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

Wei Wang, Tiantian Xie, Xin Liu and Ting Zhu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Email: {ax29092, xtiant1, xinliu1, zt}@umbc.edu

***Abstract*—Recent advances in cross-technology communication have significantly improved the spectrum efficiency in the same ISM 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 AP. 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.**

I. Introduction

The number of Internet-of-Thing (IoT) devices will exponentially increase and reach 20 billion by 2020 [1]. Most of these devices are working within the same Industrial Scientific Medical (ISM) Band [2]. Therefore, there's a pressing need to efficiently utilize the spectrum in the ISM band. To meet this requirement, researchers have proposed different cross-technology communication techniques [3], [4] to enable the communication among IoT devices without changing the devices' own protocols.

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 paper, 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 AP for reducing the packet delivery delay.

This is because, 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 600 ZB (i.e., 6×10^{23} bytes) by 2020, up from 145 ZB generated in 2015. However, it is estimated that only approximately 10% of the generated data (60ZB) is useful [5].

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/her eyes.

Traditional approaches either preset the sensors' priorities [7], [8] or decide the sensors' priorities according to the delay, energy consumption or locally available data [9], [10], [11]. 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 so as to decide their priorities. However, since lots of IoT devices are 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 so as to decide their senders' priorities as soon as possible.

Besides 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 [12]. Second, different from traditional low-duty-cycle networks, some IoT devices may introduce higher delay to the network. For example, according to the report [13] conducted by EEI, AEIC and UTC, the smart meter generates huge amounts of data and the duty-cycle of a smart

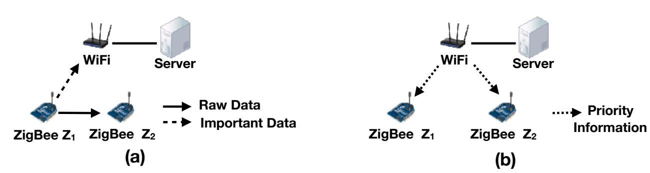


Fig. 1. An example of simplified network architecture. (a) Based on the cross-technology communication techniques [6], [4], the ZigBee node $Z1$ is able to concurrently transmit raw data and important data to the ZigBee node $Z2$ and WiFi AP using the same ZigBee packets. (b) WiFi AP disseminates the priority information to ZigBee devices.

meter is set to 1% to 5% in order to mitigate congestion to the network. As a result, such duty cycles and potential congestion introduce higher delay and thus 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], [4]. Specifically, in this paper, we introduce ECT, an innovative solution that enables a ZigBee sensor node to concurrently transmit important data and raw data to WiFi access points (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), while 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 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.

II. Models

In this section, we first introduce the cross-technology network model of the heterogeneous IoT networks that contains ZigBee nodes and WiFi devices, then we describe the working schedule model of ZigBee nodes.

A. Cross-Technology Network Model

Without loss of generality, we assume that there are N ZigBee nodes and M WiFi devices. Both ZigBee 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], [4]. Each ZigBee node has two possible states: **the receiver state** and **the dormant state**. In the receiver state, the ZigBee node can sense and receive packets from WiFi or its neighboring ZigBee nodes. In the dormant state, a ZigBee node turns off all its functions except a timer to wake up. In summary, a ZigBee node can transmit packets at any time but can only receive WiFi or ZigBee packets in the receiver state. All ZigBee nodes share their working schedules with their neighboring nodes, which can be achieved by using ENDP. Each ZigBee node is also synchronized with its neighboring nodes, which can be achieved by using FTSP [14] or FCSA [15].

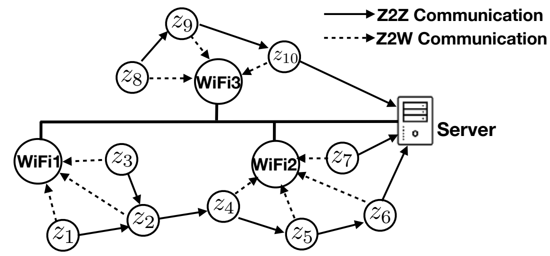


Fig. 2. An example of the IoT network

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 while $W(t)$ 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 2 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], [4]. Since WiFi APs are always active, the packets transmitted to the WiFi APs can be delivered to the server with very low latency.

B. Working Schedule Model

Since ZigBee nodes work under low-duty-cycle, 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 while

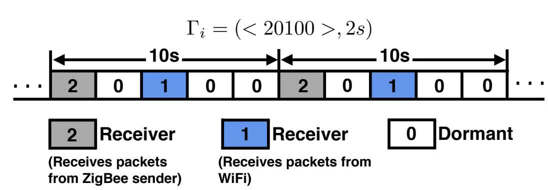
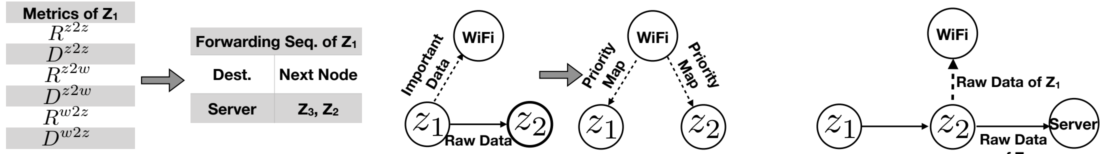
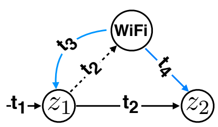


Fig. 3. The working schedule of ZigBee node i



(a) Metrics Computation and Forwarding Sequence Optimization (b) Priorities Determination (c) Concurrent Data Forwarding

Fig. 4. Design Overview



$\Gamma_1 = (< 20100 >, 2s)$
$\Gamma_1 = (< 02010 >, 2s)$

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. 5. An example of the delay introduced by the working schedule

the receiver state is denoted as 1 and 2. Specifically, 1 denotes the node i is ready to receive packets from WiFi AP and 2 denotes denotes the node i is ready to receive packets from its neighboring ZigBee nodes.

Figure 3 shows an example of the working schedule of the 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 WiFi AP. Based on this model, we can compute the delay introduced by the working schedule in this network.

As shown in Figure 5, the working schedules of ZigBee node 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 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: i) ZigBee to ZigBee **(Z2Z)**; ii) ZigBee to WiFi **(Z2W)**; and iii) WiFi to ZigBee **(W2Z)**.

III. 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: **(1) how to mathematically analyze the network with the presence of ZigBee nodes and WiFi APs**; **(2) how to reduce the delivery delay of ZigBee nodes; (3) 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. In the end, we propose a concurrent data forwarding scheme to further reduce the delivery delay from higher priority ZigBee nodes to the server.

A. Design Overview

As illustrated in Figure 4, ECT consists of three major steps:
1. Metrics Computation and Forwarding Sequence Optimization: ZigBee nodes computes 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, according to these metrics, each ZigBee node adopts a dynamic programming approach to find out its optimal forwarding sequence to reduce the delivery delay.
2. Priorities Determination: According to the optimal forwarding sequence, the ZigBee nodes concurrently transmit the important data and raw data to the WiFi APs and ZigBee receivers. Then, the WiFi APs forward 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. At last, the WiFi AP broadcasts the priority map to the ZigBee nodes during their receiver states.
3. Concurrent Data Forwarding: Each ZigBee node checks the priority map and forwards the data based on their senders' priorities. In addition, to further reduce the delivery delay from high priority ZigBee nodes to the server, the ZigBee nodes also transmit the raw data to the WiFi AP based on their senders' priorities.

B. Metrics Computation and Forwarding Sequence Optimization

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.

*1) **The Need of 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 which consists of n ZigBee nodes that can leverage ZigBee to ZigBee and ZigBee to WiFi communications to concurrently*

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 the ZigBee node i
D_i^{w2z}	Expected delivery delay for a packet transmitted from the WiFi AP and received by the ZigBee node i

TABLE I
DEFINITIONS OF METRICS

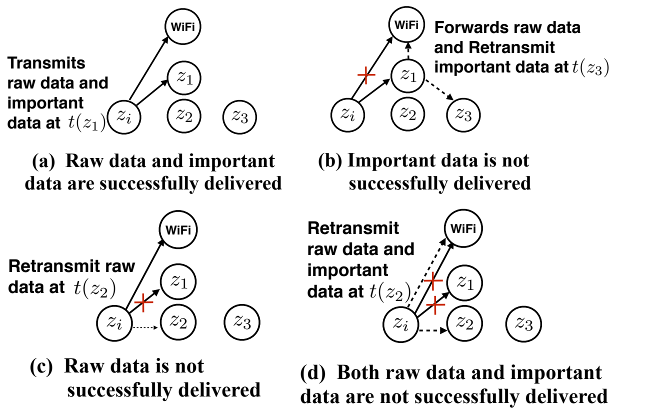


Fig. 6. An example of the transmission scheme of ZigBee node i. The forwarding sequences of z_i and z_1 are $S_i^n = (z_1, z_2)$ and $S_1^n = (z_3)$, repectively. The wake up time of each ZigBee node is denoted as $t(z_1), t(z_2)$ and $t(z_3)$, where $t(z_1) < t(z_2) < t(z_3)$.

forward packets from ZigBee node i to the server. The forwarding sequence is denoted as $S_i^n = (z_1, ..., z_n)$, which is sorted in the order of wake-up time $t(z_1) < ... < t(z_n)$.

In our design, a ZigBee sender has multiple potential forwarding nodes at each hop. As shown in Figure 6 (a), the forwarding sequence of z_i is $S_i^2 = (z_1, z_2)$. If the transmissions from z_i to 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 6 (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 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 6 (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 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 6 (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 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 to ZigBee but also need to consider the packet delivery ratio and delay from ZigBee to WiFi and WiFi to ZigBee.*

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 I.

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

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

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:

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

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 is 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 follow:

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

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 below:

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

Expected Z2W Delivery Ratio (R_i^{z2w}): As mentioned before, if the important data is 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:

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

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

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

The first term is the probability for the packet to be successfully delivered to the WiFi AP by the node i and the second term represents the probability for a packet to be delivered 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 WiFi AP but also depends on the delivery delay from the nodes in its forwarding sequence to WiFi AP.* The mathematic representation is shown below:

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

The first term represents the delay for the packet successfully delivered by the 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 is 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, *the link quality between ZigBee to 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} is can be represented as:

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

The first term represents the probability for the priority information delivered to the node i and the second term represents the probability for the priority information delivered to the nodes in the forwarding sequence of node i.

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 WiFi AP and ZigBee but also depends on the link quality between ZigBee to ZigBee.* The representation of D_i^{w2z} for node i in S_i^n is shown below:

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

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

*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 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 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 these metrics into consideration. Formally, we set the boundary conditions of the optimal forwarding sequence as follow:

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

In order to find the optimal forwarding sequence, for a ZigBee node i, the initial optimal forwarding sequence is set to: $S_i^{opt} = \emptyset$ and the maximum retransmission time is set to: 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 $\{z_{k+1}, ..., z_n\}$ and the wake up time of z_n should 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 a ZigBee node r in S_i^n into the optimal forwarding sequence, the corresponding procedure is represented as follows:

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

In order 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 $S_i^n = \{z_j, ..., z_n\}$, we need to try the sequence from $\{z_j, ..., z_n\}$ to $\{z_n\}$. 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 10.

C. Priorities Determination

After the optimization, each ZigBee node concurrently transmits the important data and raw data to the WiFi AP and the 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 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.

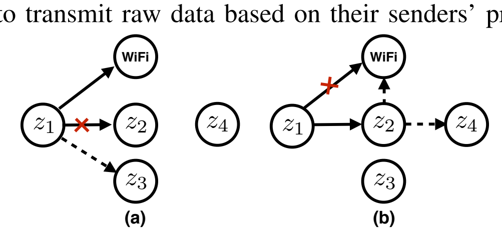


Fig. 7. The WiFi AP has no knowledge of whether the raw data is successfully delivered while the ZigBee receiver has no knowledge of whether the important data is successfully delivered.

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 the WiFi AP has no knowledge of the distribution of raw data in the network. For example, as shown in Figure 7 (a), the important data is successfully delivered to the WiFi AP from z_1 while the transmission of raw data is failed. Then, according to the transmission scheme mentioned in III-B1, 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 7 (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 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 8.

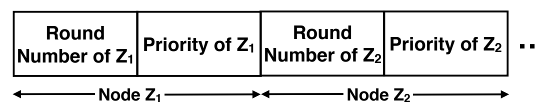


Fig. 8. The format of priority map.

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 8, 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 with the round number of raw data. There are three possible outcomes:

(1) The round number of important data and raw data are identical: This means 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 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 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.

D. Concurrent Data Forwarding

To further reduce the delivery delay for the data from high priority ZigBee nodes, in this section, we introduce a concurrent data forwarding scheme. By using this scheme, the raw data is also able to be forwarded by the WiFi AP. Specifically, during the sender state of each ZigBee node, if the transmission of the important data has finished while the raw data is still in transmission, the ZigBee nodes can transmit the raw data to the WiFi AP. To allocate the raw data transmission between ZigBee to ZigBee and ZigBee to WiFi, we introduce the following allocation process.

Formally, we denote the bit rates between ZigBee node i to ZigBee node j and ZigBee node i to WiFi AP y as b_{ij} and b_{iy}. Assuming 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 to ZigBee can be represented as:

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

Assuming 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 to WiFi communication can be represented as:

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

Then, the time difference between t_{ij}^{z2z} and t_{iy}^{z2w} can be represented as ($t_{ij}^d = t_{ij}^{z2z} - t_{iy}^{z2w}$). If t_{ij}^d is larger than 0, it means the transmission of important data has finished while the raw data is still in transmission. In this case, the ZigBee forwarder selects the raw data from highest priority to the lowest priority that can be transmitted to WiFi AP during t_{ij}^d and then forwards the raw data to the WiFi AP in the order of its senders' priorities. For the data with same priorities, the ZigBee forwarder selects the data that has been forwarded by a greater number of hops.

IV. 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. To show the advantages of our design, we use DSF [16] with predefined priorities as our baseline.

A. Experimental Setup

In this experiment, 25 ZigBee compliant TelosB nodes are randomly deployed in a $20m \times 20m$ square field. The duty cycle of each ZigBee node is set to 10%, where τ described in Section II is set to be $20ms$. We deploy WiFi compliant USRP B210 in the center to cover the whole field. We use our computer MSI GE62 6QC as the server. The USRP uses the most popular WiFi channel 1 and the transmission power is 25 dBm. 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, each ZigBee node has four possible priorities, which are shown in Table II. The nodes in DSF are randomly set with predefined priorities.

Priority 1	Urgent
Priority 2	Highly Important
Priority 3	Moderately Important
Priority 4	Less Important

TABLE II
PRIORITY ASSIGNMENT TABLE

B. Impact of Different Duty Cycles

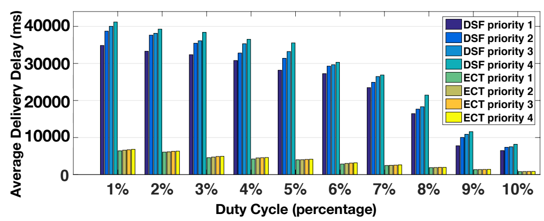


Fig. 9. Average Delivery Delay from ZigBee nodes to the Server vs. Duty Cycle

We first evaluate the performance of ECT in terms of duty cycle. During this experiment, over 5000 packets are transmitted from different ZigBee nodes to the server. As shown in Figure 9, the performance of ECT is much better than DSF. When the duty cycle increases to 10%, the average 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 WiFi AP as duty cycle increases, which increases the percentage of raw data transmitted through WiFi AP.

C. Impact of Different Network Traffic Volumes

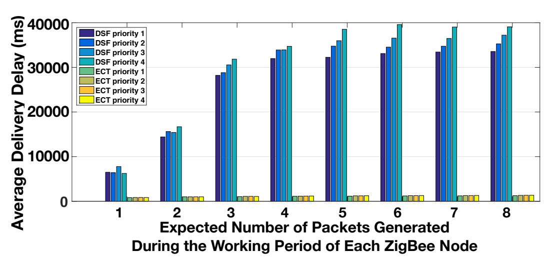


Fig. 10. Average Delivery Delay from ZigBee nodes to the Server vs. Traffic Volume.

Figure 10 evaluates the performance of ECT under different network traffic volumes. During this experiment, we ran our network for more than $2 \times 10^5 ms$. 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 $6240ms$, $6500ms$, $6400ms$, and $7780ms$, respectively while the corresponding average delivery delays of ECT are only $820ms$, $840ms$, $840ms$, and $840ms$, respectively. As the

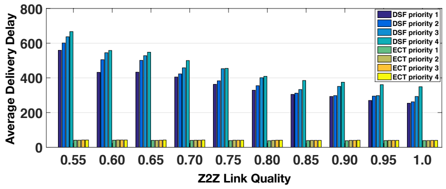


Fig. 11. Average Delivery Delay from ZigBee nodes to the Server vs. z2z link quality.

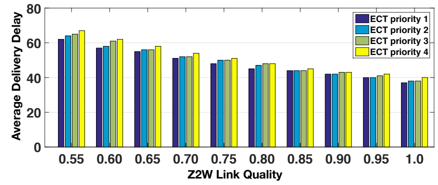


Fig. 12. Average Delivery Delay from ZigBee nodes to the Server vs. z2w link quality.

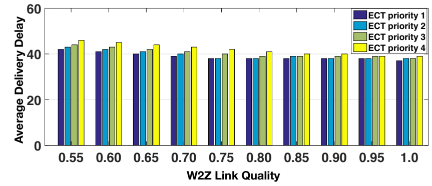


Fig. 13. Average Delivery Delay from ZigBee nodes to the Server vs. w2z link quality.

expected number of generated packets increases to 8, the average delivery delays of DSF are increased by more than 5 times while the corresponding average delivery delays of ECT are only increased to $1260ms$, $1320ms$, $1340ms$, and $1350ms$, 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 while ECT is still able to transmit packets to the server through WiFi APs.

D. Impact of Different Link Qualities

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 $100m \times 100m$ 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 WiFi AP was strictly implemented according to the experimental setup. Data collected at the server was obtained by averaging 10000 ZigBee node to server communications.

As shown in Figure 11, we can observe that the DSF suffer extremely high average delivery delay under low Z2Z link qualities, while the delivery delay of ECT just slightly increased by 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 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 12 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 13, the average delay is just slightly decreased as the W2Z link quality increases. This is because the W2Z link only transmits the priority map while the Z2W link transmits important data and raw data. That is, the link qualities have different influences on the average delivery delay.

E. 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 IV-D. 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 $100m$ to $223m$. In addition, the WiFi AP number varies from 25 to 75 to cover the whole field. As shown in Figure 14, the average delivery delay of DSF increases while 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.

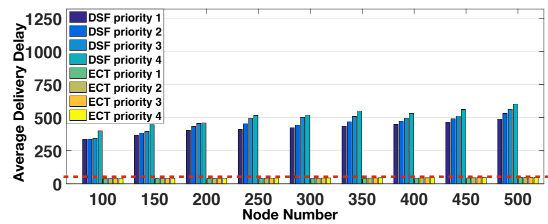


Fig. 14. Average Delivery Delay from ZigBee nodes to the Server vs. Network Size.

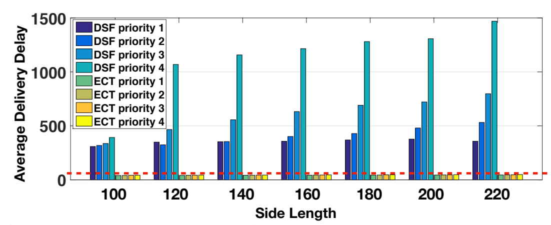


Fig. 15. Average Delivery Delay from ZigBee nodes to the Server vs. Node Density.

For different network densities, the ZigBee node number is set to 100 while the side length changes from $100m$ to $220m$. The number of 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 15, the performance of ECT only slightly decreases. This is because as the number of Z2Z retransmissions increase, they also provide more chances for ZigBee nodes to concurrently forward raw data to the WiFi APs. In addition, as shown in section IV-D, the Z2Z link quality does not have dominant influences on the average delivery delay. Therefore, the average delay is still much lower than DSF.

V. Related Work

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. As introduced in HoWiES [3], the author enables WiFi to ZigBee communication by sensing the packet length of WiFi packets. Moreover, researchers have introduced different methods to enable concurrent communication among heterogeneous IoT devices. For example, EMF [4] achieves concurrent communication between ZigBee and WiFi by shifting the packet transmission order while B2W2 [6] achieves concurrent communication between Bluetooth and WiFi.

Data Forwarding Techniques. Researchers have proposed different techniques and routing protocols [17], [18] to improve the wireless networks' performance, such as delay [19], [20], energy efficiency [21], [22], [23] and quality of service [24], [25]. Dynamic Switch-based Forwarding optimizes the data delivery ratio, delay and energy consumption [16], [26]. In [12], a forwarder selection technique and a link-quality-based backoff method are proposed to alleviate the hidden terminal problem and resolve simultaneous forwarding operations. Link correlation in wireless sensor networks is also studied to optimize the network performance [27], [28]. In [29], a correlation flooding method is introduced in low-duty-cycle networks to reduce the delivery delay. By leveraging link correlation, [30] significantly increase the network throughput by 55%.

Different from above methods, our work is the first cross-technology data forwarding design, which opens a new door for cross-technology network layer design. Specifically, 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 the cross-technology network modeling and design.

VI. Conclusion

In this paper, 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 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.

VII. Acknowledgements

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

References

[1] "http://www.gartner.com/newsroom/id/3598917."
[2] Z. Chi, Y. Li, Y. Yao, and T. Zhu, "Pmc: Parallel multi-protocol communication to heterogeneous iot radios within a single wifi channel," in *IEEE ICNP, 2017*.
[3] Y. Zhang and Q. Li, "Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices," in *INFOCOM, 2013*.
[4] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu, "Emf: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous iot devices," in *INFOCOM, 2017*.
[5] C. Systems, in *Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 White Paper*, 2014-2019.
[6] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2w2: N-way concurrent communication for iot devices," in *SenSys, 2016*.
[7] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, "Minimizing delay and maximizing lifetime for wireless sensor networks with anycast," in *TON, 2010*.
[8] M. H. Yaghmaee and D. A. Adjeroh, "Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks," in *Computer Networks, 2009*.
[9] C.-M. Yang, K.-P. Shih, and S.-H. Chang, "A priority-based energy replenishment scheme for wireless rechargeable sensor networks," in *WAINA, 2017*.
[10] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," in *IEEE Sensors Journal, 2014*.
[11] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "Rap: A real-time communication architecture for large-scale wireless sensor networks," in *RTAS, 2002*.
[12] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *MobiCom, 2009*.
[13] "http://www.eei.org/issuesandpolicy/grid-enhancements/Documents/Smart_Meters_RF_exposure.pdf."
[14] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *SenSys, 2004*.
[15] K. S. Yildirim and A. Kantarci, "Time synchronization based on slow-flooding in wireless sensor networks," in *IEEE TPDS, 2014*.
[16] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," in *TMC, 2011*.
[17] T. Zhu and M. Yu, "A secure quality of service routing protocol for wireless ad hoc networks," in *IEEE GLOBECOM, 2006*.
[18] D. Towsley and T. Zhu, "E2r: Energy efficient routing for multi-hop green wireless networks," in *IEEE INFOCOM, 2011*.
[19] L. Cheng, Y. Gu, J. Niu, T. Zhu, C. Liu, Q. Zhang, and T. He, "Taming collisions for delay reduction in low-duty-cycle wireless sensor networks," in *IEEE INFOCOM, 2016*.
[20] Z. Huang and T. Zhu, "Distributed real-time multimodal data forwarding in unmanned aerial systems," in *SECON, 2017*.
[21] Y. Gu, T. Zhu, , and T. He, "Esc: Energy synchronized communication in sustainable sensor networks," in *IEEE ICNP, 2009*.
[22] T. Zhu, S. Xiao, P. Yi, D. Towsley, and W. Gong, "A secure energy routing mechanism for sharing renewable energy in smart microgrid," in *IEEE SmartGridComm, 2011*.
[23] Y. Gu, L. He, T. Zhu, and T. He, "Achieving energy synchronized communication in energy-harvesting wireless sensor networks," in *ACM Transactions on Embedded Computing Systems ,2014*.
[24] T. Zhu and M. Yu, "A dynamic secure qos routing protocol for wireless ad hoc networks," in *IEEE Sarnoff, 2006*.
[25] A. Malvankar, M. Yu, and T. Zhu, "An availability-based link qos routing for mobile ad hoc networks," in *IEEE Sarnoff*.
[26] S. Ren, P. Yi, T. Zhu, Y. Wu, and J. Li, "A 3-hop message relay algorithm for connected dominating sets in wireless ad-hoc sensor networks," in *IEEE ICCC, 2014*.
[27] T. Zhu, Z. Zhong, T. He, and Z. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *USENIX NSDI, 2010*.
[28] J. Jun, L. Chen, L. He, Y. Gu, and T. Zhu, "Exploiting sender-based link correlation in wireless sensor networks," in *IEEE ICNP, 2014*.
[29] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *IEEE ICNP, 2011*.
[30] F. Chai, T. Zhu, and K. Kang, "A link-correlation-aware cross-layer protocol for iot devices," in *IEEE ICC, 2016*.