Portal: Transparent Cross-technology Opportunistic Forwarding for Low-power Wireless Networks

Xiaolong Zheng*, Dan Xia*, Xiuzhen Guo+, Liang Liu*, Yuan He+, Huadong Ma*
* School of Computer Science, Beijing University of Posts and Telecommunications
+ School of Software, Tsinghua University
{zhengxiaolong,xiadan,liangliu,mhd}@bupt.edu.cn,{guoxz16,heyuan}@tsinghua.edu.cn

ABSTRACT

Opportunistic forwarding seizes early forwarding opportunities in duty-cycled networks to reduce delay and energy consumption. But increasingly serious Cross-Technology Interference (CTI) significantly counteracts the benefits of opportunistic forwarding. Existing solutions try to reserve the channel for low-power networks by implicit avoidance or explicit coordination but ignore the potential of high-power CTI's superior capability. In this paper, we propose a new paradigm for low-power opportunistic forwarding in CTI environments. Instead of keeping high-power CTI devices silent, we directly involve them into the forwarding, as cross-technology forwarders. We design *Portal* to solve the challenges of realizing cross-technology opportunistic forwarding. To be transparent to the low-power networks, *Portal* adopts cross-technology rebroadcasting to enable the fast overhearing and forwarding of cross-technology data. To maximize the performance gain of using heterogeneous forwarders while minimizing the influence on legacy high-power traffic, we propose a post-forwarding forwarder selection and a traffic scheduling method. We also propose a feature-based ACK recognition method and a jamming-based ACK replying mechanism to forward the unreliable ACKs from asymmetric regions. Extensive experiments demonstrate that *Portal* not only avoids the CTI but also breaks through the existing performance limit.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Ad hoc networks**; **Routing protocols**; • **Computer systems organization** → **Sensor networks**.

KEYWORDS

Cross-technology communication, opportunistic forwarding, low-power, wireless networks

ACM Reference Format:
Xiaolong Zheng*, Dan Xia*, Xiuzhen Guo+, Liang Liu*, Yuan He+, Huadong Ma*. 2020. *Portal*: Transparent Cross-technology Opportunistic Forwarding for Low-power Wireless Networks. In *The Twenty-first ACM International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Mobihoc '20), October 11–14, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397166.3409134

1 INTRODUCTION

To save energy, low-power wireless networks often run in the duty-cycled mode [4] that nodes sleep most of the time and intermittently turn on radios to receive packets. When a sender has a packet to send, it repeatedly transmits the packet until the intended receiver wakes up and replies the ACK. To shorten the delay of waiting receiver's wake-up, Opportunistic Forwarding (OF, also known as opportunistic routing) [3, 7, 17, 19–21, 23, 25] exploits nodes that wake up earlier and probably have different link qualities as forwarders. By seizing the early and spatially diverse forwarding opportunities, OF is promising to reduce the delay and energy consumption.

However, Cross-Technology Interference (CTI), caused by devices operating on the shared spectrum but following different wireless technologies, counteracts OF's benefits. For example, the high-power CTI such as WiFi in Fig. 1 can influence a large area and corrupt the spatial diversity of low-power links. Hence, ZigBee receivers suffer similar CTI. The chances to find a detour are significantly reduced.

Most of the existing solutions are channel reserving based methods that adjust transmissions based on implicit observations of channel usage patterns or explicit coordinations with CTI devices. For example, Smoggy-link [15] profiles the link qualities under different CTIs and selects ("reserve") the less-affected periods to transmit. Weeble [22] and ECC [30] explicitly stop the high-power CTI to reserve the channel. These methods have to change the original operation of low-power networks, incurring additional delay and non-negligible control overhead. Even not considering any negatives, the optimal performance of these reserving based methods will not exceed the performance in interference-free environments.

In this paper, we propose *Portal*, a new paradigm for low-power opportunistic forwarding in CTI environments. Different from existing works that treat CTI devices as enemies [15] to avoid or as bystanders [30] that just hand over the channel, we turn CTI devices into friends and directly involves them into the forwarding process, as active heterogeneous forwarders. Our key insight is: *rather than keeping*

Mobihoc '20, October 11–14, 2020, Boston, MA, USA

ACM ISBN 978-1-4503-8015-7/20/10…$15.00
https://doi.org/10.1145/3397166.3409134

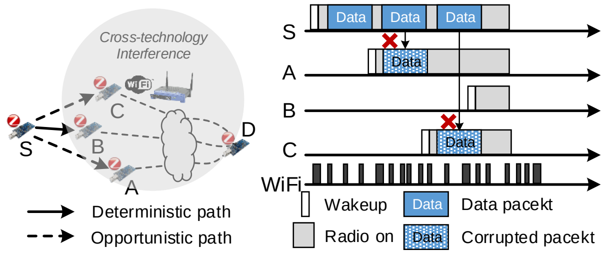


Figure 1: CTI seriously influences the low-power opportunistic forwarding. WiFi interference occupies the channel and interferes multiple opportunistic links at the same time, destroying early wakeup opportunities and spatial diversity.

heterogeneous CTI devices silent, leveraging their superior communication capability to help the low-power opportunistic forwarding is better, for both low-power and high-power networks. On one hand, low-power networks not only avoid CTI by using CTI devices to forward meaningful data, but also can break through the performance limit because high-power CTI forwarders can establish long opportunistic paths that don't exist in original low-power networks. On the other hand, by forwarding data out of the local area faster, high-power networks can reduce the number of channel competitors and obtain better network performance for themselves.

Though attractive, to leverage CTI devices as heterogeneous forwarders is challenging because of the asymmetry of transmission power and channel bandwidth between high-power and low-power networks. First, OF relies on the overhearing ability of early wake-up forwarders to speed up the forwarding with limited overhead. But due to the radio incompatibility between CTI and low-power devices, enabling transparent cross-technology forwarding is challenging. The emerging Cross-Technology Communication (CTC) technique may provide an option. But existing CTC has either a low bit rate or only one-way communication ability. How to achieve fast two-way CTC without affecting the original OF is not clear. Second, OF relies on the in-network routing metric to estimate the forwarding progress and forward in the right direction. But CTI forwarders have no such information. Besides, considering the forwarder's own traffic, it is non-trivial to make beneficial forwarding decisions. Third, OF needs ACK to stop redundant transmissions. However, due to the communication asymmetry, ACK from the low-power receiver may not be received by the high-power forwarder. How to forward the ACK in the reverse direction is also a challenge.

To address those challenges, we propose several designs for *Portal*. First, we propose Cross-Technology Rebroadcasting (CTR) to overhear and forward the cross-technology data. Inspired by LEGO-Fi [11] and WEBee [18], we reuse the WiFi modules to overhear ZigBee signals and generate emulated signals accordingly. CTR is fast because it directly uses the overheard ZigBee signals as the template without running the time-consuming decoding process. Second, *Portal* adopts a post-forwarding forwarder selection and a deadline-driven traffic scheduling method to maximize the performance gain. A CTI forwarder aggressively rebroadcasts the overheard packets and estimates the forwarding gains of next-hop receivers based on the Received Signal Strength (RSS) of the replied ACKs. Third, we propose a feature-based ACK recognition method to overcome the ACK loss problem caused by the communication asymmetry. After receiving or recognizing the replied ACK, the CTI forwarder will stop the local sender by intentional jamming, to avoid duplicate packets.

The contributions of this work are summarized as follows.

- We propose *Portal*, a new paradigm for low-power opportunistic forwarding in CTI environments. *Portal* enables unexplored cross-technology opportunistic paths by exploiting CTI devices as heterogeneous forwarders. Hence, *Portal* not only avoids CTI but also breaks through the performance limit of existing methods.
- We address several challenges of realizing *Portal*, including (i) designing a fast cross-technology rebroadcasting method to forward overheard data quickly, (ii) maximizing the performance gain and minimizing the influence on legacy traffic, (iii) guaranteeing the ACK reliability in asymmetric communication regions.
- We implement a prototype of *Portal* on the software radio platform and commercial ZigBee devices. The experimental results in various environments show that *Portal* can achieve a delay 125× faster than the existing low-power opportunistic forwarding method.

The rest of this paper is organized as follows. Section 2 discusses related works. Section 3 presents a preliminary study that motivates our work. We elaborate on our design of *Portal* in Section 4 and show the evaluation results in Section 5. Finally, we conclude our work in Section 6.

2 RELATED WORK

Opportunistic Forwarding/Routing (OF). The early studies on OF focus on forwarder selection [3, 7, 17], forwarder coordination [19, 25], and duplicate reduction [20]. ExOR [3] is a pioneer work of using opportunistic forwarding for wireless networks. ORW [17] is a representative protocol dedicated to low-power wireless networks that selects the first wake-up receiver as the forwarder to reduce delay. All those works focus on improving OF performance in a homogenous network. While in our work, CTI devices are utilized as heterogeneous forwarders. A few existing works have considered the node heterogeneity in terms of energy budget [21] and traffic load [23]. But none of them considers the incompatible radios. Our work is dedicated to enabling cross-technology opportunistic forwarding for low-power wireless networks.

Anti-CTI. the authors in [25] take link correlation into consideration and point out the correlated links diminishes the benefits of OF. COF [19] avoids the in-network homogenous interference. The authors in [31] recovers the bit errors caused by CTI. Weeble [22] and ECC [30] actively intervene high-power transmissions to aggregate long white spaces for low-power transmissions. The low-power networks wait for the instructions from high-power WiFi devices to transmit.

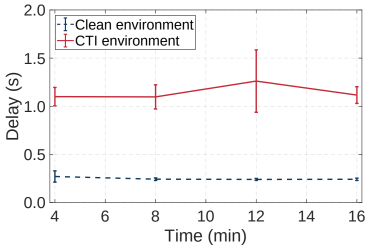


Figure 2: End-to-end delay of ORW in clean and CTI environments.

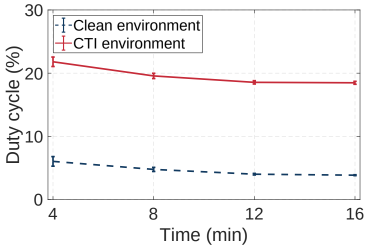


Figure 3: Duty cycle of ORW in clean and CTI environments.

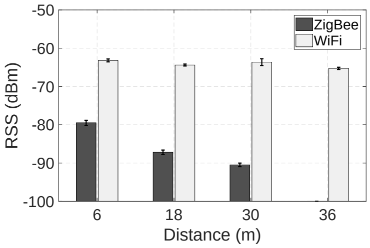


Figure 4: WiFi can extend communication range.

Different from these methods that treat CTI devices as enemies to avoid or as bystanders that hand over the channel, *Portal* directly involves CTI devices into the forwarding by enabling them as heterogeneous forwarders.

Cross-technology Communication (CTC). To achieve the cross-technology forwarding, *Portal* exploits the emerging CTC technique [2, 6, 8–12, 14, 16, 18, 28, 29, 32] that enables the direct communication between incompatible wireless technologies. The physical-layer CTC [2, 8, 11, 14, 18] emulates the signals of another technology to achieve fast CTC. But they are usually one-way because the feasible emulation in the reverse communication direction is different. The packet-level CTC [9, 10, 12, 16, 29, 32] leverages the influence of transmissions of one technology on the other one to convey data. Hence, they can be bidirectional but have a low data rate, introducing too larger latency for OF. Recently, a few works [5, 26–28] start focusing on the networking and application designs of CTC. ECT [28] leverages the raw ZigBee transmissions to encode and upload the high-priority data to WiFi. CRF [27] uses WiFi data to flooding ZigBee packets. Amphista [5] uses ZigBee packets to achieve concurrent transmissions from ZigBee to WiFi and ZigBee. Different from these works, our work is a cross-layer design dedicated to low-power OF that leverages the rebroadcast of WiFi forwarders to facilitate the low-power forwarding process.

3 MOTIVATION

In this section, we first study the impacts of CTI on low-power OF and then analyze the potential performance gain of using CTI heterogeneous forwarders to motivate our work.

3.1 Impacts of CTI on OF

We first study the impacts of CTI on ORW [17], a representative low-power OF method. We deploy five ZigBee nodes in a hallway and control the Tx power at level 3 (-25dBm) to obtain a 4-hop network with linear topology. Each node can reliably communicate with the adjacent node and opportunistically reaches the nodes within two hops. We use channel 26 and 23 to obtain clean and CTI environments, respectively. The sleep interval is set to $2048ms$. The farthest node transmits a packet every $2s$ to $6s$, to the sink node.

Fig. 2 and Fig. 3 present the end-to-end delay and energy consumption (duty cycle) of ORW in clean and CTI environments. We omit the presentation of Packet Reception Ratio (PRR) because PRR is 1 in both scenarios. From the results, we can clearly observe performance degradation in CTI environments. The average delay in CTI environments is $2.29s$, 4.6× larger than the delay in clean environments. The average duty cycle increases from 4.68% in clean environments to 19.59% in CTI environments. This is because the CTI may even cause the failures of deterministic forwarding links, not mentioned the opportunistic links. Even though ORW has multiple forwarding opportunities, it still needs more retransmissions to find an available opportunistic path.

3.2 Underutilized Coexisting CTI

The experimental results reveal the serious impact of CTI on low-power OF. Some recent work such as ECC [30] stops the WiFi devices to reserve the channel for the ZigBee networks. However, due to the duty-cycled operation of ZigBee, the transmission of one packet in a single hop can take hundreds of milliseconds and even seconds, depending on the length of sleep interval. Such a long reserving time will significantly harm the WiFi performance. Even not considering any negative influences on WiFi, the best achievable performance of such a reserving based method will not exceed the performance when running in clean environments.

We argue that the superior capability of the coexisting high-power CTI devices is underutilized. The WiFi Tx power can be 20dBm and the ZigBee Tx power is only 0dBm. Such a large difference leads to a huge difference in communication range. According to Friis transmission formula, the power at receiving antenna is $P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2}$, where P_t is the Tx power, G_t and G_r are the gains of Tx and Rx antennas, λ is wavelength, and d is the distance between Tx and Rx. Given the same parameters, WiFi theoretically has a 4.5× longer communication range than ZigBee. We also measure the RSS of WiFi and ZigBee senders at different distances in a real indoor environment. Due to the limited experiment space, we scale down the Tx power of ZigBee and WiFi to -15dBm and 5dBm, at the same proportion of the highest Tx power. From results in Fig. 4, we can find when the distance is $36m$ where the ZigBee transmission is hard to be distinguished from the noise, the average RSS of WiFi is still above -66dBm.

If a WiFi device can forward ZigBee packets, as shown in Fig. 5, nodes outside the ZigBee sender's communication

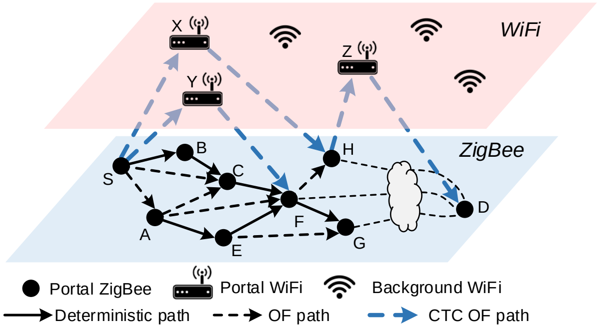


Figure 5: *Portal* exploits the underutilized coexisting CTI devices as heterogeneous forwarders. The WiFi devices with superior communication capability can help to establish new cross-technology shortcuts.

range can also receive the packets. We can establish new short paths that do not exist in original ZigBee networks. The end-to-end delay will be reduced and even much less than the best performance of using only ZigBee nodes. Besides, WiFi devices operate in always-on mode, which means they will be the earliest "wake-up" forwarders. The delay of waiting for the receiver's wake-up can also be reduced. Though attractive, it is challenging to use CTI devices as heterogeneous forwarders due to the inherent incompatibility of wireless technologies.

4 DESIGN

Portal is a new paradigm for low-power opportunistic forwarding in CTI environments. *Portal* exploits the high-power CTI devices as heterogenous forwarders to facilitate the low-power forwarding. In this section, we first present an overview and then introduce the design details of *Portal*.

4.1 Overview

The forwarding procedure in *Portal* is shown in Fig. 6. The source node S in the ZigBee network follows an existing opportunistic forwarding protocol (e.g., ORW [17]. The *Portal* forwarder, WiFi X, overhears S's transmissions and then generates the WiFi transmissions containing emulated ZigBee signals (Section 4.2). The emulated packet will be rebroadcasted by X during the interval between two repeated ZigBee transmissions. To facilitate the low-power data forwarding process while influencing the legacy WiFi transmissions as less as possible, a post-forwarding forwarder selection method and a deadline-driven scheduling method (Section 4.3) are designed. Then ZigBee node D that overhears the rebroadcasted packet will regard it as a packet from node S and reply an ACK. Due to the asymmetry of Tx power, the ZigBee ACK may have a too low SINR at the WiFi forwarder to be forwarded. Hence, *Portal* adopts a feature-based ACK recognition method (Section 4.4) to detect the ACKs. Once overhearing or detecting the replied ACK, the heterogenous forwarder will rebroadcast the ACK or intentionally jam the ZigBee sender to stop the unnecessary local forwarding. By this way, *Portal* establishes a virtual shortcut from node S to D which does not exist in the original ZigBee network and therefore shortens the delivery delay.

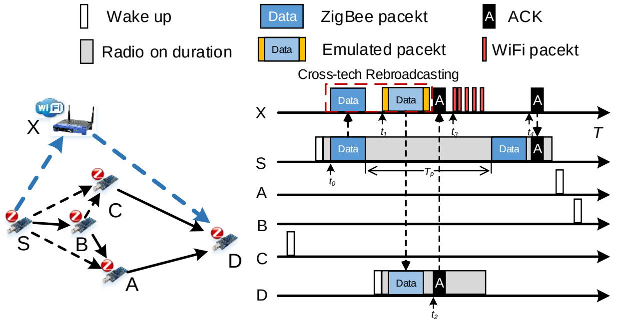


Figure 6: Low-power forwarding procedure in *Portal*.

In the above introduction, we use one WiFi forwarder as an example to illustrate. But *Portal* supports using multiple WiFi forwarders. If there are more than one WiFi forwarders in a local area such as node X and Y in Fig. 5, the first WiFi that schedules ZigBee's rebroadcast and accesses to the channel by CSMA will broadcast first. And other WiFi will drop the same rebroadcast. But notice that for WiFi nodes in adjacent areas such as node X and Z in Fig. 5, they naturally cooperate to achieve multi-hop cross-technology forwarding by *Portal*. The rebroadcasts of X will be received by node H and the H's forwarding will be overheard and rebroadcasted by Z. In the following, without losing generality, we introduce our design using one WiFi forwarder for simplicity.

4.2 Cross-technology Rebroadcasting

Inspired by LEGO-Fi [11] and WEBee[18], we propose a CTC method tailored to low-power opportunistic forwarding. Different from existing physical-layer CTC that focuses on emulating the signal of another technology to achieve one-way communication, we propose the two-way CTC that uses the ZigBee signal overheard by the WiFi RF frontend as a template to generate the rebroadcast signal. By CTR, the heterogenous forwarding can be transparent to the low-power networks and compatible with existing opportunistic forwarding methods.

For the link from ZigBee to WiFi (uplink), we reuse the long preamble detection module of WiFi to detect the deterministic Start of Frame Delimiter (SFD) of a ZigBee packet, similar to LEGO-Fi. When an ongoing transmission is detected but fail to pass the WiFi decoding chain, we compare the received signal r_n and SFD template x_k by using the long preamble detection module to calculate their moving correlation, as follows:

$$corr_i = \sum_{k=0}^{127} r_{i+k} x_k^*, i = 1, 2, ..., n - 127. \tag{1}$$

If a ZigBee SFD exists, a peak larger than a threshold will occur at the start position of SFD. Then we record the ZigBee signal segment for later rebroadcasting. Otherwise, we drop the received signals. The threshold is empirically set at 20.

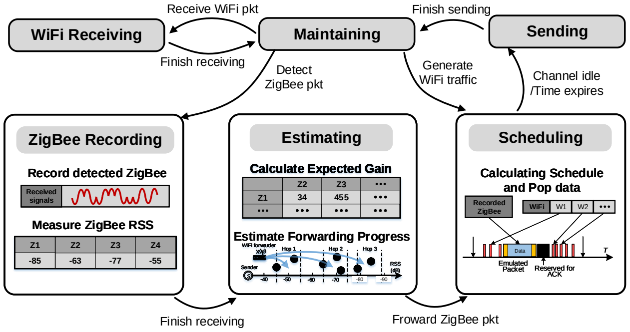


Figure 7: State Machine Diagram of the *Portal* forwarder.

For the link from WiFi to ZigBee (downlink), we leverage the signal emulation technique in WEBee [18]. We use the recorded ZigBee as the template to generate the WiFi signals that contain the interested ZigBee signals. Because we directly use the recorded ZigBee transmission as a template, we can skip a few time-consuming calculations of the cross-technology emulation. The whole process is just like rebroadcasting the received ZigBee signals. Hence, we call our two-way CTC as cross-technology rebroadcasting (CTR).

4.3 Portal's Forwarding Protocol

To maximize the performance gain of using heterogenous forwarders while minimizing the influence on legacy WiFi traffic, an elaborate forwarding protocol is needed. In this subsection, we first present the procedure of *Portal*'s forwarding protocol and then introduce the major components.

4.3.1 Overall Procedure. Fig. 7 presents the state machine diagram of the forwarding protocol running on the *Portal* WiFi forwarder. Initially, a WiFi forwarder enters Maintaining state. When receiving WiFi packets, the WiFi forwarder enters WiFi Receiving state and returns to Maintaining state after finishing receiving WiFi packets by its default WiFi settings. If a ZigBee packet is detected by CTR, the WiFi forwarder will enter ZigBee Recording state and store the received ZigBee for later rebroadcasting. The WiFi forwarder then enters Estimating state and estimates the expected gains of heterogeneous forwarding to select the best next-hop forwarders. Then the WiFi forwarder enters Scheduling state and schedules the transmissions of the emulated ZigBee packet and legacy WiFi packets. When the channel is idle or the waiting time of WiFi traffic expires, the WiFi forwarder enters Sending state and transmits the packets according to the calculated schedule, and then returns to Maintaining state after finishing sending.

The WiFi Receiving and Sending states are WiFi's own procedures. The methods running in ZigBee Recording state has been introduced in Section 4.2. Hence, in the following, we introduce the designs in Estimating and Scheduling states.

4.3.2 Estimation State. To facilitate the forwarding progress, a forwarder has to learn the expected gains to select the best next-hop forwarders. For example, ORW uses the expected duct-cycled wake-ups (EDC) to measure the distance to the destination. But such an in-network routing metric is not available on the WiFi forwarders. In *Portal*, we design a *post-forwarding* gain estimation and forwarder selection method. In the following, we first analyze the theoretical gain of using heterogeneous forwarders and then introduce how we estimate the gain online and select the good next-hop forwarders.

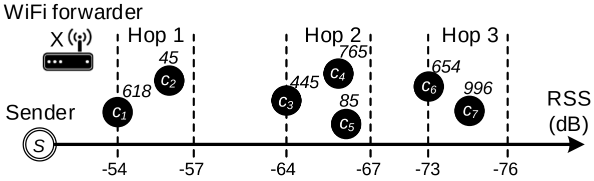


Figure 8: Illustration of RSS-based progress estimation and forwarder selection. The next-hop forwarder candidates are grouped by RSS into multiple hops. The number on top of the ZigBee node is the measured delivery delay (ms) when forwarded by WiFi forwarder, $ETD_{out}(S, C_m)$. The sleep interval of ZigBee nodes T_s is $1000ms$.

Analyze the performance gain. The forwarding progress from node i to j can be measured by the expected delivery delay, $ETD(i, j)$. When using only low-power forwarders, the expected delivery delay in the network is:

$$ETD_{in}(i, j) = (EDC_j - EDC_i) \cdot (\frac{T_s}{2} + T_{tx}) \qquad (2)$$

where T_{tx} is the packet transmission time for a ZigBee sender, T_s is the sleep interval, and $T_s/2$ is the expected waiting time for one wake-up. When using the heterogeneous forwarder X, the expected delay from node i to j is:

$$ETD_{out}(i, j) = T_{tx} \cdot \frac{1}{p_{iX}} + T_{wait} + T_{re} \cdot \frac{1}{p_{Xj}} + \frac{T_s}{2} \qquad (3)$$

where T_{re} is packet rebroadcasting time for WiFi forwarder, p_{iX} is PRR from ZigBee node i to WiFi X, p_{Xj} is PRR from WiFi X to ZigBee node j, and T_{wait} is the scheduled waiting time of X's rebroadcast. Then the performance gain of forwarding by heterogeneous forwarder X is:

$$Gain_X(i, j) = ETD_{in}(i, j) - ETD_{out}(i, j) \qquad (4)$$

To maximize the benefits of using heterogeneous forwarders, *Portal* selects nodes with top-ranked gains as next-hop forwarders. We take the example in Fig. 8 to illustrate how we estimate the gain online and select the next-hop forwarders.

Find the beneficial candidate set. Initially, the WiFi forwarder X aggressively rebroadcasts all the overheard ZigBee packets and listens to the ACKs, until the sender S receives an ACK from a local homogenous forwarder. Notice that only when enough forwarding progress is made for the original sender, a next-hop node replies ACK. Hence, nodes that reply ACKs to the WiFi forwarder before the local homogenous forwarder are the beneficial candidates that can

Table 1: Ranking of the candidate forwarders

Ranking	Node	RSS (dB)	ETD_{in} (ms)	ETD_{out} (ms)	$Gain$ (ms)
1	c_5	-66	1000	85	915
2	c_6	-73	1500	654	846
3	c_3	-64	1000	445	555
4	c_7	-75	1500	996	504
5	c_2	-56	500	45	455
6	c_4	-65	1000	753	247
7	c_1	-54	500	618	-118

both reduce the wakeup waiting delay and make forwarding progress. By this way, we can obtain a set of beneficial forwarders, C. In the example of Fig. 8, $C = \{c_1, c_2, ..., c_7\}$.

Estimate the in-network progress. $ETD_{out}(i, c_m)$ for node $c_m \in C$ can be measured from the above aggressive forwarding, as shown by the number on top of each node in Fig. 8. But to calculate the gain of each candidate, we have to know the in-network metric EDC, which is not available on the WiFi forwarder. To solve this problem, we propose an RSS-based in-network progress estimation method. Notice that we want to rank the candidates rather than calculate the exact EDCs. We observe that $ETD_{in}(i, c_m)$ is highly related to the hop distance between node i and c_m. One-hop progress will reduce the delay by $T_s/2$ on average. Hence, we estimate the relative progress of node c_m to a reference node by measuring the difference of their hop distances. We first select the node with the largest RSS, c_{ref}, as the baseline and then estimate $ETD_{in}(i, c_m)$ as:

$$ETD_{in}(i, c_m) = ETD_{in}(i, c_{ref}) + K(c_{ref}, c_m) \cdot \frac{T_s}{2} + \delta \cdot \frac{T_s}{2} \tag{5}$$

where $K(c_{ref}, c_m)$ is the hop difference between node c_{ref} and c_m, and $\delta \in [-0.3, +0.3]$ is a random variable to estimate the progress difference of nodes with the same hop distance. We start from c_{ref} and group the nodes by an RSS window, i.e., $RSS_i - RSS_{ref} < th_{RSS}$. For example, in Fig. 8, when th_{RSS} is 3dB, c_1 is c_{ref} and forms the hop-1 group with c_2. Then c_3, whose RSS is -64dB, will be the new starting point for grouping. Hence, we can get the hop-2 group, $\{c_3, c_4, c_5\}$ and the hop-3 group, $\{c_6, c_7\}$.

Select the next-hop forwarders. After obtaining the estimated progress, *Portal* calculates the gain of each candidate based Eq. (4) and selects the top-2 ranked nodes as next-hop forwarders. The results of the example in Fig. 8 are shown in Table 1. For simplicity of illustration, we regard the baseline $ETD_{in}(S, c_1)$ as $T_s/2 = 500ms$, and ignore δ, the progress difference in the same group. From Table 1, we can find neither the earliest wake-up node (c_2) or the farthest node (c_7) has the biggest gain. Hence, estimation-based selection is necessary.

4.3.3 Scheduling State. To minimize the influence on the legacy WiFi traffic, *Portal* exploits a deadline-driven scheduling that sends out the WiFi packets with a tolerable latency.

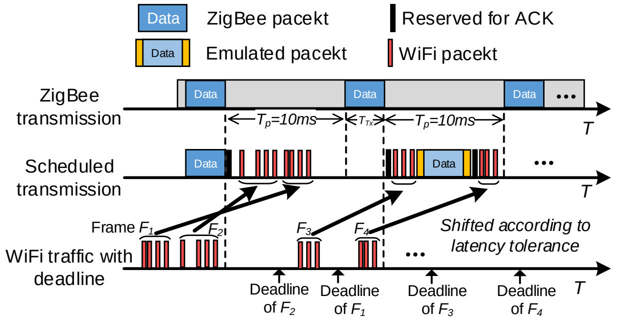


Figure 9: *Portal*'s forwarding scheduling.

Most of the network applications tolerate a latency from tens to hundreds of milliseconds, even for the high user experience desired action games such as DOTA2 and first-person shooting games [1]. Hence, *Portal* schedules the transmissions into the predictable whitespace between two consecutive ZigBee transmissions, which is $T_p = 10ms$ in TinyOS.

We use the example in Fig. 9 to illustrate our scheduling method. The WiFi forwarder has ZigBee and WiFi packets to transmit. We first set a deadline for each WiFi frame according to its network application and then schedule the transmissions to minimize the number of packets missing their deadlines. If there are many packets that cannot be delayed to the next whitespace, the WiFi forwarder will delay the ZigBee rebroadcast and transmit WiFi packets preferentially. For example, in Fig. 9, both frame F_1 and F_2 have deadlines before the next whitespace and F_2 has an earlier deadline. Then *Portal* will transmit F_2 first and then F_1 in the first whitespace, and delay the ZigBee packet to the second whitespace because the remaining period in the first whitespace is not long enough.

Right after the ZigBee rebroadcast, the channel should be reserved for receiving an ACK. According to the IEEE 802.15.4 standard [13], the ACK is transmitted 12 symbol periods ($192\mu s$ in CC2420 [24]) after the last received symbol. The length of an ACK frame is 11 bytes and its transmission takes $352\mu s$. Considering the software delay of the operating system, we set the reserving period as $1ms$.

4.4 ACK Replying

ACK is also necessary to stop redundant forwarding. However, due to the communication asymmetry, a WiFi forwarder is not always able to rebroadcast ACKs from next-hop ZigBee receivers, as illustrated in Fig. 10. In the symmetric region where WiFi and ZigBee can receive each other's packet, ACK can be replayed by CTR just like forwarding data packets. But in the asymmetric region, SINR at the WiFi forwarder will be too low to successfully rebroadcast the ACK.

To solve the problem, we propose a feature-based ACK recognition method to detect the ACKs replied by nodes in the asymmetric region. According to the standard, an 11-Byte ACK ($352\mu s$) is transmitted 12 symbol periods ($192\mu s$) after receiving a packet. Considering the software delay, the Tx timing of ACK is around $1ms$ after receiving the packet.

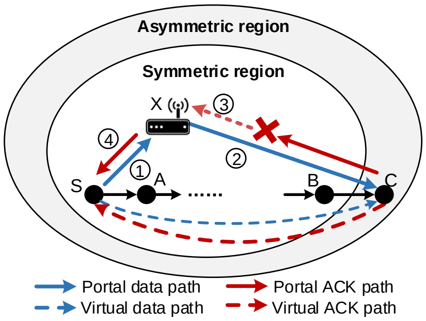


Figure 10: ACK replying in the asymmetric region: ① concurrent overhearing, ② transparent forwarding, ③ feature-based ACK recognition, ④ jamming-based ACK replying.

Hence we use the transmission period and interval between the data packet and ACK as two features to recognize ACKs. Even though the SINR is too low to decode in the asymmetric region, the radio energy is still detectable at the WiFi forwarder. Hence, a WiFi forwarder will listen to the channel after rebroadcasting, and search for the RSS sequence that matches the ACK's features with an allowed error, ε, which is set to $80\mu s$ in our current implementation. To avoid redundant transmissions and duplicates, after getting ACKs from the next-hop forwarders, the WiFi forwarder will intentionally jam local ZigBee forwarders by its traffic.

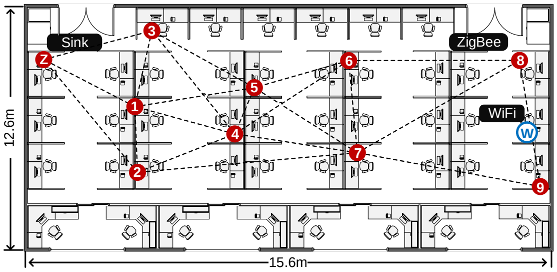
5 EVALUATION

In this section, we extensively evaluate *Portal* in different environments and present the experimental results.

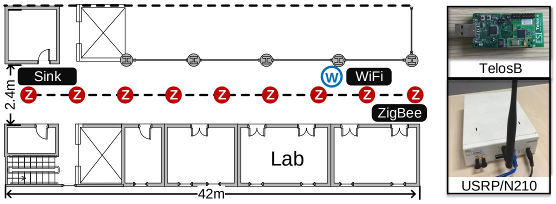
5.1 Experiment Setup

We implement *Portal* on the WiFi compliant USRP N210 and the off-the-shelf ZigBee platform, TelosB, as shown in Fig. 11. ZigBee nodes work in the duty-cycled mode, which is low power listening in TinyOS 2.1.2. We conduct experiments in two environments, the lab and the hallway, the floor plans of which are shown in Fig. 11(a) and Fig. 11(b).

For comparison, we use ORW [17] as the representative work of opportunistic forwarding. We also implement a simplified version of ECC [30], as a representative work of anti-CTI forwarding. The key principle of ECC is grouping the WiFi traffic to leave more whitespace for ZigBee. Hence, we aggressively stop the WiFi when there is ZigBee transmission to obtain the best performance of ECC-based low-power opportunistic forwarding, ECC-ORW for short. We compare the performance in terms of end-to-end delay, energy consumption, and PRR. The energy consumption is measured by the radio duty cycle which is the ratio of radio-on time to the live time because the major energy consumption on ZigBee is from radio activities.



(a) Floor plan of lab



(b) Floor plan of hallway (c) Platform

Figure 11: Experiment settings.

5.2 Benchmarks

We evaluate CTR and feature-based ACK recognition to validate the effectiveness of the forward and reverse links.

5.2.1 Performance of CTR. We first compare CTR and legacy ZigBee to show *Portal* can extend the communication range. We put the WiFi and ZigBee sender at the same location and move the ZigBee receiver from $6m$ to $36m$. From the results shown in Fig. 12, we can find that CTR clearly has a much better PRR when the distance is large. When the distance is $36m$, the ZigBee sender fails to reach the receiver but CTR still has an average PRR of 0.53, demonstrating the ability of CTR to extend communication range. We can also find when the distance is $6m$, CTR has a lower PRR than the legacy ZigBee. This is because the emulated signals of CTR has distortions, compared with the signals from standard radio.

We also investigate the impact of ZigBee packet length. The distance between the *Portal* forwarder and the ZigBee node is $18m$. Tx powers of WiFi and ZigBee sender are 5dBm and -15dBm. We vary the packet length and measure the PRR of CTR in the view of uplink, downlink, and whole link. The results are shown in Fig. 13. The PRR decreases when increasing the packet length. But the downlink is more reliable than the uplink because of the higher Tx power of the WiFi forwarder. For the whole link, the average PRR drops from 0.894 to 0.639 when the packet length increases from 16 Bytes to 80 Bytes. The PRR of *Portal* is good enough because the repeated transmissions of a packet in low duty-cycled mode can work as retransmissions.

5.2.2 Accuracy of Feature-based ACK Recognition. We conduct experiments in the hallway to study the feature-based ACK recognition accuracy and the effectiveness of the reverse links. The ZigBee Tx power is -15dBm. We vary the distance between ZigBee receiver and WiFi sender from $6m$ to $42m$ and measure the recognition performance. We omit

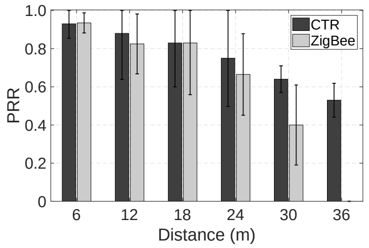


Figure 12: PRR of CTR and ZigBee vs. distance.

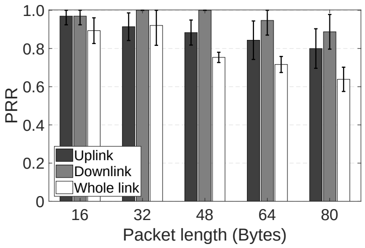


Figure 13: PRR of *Portal* vs. packet length.

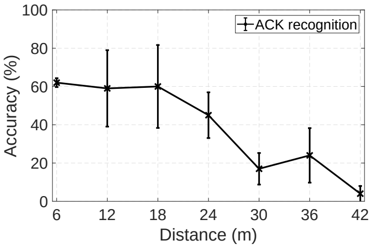
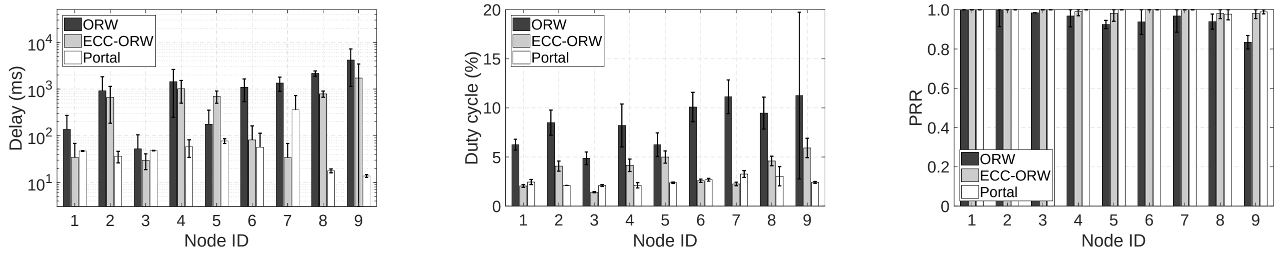


Figure 14: Accuracy of the ACK recognition method.



(a) End-to-end delay (b) Duty cycle (c) PRR

Figure 15: Performance comparison of ORW, ECC-ORW, and *Portal*, in the lab environment.

the presentation of false-positive ratio because it is always 0 during our experiments. This because *Portal* only recognizes ACK after its own transmissions and it is uncommon that an interference just transmits with a fixed interval and packet length same to the ACK. The recognition accuracy (true positive ratio) is shown in Fig. 14. The accuracy decreases with the increases in distance. *Portal* achieves an accuracy of 60% at $18m$, and 22% at $36m$. When the distance increases to $42m$, the recognition becomes unreliable. However, we can rely on the repeated transmissions in ZigBee duty-cycled mode as the retransmissions. Three-times retransmission can significantly improve the recognition accuracy with only additional $20ms$ delay, which is much smaller than the delay of multi-hop forwarding in the original low-power networks.

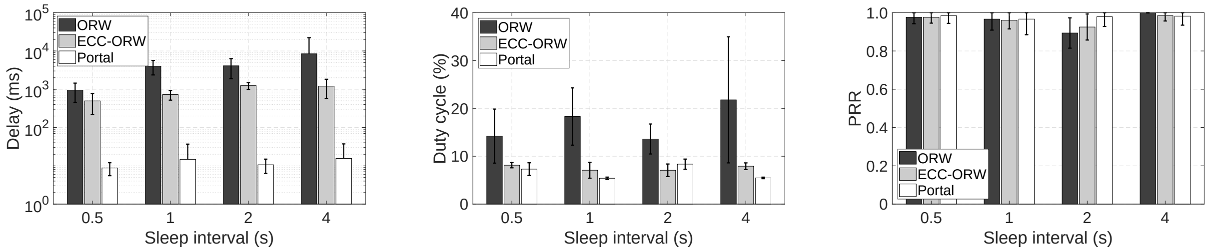
5.3 Performance in Real Environments

We deploy a 10-node low-power ZigBee network in the lab environment, as shown in Fig. 11(a). The sleep interval T_w is $1s$ and each node generates a packet randomly in every 30 seconds. The operating channel is 23, overlapping with campus WiFi networks and our WiFi forwarder. To establish a multi-hop network, we control ZigBee's Tx power to level 4 (-22dBm). WiFi's Tx power is 5dBm. We repeat the experiments five times and each experiment runs one hour.

We compare the performance of ORW, ECC-ORW, and *Portal* in Fig. 15. It is clear that *Portal* significantly reduces the end-to-end delay and duty cycle, especially for nodes far away from the sink. For node 9, the average delays of ECC and ECC-ORW are $4182ms$ and $1722ms$. The average delay of *Portal* is $13.74ms$, which is **304**× and **125**× faster than ORW and ECC-ORW, respectively. The low delay of *Portal* indicates the WiFi forwarder helps forward the packet directly to the sink, jumping over 3 to 4 hops in ZigBee networks. The average duty cycle of *Portal* for all nodes is 2.5%, which is 70.2% and 30.6% smaller than ORW and ECC-ORW because of less forwarding. The results reveal that *Portal* significantly improves performance in real environments.

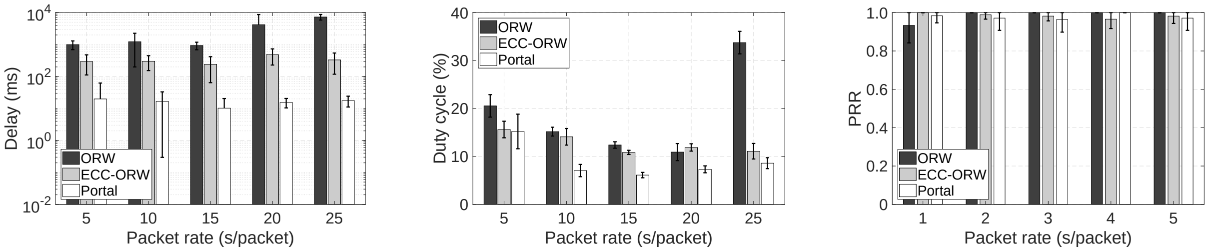
5.4 Different Sleep Intervals

We vary T_w from $0.5s$ to $4s$ to study the impact of sleep interval on performance. We let node 10 transmits a packet randomly every $30s$. The other settings are consistent with the settings in Section 5.3. The results are shown in Fig. 16. From Fig. 16(a), we can find that when the sleep interval increases from $0.5s$ to $4s$, the end-to-end delay of ORW and ECC-ORW increases by 788% and 143%, respectively. This is because CTI in the lab has a serious influence on low-power networks and causes packet retransmissions. The expectation of each retransmission delay is half of the sleep interval. Hence, ORW experiences significant performance degradation. Even though ECC-ORW avoids the CTI influence, it still has a long waiting time of waiting for next-hop receiver's wake-up. This is the born performance limit of the low-power multi-hop forwarding. However, *Portal* can jump multiple hops by a single-hop heterogeneous forwarding. And due to the always-on working mode of WiFi and the sink node, the delay is quite small and stable. Due to the longer waiting delay, the duty cycle of ORW and ECC-ORW also increase but the duty cycle of *Portal* remains in a similar range.



(a) End-to-end delay (b) Duty cycle (c) PRR

Figure 16: Performance of ORW, ECC-ORW, and *Portal* with different sleep intervals.

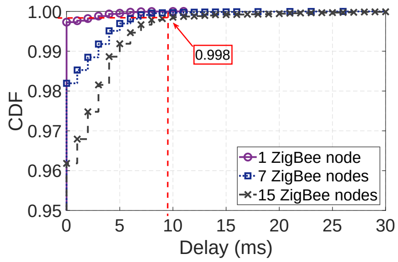


(a) End-to-end delay (b) Duty cycle (c) PRR

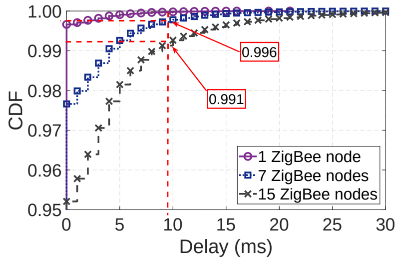
Figure 17: Performance of ORW, ECC-ORW, and *Portal* with different packet transmission rates.

5.5 Different Packet Transmission Rates

We vary T_{pkt} from 5 to 25s/packet to investigate the performance under different packet transmission rates. We let node 10 transmit a packet every T_{pkt} seconds and other nodes forward the packet for node 10. The sleep interval T_w is $512ms$ and other settings are consistent with the settings in Section 5.3. From the results in Fig. 17, we can find *Portal* obtains the smallest end-to-end delay and duty cycle with a similar PRR in all settings. The average delay of *Portal* is $16ms$. Compared with ECC-ORW and ORW, the average delays of which are $330ms$ and $2932ms$, *Portal* improves the performance by 95.1% and 99.5%, respectively. Thanks to the reduced delay, energy consumption is also reduced. The average duty cycle of *Portal* is 8.86%, which is 52% and 30% smaller than ORW and ECC-ORW, respectively.



(a) 100 WiFi packets/s.



(b) 500 WiFi packets/s.

Figure 18: The WiFi packet delay when suing *Portal*.

5.6 Impact on WiFi Traffic

The rebroadcast of ZigBee packets may incur the delay of legacy WiFi traffic. We conduct a trace-based evaluation to study the influence. We record the behaviors of *Portal* with a different number of ZigBee nodes, including the ZigBee packet generated time and rate. Then we simulate the WiFi traffic with two packet rates and calculate the delay, compared with transmitting in interference-free environments. The CDF of delays is shown in Fig. 18. When the WiFi packet rate is 100 packets/s, *Portal* has no influence on more than 95% traffic. Only when 15 ZigBee nodes need rebroadcast, 0.2% of the WiFi packets has delay larger than $10ms$. This is because both WiFi and ZigBee have low data rates and WiFi can focus on its own traffic when there is no ZigBee packet. When WiFi's packet rate is 500 packets/s, 95% of the packets are not influenced. Only 0.4% and 0.9% of the packets have more than $10ms$ delay when the WiFi forwarder covers 7 and 15 ZigBee nodes, respectively. The experimental results reveal that *Portal* has a negligible impact on legacy WiFi traffic.

6 CONCLUSION

We propose *Portal*, a transparent cross-technology opportunistic forwarding method that directly uses CTI devices as heterogeneous forwarders for low-power wireless networks. As a new paradigm for low-power opportunistic forwarding in CTI environments, *Portal* not only avoids the CTI but also breaks through the performance limit by exploiting the superior capability of CTI devices. We design cross-technology rebroadcast, a transparent CTC method, to enable the fast two-way CTC without influencing the original low-power opportunistic forwarding. We propose a post-forwarding forwarder selection and a scheduling method to maximize the performance gain while minimizing the influence on legacy WiFi traffic. We adopt a feature-based ACK recognition method and a jamming-based ACK replying method to establish the reverse links. We extensively evaluate *Portal* in various environments. The experimental results show that *Portal* can achieve the forwarding 125× faster than the state-of-the-art low-power opportunistic forwarding method. *Portal* breaks through the performance limit of existing methods with negligible influence on the legacy WiFi traffic.

ACKNOWLEDGMENTS

This work is supported in part by the Innovation Research Group Project of NSFC (61921003), NSFC (61672320, 61722201, 61720106007, 61772306, 61632008), National Key R&D Program of China (2017YFB1003000), the 111 Project (B18008), and the R&D Project of Key Core Technology and Generic Technology in Shanxi Province (2020XXX007).

REFERENCES

[1] 2012. Some Interesting Bits About Latency. https://www.citycloud.com/some-interesting-bits-about-latency/

[2] Zhenlin An, Qiongzheng Lin, and Lei Yang. 2018. Cross-Frequency Communication: Near-Field Identification of UHF RFIDs with WiFi. In *Proceedings of ACM MobiCom*.

[3] Sanjit Biswas and Robert Morris. 2005. ExOR: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM computer communication review* 35, 4 (2005), 133–144.

[4] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. 2006. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of ACM SenSys*.

[5] Zicheng Chi, Yan Li, Zhichuan Huang, Hongyu Sun, and Ting Zhu. 2019. Simultaneous Bi-directional Communications and Data Forwarding using a Single ZigBee Data Stream. In *Proceedings of IEEE INFOCOM*.

[6] Zicheng Chi, Yan Li, Yao Yao, and Ting Zhu. 2017. PMC: Parallel multi-protocol communication to heterogeneous IoT radios within a single WiFi channel. In *Proceedings of IEEE ICNP*.

[7] Simon Duquennoy, Fredrik Österlind, and Adam Dunkels. 2011. Lossy links, low power, high throughput. In *Proceedings of ACM SenSys*.

[8] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. 2019. WIDE: physical-level CTC via digital emulation. In *Proceedings of ACM/IEEE IPSN*.

[9] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu, and Omprakash Gnawali. 2018. Zigfi: Harnessing channel state information for cross-technology communication. In *Proceedings of IEEE INFOCOM*.

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

[11] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Zihao Yu, and Yunhao Liu. 2019. Lego-fi: Transmitter-transparent ctc with cross-demapping. In *Proceedings of IEEE INFOCOM*.

[12] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. Wizig: Cross-technology energy communication over a noisy channel. In *Proceedings of IEEE INFOCOM*.

[13] IEEE. 2003. IEEE std. 802.15.4-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs). (2003).

[14] Wenchao Jiang, Song Min Kim, Zhijun Li, and Tian He. 2018. Achieving receiver-side cross-technology communication with cross-decoding. In *Proceedings of ACM MobiCom*.

[15] Meng Jin, Yuan He, Xiaolong Zheng, Dingyi Fang, Dan Xu, Tianzhang Xing, and Xiaojiang Chen. 2016. Smoggy-link: Fingerprinting interference for predictable wireless concurrency. In *Proceedings of IEEE ICNP*.

[16] Song Min Kim and Tian He. 2015. Freebee: Cross-technology communication via free side-channel. In *Proceedings of ACM MobiCom*.

[17] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. 2012. Low power, low delay: opportunistic routing meets duty cycling. In *Proceedings of ACM IPSN*.

[18] Zhijun Li and Tian He. 2017. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of ACM MobiCom*.

[19] Daibo Liu, Mengshu Hou, Zhichao Cao, Yuan He, Xiaoyu Ji, and Xiaolong Zheng. 2015. COF: Exploiting concurrency for low power opportunistic forwarding. In *Proceedings of IEEE ICNP*.

[20] Daibo Liu, Mengshu Hou, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. 2016. Duplicate detectable opportunistic forwarding in duty-cycled wireless sensor networks. *IEEE/ACM Transactions on Networking* 24, 2 (2016), 662–673.

[21] Juan Luo, Jinyu Hu, Di Wu, and Renfa Li. 2015. Opportunistic routing algorithm for relay node selection in wireless sensor networks. *IEEE Transactions on Industrial Informatics* 11, 1 (2015), 112–121.

[22] Božidar Radunović, Ranveer Chandra, and Dinan Gunawardena. 2012. Weeble: Enabling low-power nodes to coexist with high-power nodes in white space networks. In *Proceedings of ACM CoNEXT*.

[23] Jungmin So and Heejung Byun. 2017. Load-balanced opportunistic routing for duty-cycled wireless sensor networks. *IEEE Transactions on Mobile Computing* 16, 7 (2017), 1940–1955.

[24] Texas Instruments. 2007. CC2420 datasheet. http://www.ti.com/lit/ds/symlink/cc2420.pdf.

[25] Shuai Wang, Anas Basalamah, Song Min Kim, Shuo Guo, Yoshito Tobe, and Tian He. 2015. Link-correlation-aware opportunistic routing in wireless networks. *IEEE Transactions on Wireless Communications* 14, 1 (2015), 47–56.

[26] Shuai Wang, Zhimeng Yin, Zhijun Li, and Tian He. 2018. Networking support for physical-layer cross-technology communication. In *Proceedings of IEEE ICNP*.

[27] Wei Wang, Xin Liu, Yao Yao, Yan Pan, Zicheng Chi, and Ting Zhu. 2019. CRF: Coexistent Routing and Flooding using WiFi Packets in Heterogeneous IoT Networks. In *Proceedings of IEEE INFOCOM*.

[28] Wei Wang, Tiantian Xie, Xin Liu, and Ting Zhu. 2018. ECT: Exploiting cross-technology concurrent transmission for reducing packet delivery delay in IoT networks. In *Proceedings of IEEE INFOCOM*.

[29] Dan Xia, Xiaolong Zheng, Liang Liu, Chaoyu Wang, and Huadong Ma. 2020. c-Chirp: Towards Symmetric Cross-technology Communication over Asymmetric Channels. In *Proceedings of IEEE SECON*.

[30] Zhimeng Yin, Zhijun Li, Song Min Kim, and Tian He. 2018. Explicit channel coordination via cross-technology communication. In *Proceedings of ACM MobiSys*.

[31] Zhiwei Zhao, Wei Dong, Gonglong Chen, Geyong Min, Tao Gu, and Jiajun Bu. 2017. Embracing corruption burstiness: Fast error recovery for zigbee under wi-fi interference. *IEEE Transactions on Mobile Computing* 16, 9 (2017), 2518–2530.

[32] Xiaolong Zheng, Yuan He, and Xiuzhen Guo. 2018. StripComm: Interference-resilient cross-technology communication in coexisting environments. In *Proceedings of IEEE INFOCOM*.