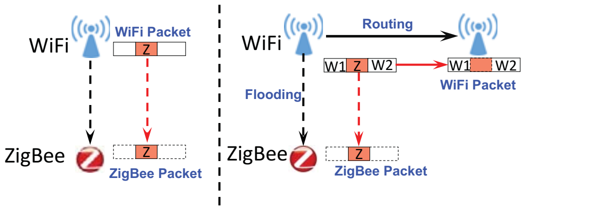
I. Introduction

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]–[9].

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 [10]), which uses WiFi signals to emulate ZigBee signals. However, in WEBee, the WiFi data is not utilized.

Different from WEBee (see Figure 1), we propose a novel physical layer design that enables the 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), that is concurrently conducting routing within the WiFi network and flooding among ZigBee nodes using a single stream of WiFi packets.



(a) WEBee (b) CRF

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.

With the exponentially increasing number of internet-of-things (IoT) devices [11], 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 existing infrastructure; 2) our network layer design enables the coexistent routing and flooding, which effectively avoid 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

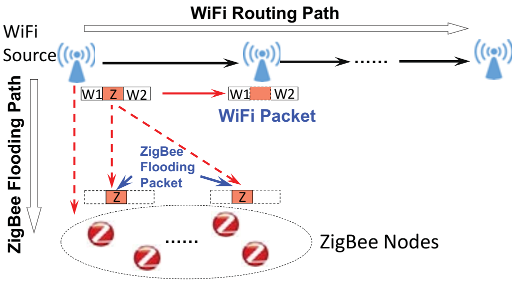


Fig. 2: **Network Architecture.** The WiFi source embeds a ZigBee flooding packet (Z) into its own packet for broadcasting. Other WiFi devices can receive the original WiFi data (W_1 and W_2) while the ZigBee nodes simultaneously receive the ZigBee flooding packet (Z).

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 networks by 1.2 times than the state-of-the-art routing protocols; and ii) significantly reduces the flooding delay in ZigBee networks (i.e., 31 times faster than the state-of-the-art flooding protocol).

II. 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 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 of its functions to save energy.

III. Challenges and Solutions

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

1) 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 [10], [12]. Specifically, the WiFi device controls the payload of its packet to emulate a 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).

2) How to preserve the throughput in the WiFi network and protect the embedded flooding information? Retransmission is one of the major problems that affect the network throughput. Different from traditional wireless 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).

3) How to terminate the flooding? Current ZigBee to WiFi communication techniques mainly use packet-level modulation [13], [14]. 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 [15], [16]. In Section VII-A, we introduce a CTC flooding termination scheme to overcome this challenge.

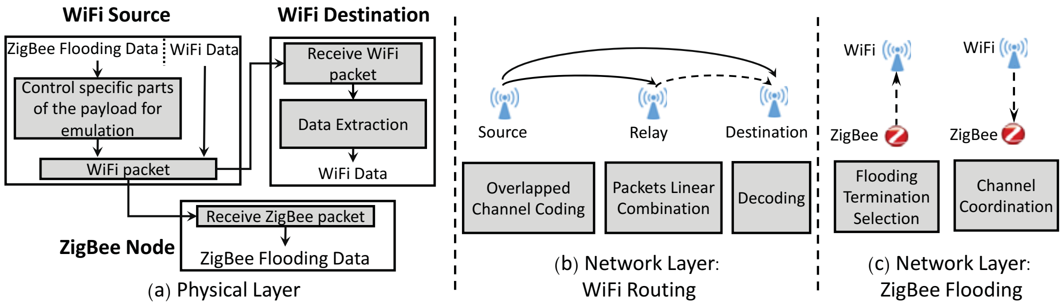
4) How to reduce the flooding delay and fully leverage the WiFi to ZigBee communication capability? Due to unreliable radio links from WiFi to ZigBee, the retransmissions of flooding packets introduce flooding delay. Moreover, since a ZigBee node has multiple channels that are overlapped with WiFi while it can only receive the packets on its current channel, which wastes the communication capability of the WiFi devices. In Section VII-B, we introduce a flooding channel coordination mechanism to improve the flooding reliability and fully leverages the CTC capabilities to transmit additional packets (i.e., control commands).

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:

1) 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 to get the original WiFi and ZigBee flooding data.

2) 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



(a) Physical Layer

(b) Network Layer: WiFi Routing

(c) Network Layer: ZigBee Flooding

Fig. 3: Design Overview

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.

3) Network Layer: ZigBee Flooding (see Figure 3 (c)). Our flooding scheme contains two parts: flooding termination selection and channel coordination schemes. In CRF, only a limited number of ZigBee nodes are selected to transmit feedback to the WiFi source, which significantly reduces the interference with the WiFi network. The WiFi source also applies a channel coordination mechanism to fully leverage the communication capability of WiFi-to-ZigBee.

V. Physical Layer of CRF

This section introduces the method of achieving coexistent WiFi-to-WiFi and WiFi-to-ZigBee communications.

A. *WiFi Data Transmission and Extraction*

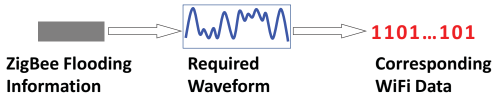


Fig. 4: Based on the ZigBee flooding information, the WiFi source calculates the required waveform and generates the corresponding WiFi data.

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 each other. 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.

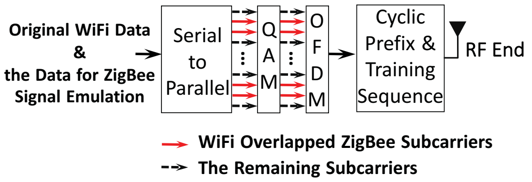

Original WiFi Data & the Data for ZigBee Signal Emulation
Serial to Parallel
QAM
OFDM
Cyclic Prefix & Training Sequence
RF End
WiFi Overlapped ZigBee Subcarriers
The Remaining Subcarriers

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

Specifically, as shown in Figure 5, the WiFi source first divides the original WiFi data and the data for 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 an Inverse Fast Fourier Transform (IFFT). Finally, a cyclic prefix and training sequence are applied to reduce Inter-Symbol Interference (ISI) and conduct synchronization between the source and destination.

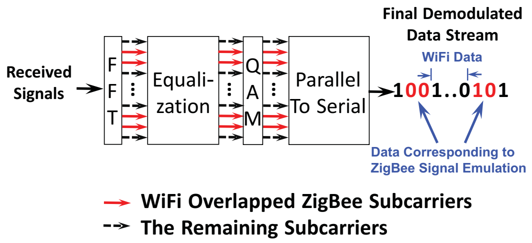

Final Demodulated Data Stream
WiFi Data
Received Signals
FFT
Equalization
QAM
Parallel To Serial
1001..0101
Data Corresponding to ZigBee Signal Emulation
WiFi Overlapped ZigBee Subcarriers
The Remaining Subcarriers

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.

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

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*

Before introducing overlapped channel coding, it is helpful to consider a simple WiFi network in Figure 7. 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

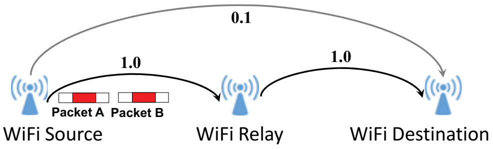


Fig. 7: The WiFi source can transmit the packets with embedded flooding information directly. It can also utilize the relay to conduct reliable transmissions.

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 have some correct parts. Moreover, since the WiFi packets contain flooding information, the flooding at the WiFi destination side is also affected. In CRF, 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

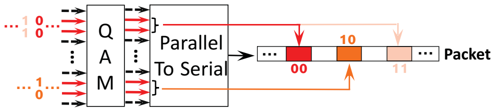


Fig. 8: The emulation data blocks are in the predictable positions.

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 8. 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.*

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 the 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.

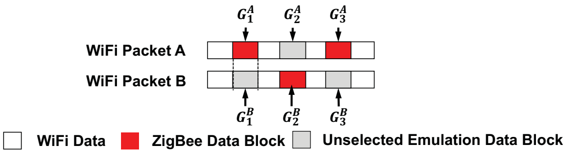


Fig. 9: An example of the coded packet.

As shown in Figure 9, assume the WiFi source transmits packets 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 β_1 and β_2 and linearly combines the data in G_1^B and the data in G_1^A together, which can be represented as $G_1^B = \beta_1 a_1 + \beta_2 b_1$. β_1 and β_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:

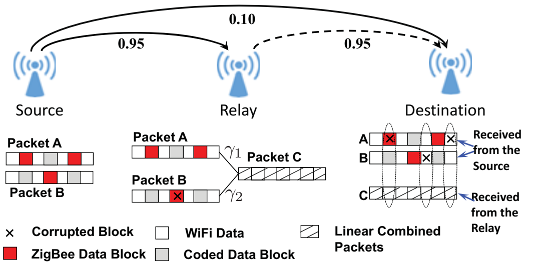


Fig. 10: The source broadcasts A and B to the relay and destination. The relay linearly combine A and B for transmission. The destination can recover the flooding information even if A and B are corrupted.

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

For the WiFi relays. 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 $\vec{v}_I = (\gamma_1, ..., \gamma_u)$ to combine this packet with other randomly selected decoded packets U. Then, the coded data blocks j in packet I can be represented as $G_j^I = \gamma_1 G_j^A + ... + \gamma_u G_j^U$.

As shown in Figure 10, after receiving the coded packets 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.

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 [17], 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 10, 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 introduce the flooding termination scheme and then describe the Flooding Channel Coordination approach.

A. CTC Flooding Termination Scheme

As mentioned in section III, the ZigBee nodes may not immediately transmit acknowledgments (ACKs) back to the WiFi source after successfully receiving the flooding packets. In our design, to terminate the flooding with limited number of acknowledgments, the ZigBee nodes are coordinate together by utilizing the non-overlapping channels. Formally, we denote the channel that are not overlapped with current WiFi channel as C_h. After successfully receiving the flooding packets, the ZigBee nodes will switch to the channel C_h. Then, these ZigBee nodes share their ID and ZigBee to WiFi (Z2W) link quality and the node with highest Z2W link quality will transmit a single ACK to the WiFi source. This ACK will indicate which ZigBee nodes have successfully received the packets. Based on the received ACK, the WiFi source can decide whether to terminate the flooding.

B. Flooding Channel Coordination

Traditionally, the WiFi source should conduct the retransmissions for the ZigBee nodes that did not receive the flooding packets. However, since the ZigBee node may not inform the transmission status to the WiFi device immediately, the retransmissions of the flooding packers will suffer a higher delay. In our design, we utilize Luby Transform codes (LT Codes) and develop a flooding channel coordination approach, which has the following benefits: *i) It improves the flooding reliability and reduces the flooding delay;* and *ii) It can transmit additional information to the ZigBee nodes.*

1) Preliminaries on LT Codes: LT codes have been utilized to achieve reliable communication and reduce the network overhead [15]. 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 number of packets.

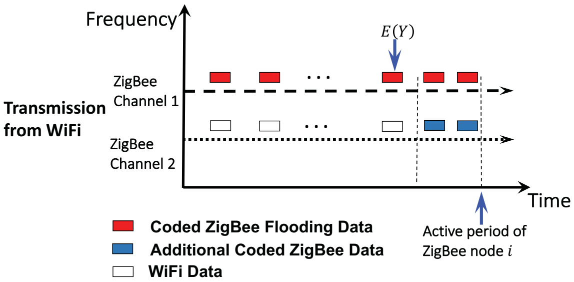

Frequency
$E(Y)$
ZigBee Channel 1
Transmission from WiFi
ZigBee Channel 2
Time
🟥 Coded ZigBee Flooding Data
🟦 Additional Coded ZigBee Data
□ WiFi Data
Active period of ZigBee node i

Fig. 11: After transmitting $E(Y)$ packets, the WiFi source simultaneously uses channel 2 to transmit additional packets.

2) Simple Solutions and Limitations: 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 number of packets, the ZigBee nodes can decode the flooding information. However, this approach may waste the opportunity for transmitting additional information to the ZigBee nodes. Specifically, it is possible for the ZigBee node i to decode the flooding packets before the end of its active period. Then, it will switch to channel C_h to conduct the flooding termination scheme and wait for other nodes to switch to the channel C_h. Meanwhile, since the WiFi source does not know the transmission status of the coded flooding packets, it will terminate the transmission at the end of the node i's active period. In this case, *both ZigBee and WiFi waste their communication resources*. To better show this concept, we have the following analysis.

Formally, based on the LT codes theory [18], a ZigBee node needs to receive Y packets to have the probability ϵ 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 $c \in (0, 1)$.

In practice, due to 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 the link quality between WiFi and the ZigBee node i. Since the ZigBee node will switch to a different channel after successfully decoding the flooding packets, the transmissions from WiFi after the switch are redundant. If we denote the number of packets that can be transmitted from WiFi to ZigBee during its active period τ_i as N_{w2z}, the number of redundant packets transmitted from WiFi can be calculated as follow:

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

Where $E(N_{rp})$ 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. The corresponding probability can be represented as:

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

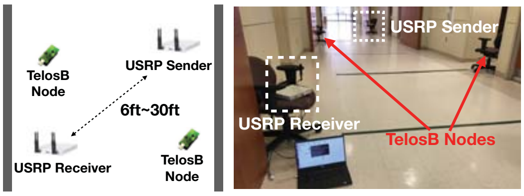


Fig. 12: The deployment in a LoS scenario

Where $\rho(n)$ is the robust soliton distribution ($\rho \leq 1$). 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, ***the ZigBee nodes can receive additional packets during their active periods.***

3) Flooding Channel Coordination: Based on the above analysis, to fully leverage the communication capability of WiFi-to-ZigBee, we leverage the feature that the WiFi has two overlapped channels that can directly communicate with ZigBee nodes.

For the WiFi source. It transmits the flooding packets and additional configuration packets on two WiFi overlapped ZigBee channels, respectively. 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 additional coded packets to the ZigBee node, which is shown in Figure 11. The WiFi source will stop the transmission based on the ZigBee nodes' working schedule.

For the ZigBee node. It senses and receives the coded flooding packets on its working channel. After successfully decoding the packets, it switches to the other overlapped channel to receive additional coded packets. After successfully decoding the additional packets, it will switch to the channel C_h to coordinate with other ZigBee nodes.

By leveraging this approach, for the WiFi source, it fully leverages its communication capability to communicate with ZigBee nodes. For the ZigBee nodes, it receives additional packets on the other overlapped channel by leveraging the possibilities of decoding the flooding packets before the end of the active period.

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, provides no appropriate baselines for comparison. To show the advantages of CRF, we use **PANDO** [15] and **Opportunistic Routing (OPPO)** [19] as the baselines for ZigBee flooding and WiFi routing, respectively.

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 original flooding packets to ZigBee nodes. After successfully receiving the flooding packets, the ZigBee nodes directly transmit ACKs back to the WiFi source by using CTC packet-level modulation scheme.

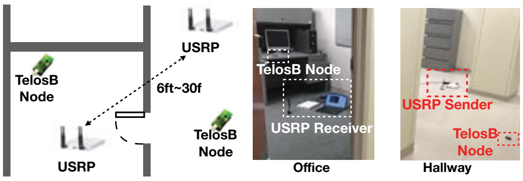
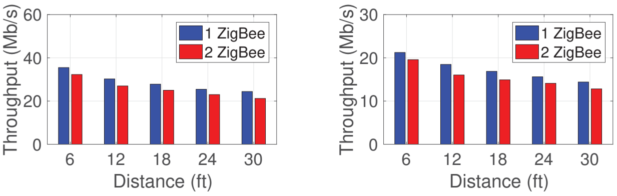


Fig. 13: The deployment in a NLoS scenario



(a) LoS (b) NLoS

Fig. 14: The throughput of WiFi to WiFi

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:

Line-of-sight (LoS): The sender and receivers are within the line-of-sight and the distance varies from 6ft to 30ft, which is shown in Figure 12.

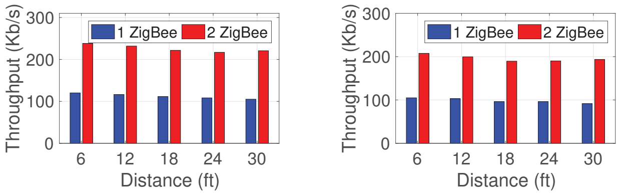
Non-Line-of-sight (NLoS): The sender and receivers are within the non-line-of-sight and the distance varies from 6ft to 30ft, which is shown in Figure 13.

As shown in Figure 14, when communicating with one ZigBee node, the throughput of WiFi to WiFi at 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 15, the throughput of WiFi to one ZigBee node at 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 Figures 16 and 17). 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 packet size varies from 100 bytes to 1,400 bytes. Each experiment is repeated



(a) LoS

(b) NLoS

Fig. 15: The throughput of WiFi to ZigBee

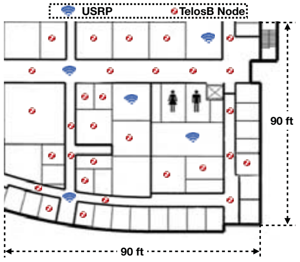


Fig. 16: Deployment in smart building scenario

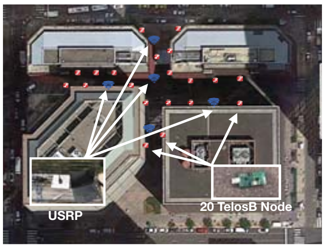


Fig. 17: Deployment in smart city scenario

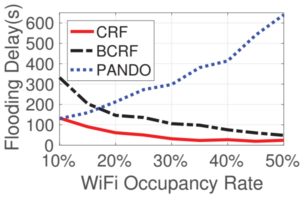


Fig. 18: Flooding Delay vs. WiFi Occupancy Rate (Smart Building)

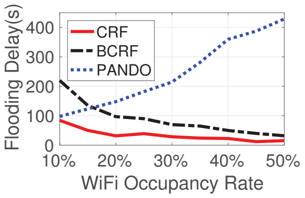


Fig. 19: Flooding Delay vs. WiFi Occupancy Rate (Smart City)

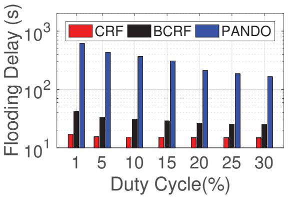


Fig. 20: Flooding Delay vs. Duty cycle (Smart Building)

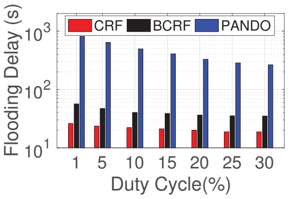


Fig. 21: Flooding Delay vs. Duty cycle (Smart City)

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 18 and Figure 19, CRF shows great advantages over 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 that of 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 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* ***27*** *times lower than the state-of-the-art solution PANDO.*

2) Flooding Delay vs. ZigBee Duty cycle

Figure 20 and Figure 21 show the flooding delay under different duty cycles. As the duty cycle increases, the flooding delay of BCRF and PANDO decreases. 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 $612s$ and $811s$, respectively. In contrast, the corresponding delays of CRF are $17.10s$ and $26.08s$, which is much lower than PANDO. In addition, the flooding delays of BCRF are also as low as $41.64s$ and $56.71s$. This result shows the advantages of CRF.

We also observe that the performance of CRF is much better than BCRF. This is because CRF conducts reliable flooding during the duty-cycle of each ZigBee node. In the contrast, BCRF shows a higher flooding delay since the 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 PANDO and BCRF, respectively.*

3) Reliability Progress vs. Packets Dissemination Time

Figure 22 and Figure 23 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 a scenario.

The dissemination time of BCRF is also higher than 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 decreases 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.*

4) WiFi Network Throughput vs. Number of Transmitted Flooding Packets per Duty Cycle

As shown in Figure 24 and Figure 25, we evaluate the WiFi network throughput under different number of transmitted flooding packets. The throughput of OPPO decreases much faster than CRF and BCRF. This is because OPPO suffers

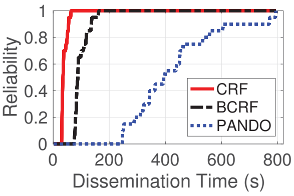


Fig. 22: Reliability Progress vs. Dissemination Time (Smart Building)

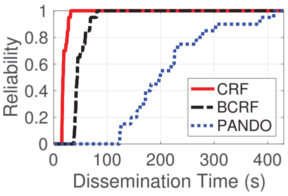


Fig. 23: Reliability Progress vs. Dissemination Time (Smart City)

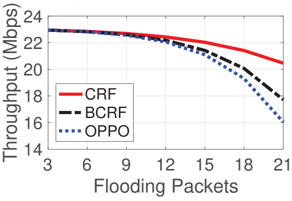


Fig. 24: WiFi Network Throughput vs. Flooding Packets per Duty Cycle (Smart Building)

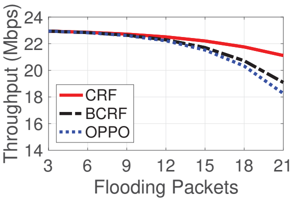


Fig. 25: WiFi Network Throughput vs. Flooding Packets per Duty Cycle (Smart City)

the interference from the ZigBee network. In contrast, CRF and BCRF use 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 $18.26Mbps$ 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 1.2 times higher than OPPO.

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

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 the ZigBee compliant TelosB node [20].

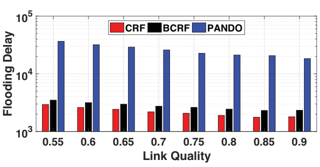


Fig. 26: Flooding Delay vs. WiFi to ZigBee Link Quality

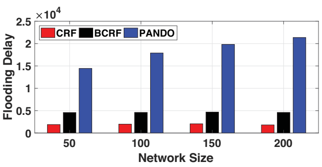


Fig. 27: Flooding Delay vs. Network Size

1) 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 are the flooding delays under different ZigBee to ZigBee link qualities. As shown in Figure 26, CRF shows great advantages to conduct 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, the 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.*

2) Flooding Delay vs. Network Size

We evaluate the flooding delay under different network sizes. As we can see from Figure 27, 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 CRF (21356 time units for 200 nodes). BCRF cannot ensure reliable flooding and routing, so the performance is worse than CRF. **In summary,** *the flooding delay of CRF is almost stable, which is 11.75 times lower than PANDO.*

X. Related Work

The related work can be divided into two categories:

A. Routing & Flooding

Wireless Networks have been investigated to support different smart applications [21]–[26]. **Routing** is one of the key function in this topic. Researchers have proposed various routing protocols for different types of wireless networks, such as the wireless mesh networks [27], [28], the intermittently connected sensor networks [29] and the wireless ad hoc networks [30]. The diversity of wireless networks gives the researchers various features that could help the design of the routing protocols. SocialCast [31] utilizes locations of acquaintances in the social network for routing. R3 [32] is a routing protocol that self-adapts replication for robust routing. Unnecessary forwarding [33] and network coding [34] can significantly improve the network performance in opportunistic routing.

Flooding protocols have been proposed in various wireless networks [35]–[38]. Chorus [39] 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 [40] can be estimated. The reliability of flooding has also been improved when facing unreliable links in low-duty-cycle networks [41].

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 proposed cross-technology communication (CTC) techniques. FreeBee [42] utilizes RSS for communication between WiFi and ZigBee devices. EMF [13] and B^2W^2 [14] embeds information in the existing traffic for concurrent communication among heterogeneous devices. WEBee [10] uses WiFi to emulate ZigBee signals for cross-technology communication. PMC [43] and Chiron [44] enable communication between WiFi and multiple ZigBee devices simultaneously. However, these two approaches require some specific hardwares, 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.

XI. 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.

XII. Acknowledgements

This project is supported by NSF grants CNS-1652669, CNS-1824491, and CNS-1539047.

References

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

[2] K. Lin and P. Levis, "Data discovery and dissemination with dip," in *IPSN 2008*.

[3] K. Whitehouse and D. Culler, "A robustness analysis of multi-hop ranging-based localization approximations," in *IPSN 2006*.

[4] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal clock synchronization in networks," in *Sensys 2009*.

[5] W. Lou and J. Wu, "Double-covered broadcast (dcb): a simple reliable broadcast algorithm in manets," in *INFOCOM 2004*.

[6] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza, "Rbp: Robust broadcast propagation in wireless networks," in *Sensys 2006*.

[7] S. Yun and L. Qiu, "Supporting wifi and lte co-existence," in *INFOCOM 2015*.

[8] Y. He, J. Fang, J. Zhang, H. Shen, K. Tan, and Y. Zhang, "Mpap: Virtualization architecture for heterogenous wireless aps," in *SIGCOMM 2010*.

[9] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving mac layer fairness in wireless packet networks," in *MobiCom 2010*.

[10] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *ACM MobiCom, 2017*.

[11] "http://www.gartner.com/newsroom/id/3598917."

[12] Y. Chen, Z. Li, and T. He, "Twinbee: Reliable physical-layer cross-technology communication with symbol-level coding," in *ACM MobiCom, 2017*.

[13] 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 *INFOCOM*, 2017.

[14] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2W2: N-Way Concurrent Communication for IoT Devices," in *SenSys*, 2016.

[15] W. Du, J. C. Liando, H. Zhang, and M. Li, "When pipelines meet fountain: Fast data dissemination in wireless sensor networks," in *ACM SenSys, 2015*.

[16] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *NSDI, 2010*.

[17] A. Miu, H. Balakrishnan, and C. E. Koksal, "Improving loss resilience with multi-radio diversity in wireless networks," in *MobiCom, 2005*.

[18] M. Luby, "Lt codes," in *FOCS, 2002*.

[19] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," in *SIGCOMM, 2004*.

[20] "http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf."

[21] Z. Huang and T. Zhu, "Real-time data and energy management in microgrids," in *RTSS, 2016*.

[22] Z. Chi, Y. Yao, T. Xie, Z. Huang, M. Hammond, and T. Zhu, "Harmony: Exploiting coarse-grained received signal strength from iot devices for human activity recognition," in *ICNP, 2016*.

[23] Q. Zhang, Z. Zhou, W. Xu, J. Qi, C. Guo, P. Yi, T. Zhu, and S. Xiao, "Fingerprint-free tracking with dynamic enhanced field division," in *INFOCOM, 2015*.

[24] Z. Huang and T. Zhu, "Distributed real-time multimodal data forwarding in unmanned aerial systems," in *SECON, 2017*.

[25] P. Yi, T. Zhu, N. Liu, Y. Wu, and J. Li, "Cross-layer detection for black hole attack in wireless network," *Journal of Computational Information Systems, 2012*.

[26] Z. Huang and T. Zhu, "Leveraging multi-granularity energy data for accurate energy demand forecast in smart grids," in *Big Data, 2016*.

[27] S. Miskovic and E. W. Knightly, "Routing primitives for wireless mesh networks: Design, analysis and experiments," in *INFOCOM 2010*.

[28] R. K. Sheshadri and D. Koutsonikolas, "Comparison of routing metrics in 802.11n wireless mesh networks," in *INFOCOM 2013*.

[29] L. Su, C. Liu, H. Song, and G. Cao, "Routing in intermittently connected sensor networks," in *ICNP 2008*.

[30] C. Sengul and R. H. Kravets, "Bypass routing: An on-demand local recovery protocol for ad hoc networks," *Ad Hoc Networks*, 2006.

[31] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, "Socially-aware Routing for Publish-Subscribe in Delay-tolerant Mobile Ad Hoc Networks," *JSAC 2008*.

[32] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: Robust replication routing in wireless networks with diverse connectivity characteristics," in *MobiCom 2011*.

[33] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *SIGCOMM 2007*.

[34] M. K. Han, A. Bhartia, L. Qiu, and E. Rozner, "O3: Optimized overlay-based opportunistic routing," in *MobiHoc*, 2011.

[35] Z. Li, M. Li, J. Liu, and S. Tang, "Understanding the flooding in low-duty-cycle wireless sensor networks," in *ICPP 2011*.

[36] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, "Green firewall: An energy-efficient intrusion prevention mechanism in wireless sensor network," in *(GLOBECOM, 2012*.

[37] Y. Gu, L. He, T. Zhu, and T. He, "Achieving energy-synchronized communication in energy-harvesting wireless sensor networks," *TECS, 2014*.

[38] S. Zhang, Q. Zhang, S. Xiao, T. Zhu, Y. Gu, and Y. Lin, "Cooperative data reduction in wireless sensor network," *TECS, 2015*.

[39] X. Zhang and K. G. Shin, "Chorus: Collision resolution for efficient wireless broadcast," in *INFOCOM 2010*.

[40] M. Zúñiga and B. Krishnamachari, "Optimal transmission radius for flooding in large scale sensor networks," *Cluster Computing*, 2005.

[41] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *MobiCom 2009*.

[42] S. M. Kim and T. He, "Freebee: Cross-technology communication via free side-channel," in *MobiCom, 2015*.

[43] Z. Chi, Y. Li, Y. Yao, and T. Zhu, "PMC: Parallel Multi-protocol Communication to Heterogeneous IoT Radios within a Single WiFi Channel," in *ICNP*, 2017.

[44] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent High Throughput Communication for IoT Devices," in *MobiSys 2018*.