ECT: Exploiting Cross-Technology Transmission for Reducing Packet Delivery Delay in IoT Networks

WEI WANG, TIANTIAN XIE, XIN LIU, YAO YAO, and TING ZHU, University of Maryland, Baltimore County

Recent advances in cross-technology communication have significantly improved the spectrum efficiency in the same Industrial, Scientific, and Medical band among heterogeneous wireless devices (e.g., WiFi and ZigBee). However, further performance improvement in the whole network is hampered because the cross-technology network layer is missing. As the first cross-technology network layer design, our work, named *ECT*, opens a promising direction for significantly reducing the packet delivery delay via collaborative and concurrent cross-technology communication between WiFi and ZigBee devices. Specifically, ECT can dynamically change the nodes' priorities and reduce the delivery delay from high-priority nodes under unreliable links. The key idea of ECT is to leverage the concurrent transmission of important data and raw data from ZigBee nodes to the WiFi access point. We extensively evaluate ECT under different network settings, and results show that our ECT's packet delivery delay is more than 29 times lower than the current state-of-the-art solution.

CCS Concepts: • **Networks** → **Routing protocols**; **Sensor networks**; *Network performance modeling*; Packet classification; • **Computer systems organization** → **Sensor networks**;

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

ACM Reference format:
Wei Wang, Tiantian Xie, Xin Liu, Yao Yao, and Ting Zhu. 2019. ECT: Exploiting Cross-Technology Transmission for Reducing Packet Delivery Delay in IoT Networks. *ACM Trans. Sen. Netw.* 15, 2, Article 20 (February 2019), 28 pages.
https://doi.org/10.1145/3293536

1 INTRODUCTION

Based on Gartner, the number of Internet of Things (IoT) devices will exponentially increase and reach 20 billion by 2020 [12]. Most of these devices are working within the same Industrial, Scientific, and Medical (ISM) band, which introduces severe communication interference among these devices [8]. Therefore, there is a pressing need to efficiently utilize the spectrum in the ISM band. To meet this requirement, researchers have proposed different cross-technology communication techniques [6, 48] to enable the communication among IoT devices without changing the devices' own protocols.

This project was supported by NSF grants CNS-1652669, CNS-1824491, and CNS-1539047.
Authors' addresses: W. Wang, T. Xie, X. Liu, Y. Yao, and T. Zhu (corresponding author), Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 1000 Hilltop Cir, Baltimore, MD 21250; emails: {ax29092, xtiant1, xinliu1, yaoyaoumbc, zt}@umbc.edu.

1550-4859/2019/02-ART20 $15.00
https://doi.org/10.1145/3293536

These cross-technology communication techniques have the potential to enable the collaboration among IoT devices to more efficiently utilize the spectrum. However, these techniques only provide 1-hop communication among IoT devices. The network performance improvement is hampered because of the lack of network layer design among heterogeneous IoT devices.

In this article, we introduce a new direction for cross-technology communication: a network layer design named *ECT*, which leverages the unique feature of cross-technology communication (i.e., concurrent transmission of data from a ZigBee node to another ZigBee node and a WiFi device). Specifically, ECT leverages the concurrent transmission of important data and raw data from ZigBee nodes to the WiFi access point (AP) for reducing the packet delivery delay.

Although IoT devices will create huge amounts of wireless traffic, only a small amount of data is useful. Based on the Cisco Global Cloud Index, the data created by IoT devices will reach 600ZB (i.e., 6×10^{23} bytes) by 2020, up from 145ZB generated in 2015. However, it is estimated that only approximately 10% of the generated data (60ZB) is useful [37].

Furthermore, the value and priorities of useful data are varying for different devices over time based on the occupant's preference. For example, in a smart building scenario, when the occupants are doing exercise, the air quality sensor's data has a higher priority than the other sensing data to be delivered to the server to make sure that the indoor air quality remains at a healthy level. When the occupant is reading a book, the light intensity sensor's data has a higher priority to make sure that the occupant uses as much natural light as possible while maintaining enough light intensity for his or her eyes.

Traditional approaches either preset the sensors' priorities [26, 40] or decide the sensors' priorities according to the delay, energy consumption, or locally available data [30, 33, 41]. The first approach is not suitable for the network with dynamic priorities while the nodes using the second approach decide their priorities only based on the limited knowledge of the network, which is inaccurate. Therefore, it is better to analyze the data from each IoT device to decide their priorities. However, considering that lots of IoT devices are of low duty cycle, simply transmitting all the data to the server and waiting for the response will introduce higher delay. Then the challenge is how to make the network recognize the worth of every data to decide their senders' priorities as soon as possible.

In addition to the dynamic changing of priorities issue, there are two challenges to be solved. First, most IoT devices are wirelessly connected with each other, which introduces unreliable links [16]. Second, different from traditional low-duty-cycle networks, some IoT devices may introduce higher delay to the network. For example, according to the report [23] conducted by Edison Electric Institute (EEI), Association of Edison Illuminating Companies (AEIC) and Utilities Telecom Council (UTC), the smart meter generates huge amounts of data and the duty cycle of a smart meter is set to 1% to 5% to mitigate congestion to the network. As a result, such duty cycles and potential congestion introduce higher delay, which affects the quality of experience. Due to the constrains of the physical layer (i.e., limited throughput and low duty cycle) in the traditional wireless sensor networks, the combination of dynamic priority determination, unreliable links, and potential high delay makes the problem a new challenging issue.

To overcome these challenges, we leverage the unique feature in cross-technology communication techniques (i.e., concurrent transmission of data from a ZigBee node to another ZigBee node and a WiFi device) [6, 7]. Specifically, in this article, we introduce ECT, an innovative solution that enables a ZigBee sensor node to concurrently transmit important data and raw data to WiFi APs and ZigBee receivers. The important data is generated from raw data, which contains the core information. For example, the important data of a smart meter can be a simple value (e.g., high power consumption), whereas the raw data can be the actual values of power consumption, frequency, and phase angle. When the server receives the important data, it can decide whether it needs the

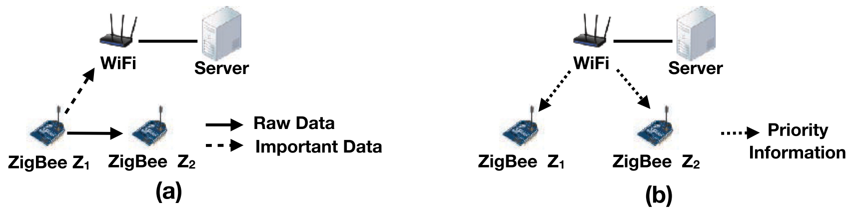


Fig. 1. An example of simplified network architecture. (a) Based on the cross-technology communication techniques [6, 7], the ZigBee node $Z1$ is able to concurrently transmit raw data and important data to ZigBee node $Z2$ and the WiFi AP using the same ZigBee packets. (b) The WiFi AP disseminates the priority information to ZigBee devices.

raw data immediately or not. An example of network architecture is shown in Figure 1. Since the WiFi AP is connected to the server and always active, the important data can arrive at the server with minimum delay, which enables the server to decide the priorities of ZigBee sensors before receiving the raw data. The priority information is then delivered to ZigBee sensors through WiFi nodes. After receiving the priority information, the ZigBee sensors forward the raw data based on the priority of their senders. Therefore, ECT is able to dynamically change the priorities of IoT devices (ZigBee) and reduce the delivery delay for the data from high-priority ZigBee nodes to the server.

In summary, our contributions are as follows:

- To the best of our knowledge, this is the first cross-technology data forwarding method that enables each ZigBee node to concurrently transmit important data and raw data to the WiFi and other ZigBee nodes. Our design not only allows the server to dynamically decide the priorities of the ZigBee nodes but also decreases the packet delivery delay of the higher-priority ZigBee nodes.
- Our network model is the first in-depth model of the cross-technology low-duty-cycle ZigBee network with presence of other heterogeneous IoT devices (i.e., WiFi APs).
- We extensively evaluate our design under various network settings, and the results show that our ECT's packet delivery delay is more than 29 times lower than the state-of-the-art solution.

The rest of the article is organized as follows. Section 2 introduces the network model. Section 3 describes the main design of ECT. In Section 4, we talk about the design issues and optimization. In Sections 5 and 6, we introduce our evaluation and simulation results. Section 7 presents the related work. In Section 8, we conclude the article.

2 MODELS

In this section, we first briefly introduce the cross-technology communication techniques. Then, we present our cross-technology network model of the heterogeneous IoT networks that contains ZigBee nodes and WiFi devices. Last, we describe the working schedule model of ZigBee nodes.

2.1 Cross-Technology Communication Techniques

Cross-technology communication techniques have been proposed to improve the spectrum utilization efficiency between heterogeneous IoT devices (e.g., ZigBee, WiFi, and Bluetooth) with different communication techniques. Normally, cross-technology communication mainly leverages the overlapped channel between different devices and WiFi to enable cross-technology communication. As shown in Figure 2, WiFi channel 6 is overlapped with ZigBee channels from 16 to 19. For ZigBee to WiFi communication, the ZigBee node needs to operate on the overlapped

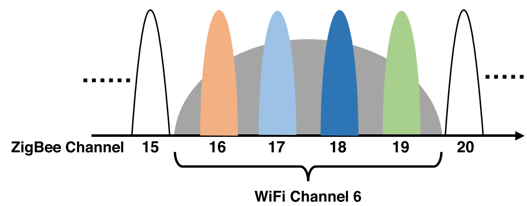


Fig. 2. The overlapped channels between ZigBee and WiFi.

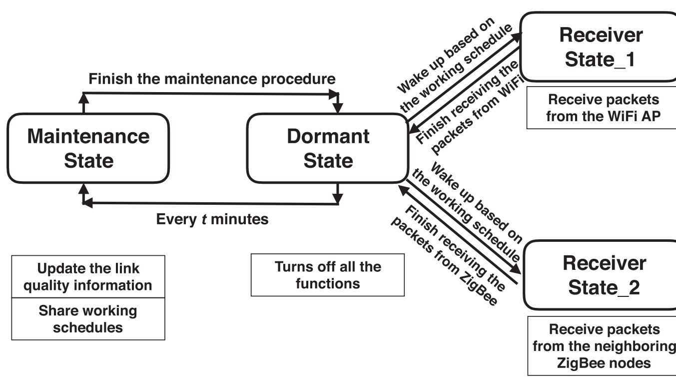


Fig. 3. The state transition diagram of ZigBee nodes.

channel. To make WiFi understand the packets transmitted from the ZigBee nodes, the ZigBee node can utilize shifted beacons [27], transmission power [18], and modified data traffic [24]. The WiFi receiver can detect the Received Signal Strength (RSS) information or channel state information to demodulate the packets from a ZigBee sender. To enable WiFi to ZigBee communications, the WiFi sender normally transmits shifted beacons or packets, and the message to ZigBee is embedded into the time interval difference between each beacon or packets [45]. The ZigBee receiver will detect such difference and demodulate the packets. Since the communication from ZigBee to WiFi and WiFi to ZigBee does not change the original packet from ZigBee to ZigBee and WiFi to WiFi, the ZigBee sender and WiFi sender are able to concurrently communicate with heterogenous IoT devices with different communication techniques.

2.2 Cross-Technology Network Model

Without loss of generality, we assume that there are N ZigBee nodes and M WiFi devices. Both ZigBee nodes and WiFi devices can communicate to the server over multiple hops. ZigBee nodes and WiFi devices can communicate with each other using cross-technology communication techniques [6, 7]. Each ZigBee node has four possible states: the maintenance state, the receiver state 1, the receiver state 2, and the dormant state.

The state transition diagram is shown in Figure 3. In the dormant state, a ZigBee node turns off all of its functions except a timer to wake up. Base on the working schedule, a ZigBee node can wake up and receive packets from WiFi or its neighboring ZigBee nodes. Specifically, in the receiver state 1, a ZigBee node will sense and receive packets from the WiFi AP. After receiving the packets, it will go back to the dormant state. Similarly, in the receiver state 2, a ZigBee node will receive packets from its neighboring ZigBee nodes. Every t minutes, a ZigBee node will switch to the maintenance state to update the link quality between its neighboring ZigBee nodes and the WiFi AP. In addition, during this time, a ZigBee node will share its working schedules with its neighboring nodes. Each ZigBee node is also synchronized with its neighboring nodes, which can

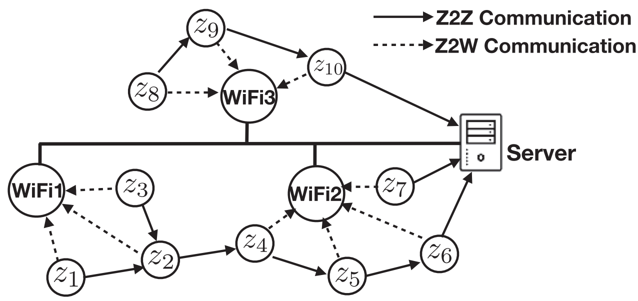


Fig. 4. An example of the IoT network. ZigBee nodes can concurrently communicate with its neighboring ZigBee nodes and the WiFi device.

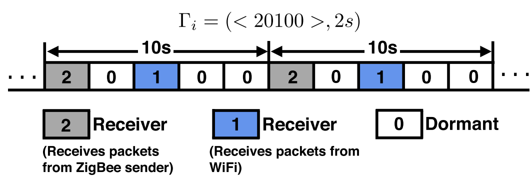


Fig. 5. The working schedule of ZigBee node i.

be achieved by using FTSP [32] or FCSA [44]. In summary, a ZigBee node can transmit packets at any time but can only receive WiFi or ZigBee packets in the receiver state 1 and receiver state 2.

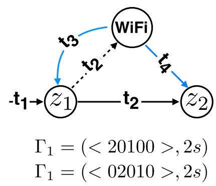
The network status at time t can be represented as $G_k(t) = (V, y_k, Z(t), W(t))$, where V is a complete set of N ZigBee nodes in the network and y_k represents a WiFi device. $Z(t)$ is the set of directed edges between ZigBee nodes at time t, whereas $W(t)$ is the corresponding set of directed edges between ZigBee and WiFi devices. An edge $z(i, j)$ belongs to $Z(t)$ if and only if (1) node j is the neighboring node of i and (2) node j is in the receiver state. The edge $w(i, y)$ belongs to $W(t)$ and denotes the link between ZigBee node i and the WiFi device y. In summary, $G_k(t)$ is a time-dependent network, and the traffic flow in the network varies over time t.

Figure 4 shows an example of the network. It has three WiFi APs and 10 ZigBee nodes. The ZigBee nodes can concurrently transmit packets to their neighboring ZigBee nodes and WiFi devices using cross-technology communication techniques [6, 7]. Since the WiFi APs are always active, the packets transmitted to the WiFi APs can be delivered to the server with very low latency.

2.3 Working Schedule Model

Since ZigBee nodes work under a lowdutycycle, the working schedule of a ZigBee node i can be represented as $\Gamma_i = (\omega, \tau)$, where ω denotes the states of the node and τ denotes the time duration of each state. ω has three possible values: 0, 1, and 2. The dormant state is denoted as 0, whereas the receiver state is denoted as 1 and 2. Specifically, 1 denotes that node i is ready to receive packets from the WiFi AP, and 2 denotes denotes that node i is ready to receive packets from its neighboring ZigBee nodes.

Figure 5 shows an example of the working schedule of ZigBee node i. $\Gamma_i = (< 20100 >, 2s)$ denotes the working schedule of node i. The working period of the node is 10s, which is divided equally into five time slots and each time slot lasts for 2s. During the first time slot, the node is in receiver state 2 and ready to sense and receive packets from its neighboring ZigBee nodes. Then, the node goes back to the dormant state. In the third time slot, the node wakes up again and switches to the receiver state 1 to receive packets from the WiFi AP. Based on this model, we can compute the delay introduced by the working schedule in this network.



Time Slot	Z_1 Schedule	Z_2 Schedule	Event
t_1	2	0	Neighboring Node –> Z_1
t_2	0	2	Z_1 –> { Z_2, WiFi }
t_3	1	0	WiFi –> Z_1
t_4	0	1	WiFi –> Z_2
t_5	0	0	Dormant

Fig. 6. An example of the delay introduced by the working schedule.

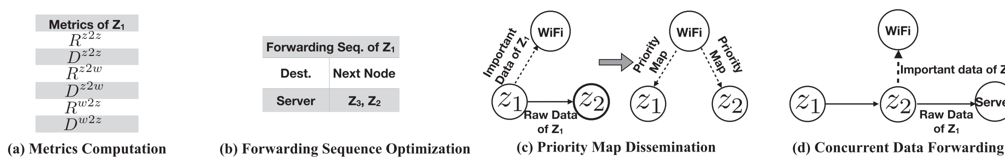


Fig. 7. Design overview.

As shown in Figure 6, the working schedules of ZigBee nodes z_1 and z_2 are $\Gamma_1 = (< 20100 >, 2s)$ and $\Gamma_2 = (< 02010 >, 2s)$. As shown in this example, z_1 receives packets from its neighboring ZigBee nodes at t_1 and tries to forward them to z_2. However, according to the working schedule of z_2, those packets are delayed because z_1 has to wait until z_2 is in the receiver state at t_2. Similarly, if the WiFi AP wants to transmit priority information to z_1 and z_2, it has to wait until z_1 and z_2 are in the receiver states at t_3 and t_4, respectively. Therefore, different from the traditional low-duty-cycle sensor network, our network has three types of delay: ZigBee to ZigBee (Z2Z), ZigBee to WiFi (Z2W), and WiFi to ZigBee (W2Z).

3 MAIN DESIGN

The design goal of ECT is to reduce the data delivery delay from high-priority ZigBee nodes to the server under the constraint that the priorities of ZigBee nodes are dynamically changing. To achieve this goal, we encounter three major challenges: how to mathematically analyze the network with the presence of ZigBee nodes and WiFi APs, how to mathematically analyze the network with the presence of ZigBee nodes and WiFi APs, how to reduce the delivery delay of ZigBee nodes, and how to update priorities of each ZigBee node. To overcome these challenges, we propose six metrics to evaluate the network performance. Then, based on these metrics, we adopt a dynamic programming approach to find the optimal forwarding sequence for each ZigBee node to reduce the delivery delay. After that, a priority map is introduced to dynamically update the priorities of each ZigBee node.

3.1 Design Overview

As illustrated in Figure 7, ECT consists of four major steps:

(1) ***Metrics Computation***: ZigBee nodes compute their own the Expected Z2Z Delivery Ratio (R^{z2z}), Expected Z2Z Delivery Delay (D^{z2z}), Expected Z2W Delivery Ratio (R^{z2w}), Expected Z2W Delivery Delay (D^{z2w}), Expected W2Z Delivery Ratio (R^{w2z}), and Expected W2Z Delivery Delay (D^{w2z}). Then, the ZigBee nodes share these metrics with their neighboring nodes.

(2) ***Forwarding Sequence Optimization***: According to the metrics computation results, each ZigBee node adopts a dynamic programming approach to find out its optimal forwarding sequence to reduce the delivery delay.

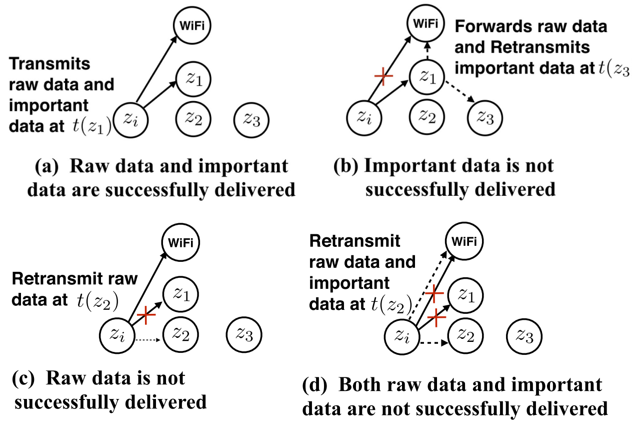


Fig. 8. An example of the transmission scheme of ZigBee node i. The forwarding sequences of z_i and z_1 are $S_i^n = (z_1, z_2)$ and $S_1^n = (z_3)$, respectively. The wake-up time of each ZigBee node is denoted as $t(z_1), t(z_2)$, and $t(z_3)$, where $t(z_1) < t(z_2) < t(z_3)$.

(3) ***Priority Map Dissemination***: Based on the optimal forwarding sequence, the ZigBee nodes concurrently transmit the important data and raw data to the WiFi APs and the ZigBee forwarders. Then, the WiFi APs transmit the important data to the server for priority determination. Based on the important data, a priority map is generated by the server and then transmitted back to the WiFi AP. Last, the WiFi APs disseminate the priority map to the ZigBee nodes during their receiver states.

(4) ***Concurrent Data Forwarding***: Each ZigBee node checks the priority map and forwards the raw data based on its senders' priorities.

The first and second steps can be done during the maintenance state of each ZigBee node. For the third step, the ZigBee nodes will receive the priority maps during the receiver state 1. The raw data and important data are forwarded to the ZigBee forwarders and the WiFi APs during the ZigBee forwards' receiver state 2.

3.2 Metrics Computation

In this section, we first show why we need new metrics. Then, we propose our metrics and elaborate on their meanings. Finally, we find the optimal forwarding sequence based on these metrics.

3.2.1 The Need for New Metrics. To show why we need new metrics, we need to understand the forwarding sequence and transmission scheme in our design. Formally, we define the forwarding sequence of a ZigBee node i as follows.

Definition 1 (Forwarding Sequence (S_i^n)). S_i^n is the sequence that consists of n ZigBee nodes that can leverage ZigBee to ZigBee and ZigBee to WiFi communications to concurrently forward packets from ZigBee node i to the server. The forwarding sequence is denoted as $S_i^n = (z_1, \ldots, z_n)$, which is sorted in the order of wake-up time $t(z_1) < \cdots < t(z_n)$.

In our design, a ZigBee sender has multiple potential forwarding nodes at each hop. As shown in Figure 8(a), the forwarding sequence of z_i is $S_i^2 = (z_1, z_2)$. If the transmissions from z_i to the WiFi AP (important data) and z_i to z_1 (raw data) are successful, z_i ends the transmission and goes back to the dormant state. However, since the radio links are unreliable, it is possible that the important data or raw data is not successfully delivered to the WiFi AP or ZigBee receiver. In this case, we introduce the following retransmission scheme.

Retransmission of important data. If the important data is not successfully delivered to the WiFi AP while the raw data is successfully delivered to the ZigBee receiver, the ZigBee receiver retransmits the important data to the WiFi AP and concurrently forwards the raw data to the node in its forwarding sequence. As shown in Figure 8(b), the important data is not successfully delivered to the WiFi AP by z_i. Then, z_1 forwards the raw data to z_3 and retransmits the corresponding important data to the WiFi AP at z_3's receiver state. In this case, the important data is delayed by time $t(z_3) - t(z_1)$. This is because the cross-technology communication techniques require the Z2W traffic to be embedded in Z2Z traffic. Since the raw data has been successfully delivered to z_1, only z_1 can generate Z2Z traffic (i.e., forwards the raw data).

Retransmission of raw data. If the transmission of raw data is failed while the important data is successfully delivered to the WiFi AP, instead of waiting for the particular ZigBee receiver to wake up again, the ZigBee sender retransmits the raw data according to the working schedule of the next node in its forwarding sequence. As shown in Figure 8(c), the raw data is not successfully delivered to z_1. Then, z_i retransmits the raw data to z_2 at time $t(z_2)$. In this case, the raw data is delayed by time $t(z_2) - t(z_1)$. The major advantage of this scheme is that the ZigBee sender is able to reduce the waiting time for a particular ZigBee receiver to wake up again.

Retransmission of both data. If the transmission of raw data and important data failed, the ZigBee sender retransmits the raw data and important data according to the working schedule of the next node in its forwarding sequence. As shown in Figure 8(d), both raw data and important data are not successfully delivered. Then, z_i concurrently retransmits the raw data and important data to z_2 and the WiFi AP at time t_2. In this case, both raw data and important data is delayed by time $t(z_2) - t(z_1)$.

To inform a ZigBee node to forward the data based on its senders' priorities, the WiFi AP transmits the priority information during the receiver state of the target ZigBee node. However, due to the unreliable links, if the priority information is not successfully delivered, the ZigBee node has to forward the raw data according to its senders' original priorities, which may introduce delay for the raw data from high-priority ZigBee nodes.

Therefore, to reduce the delivery delay of raw data from a high-priority ZigBee node, we not only need to consider the packet delivery ratio and delay between ZigBee and ZigBee but also need to consider the packet delivery ratio and delay from ZigBee to WiFi and WiFi to ZigBee.

3.2.2 Metrics Computation. We propose Expected Z2Z Delivery Ratio (R_i^{z2z}), Expected Z2Z Delivery Delay (D_i^{z2z}), Expected Z2W Delivery Ratio (R_i^{z2w}), Expected Z2W Delivery Delay (D_i^{z2w}), Expected W2Z Delivery Ratio (R_i^{w2z}), and Expected W2Z Delivery Delay (D_i^{w2z}) to analyze the network. These metrics are defined in Table 1.

To show how to compute these metrics, we take ZigBee node i as an example. Assuming that the forwarding sequence of ZigBee node i is $S_i^n = (z_j, \ldots, z_n)$, these metrics can be calculated as follows.

Expected Z2Z Delivery Ratio (R_i^{z2z}). We first denote the link quality between ZigBee i and j as p_{ij}^{z2z}. Then, the probability for a packet transmitted by the node i and successfully received by the ZigBee node j in its forwarding sequence S_i^n can be represented as follows:

$$P_{ij}^{z2z} = \left(\prod_{k=1}^{j-1} \left(1 - p_{ik}^{z2z}\right)\right) p_{ij}^{z2z} \tag{1}$$

Table 1. Definitions of Metrics

Metrics	Definitions
R_i^{z2z}	Expected delivery ratio for a packet transmitted from node i and received by the server through Z2Z communication
D_i^{z2z}	Expected delivery delay for a packet transmitted from node i and received by the server through Z2Z communication
R_i^{z2w}	Expected delivery ratio for a packet transmitted from node i and received by the WiFi AP
D_i^{z2w}	Expected delivery delay for a packet transmitted from node i and received by the WiFi AP
R_i^{w2z}	Expected delivery ratio for a packet transmitted from the WiFi AP and received by ZigBee node i
D_i^{w2z}	Expected delivery delay for a packet transmitted from the WiFi AP and received by ZigBee node i

By leveraging Equation (1), the Expected Z2Z Delivery Ratio R_i^{z2z} is the summation of the product of the probability for a packet successfully received by the neighboring ZigBee node j and its corresponding R_j^{z2z}, which can be represented as follows:

$$R_i^{z2z} = \sum_{j=1}^{n} P_{ij}^{z2z} R_j^{z2z} \tag{2}$$

Expected Z2Z Delivery Delay (D_i^{z2z}). To calculate the Expected Z2Z Delivery Delay (D_i^{z2z}), we first need to understand that the raw data successfully forwarded to the server is under the probability that the raw data is forwarded by one of the nodes in S_i^n, which can be represented as follows:

$$P_{ij|S_i^n}^{z2z} = \frac{P_{ij}^{z2z} R_j^{z2z}}{R_i^{z2z}} \tag{3}$$

Then, let t_i^j denote the delay for node i to wait for node j to switch to the receiver state. The Expected Z2Z Delivery Delay (D_i^{z2z}) for node i can be represented as follows:

$$D_i^z = \sum_{j=1}^{n} \left(\left(t_i^j + D_j^{z2z} \right) P_{ij|S_i^n}^{z2z} \right) \tag{4}$$

Expected Z2W Delivery Ratio (R_i^{z2w}). As mentioned previously, if the important data are not successfully delivered to the WiFi AP while the raw data is successfully delivered to the ZigBee receiver, the receiver will retransmit the important data to the WiFi AP. Therefore, R_i^{z2w} not only depends on the link quality from node i to the WiFi AP but also depends on the link qualities from node i to the nodes in its forwarding sequence and their corresponding Z2W link qualities.

Formally, we denote the link quality from ZigBee node i to WiFi AP y as p_{iy}^{z2w}. Then, we can represent the probability for a packet successfully delivered from node i to WiFi AP y after k times transmissions as $P_{iy}^{z2w}(k)$, which is the product of the probability for a packet successfully delivered to the WiFi AP at kth attempts and the probability for the raw data failed to be delivered to the nodes in S_i^n at $(k-1)$th attempts:

$$P_{iy}^{z2w}(k) = p_{iy}^{z2w} \left(1 - p_{iy}^{z2w}\right)^{k-1} \prod_{j=1}^{k-1} \left(1 - p_{ij}^{z2z}\right) \tag{5}$$

Then, the Expected Z2W Delivery Ratio R_i^{z2w} for ZigBee node i can be represented as follows:

$$R_i^{z2w} = \sum_{j=1}^{n} P_{iy}^{z2w}(j) + \sum_{j=1}^{n} P_{ij}^{z2z}\left(1 - p_{iy}^{z2w}\right)^j R_j^{z2w} \tag{6}$$

The first term is the probability for the packet to be successfully delivered to the WiFi AP by node i. The second term represents the probability for a packet to be delivered by the node in S_i^n, which is the probability for the packet failed to be delivered to the WiFi AP by node i and successfully delivered to the node in S_i^n multiplied by the probability for the packet to be successfully delivered to the WiFi by the node in S_i^n.

***Expected Z2W Delivery Delay* (D_i^{z2w}).** Similar to the Expected Z2W Delivery Ratio, D_i^{z2w} not only depends on the delivery delay from ZigBee node i to the WiFi AP but also depends on the delivery delay from the nodes in its forwarding sequence to the WiFi AP. The mathematic representation is shown next:

$$\begin{aligned} D_i^{z2w} &= \sum_{j=1}^{n} t_i^j P_{iy}^{z2w}(j) \\ &\quad + \sum_{j=1}^{n} P_{ij|S_i^n}^{z2z}\left(1 - p_{iy}^{z2w}\right)^{j-1}\left(t_i^j + D_j^{z2w}\right) \end{aligned} \tag{7}$$

The first term represents the delay for the packet successfully delivered by ZigBee node i, and the second term represents the delay for the packet successfully delivered by the nodes in node i's forwarding sequence.

***Expected W2Z Delivery Ratio* (R_i^{w2z}).** To forward raw data according to its sender's priority, the priority information should be delivered to the node that receives that data. If the priority information failed to be delivered to the target ZigBee node i, the raw data may be delivered to the node in S_i^n with its original priorities. Therefore, ideally, the link quality between ZigBee and ZigBee also affects the Expected W2Z Packet Delivery Ratio (R_i^{w2z}). We denote the link quality from WiFi AP y to ZigBee node i as p_{yi}^{w2z}. Then, R_i^{w2z} can be represented as follows:

$$\begin{aligned} R_i^{w2z} &= \sum_{j=1}^{n}\left(p_{yi}^{w2z}\left(1 - p_{yi}^{w2z}\right)^{j-1}\prod_{m=1}^{j-1}\left(1 - p_{im}^{z2z}\right)\right) \\ &\quad + \sum_{j=1}^{n}\left(P_{ij}^{z2z}\left(1 - p_{yi}^{w2z}\right)^j R_j^{w2z}\right) \end{aligned} \tag{8}$$

The first term represents the probability for the priority information delivered to node i, which is the probability for the priority successfully delivered to the ZigBee sender multiplied by the probability for the raw data failed to be delivered to the ZigBee receiver in S_i^n. The second term represents the probability for the priority information delivered to the ZigBee receiver in the forwarding sequence S_i^n, which is the probability for the raw data successfully delivered to the ZigBee receiver multiplied by the probability for the priority information successfully delivered to that ZigBee receiver.

***Expected W2Z Delivery Delay* (D_i^{w2z}).** Similar to R_i^{w2z}, the Expected W2Z Delivery Delay not only depends on the link quality between the WiFi AP and ZigBee but also depends on the link quality between ZigBee and ZigBee. The representation of D_i^{w2z} for node i in S_i^n is shown next:

$$\begin{aligned} D_i^{w2z} = &\sum_{j=1}^{n} \left(t_i^j p_{yi}^{w2z} \left(1 - p_{yi}^{w2z}\right)^{j-1} \prod_{m=1}^{j-1} \left(1 - p_{im}^{z2z}\right)\right) \\ &+ \sum_{j=1}^{n} \left(\left(t_i^j + D_j^{w2z}\right) P_{ij}^{z} \left(1 - p_{yi}^{w2z}\right)^{j-1} R_j^{w2z}\right) \end{aligned} \tag{9}$$

The first term represents the delay for the priority information to be delivered to the node i. The second term represents the delay for the priority information to be delivered to the nodes in the forwarding sequence S_i^n.

3.3 Forwarding Sequence Optimization

Based on the proposed metrics, we find the optimal forwarding sequence S_i^{opt} by using a dynamic programming approach to reduce the Expected Z2Z Delivery Delay.

Obviously, the easiest solution is to select only one node from S_i^n into S_i^{opt} and make sure that the Expected Z2Z Delivery Delay of the selected node is minimum. However, in this case, it is possible that the forwarding sequence has low R^{z2z}, R^{z2w}, or R^{w2z} or high D^{z2w} or D^{w2z}. If R^{z2z} is low, the raw data may not be successfully delivered to the sink node. If D^{z2w} is high or R^{z2w} is low, the important data may be delivered to the WiFi AP with longer delay or even may not be able to be received by the WiFi AP. Similarly, if R^{w2z} is low or D^{w2z} is high, the priority information may not be delivered to the ZigBee nodes. Therefore, it is important to take all of these metrics into consideration. Formally, we set the boundary conditions of the optimal forwarding sequence as follows:

$$\begin{aligned} &R_i^{z2z} > \epsilon, R_i^{z2w} > \gamma, D_i^{z2w} < \delta, \\ &R_i^{w2z} > \eta, D_i^{w2z} < \sigma \end{aligned} \tag{10}$$

To find the optimal forwarding sequence, for a ZigBee node i, the initial optimal forwarding sequence is set to $S_i^{opt} = \emptyset$. Normally, a ZigBee sender in low-duty-cycle networks does not have infinite retransmission times. Therefore, we set the maximum retransmission time as T_r. Then, starting from the initial optimal sequence, we select nodes from S_i^n in the order of wake-up time under the restriction of T_r. For example, if the node z_k in S_i^n has been selected into S_i^{opt}, the next node should be selected from $\{z_{k+1}, \ldots, z_n\}$ and the wake-up time of z_n should be smaller than T_r. To determine whether a node should be included into the optimal forwarding sequence, we introduce the following inclusion conditions:

- If the inclusion of a node z_j decreases D_i^{z2z}, we select this node into the optimal forwarding sequence and try the next node.
- If the inclusion of the next node in S_i^n does not decrease D_i^{z2z}, we discard this node and try the next node.

Formally, to decide whether to include ZigBee node r in S_i^n into the optimal forwarding sequence, the corresponding procedure is represented as follows:

$$S_i^{opt} = \begin{cases} S_i^{opt} \cup z_r, & D_i^{z2z}\left(S_i^{opt} \cup z_r\right) < D_i^{z2z}\left(S_i^{opt}\right) \\ S_i^{opt}, & Otherwise \end{cases} \tag{11}$$

To get the optimal solution, we need to try every node in S_i^n as the first node in S_i^{opt}. Specifically, for the forwarding sequence $S_i^n = \{z_j, \ldots, z_n\}$, we need to try the sequence from $\{z_j, \ldots, z_n\}$ to

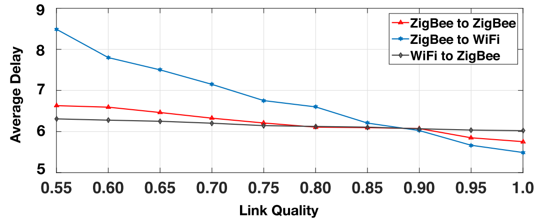


Fig. 9. Average delay for ZigBee nodes to receive priority information vs. link quality.

$\{z_n\}$. After getting all the potential forwarding sequences, the optimal forwarding sequence is the sequence that has minimal D_i^{z2z} and satisfies the boundary conditions in Equation (10).

3.3.1 Optimization Issue. As mentioned previously, in ideal settings, each ZigBee node needs to apply six boundary conditions to find the optimal forwarding sequence. However, in real-world settings, it is difficult for a ZigBee node to find the optimal forwarding sequence under these six conditions. In other words, we need to carefully set ϵ, γ, δ, η, and σ to make the optimization process easier for a ZigBee node. This procedure is time consuming. In addition, even if we carefully set the values of boundary conditions, it is still possible that few ZigBee nodes cannot find the forwarding sequence. This is because the number of potential forwarding sequences is limited while there are too many boundary conditions. Therefore, we need to reduce the number of boundary conditions to make sure each ZigBee node is able to find the optimal forwarding sequence.

Based on our transmission scheme, a ZigBee node requires the priority information to forward the raw data based on its sender's priority. Without the priority information, a ZigBee node has to forward the raw data based on the first-in first-out principle. Therefore, to find out which boundary conditions can be ignored, it is essential to know the impacts of the link qualities on the average delivery delay for the priority information to be received by a ZigBee node.

We conduct a series of simulations by changing the link qualities of ZigBee to ZigBee, ZigBee to WiFi, and WiFi to ZigBee from 55% to 100%. In this simulation, 100 ZigBee nodes with 10% duty cycles were randomly deployed in a 100×100 m square field, and 25 WiFi APs were uniformly deployed in this area. The server was positioned in the center of the field. The simulation was repeated 100 times with different random seeds, and the average value is reported in Figure 9. Counterintuitively, the link quality of Z2W has the maximum influence. With the increasing of Z2W link quality, the delay drops from 8.5 to 5.5 time units. As Z2Z link quality increases, the delay drops from 6.5 to 5.7 time units. In contrast, the W2Z link quality only has minimum influences on the delay. The delay drops from 6.3 to 6 time units as W2Z link quality increases. Therefore, we can conclude that the boundary conditions of the W2Z link can be ignored if a ZigBee node cannot find the optimal forwarding sequence.

The final algorithm is shown in Algorithm 1. To find the optimal forwarding sequence, we need to try every node in the forwarding sequence S_i^n as the first node (Line 4 to Line 11). Then, we apply six boundary conditions to find the potential forwarding sequences (Line 13 to Line 18). From Line 20 to Line 27, we apply only four boundary conditions to make sure a ZigBee node can find the optimal forwarding sequence.

3.4 Priority Map Dissemination

After the optimization, each ZigBee node concurrently transmits the important data and raw data to the WiFi AP and ZigBee nodes in its optimized forwarding sequence. Then, the WiFi AP forwards the important data to the server. According to the important data, the server decides the

ALGORITHM 1: Forwarding sequence optimization for ZigBee node i

Require: The forwarding sequence S_i^n under the restriction of T_r
Require: Boundary conditions: $\epsilon, \gamma, \delta, \eta, \sigma$
Ensure: The initial optimal forwarding sequence S_i^{opt}

$S_i^{opt}(0) \leftarrow \emptyset$
$D_i^{z2z}(S_i^{opt}(0)) \leftarrow \infty$
for $k = 1$ to n **do**
  for j from k to n **do**
    if $D_i^{z2z}(S_i^{opt}(k) \cup z_j) < D_i^{z2z}(S_i^{opt}(k-1))$ **then**
      $S_i^{opt}(k) \cup z_j$
    else
      $S_i^{opt}(k) \cup \emptyset$
    end if
  end for
end for
$l \leftarrow 0$
for m = 1 to n **do**
  if $R_i^{z2z}(S_i^{opt}(m)) > \epsilon$ and $R_i^{z2w}(S_i^{opt}(m)) > \gamma$ and $D_i^{z2w}(S_i^{opt}(m)) < \delta$ and $R_i^{w2z}(S_i^{opt}(m)) > \eta$ and $D_i^{w2z}(S_i^{opt}(m)) < \sigma$ **then**
    $l \leftarrow l + 1$
    $S_i^{opt}(l) \leftarrow S_i^{opt}(m)$
  end if
end for
$S_i^{opt} \leftarrow Min(D_i^{z2z}(S_i^{opt}(l)))$
if l == 0 **then**
  $a \leftarrow 0$
  for b = 1 to n **do**
    if $R_i^{z2z}(S_i^{opt}(m)) > \epsilon$ and $R_i^{z2w}(S_i^{opt}(m)) > \gamma$ and $D_i^{z2w}(S_i^{opt}(m)) < \delta$ **then**
      $a \leftarrow a + 1$
      $S_i^{opt}(a) \leftarrow S_i^{opt}(b)$
    end if
  end for
  $S_i^{opt} \leftarrow Min(D_i^{z2z}(S_i^{opt}(l)))$
end if

priorities of the ZigBee nodes and then transmits the priority information back to the WiFi AP. Finally, the WiFi AP informs the ZigBee nodes to transmit raw data based on their senders' priorities.

However, the radio links in the network are unreliable and the transmission status between ZigBee to ZigBee and ZigBee to WiFi are independent, which means that the WiFi AP has no knowledge of the distribution of raw data in the network. For example, as shown in Figure 10(a), the important data is successfully delivered to the WiFi AP from z_1, whereas the transmission of raw data is failed. Then, according to the transmission scheme mentioned in Section 3.2.1, the raw data needs to be retransmitted to z_3. In this case, even if the WiFi AP has the priority information, it is difficult to decide which ZigBee node to inform. Similarly, as shown in Figure 10(b), if the important data is not successfully delivered to the WiFi AP by z_1, z_2 needs to retransmit the important data to the WiFi AP. However, since z_2 does not know the transmission status of the important data from z_1, it is difficult for z_2 to decide whether to start the retransmission process.

Intuitively, a ZigBee sender can transmit redundant information to inform the WiFi AP whether the raw data is successfully delivered, and the WiFi AP can inform the ZigBee receiver whether it

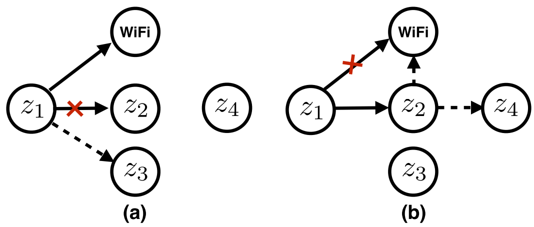


Fig. 10. The WiFi AP has no knowledge of whether the raw data is successfully delivered, whereas the ZigBee receiver has no knowledge of whether the important data is successfully delivered.

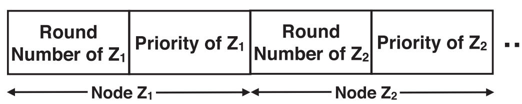


Fig. 11. The format of priority map.

has successfully received the important data. However, this solution introduces a huge overhead and significant delay to the network.

To efficiently update the transmission status and priority information at the same time, we propose a priority map, which is shown in Figure 11.

The priority map contains the current round number and the priority information of each ZigBee node. The round number and the priority information are mapped to the predefined columns in the priority map. For example, as shown in Figure 11, z_1 is mapped to the first two columns. The first column contains the current round number and the second column contains the corresponding priority of z_1.

For a ZigBee sender, the initial round number is set to 0, and it adds 1 when it generates and transmits raw data and important data to the network. For a ZigBee receiver, it receives the priority map during its receiver state. Then, according to the original sender of the raw data in its buffer, the ZigBee receiver checks the corresponding columns in the priority map and compares the round number of important data to the round number of raw data. There are three possible outcomes:

(1) *The round number of important data and raw data is identical*: This means that the important data was successfully delivered to the WiFi AP. Then, the ZigBee receiver checks the corresponding priority information and forwards the raw data based on its sender's priority.
(2) *The round number of important data is smaller*: This means that the important data was not successfully delivered to the WiFi AP. Then, the ZigBee receiver retransmits the corresponding important data in the sender state and forwards the raw data based on its sender's original priority.
(3) *The round number of important data is larger*: This means that the sender has generated and successfully delivered new important data to the WiFi AP. In this case, the server determines the sender's priority based on the new important data and the ZigBee receiver forwards the original raw data based on the updated priority information in the priority map.

3.5 Concurrent Data Forwarding

Based on the priority map, the ZigBee node can decide which data to forward. As shown in Figure 12, z_1 successfully forwards the important data and raw data to the WiFi AP and its

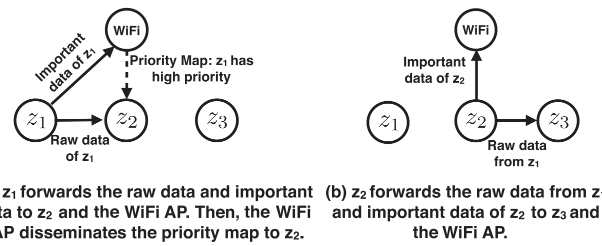


Fig. 12. An example of concurrent data forwarding.

neighboring ZigBee node z_2 during z_2's receiver state 2. Then, the WiFi AP disseminates the priority map during the receiver state 1 of z_2. Since z_1 has higher priority, z_2 forwards the raw data from z_1 during z_3's receiver state 2. Meanwhile, z_2 leverages the traffic between z_2 and z_3 to concurrently forward the important data of z_2 to the WiFi AP for priority determination. By leveraging this scheme, the raw data from z_1 can reach the server with lower delay.

However, it is possible that the transmission of the important data has finished while the raw data is still in transmission. Simply transmitting all the raw data through ZigBee to ZigBee communication will waste the link between ZigBee and WiFi. To further reduce the delivery delay for raw data from high-priority ZigBee nodes, we introduce a concurrent data forwarding scheme to forward the raw data through the WiFi AP. The key idea of this forwarding scheme is that during the sender state of each ZigBee node, if the transmission time of the important data is smaller than the transmission time of raw data, the ZigBee nodes can transmit the raw data to the WiFi AP. Specifically, in this section, we introduce (1) how to determine the size of raw data than can be transmitted through the ZigBee to WiFi link and (2) how to allocate which packet can be transmitted to the WiFi.

3.5.1 Raw Data Size Determination. To determine the size of raw data that can be transmitted, we introduce the following allocation process. Formally, the bit rates between ZigBee node i to ZigBee node j and ZigBee node i to WiFi AP y are denoted as b_{ij} and b_{iy}. Assuming that the overall packet number of raw data is n_r and the corresponding size of each packet is s_r, the overall transmission time between ZigBee and ZigBee can be represented as follows:

$$t_{ij}^{z2z} = \frac{\sum_{k=1}^{n_r} s_r^k}{b_{ij}} \tag{12}$$

Assuming that the important data packet size is s_m and the number of important data packets is n_m, the transmission time of important data between ZigBee and WiFi communication can be represented as follows:

$$t_{iy}^{z2w} = \frac{\sum_{k=1}^{n_m} s_m^k}{b_{iy}} \tag{13}$$

Then, the time difference between t_{ij}^{z2z} and t_{iy}^{z2w} can be represented as follows:

$$t_{ij}^{d} = t_{ij}^{z2z} - t_{iy}^{z2w} \tag{14}$$

If t_{ij}^d is larger than 0, it means that the transmission of important data has finished while the raw data is still in transmission. In this case, ZigBee forwarder i can transmit the raw data through

(a) The raw data from z_1 has high priority. The raw data of z_2 may have low priority.

(b) z_2 forwards the important data of z_2 to the WiFi AP and concurrently forward raw data from z_1 and z_2 to z_3.

Fig. 13. An example of the delay introduce by the easy solution.

the ZigBee to WiFi link. Therefore, the maximum size for the raw data can be transmitted to the WiFi AP is shown next:

$$S_i^{raw} = \frac{t_{ij}^d}{b_{iy}} \tag{15}$$

3.5.2 Packet Allocation. Based on the maximum size of the raw data in Equation (15), the ZigBee forwarder can allocate the raw data transmissions between ZigBee to ZigBee and ZigBee to WiFi links. Intuitively, a ZigBee forwarder should first transmit raw data packets to the WiFi based on their senders' priorities. However, this easy solution may introduce the delay for the raw data from the high-priority node to the server. For example, as shown in Figure 13(a), the WiFi AP disseminates the priority map at z_2's receiver state 1. According to the easy solution, during z_3's receiver state 2, z_2 should transmit the important data packet of z_2 and the raw data packets from z_1 to the WiFi AP. However, if the maximum size for the raw data that can be transmitted to the WiFi AP S_i^r is smaller than any raw data packets from z_1, there is no way for raw data to be delivered to the WiFi AP. In this case, z_2 either has to wait until it has enough traffic to transmit the raw data to the WiFi or it can simply transmit the raw data from z_1 to z_3 and hope that z_3 has enough traffic to transmit the raw data from z_1 to the WiFi AP. Both of these solutions will increase the delivery delay for the raw data to the server.

To increase the size of raw data that can be transmitted to the WiFi AP, we introduce the following transmission scheme. If the priority map indicates that the raw data from a typical sender has high priority, the ZigBee forwarder should first forward the raw data and then forward the important data generated by itself. In this case, the maximum size for the raw data to be transmitted to the WiFi AP can be represented as follows:

$$S_i^{raw}(high) = \frac{t_{ij}^{z2z}}{b_{iy}} \tag{16}$$

In this case, the actual size of raw data that can be transmitted to the WiFi AP is increased, which may reduce the delay for the raw data from the high-priority sender. For data with the same priorities, the ZigBee forwarder selects the data that has been forwarded by a greater number of hops. If the raw data packet from the highest-priority sender still cannot transmitted to the WiFi AP, the ZigBee node will select the raw data packets from lower-priority senders. If there are no raw data packets from high-priority senders that can be transmitted through the ZigBee to WiFi link, the ZigBee node will forward the important data to the WiFi AP. The detailed transmission scheme is shown in Algorithm 2 to clearly describe our forwarding scheme.

As mentioned previously, if there is no raw data packet from the high-priority ZigBee node, the ZigBee forwarder first transmits its own important data to the WiFi AP (Line 6 to Line 10). Then, it will check if there exists any row data packet that can be transmitted through the ZigBee to WiFi link (Line 11 to Line 16). If the priority map indicates that there are row data packets from

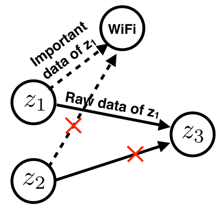
ALGORITHM 2: Concurrent data forwarding scheme for ZigBee node i

Require: The optimal forwarding sequence S_i^{opt} and priority map
Require: The important data packet size: $S_i^{important}$
Require: The maximum size: S_i^{raw} and $S_i^{raw}(high)$
Ensure: The forwarding packets set Ω
1: Select the raw data packets from high-priority senders into Ψ and count the number of these packets as n
2: Sort the raw data packet in Ψ from the highest-priority ρ_{high} to the lowest-priority ρ_{low}.
3: Calculate the size S_Ψ^r(n) of each raw data packet in Ψ
4: For the raw data packet with the same priority, sort the packet according to the packet size from high to low.
5: $\Omega \leftarrow \emptyset$
6: **if** All the senders in Ψ have lowest priority **then**
7:     **if** $S_i^{important} < S_i^{raw}$ **then**
8:         $Omega \leftarrow$ the important data packet of z_i
9:         $S_i^{raw} \leftarrow S_i^{raw} - S_\Psi^{important}$
10:     **end if**
11:     **for** $k = 1$ to n **do**
12:         **if** $S_\Psi^r(k) < S_i^{raw}$ **then**
13:             $\Omega \leftarrow S_\Psi^r(k)$
14:             $S_i^{raw} \leftarrow S_i^{raw} - S_\Psi^r(k)$
15:         **end if**
16:     **end for**
17: **else**
18:     **for** $m = 1$ to n **do**
19:         **if** $\rho(S_\Psi^r(m)) \neq \rho_{low}$ **then**
20:             **if** $S_\Psi^r(m) < S_i^{raw}(high)$ **then**
21:                 $\Omega \leftarrow S_\Psi^r(m)$
22:                 $S_i^{raw}(high) \leftarrow S_i^{raw}(high) - S_\Psi^r(m)$
23:             **end if**
24:         **else**
25:             break
26:         **end if**
27:     **end for**
28:     **if** $S_i^{important} < S_i^{raw}(high)$ **then**
29:         $Omega \leftarrow$ the important data packet of z_i
30:     **end if**
31: **end if**

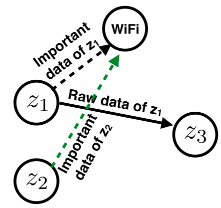
high-priority ZigBee nodes, the ZigBee forwarder will first transmit the row data packets (Line 18 to Line 26). Then, it will check if there are still some spaces for its own data to be transmitted to the WiFi AP.

4 DESIGN ISSUES AND OPTIMIZATION

In this section, we introduce several design issues. First, we describe the optimization of sharing working schedules and link qualities with neighboring ZigBee nodes and the WiFi AP. Then, we talk about issue of simultaneous transmissions. Last, we talk about how to deal with the interference from WiFi traffic.



(a) z_1 forwards the raw data and important data to z_3 and the WiFi AP. z_2 cannot transmit data to ZigBee and WiFi.



(b) z_1 forwards the raw data and important data to z_3 and the WiFi AP. z_2 transmits important data to WiFi on a different channel.

Fig. 14. An example of simultaneous transmissions.

4.1 Sharing Working Schedules and Link Qualities With Neighboring ZigBee Nodes and the WiFi AP

As introduced in Section 2, during the maintenance state of the ZigBee node, it will share its working schedule and link quality information with its neighboring ZigBee nodes and the WiFi AP. In low-duty-cycle networks, traditional technologies require a ZigBee node to transmit redundant information for working schedule and link quality information updating. This process normally is time consuming and inefficient. In this section, we introduce WiFi-based information updating techniques for the ZigBee node to share its working schedule and link quality information with its neighboring ZigBee nodes and the WiFi AP.

The basic idea is that each ZigBee node can transmit the new working schedule and link quality information to the WiFi AP. Then, the WiFi AP will broadcast the new information during the receiver state 1 of each ZigBee node. As shown in Figure 2, the WiFi AP has four overlapped channels with the ZigBee nodes. Therefore, the WiFi AP can support up to four ZigBee nodes that work on different channels to simultaneously update their working schedules and link quality information. Formally, we assign each ZigBee node a maintenance tag $\vartheta = (D, C)$, where D is the group ID and C is the channel number. Each group number is assigned to up to four ZigBee nodes with different channel numbers. The group number is assigned with a predefined updating time and period. Then, the ZigBee node with the same group number can communicate with WiFi. Each ZigBee node communicates with the WiFi AP on the predefined channel C. Therefore, four ZigBee nodes can simultaneously update their working schedules and link qualities to the WiFi AP. We are aware that it is possible that the duty cycles of the ZigBee nodes are dynamically changing. Instead of informing the duty cycle to its neighboring nodes and WiFi directly, the node with a changed duty cycle should transmit the new duty cycle information to the WiFi AP when forwarding the important data and raw data.

On receiving the working schedule and link quality information, the WiFi AP will then broadcast this information to the ZigBee nodes. This procedure does not require each ZigBee node to stay in the maintenance state. The WiFi AP also can broadcast this information during the receiver state 1 of each ZigBee node. The advantage of this method is that the ZigBee does not need to wait for the updated working schedules and link qualities from its neighboring nodes. The ZigBee nodes can switch back to the dormant state to save energy. During the receiver state, the ZigBee node can receive the update information, which is more efficient than with traditional techniques.

4.2 Simultaneous Transmissions

In previous sections, we introduced our design under the assumption that no multiple senders simultaneously transmit packets to the ZigBee receiver and the WiFi AP. Simply using CSMA may waste the link between ZigBee and WiFi. As shown in Figure 14(a), z_1 is forwarding raw data and important data to ZigBee node z_3 and the WiFi AP. In this case, z_2 has to wait until the transmission of z_1 has been finished. However, even if ZigBee node z_3 cannot receive raw data from z_2, the WiFi AP is still able to receive important data from z_2 on other channels. Therefore, in our design, if a ZigBee node finds that the channel is occupied, it will switch its channel according to the predefined channel information in $\vartheta = (D, C)$, which is shown in Figure 14(b). If the current channel is the same as its predefined channel information C, it can randomly select the other two channels to transmit important data to the WiFi AP. The detailed procedure is shown in Algorithm 3.

If more than two ZigBee nodes simultaneously transmit packets to the same ZigBee receiver, this approach may introduce collision at the WiFi AP. However, in low-duty-cycle networks, this situation happens rarely, which makes our approach sufficient to solve this problem.

ALGORITHM 3: Simultaneous transmissions

Require: $\vartheta = (D, C)$
Require: Current transmission channel c_1
Require: Four overlapped channels: c_1, c_2, c_3, and c_4
Ensure: The Z2W transmission channel

```
if c1 ≠ C then
    Switch the Z2W transmission channel to C
else if C = c2 then
    Randomly switch the Z2W transmission channel between c3 and c4
else if C = c3 then
    Randomly switch the Z2W transmission channel between c2 and c4
else
    Randomly switch the Z2W transmission channel between c2 and c3
end if
```

4.3 Interference from WiFi Traffic

In a cross-technology communication network, the interference from the WiFi traffic may affect the performance of the ZigBee network. Specifically, since the ZigBee to WiFi transmission is the packet-level cross-technology communication, the ZigBee node needs to transmit multiple packets to the ZigBee receiver to concurrently transmit packets to the WiFi AP. If other WiFi devices are also transmitting packets, the communication from ZigBee to the WiFi AP will be hampered. The CSMA protocol is limited in such scenarios because it cannot explicitly coordinate the WiFi and ZigBee devices at the same time, which will limit network performance.

In our design, the WiFi AP will broadcast a ZigBee transmission (ZT) request to the network before the ZigBee sender switches to the active state. The ZT request contains the active time and duration information ($active\_time, duration$). After receiving the ZT request, the WiFi devices will back off based on the duration information in the ZT request. We aware that this mandatory backoff request may affect the performance of the WiFi network. Fortunately, due the the low duty cycle of the ZigBee nodes, the backoff time only introduces negligible influences on the WiFi throughput, which is evaluated in Section 5.6.

4.4 Uncovered ZigBee Nodes

We aware that it is possible that some of the ZigBee nodes may not be in the coverage range of the WiFi AP in the real-world settings. In this case, the important data and raw data generated by these ZigBee nodes have to be transmitted to their neighboring nodes until it is transmitted to the WiFi AP. We also need to mention that when these ZigBee nodes are transmitting a high volume of data (based on EEI, AEIC, and UTC), the congestion introduced to the network will be extremely high, which significantly reduces network performance. Therefore, when facing congestion, the

Table 2. Priority Assignment Table

Priority 1	**Urgent:** High energy consumption currently
Priority 2	**Highly Important:** Low energy consumption currently
Priority 3	**Moderately Important:** May consume high energy in the near future
Priority 4	**Less Important:** Very low energy consumption

ZigBee nodes that are not in the WiFi AP's coverage range will first transmit the important data to their neighboring nodes.

To find the optimized forwarding sequence for these ZigBee nodes, we can still leverage the forwarding sequence optimization Algorithm 1. Specifically, the uncovered ZigBee nodes should set the metrics $R_i^{z2w}, R_i^{w2z}, D_i^{z2w}, D_i^{w2z}$ of its neighboring nodes to infinity. By using this approach, the forwarding sequence of uncovered ZigBee nodes can be computed.

5 EVALUATION

In this section, we extensively evaluate our design under various network settings. Since this work is the first one investigating concurrent cross-technology transmission, the current state of the art is complementary; however, it provides no appropriate baselines for comparison. DSF [14] is one of the famous data forwarding techniques in low-duty-cycle networks, and it is typically designed for ZigBee nodes. To show the advantages of our design, we use DSF with predefined priorities as our baseline.

5.1 Experimental Setup

In this experiment, 25 ZigBee-compliant TelosB nodes are randomly deployed in the hallway of our building (45×20 m, maximum number of hops is 10). The duty cycle of each ZigBee node is set to 10%, where τ is described in Section 2 and set to be 20 ms. Although the current commodity device can achieve WiFi to ZigBee communication by using a physical-layer emulation technique [29], it requires a high-granularity RSS information value to achieve ZigBee to WiFi communication. However, current off-the-shelf WiFi devices do not provide API to get such information. Therefore, we deploy the WiFi-compliant USRP B210 in the center to cover the whole field. The implementation of the USRP B210 strictly follows the IEEE 802.11n standard. We use our computer, the MSI GE62 6QC, as the server. The USRP uses the most popular WiFi channel 1, and the transmission power is 25dBm. The ZigBee channel is set to channel 12, which is overlapped with WiFi channel 1. The ZigBee devices sample the received signal strength and establish the communication by applying beacon folding.

In our experiment, we use the power consumption data collected from 726 smart homes for more than 1 year. We use the voltage data and frequency data as the important data because it reflects the current electricity and power transmission levels, which can be directly used for smart grid management and demand forecast [19, 20, 36]. Each ZigBee node has four possible priorities. The details are shown in Table 2.

The Urgent priority means that the current energy consumption level and the demand in a smart home is high. The Less Important priority means that a smart home does not have an urgent power consumption request. In contrast, the nodes in DSF are randomly set with predefined priorities.

5.2 Impact of Different Duty Cycles

We first evaluate the performance of ECT in terms of duty cycle. During this experiment, over 5,000 packets are transmitted from different ZigBee nodes to the server. As shown in Figure 15, the performance of ECT is much better than DSF. When the duty cycle increases to 10%, the average

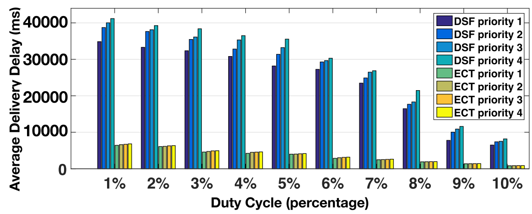


Fig. 15. Average delivery delay from ZigBee nodes to the server vs. duty cycle.

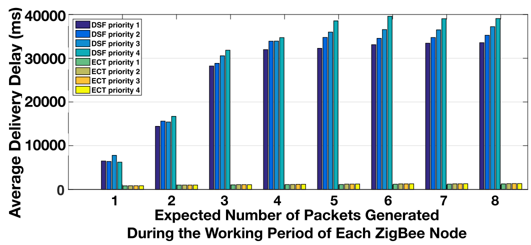


Fig. 16. Average delivery delay from ZigBee nodes to the server vs. traffic volume.

delivery delays of ECT from priority 1, priority 2, priority 3, and priority 4 nodes are 7.9, 8.9, 8.6, and 9.5 times lower than the delay of DSF, respectively. Even in the worse cases (duty cycle 1%, priority 1), the average delivery delay of ECT is still 5.4 times lower than the delay of DSF. We can also observe that the average delivery delay of ECT drops much faster than DSF with the increasing of duty cycle. This is because ZigBee nodes have more chances to communicate with the WiFi AP as duty cycle increases, which increases the percentage of raw data transmitted through the WiFi AP.

5.3 Impact of Different Network Traffic Volumes

Figure 16 evaluates the performance of ECT under different network traffic volumes. During this experiment, we ran our network for more than 2×10^5 ms. When the expected number of packets generated during each node's working period is 1, the average delivery delays of the raw data for DSF from priority 1, priority 2, priority 3, and priority 4 nodes are 6,240 ms, 6,500 ms, 6,400 ms, and 7,780 ms, respectively, whereas the corresponding average delivery delays of ECT are only 820 ms, 840 ms, 840 ms, and 840 ms, respectively. As the expected number of generated packets increases to 8, the average delivery delays of DSF are increased by more than 5 times, whereas the corresponding average delivery delays of ECT are only increased to 1,260 ms, 1,320 ms, 1,340 ms, and 1,350 ms, which is more than 29.1 times lower than DSF for the best case (priority 4). We observe that the delay of ECT increases much slower than DSF. This is because the DSF may face congestion as the traffic volume increases, whereas ECT is still able to transmit packets to the server through the WiFi APs.

5.4 Impact of Dynamic Working Schedules

We study the performance of ECT when the working schedules of ZigBee nodes are dynamically changing (varies from 1% to 10%) based on their current energy level. As shown in Figure 17, the delay of ECT is still much lower than DSF. Specifically, the average delay of ECT is around 5.9 times lower than DSF under different priorities. This is because that the working schedule updating process of ECT is much more efficient than DSF (up to 4 devices can update their working schedule to the WiFi device at the same time). In contrast, DSF has to run the node discovery process to conduct forwarding, which degrades network performance.

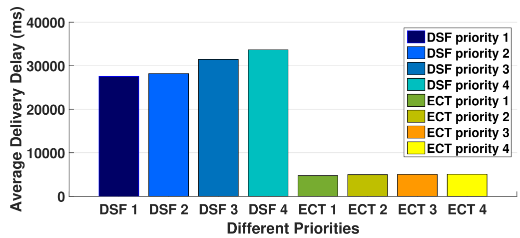


Fig. 17. Average delivery delay from ZigBee nodes to the server vs. dynamic duty cycles.

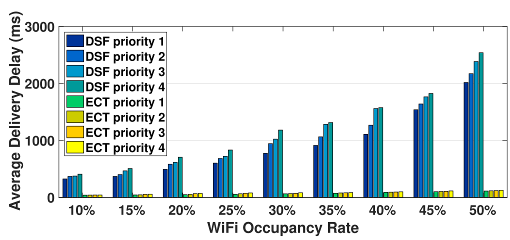


Fig. 18. Average delivery delay from ZigBee nodes to the server vs. WiFi occupancy rate.

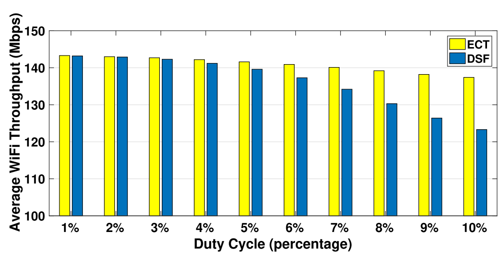


Fig. 19. Average WiFi throughput (Mbps) vs. WiFi occupancy rate.

5.5 Impact of WiFi Traffic

We show the performance under different WiFi occupancy rates when the duty cycles of the ZigBee nodes are set to 10%. As shown in Figure 18, both ECT and DSF are suffering high interference when the traffic volume is high. However, the delay of ECT increases more slowly than in ECT. This is because the WiFi APs in ECT will broadcast a short ZT request before receiving the transmissions from the ZigBee nodes. By using this approach, the important data (voltage and frequency) and raw data are transmitted with much lower CTI. However, DSF only leverages CSMA to schedule the transmissions, which limits network performance. As a result, ECT is able to reduce the delivery delay by 20 times over DSF when the WiFi traffic percentage is as high as 50%.

5.6 Impact on WiFi Throughput

We study the impact on the WiFi throughput for ECT and DSF. As shown in Figure 19, the throughput of WiFi degrades as the duty cycles of the ZigBee nodes increase. However, the performance of WiFi under the design of ECT is slightly better than DSF. This is because the WiFi devices for DSF suffer interference from the ZigBee network. In contrast, ECT leverages the WiFi to conduct transmission, which significantly reduces the traffic in the ZigBee network. As the duty cycle increases to 10%, the throughput of WiFi for ECT is 137.4Mbps, whereas the throughput of WiFi for DSF is 123.3Mbps.

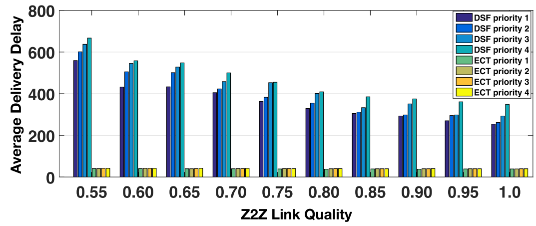


Fig. 20. Average delivery delay from ZigBee nodes to the server vs. Z2Z link quality.

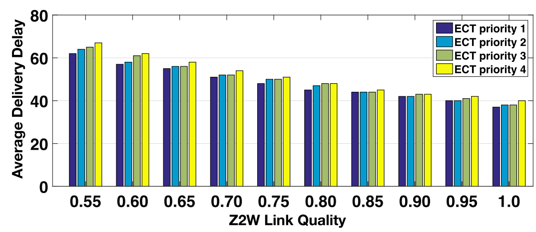


Fig. 21. Average delivery delay from ZigBee nodes to the server vs. W2Z link quality.

6 SIMULATION

We simulate our system to study average delivery delay under the different Z2Z, Z2W, and W2Z link qualities. In this experiment, 100 ZigBee nodes with 10% duty cycles were randomly deployed in a 100×100 m square field, and 25 WiFi APs were uniformly deployed to cover the whole area. Each experiment was repeated 20 times with different random seeds, ZigBee node deployments, and working schedules. The server was positioned in the center of the deployment field. The physical layer of ZigBee and the WiFi AP was strictly implemented according to the experimental setup. Data collected at the server was obtained by averaging 10,000 ZigBee node to server communications.

6.1 Impact of Different Link Qualities

As shown in Figure 20, we can observe that the DSF suffer extremely high average delivery delay under low Z2Z link qualities, whereas the delivery delay of ECT just slightly increased by a few time units. This is because as Z2Z link quality decreases, the ZigBee nodes have to retransmit the raw data multiple times to deliver it to the next hop. In this case, the ZigBee nodes have more opportunities to communicate with the WiFi AP and forward the data through Z2W links. Therefore, even if Z2Z link qualities are low, the average delivery delay is still much better than DSF.

Figure 21 studies the delivery delay under different Z2W link qualities. As we can see from this figure, even if Z2W link quality is low, for each priority, the average delivery delay is still as low as 60 time units. As the Z2W link quality increases from 0.55 to 1.0, the average delivery delay is lower than 40 time units.

Interestingly, as shown in Figure 22, the average delay is just slightly decreased as the W2Z link quality increases. This is because the W2Z link only transmits the priority map, whereas the Z2W link not only transmits important data but also transmits raw data. This observation supports our observation in Figure 9. Therefore, we can conclude that the link qualities have different influences on the average delivery delay for the raw data to the server.

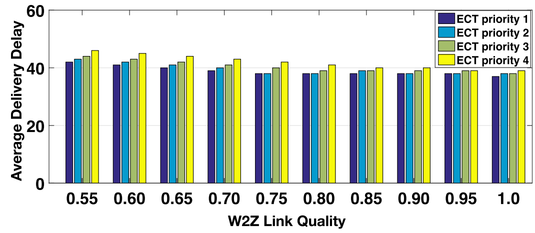


Fig. 22. Average delivery delay from ZigBee nodes to the server vs. Z2Z link quality.

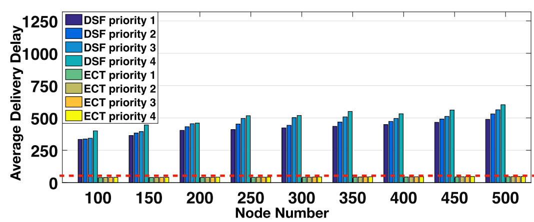


Fig. 23. Average delivery delay from ZigBee nodes to the server vs. network size.

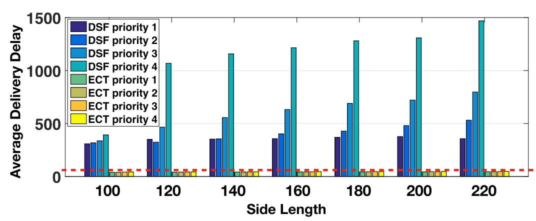


Fig. 24. Average delivery delay from ZigBee nodes to the server vs. node density.

6.2 Impact of Different Network Sizes and Densities

We also simulate the performance of our design and DSF under different network sizes and densities with the same simulation setup mentioned in Section 6.1. For different network sizes, the ZigBee node number varies from 100 to 500. To keep the density similar, the side length of the area changes from 100 to 223 m. In addition, the WiFi AP number varies from 25 to 75 to cover the whole field. As shown in Figure 23, the average delivery delay of DSF increases, whereas the performance of ECT almost remains the same. This is because as the network size increases, the raw data is forwarded to the server with more hops, which also increase the opportunities for the raw data to be transmitted to the WiFi APs.

For different network densities, the ZigBee node number is set to 100 while the side length changes from 100 to 220 m. The number of the WiFi AP also varies from 25 to 75 to cover the whole area. With the decreasing of network densities, the number of potential forwarding nodes for each ZigBee node is decreasing and the distance between each ZigBee node is increasing, which introduces lower Z2Z link quality and a higher number of retransmissions. However, as shown in Figure 24, the performance of ECT only slightly decreases. This is because as the number of Z2Z retransmissions increases, they also provide more chances for ZigBee nodes to concurrently forward raw data to the WiFi APs. In addition, as shown in Section 6.1, the Z2Z link quality does not have dominant influences on the average delivery delay. Therefore, the average delay is still much lower than DSF.

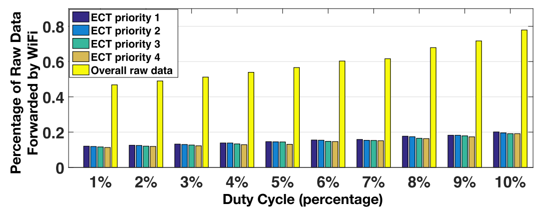


Fig. 25. Percentage of raw data vs. duty cycle.

6.3 Insights

To understand why ECT provides better performance than the state-of-the-art solutions, we analyze the percentage of raw data forwarded by WiFi under different duty cycles and the impact of different network sizes and densities.

As shown in Figure 25, with the increasing of duty cycle, the percentage of raw data forwarded by WiFi AP also increases. This is because the ZigBee nodes have more time to concurrently transmit raw data through the link from ZigBee to WiFi AP. This observation indicates the reason why the performance of ECT is much better than the state-of-the-art solution.

7 RELATED WORK

The originality of our work lies in the intersection of multiple novel researches for cross-technology communication and wireless sensor networks.

Cross-technology communication. Due to the exponentially increasing number of IoT devices and their heterogeneous wireless communication technologies, researchers have introduced several techniques to enable ZigBee, WiFi, and Bluetooth communications. By sensing and interpreting energy profiles, Esense can enable the communication between WiFi and ZigBee devices [4]. One of the most famous works is FreeBee, which enables the communication among ZigBee, Bluetooth, and WiFi devices by using shifted beacons [27]. The FreeBee is able to support mobile sensor nodes, and the maximum speed is up to 30mph. As introduced in HoWiES [48], the author enables WiFi to ZigBee communication by sensing the packet length of WiFi packets. WEBee enables cross-technology communication via channel emulation. Specifically, WEBee enables the communication from WiFi to ZigBee. It uses a high-speed wireless radio (e.g., WiFi) to emulate the desired signals (e.g., ZigBee) by controlling the WiFi payload [29]. However, since the WiFi packet is changed, the WiFi to WiFi communication is hampered. Recently, researchers introduced different methods to enable concurrent communication among heterogeneous IoT devices. One famous work is EMF [6], which achieves concurrent communication between ZigBee and WiFi. Specifically, during the modulation process, EMF slightly shifts and flips the original packet. The receiver demodulates the packets by detecting the RSS. By doing this, EMF embeds different pieces of information in the existing traffic (i.e., ZigBee to ZigBee or WiFi to WiFi) and concurrently transmits this information from one sender to multiple receivers with different communication technologies. Another example is B^2W^2 [7], which achieves N-way concurrent communication between Bluetooth and WiFi. PMC and Chiron enable multiprotocol high-speed communication to WiFi and ZigBee at the cost of requiring specific design hardware [8, 10].

Data forwarding techniques. Wireless sensor networks have been applied to support lots of different applications [9, 22, 39, 46]. To improve system performance, researchers have proposed different techniques [1, 11, 28] and routing protocols [35, 49, 52] from different perspectives, such as delay [5, 21], network capacity [1, 11, 47], energy efficiency [13, 15, 50], security [42, 43], and quality of service [31, 51]. Dynamic Switch-Based Forwarding optimizes the data delivery ratio, delay, and energy consumption [14, 34]. In Guo et al. [16], a forwarder selection technique and a link-quality–based backoff method are proposed to alleviate the hidden terminal problem and resolve simultaneous forwarding operations. As presented in Baysan et al. [2], a polynomial time algorithm is designed to solve the minimum forwarding set problem for a wireless network under-unit disk coverage model. Link correlation is another interesting topic that has been studied in wireless sensor networks to optimize network performance [53]. In Guo et al. [17], a correlation flooding method is introduced in low-duty-cycle networks to reduce the delivery delay. By leveraging link correlation, Chai et al. [3] significantly increase the network throughput by 55%. In Jun et al. [25], a new sender-based correlation is demonstrated and utilized to reduce the transmission energy consumption in data forwarding.

Different from the preceding methods, our work is the first cross-technology data forwarding design, which opens a new door for cross-technology network layer design. Specifically, our previous work has shown the advantages of the cross-technology communication network when forwarding data from some low-duty-cycle nodes [38]. In this article, we exploit the unique feature in cross-technology communication by concurrently transmitting raw data and important data to the server. Moreover, our design can dynamically change the priority of ZigBee nodes and significantly reduce the packet delivery delay of higher-priority nodes. To the best of our knowledge, there is no prior work on cross-technology network modeling and design.

8 CONCLUSION

In this article, we introduce ECT, which is the first cross-technology data forwarding method that leverages the cross-technology communication's unique feature (i.e., concurrent transmission of raw data and important data with a single stream of packet transmissions from a ZigBee node to a WiFi AP and another ZigBee node). ECT is able to dynamically change the priorities of ZigBee nodes and significantly reduce the delivery delay from high-priority nodes to the server. To the best of our knowledge, this is the first work that considers dynamic priorities of ZigBee nodes, unreliable links, and a low duty cycle at the same time in heterogeneous IoT networks. We extensively evaluated ECT under various network settings. The results demonstrate the advantages of our design. For example, ECT's packet delivery delay is more than 29 times lower than the state-of-art solution.

REFERENCES

[1] Apostolos Apostolaras, George Iosifidis, Kostas Chounos, Thanasis Korakis, and Leandros Tassiulas. 2016. A mechanism for mobile data offloading to wireless mesh networks. *IEEE Transactions on Wireless Communications* 15, 9, 5984–5997.

[2] Mehmet Baysan, Kamil Sarac, Ramaswamy Chandrasekaran, and Sergey Bereg. 2009. A polynomial time solution to minimum forwarding set problem in wireless networks under unit disk coverage model. *IEEE Transactions on Parallel and Distributed Systems* 20, 7 (2009), 913–924.

[3] Fangming Chai, Ting Zhu, and Kyoung-Don Kang. 2016. A link-correlation-aware cross-layer protocol for IoT devices. In *Proceedings of the IEEE International Conference on Communications (ICC'16)*. IEEE, Los Alamitos, CA, 1–6.

[4] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: Communication through energy sensing. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. ACM, New York, NY, 85–96.

[5] Long Cheng, Yu Gu, Jianwei Niu, Ting Zhu, Cong Liu, Qingquan Zhang, et al. 2016. Taming collisions for delay reduction in low-duty-cycle wireless sensor networks. In *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM'16)*. IEEE, Los Alamitos, CA, 1–9.

[6] Zicheng Chi, Zhichuan Huang, Yao Yao, Tiantian Xie, Hongyu Sun, and Ting Zhu. 2017. EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices. In *Proceedings of the 2017 IEEE International Conference on Computer Communications (INFOCOM'17)*. IEEE, Los Alamitos, CA, 1–9.

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

[8] 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 the 25th IEEE International Conference on Network Protocols (ICNP'17)*. IEEE, Los Alamitos, CA, 1–10.

[9] Zicheng Chi, Yao Yao, Tiantian Xie, Zhichuan Huang, Michael Hammond, and Ting Zhu. 2016. Harmony: Exploiting coarse-grained received signal strength from IoT devices for human activity recognition. In *Proceedings of the 24th IEEE International Conference on Network Protocols (ICNP'16)*. IEEE, Los Alamitos, CA, 1–10.

[10] Zicheng Chi, Yao Yao, Tiantian Xie, Xin Liu, Zhichuan Huang, Wei Wang, et al. 2018. EAR: Exploiting uncontrollable ambient RF signals in heterogeneous networks for gesture recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, 237–249.

[11] Florin Ciucu, Ramin Khalili, Yuming Jiang, Liu Yang, and Yong Cui. 2014. Towards a system theoretic approach to wireless network capacity in finite time and space. In *Proceedings of the 2014 IEEE International Conference on Computer Communications (INFOCOM'14)*. IEEE, Los Alamitos, CA, 2391–2399.

[12] Gartner. 2017. Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. Retrieved October 1, 2018 from http://www.gartner.com/newsroom/id/3598917.

[13] Yu Gu, Liang He, Ting Zhu, and Tian He. 2014. Achieving energy-synchronized communication in energy-harvesting wireless sensor networks. *ACM Transactions on Embedded Computing Systems* 13, 2S (2014), 68.

[14] Yu Gu and Tian He. 2011. Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks. *IEEE Transactions on Mobile Computing* 10, 12 (2011), 1741–1754.

[15] Yu Gu, Ting Zhu, and Tian He. 2009. ESC: Energy synchronized communication in sustainable sensor networks. In *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP'09)*. IEEE, Los Alamitos, CA, 52–62.

[16] Shuo Guo, Liang He, Yu Gu, Bo Jiang, and Tian He. 2014. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. *IEEE Transactions on Computers* 63, 11 (2014), 2787–2802.

[17] Shuo Guo, Song Min Kim, Ting Zhu, Yu Gu, and Tian He. 2011. Correlated flooding in low-duty-cycle wireless sensor networks. In *Proceedings of the 2011 19th IEEE International Conference on Network Protocols (ICNP'11)*. 383–392.

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

[19] Zhichuan Huang and Ting Zhu. 2016. Leveraging multi-granularity energy data for accurate energy demand forecast in smart grids. In *Proceedings of the IEEE International Conference on Big Data*. IEEE, Los Alamitos, CA, 1182–1191.

[20] Zhichuan Huang and Ting Zhu. 2016. Real-time data and energy management in microgrids. In *Proceedings of the 2016 IEEE Real-Time Systems Symposium (RTSS'16)*. IEEE, Los Alamitos, CA, 79–88.

[21] Zhichuan Huang and Ting Zhu. 2017. Distributed real-time multimodal data forwarding in unmanned aerial systems. In *Proceedings of the 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON'17)*. IEEE, Los Alamitos, CA, 1–9.

[22] Zhichuan Huang, Ting Zhu, Haoyang Lu, and Wei Gao. 2016. Accurate power quality monitoring in microgrids. In *Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'16)*. IEEE, Los Alamitos, CA, 1–6.

[23] Edison Electric Institute. 2017. A Discussion of Smart Meters And RF Exposure Issues. Retrieved February 3, 2019 from https://aeic.org/wp-content/uploads/2013/07/smartmetersandrf031511.pdf.

[24] Wenchao Jiang, Zhimeng Yin, Song Mim Kim, and Tian He. 2017. Transparent cross-technology communication over data traffic. In *Proceedings of the 2017 IEEE International Conference on Computer Communications (INFOCOM'17)*. IEEE, Los Alamitos, CA, 1–9.

[25] Junghyun Jun, Long Cheng, Liang He, Yu Gu, and Ting Zhu. 2014. Exploiting sender-based link correlation in wireless sensor networks. In *Proceedings of the 22nd IEEE International Conference on Network Protocols (ICNP'14)*. IEEE, Los Alamitos, CA, 445–455.

[26] Joohwan Kim, Xiaojun Lin, Ness B. Shroff, and Prasun Sinha. 2010. Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Transactions on Networking* 18, 2 (2010), 515–528.

[27] Song Min Kim and Tian He. 2015. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, New York, NY, 317–330.

[28] Yanhua Li and Zhi-Li Zhang. 2013. Random walks and Green's function on digraphs: A framework for estimating wireless transmission costs. *IEEE/ACM Transactions on Networking* 21, 1 (2013), 135–148.

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

[30] Chenyang Lu, Brian M. Blum, Tarek F. Abdelzaher, John A. Stankovic, and Tian He. 2002. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *Proceedings of the 2002 8th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, Los Alamitos, CA, 55–66.

[31] Aniket Malvankar, Ming Yu, and Ting Zhu. 2006. An availability-based link QoS routing for mobile ad hoc networks. In *Proceedings of the 2006 IEEE Sarnoff Symposium*. IEEE, Los Alamitos, CA, 1–4.

[32] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. 2004. The flooding time synchronization protocol. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, 39–49.

[33] S. Mini, S. K. Udgata, and S. L. Sabat. 2014. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sensors Journal* 14, 3 (2014), 636–644.

[34] Sijun Ren, Ping Yi, Ting Zhu, Yue Wu, and Jianhua Li. 2014. A 3-hop message relay algorithm for connected dominating sets in wireless ad-hoc sensor networks. In *Proceedings fo the IEEE/CIC International Conference on Communications in China (ICCC'14)*. IEEE, Los Alamitos, CA, 829–834.

[35] Hulya Seferoglu and Eytan Modiano. 2016. Separation of routing and scheduling in backpressure: Based wireless networks. *IEEE/ACM Transactions on Networking* 24, 3 (2016), 1787–1800.

[36] Yu Sui, Ping Yi, Xin Liu, Wei Wang, and Ting Zhu. 2017. Optimization for charge station placement in electric vehicles energy network. In *Proceedings of the Workshop on Smart Internet of Things*. ACM, New York, NY, 1.

[37] Cisco Systems. 2014-2019. In *Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 White Paper*. Cisco Systems.

[38] 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 the 2018 IEEE International Conference on Computer Communications (INFOCOM'18)*. IEEE, Los Alamitos, CA, 369–377.

[39] Tiantian Xie, Zhichuan Huang, Zicheng Chi, and Ting Zhu. 2017. Minimizing amortized cost of the on-demand irrigation system in smart farms. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. ACM, New York, NY, 43–46.

[40] Mohammad Hossein Yaghmaee and Donald A. Adjeroh. 2009. Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks. *Computer Networks* 53, 11 (2009), 1798–1811.

[41] Chi-Ming Yang, Kuei-Ping Shih, and Shih-Hao Chang. 2017. A priority-based energy replenishment scheme for wireless rechargeable sensor networks. In *Proceedings of the 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA'17)*. IEEE, Los Alamitos, CA, 547–552.

[42] Ping Yi, Ting Zhu, Ning Liu, Yue Wu, and Jianhua Li. 2012. Cross-layer detection for black hole attack in wireless network. *Journal of Computational Information Systems* 8, 10 (2012), 4101–4109.

[43] Ping Yi, Ting Zhu, Qingquan Zhang, Yue Wu, and Jianhua Li. 2012. Green firewall: An energy-efficient intrusion prevention mechanism in wireless sensor network. In *Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM'12)*. IEEE, Los Alamitos, CA, 3037–3042.

[44] Kasim Sinan Yildirim and Aylin Kantarci. 2014. Time synchronization based on slow-flooding in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 25, 1 (2014), 244–253.

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

[46] Qingquan Zhang, Ziqiao Zhou, Wei Xu, Jing Qi, Chenxi Guo, Ping Yi, et al. 2015. Fingerprint-free tracking with dynamic enhanced field division. In *Proceedings of the 2015 IEEE International Conference on Computer Communications (INFOCOM'15)*. IEEE, Los Alamitos, CA, 2785–2793.

[47] Shiwen Zhang, Qingquan Zhang, Sheng Xiao, Ting Zhu, Yu Gu, and Yaping Lin. 2015. Cooperative data reduction in wireless sensor network. *ACM Transactions on Embedded Computing Systems* 14, 4 (2015), 84.

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

[49] Ting Zhu and Don Towsley. 2011. E2R: Energy efficient routing for multi-hop green wireless networks. In *Proceedings of the 2011 IEEE International Conference on Computer Communications Workshops (INFOCOM WKSHPS'11)*. IEEE, Los Alamitos, CA, 265–270.

[50] Ting Zhu, Sheng Xiao, Yi Ping, Don Towsley, and Weibo Gong. 2011. A secure energy routing mechanism for sharing renewable energy in smart microgrid. In *Proceedings of the IEEE International Conference on SmartGridComm*. IEEE, Los Alamitos, CA, 143–148.

[51] Ting Zhu and Ming Yu. 2006. A dynamic secure QoS routing protocol for wireless ad hoc networks. In *Proceedings of the 2006 IEEE Sarnoff Symposium*. IEEE, Los Alamitos, CA, 1–4.

[52] Ting Zhu and Ming Yu. 2006. A secure quality of service routing protocol for wireless ad hoc networks. In *Proceedings of the 49th Annual IEEE GLOBECOM Technical Conference and IEEE Communications Expo (GLOBECOM'06)*.

[53] Ting Zhu, Ziguo Zhong, Tian He, and Zhi-Li Zhang. 2010. Exploring link correlation for efficient flooding in wireless sensor networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10)*, Vol. 10. 1–15.

Received April 2018; revised October 2018; accepted November 2018