MCTS: Multi-Channel Transmission Simultaneously using Non-Feedback Fountain Code

Jianhang Liu¹  Shuqing Wang¹  Shibao Li¹  Xuerong Cui¹  Yan Pan²,³  Ting Zhu²

1. College of Computer and Communication Engineering in China University of Petroleum, Qingdao 266555, China;
2. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, USA
3. School of Computer Science and Engineering, Northwestern Polytechnical University, Xi’an, PR China, 710072
{lujianhang wangshuqing lishibao,xrcui}@upc.edu.cn, panyan@mail.nwpu.edu.cn, zt@umbc.edu

Abstract—Data dissemination is an essential application in IoT network. It is generally relayed hop by hop due to the limitation of the communication distance of IoT devices. However, multi-hop transmission increases dissemination time, especially in a low duty-cycle network. Channel contention is another factor result in long delay, because most of IoT devices and WiFi equipment work within the same bandwidth (2.4GHz). To reduce transmission delay of data dissemination, our group proposes a method of Multi-Channel Transmission Simultaneously using no-feedback fountain code (MCTS). Instead of terminating sending data by feedback as traditional fountain code protocol, in the MCTS, the sink node pours a certain amount of packets into the network according to the distance of object nodes to guarantee reliable transmission. By making use of channel encoding, the data to ZigBee nodes can be carried by some subcarriers of WiFi in multi ZigBee channels simultaneously when WiFi sends itself data. This method technically avoids the delay from channel contention. Compared with flooding method, the delay of the MCTS sending data from the sink node to all ZigBee nodes is decreased to 1/4.

Keywords—coexistence network, fountain code, multi-channel transmission, no-feedback transmission

I. INTRODUCTION

The ZigBee communication protocol using IEEE802.15.4 has been an effective measure on Internet of things (IoT). Information interaction between the sink node and ZigBee nodes is performed on certain applications, such as inspection of temperature, electronic pressure and environment [1, 2]. Due to the limitation of the transmission range of ZigBee nodes, a multi-hop relay is used for most of the data dissemination application. Generally, the sink node broadcasts data to its neighbor ZigBee nodes. These nodes flood the data to other nodes hop by hop. The method guarantees that all the nodes can receive the data. In low duty-cycle networks, however, one node cannot relay data until next hop nodes are in its wake slot. Consequently, the multi-hop retransmission causes a long delay.

With the increasing number of WiFi hot-spots, WiFi has become a preferred way to access the Internet. Compared with ZigBee network, the access point (AP) of WiFi transmits data faster (more than 10 times) than ZigBee network. However, transmission collision is inevitable in the coexistence of WiFi and ZigBee network, because most of WiFi and ZigBee devices work within 2.4GHz. Although CSMA/CA protocol can be used to avoid transmission collision, it increases the delay of data dissemination from the sink node to ZigBee nodes, especially when WiFi network occupies the channel frequently. Figure 1 shows an example of coexistence network of WiFi and ZigBee. The sink node disseminates data to the ZigBee nodes using flooding method. If the Z₁ and the Z₃ fail for channel contention, the transmission would not be finished until next wake slot, which causes a long delay. The finish time is at t₁ in this example.

Fig.1. Data dissemination in coexistence Network of WiFi and ZigBee

Fig.2. Comparison of the delay when data dissemination using traditional fountain code and the MCTS methods

It is not only a challenge but also an opportunity that the coexistence of WiFi and ZigBee network both working within 2.4GHz. If an AP can piggyback ZigBee data while sending...
WiFi data, the loss from channel contention of the two networks would be reduced dramatically and the effective throughput would be increased greatly. Prospectively, the method of hop by hop retransmission could be replaced by one hop broadcast that helps to reduce the delay. The incidental transmission can be achieved by hybrid coding in physical layer because the signals of WiFi and ZigBee can both use phase modulation to send data in this layer. Figure 2(a) shows data dissemination in ZigBee network using WEBee[3] or PMC method [4]. The finish time $t_2 < t_1$ (as shown in figure 1) is due to no channel contention.

Due to different communication ability of WiFi and ZigBee, incidental transmission is asymmetric in the network layer. Specifically, the sink node broadcasts data to ZigBee nodes by one hop, while ZigBee nodes must return ACK to sink node for guaranteeing reliable transmission by multi-hop as shown in Figure 2. Multi-hop transmission in low-duty cycle would inevitably increase delay. To resolve this problem, we propose multi-channel transmission simultaneously using no-feedback fountain code (MCTS). No-feedback mechanism avoids the time of returning ACK from the nodes to the sink node. Figure 2(b) shows the data dissemination procedure using the MCTS method. It is obvious that the finish time $t_2 < t_1$ because ACK feedback mechanism is canceled. Furthermore, multi-channel transmission simultaneously further improves transmission efficiency.

We make the following three contributions in this paper:

1. We modify fountain code mechanism and cancel the feedback from the destination to the source. According to link quality and packets loss rate that is related to the distance of the destination, the sink node determines how many encoding packets it will pour into the network in one transmission. In the MCTS, the sink node has several transmissions with different encoding packets to cover all ZigBee nodes. Therefore, even if a ZigBee node does not receive sufficient packets to decode in its wake slot, it can get new packets from its neighbors that have different wake slot.

2. We refine the transmission procedure of a wake slot of a ZigBee node and make use of subcarriers of WiFi to send encoding packets to up to four ZigBee nodes in four different channels simultaneously. Accordingly, four neighboring ZigBee nodes are grouped that have the same wake slot. They receive different packets simultaneously in four different channels and then share the packets each other. This method decreases the time period of which the sink node sends encoding packets.

3. To finish the task of data dissemination in ZigBee network as soon as possible, multi-broadcast of one hop is performed to replace multi-hop relay among ZigBee nodes. We prove that covering the wake slot of all nodes is NP-Hard. And we propose a self-adapting algorithm of approximate optimum solution which schedules transmission time of the AP according to the wake slots of ZigBee nodes. The algorithm traverses all ZigBee nodes using less transmission time for the AP.

II. RELATED WORK

Delay is an important metric of data dissemination for ZigBee network. Traditional data dissemination protocols [5~8] explore flooding method that send packets by multi-hop. Although these methods guarantee reliable transmission, it causes long back off delay. Cooperative broadcasting in physical layer [9] allows several senders to send same packets simultaneously and decode at the receivers, which improve the transmission efficiency. Glossy[10] experimentally proved that channel contention could be removed by constructive interference. The proposed method improves the flooding performance. Based on Glossy study, Splash[11] is proposed to enable data dissemination over parallel pipelines which improve the channel utilization greatly. Pando [12] makes use of fountain code to send disseminate data and propose silence-based feedback scheme by inspecting Received Signal Strength Indicator (RSSI). These methods above can improve channel utility efficiency or transmission reliability successfully. However the delay due to multi-hop transmission is not decreased.

Recently, the methods, such as [13~17], are proposed to improve spectrum utilization for solving the increasingly crowded 2.4GHz band. Compared with them, our research is based on heterogeneous radios with different protocols that are communicating with together. other cross-technology communication systems are also introduced. Such as Esense[18] and HowiES[19] enable WiFi to ZigBee communication by sensing the packet length of WiFi packets. FreeBee [20] modulates periodical beacons to achieve communication among WiFi, ZigBee and Bluetooth. B2W2 [21] use CSI of WiFi system to enable BLE and WiFi transmission. CCMA [22] is proposed to enable neighboring devices to communicate on orthogonal channels to avoid channel contentions. Two defense countermeasures are proposed in [23] to resist the jamming effect brought by the DoS attack. The authors [24] classify the existing mobile data offloading technologies through WiFi networks and propose a time-ordered aggregation model to reduce a dynamic network to a series of time-ordered networks [25].

III. DESIGN OF THE MCTS

The main idea of the MCTS is that the AP of WiFi, also as sink node in ZigBee network, transmits ZigBee data in up to four channels simultaneously when it sends WiFi data. The ZigBee data are encoded by LT code [26] which is a kind of fountain code. Every four adjacent ZigBee nodes with same sleep-wake cycle are grouped. At the beginning of wake slot, they receive encoding packets in four different channels. If they do not receive sufficient packets to decode data before the AP stops to broadcast data, they adjust channel and transmit packets to each other. In this stage, each node in this
group can obtain 4 times amount of packets because they received different packets in the broadcast stage. Fountain code is a kind of rateless code, which means the AP produces infinite and non-repeated encode packets. Every ZigBee node can receive different packets. Therefore, if the packets they received are not still sufficient, the nodes can obtain new packets from their neighbors that have a different sleep-wake cycle.

The AP pours a certain amount of encoded packets into the network several times to cover all the ZigBee nodes. These nodes can obtain sufficient packets for decoding through receiving directly from AP and communicating with other nodes. As a result, feedback mechanism of fountain code from destination to the source can be canceled to reduce long delay in low-duty cycle network.

A. Using Fountain Code Without Feedback

Because the transmission ability of the sink node is greatly stronger than ZigBee nodes, the sink node can send data to the destination by one hop, while the ZigBee node must transmit data back to the sink node hop by hop. In order to avoid long delay of the relay, we propose that the sink node encodes data using fountain code before sending to the destination.

Because fountain code is a rateless code, the source can produce endless encoding packets. The destination can decode data as long as it receives sufficient amount of packets, even though some of the packets loss. Clearly, it is not necessary to send ACK from destination to source to guarantee reliable transmission using fountain code. However, in traditional fountain code protocols, the sender needs a feedback from the receiver as successfully decoded to stop producing and sending encoded packets. Due to asymmetric transmission capability of the sender and the receivers, the receivers need to send the feedback hop by hop to stop the sender.

If the sink node sends encoding packets at time $t_0$, the destination node gathers sufficient packets to decode at time $t_1$ and then return the stop message. The sink receives the message at $t_2$, and stops to produce and send encoding packets. The amount of data the sink node sent is:

$$R = wt_1 + wt_\Delta$$ (1)

The $w$ is the rate of sending data, and $t_\Delta = t_2 - t_1$. Actually, $wt_1$ in the formula (1) is valid data, but the amount of data the sink node sent at $t_\Delta$ is redundant. In the IoT network with low-duty cycle, the $t_\Delta$ is long due to sleep cycle. The multi-hop feedback not only brings long delay, but also wastes finite bandwidth. Although Pando [12] proposed silence-based feedback scheme to low delay from the feedback, it cannot eliminate the kind of delay completely. Furthermore, the feedback mechanism requires the sink node to determine if all the nodes need to receive sufficient packets. It increases the complexity of implementation.

To solve the problem, we propose non-feedback fountain code method. The sink node determines the amount of encoding packets according to packet loss rate to the destination. When the amount of sending packets reaches a threshold, the sink node stops producing and sending data. Due to the communication range of the sink node is further than that of ZigBee nodes, the sink node can broadcast the packets to all the nodes which are in its communication range. The procedure of data dissemination is not hop by hop. The sink node broadcasts the packets in different wake slot of ZigBee nodes. According to the characteristic of fountain code, the sink node could use different encoding packets to indicate same data. These packets are sent to different nodes with different wake slot. Therefore, even if a node cannot receive sufficient packets to decode data in its wake slot, it can obtain new encoding packets from its neighbors with different wake slot. Collectively, this way is more efficient and less delay than hop-by-hop transmission to guarantee reliability.

Firstly, we discuss how many encoding packets a sink node should send to the destination for decoding. Although the MCTS is not limited to any specific Fountain codes, in our current analysis, we choose LT codes [26] which have low computation overhead. Assuming that the sink node would send $K'$ original packets, $\rho(d)$ in formula (2) is the degree distribution of ideal soliton. Apparently, the degree distribution cannot produce redundancy and there must be one and only one node which degree is 1 during every iteration procedure. However, it is easy to discover the situation in which there is not any node with 1 degree using the $\rho(d)$ function because the encoding packets are produced randomly by using pseudo-random code. This causes a failure of decoding.

$$\rho(d) = \begin{cases} 
\frac{1}{K'} & \text{for } d = 1 \\
\frac{1}{d(d-1)} & \text{for } d = 2, 3, \ldots, K'
\end{cases}$$ (2)

In order to guarantee that the receiver can decode when it receives sufficient packets, there should be at least one packet with 1 degree in every iteration procedure. Robust soliton distribution is as degree distribution in LT code. We define a positive function depicted as formula (3) which guarantees the number of 1 degree in every iteration procedure. $S = c \log_e(K'/\delta) \sqrt{K'}$ is the expected value of 1 degree. The $\delta$ is the probability by which the receiver cannot decode the original data.

$$\tau(d) = \begin{cases} 
\frac{S}{K'} & \text{for } d < K'/S \\
\frac{S}{K'} \log \left( \frac{2}{\delta} \right) & \text{for } d = K'/S \\
0 & \text{for } d > K'/S
\end{cases}$$ (3)

Then, robust soliton distribution is depicted as the Formula (4) which is degree distribution function.

$$\mu(d) = \frac{\rho(d) \tau(d)}{\sum_d \rho(d) \tau(d)}$$ (4)
In order to decode to $K'$ original packets, the receiver needs to obtain at least $M$ encoding packets, which depicted as formula (5).

$$M = K' + 2\log_e(S/\delta)S$$  \hspace{1cm} (5)

Because a wireless communication channel is vulnerable from the external environment, packets loss is therefore inevitable. The sink node needs to send at least $K$ ($K> M$) encoding packets in order to guarantee $M$ encoding packets being received by the receiver.

Assuming that the bit error rate of a ZigBee node $n$ which the distance away from the sink node is $d_n$ is $P_n'$, in a noisy channel, the $P_n'$ is often expressed as a function of the normalized carrier-to-noise ratio that measures denoted $E_b/N_0$ which is the energy per bit to noise power spectral density ratio. For example, in the case of QPSK modulation, the $P_n'$ is given by [27]:

$$P_n' = \frac{1}{2} \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$  \hspace{1cm} (6)

If the length of every packet is $l$, including control information, the packets loss rate $P_n$ is given by:

$$P_n = 1 - (1 - P_n')^l$$  \hspace{1cm} (7)

Assuming the total number of sending packets from the sink node is $k$ and the total number of received packets by the receiver is $M$, the probability distribution function of $p$ follows binomial distribution:

$$P_n(M, k) = C_k^M (P_n)^k (1 - P_n)^{M-k}$$  \hspace{1cm} (8)

To reach the probability that the node $n$ can obtain $M$ packets is larger than $Q$, the number of packets $K_n$ which sent from the sink node should be:

$$K_n = \min \left\{ k | \arg \sum_{j=0}^{k-M} P_n(M, M + j) \geq Q \right\} + \varepsilon$$  \hspace{1cm} (9)

The desired $M$ is identified under the system requirement $q$ through a fixed-point iteration approach. Starting with $M = K' + 2\log_e(S/\delta)S$, we estimate the corresponding $Q$ according to the latency distribution. The $\varepsilon$ is the cost of adjusting channel of the ZigBee nodes, which will be discussed shortly.

According to the analysis above, the sink node needs to broadcast at least $K_n$ packets so that the probability of the node $n$ obtaining $M$ packets reaches $Q$. If the $M$ equals $K' + 2\log_e(S/\delta)S$, just as the description at formula (5), the node can decode the original data. If the $Q$ reaches $100\%$, the feedback from ZigBee node to the sink node is not necessary.

Although we cannot guarantee $Q=100\%$, the node can obtain most of the packets if $Q$ is very near $100\%$. If the number of receiving packets is lower than $M$, the node could obtain new encoding packets from its neighbors with different wake slot, as we discuss at next part. Delay will be decreased greatly using the method, compared with hop-by-hop feedback mechanism.

B. Sending data to up to 4 ZigBee devices simultaneously

It is well known that every channel bandwidth of WiFi working at 2.4GHz is 20MHz or 40MHz, while the bandwidth of ZigBee is 2MHz with 3MHz guard band. Therefore, a WiFi channel of 20MHz overlaps four ZigBee channels, as shown in Fig.3.

![Fig.3. a 40Hz WiFi channel overlaps four ZigBee channels](image)

In the example of figure 2, a 40MHz channel of WiFi overlaps with four ZigBee channels (i.e., $11\sim19$). Within the channel of WiFi, there are 128 subcarriers among 7 subcarriers that overlap with one of the eight ZigBee channels. Therefore, we explore the possibility of using these $7\times8=56$ subcarriers to imitate ZigBee’s OQPSK modulation while keeping the rest 72 subcarriers intact. To realize this proposal, we enable the parallel communication to 8 ZigBee devices and 1 WiFi devices with 40MHz. However, due to the implementation of six pilot tones (transmitting a predefined message for environmental adapting) and two null tone (reserved tone) in IEEE 802.11n and its successor, it can only concurrently send four different ZigBee data streams to four ZigBee devices in parallel. For 20MHz bandwidth WiFi with four pilot tones and one full tone, two ZigBee data streams to two ZigBee devices can be sent out in parallel at the same time.

To perform this method, the sink node encodes original data to the packets as we discuss at part A sends the $K$ (according to formula (9)) encoding packets in four ZigBee channels simultaneously to a group ZigBee nodes that have same wake slot. They can communicate with each other. As a result, the sending time reduces to one-fourth of the time as single channel sending. Because the fountain code is rateless code, all of the encoding packets are different. Therefore, the nodes in the group can transmit their encoding packets after receiving from the sink node to decode.
According to above analysis, we assign four ZigBee nodes which are the communication range of each other into a group. These four ZigBee nodes in a group have same wake slot. The wake slot is divided into three phases which are data receiving, transmission in the group and transmission among groups (Figure 4).

The sink node decides when the encoding packets are sent out according to the wake slots. In the MCTS, the sink node broadcasts $K_n \{x=1,2\ldots n\}$ encoding packets to the ZigBee nodes that are in wake slot. The value of $K_n$ is determined by the farthest node which has the highest packet loss rate, as we discuss in part A. The sink node sends $n$ times until all the ZigBee nodes receive the encoding packets from the sink node. We will discuss the algorithm to enable the sink node the least number of transmissions at part C.

At the beginning of wake slot, all the ZigBee nodes work in a unified channel. The nodes receive the signal from which the sink node will send encoding packets at time $t_1$. Every node in the group adjusts to an agreed channel to receive the packets at time $t_2$. The phase of data receiving finish when there are no encoding packets being received or the received data is messy code. Messy code means that the task of sending encoded packets from the sink node has finished and will transmit WiFi data using these subcarriers. Because of the different encoding style between WiFi data and ZigBee Data in physical layer, ZigBee nodes cannot resolve the data if the sink node transmits WiFi data in the channel.

Transmission in the group is finished at time $t_4$ when all the nodes have the same number of encoding packets. If the number is equal or bigger than $M$ which is described as formula (5), these nodes decode original data. Otherwise, the nodes request encoding packets from their neighbor who has different wake slot which is the phase of transmission among groups. Making use of fountain code, the sink node produces different packets at a different time. Therefore, their neighbor must obtain different packets from which they received. If their neighbors do not have encoding packets or are at sleep slot, the nodes will receive the encoding packets or request from their neighbor at next wake slot.

During the procedure, whenever $t_1$ the is in the wake slot of the node, the whole communication process must be finished before it sleeps as long as it receives the incidental encoding packets from the sink node.

C. The scheme of sending encoding packets.

In order to save energy, ZigBee nodes sleep periodically, especially in low-duty cycle network with long sleep slot. The wake slots of neighboring nodes are overlapped so that they can communicate with each other. In traditional ZigBee network, the sink node explores flooding method to disseminate data, meanwhile the ZigBee nodes relay the data till all the nodes receive. Figure 6(a) is an example to flood data from sink to ZigBee nodes. It will be $n$ times transmission in the example. Hop by hop transmission in low-duty cycle network will increase delay greatly due to long sleep slot of the nodes.

Generally, the communication radius of WiFi is more than 100 meters, while the distance of ZigBee transmission is less than 10 meters in low-duty cycle network for saving energy although the capability of ZigBee nodes is actually stronger than the distance. Therefore, the AP of WiFi can send data to the 10-hops far ZigBee nodes directly. Making use of incidental transmission by the AP of WiFi we proposed in this paper, the sink node could multiply broadcast the encoding packets to the ZigBee nodes within its communication range, as shown in figure 6(b).

Transmission in the channel is finished at time $t_6$ when the whole node broadcast the encoding packets from the sink node. 

If the only one of the $n$ ZigBee nodes can receive the broadcast data from the sink node in every broadcast, then all of the ZigBee nodes receive the data when the broadcast times $m$ equals $n$. However, as we discussed above, the wake slots of neighboring ZigBee nodes are overlapped for keeping communication each other. These packets of every broadcast will be received by several ZigBee nodes. As a result, $m$ is smaller than $n$. our object is to enable the sink node to choose a suitable time to broadcast for guaranteeing the $m$ near the smallest value.

In ZigBee networks, the node wakes at the appointed wake time and communicate with neighbor nodes. It sleeps at
the appointed sleep time if there is no data communication. Otherwise, it will be awake until the communication finish.

Figure 7. the quantization of wake-sleep cycle of a node

Based on the characteristic, we assume that the sink node knows the wake-sleep cycle of all the ZigBee nodes within its communication range. It is easy to accomplish by periodical information exchange among the ZigBee nodes. We quantize the wake-sleep cycle of every node 0 as sleep status and 1 as wake status in the time axis, as shown in figure 7.

Initially, the sink node obtains the wake-sleep status matrix of all the ZigBee nodes within its communication range, as shown in figure 8.

\[
\begin{array}{cccc}
Z_0 &=& \{000111111100000000011...\} \\
Z_1 &=& \{0011111000000000...\} \\
Z_2 &=& \{0001111100000000...\} \\
Z_3 &=& \{00000001111100000...\} \\
\end{array}
\]

Figure 8. the wake-sleep status matrix of all the ZigBee nodes

In the matrix, the column is the time. The 1 in every column means that the node is at wake status at this time while the 0 stands for sleep status. Our aim is to get the minimum number of the time slot to cover all the ZigBee nodes. Given a universe \( T \) and a family \( C \) of subsets of \( T \), it means \( C \subseteq T \) is a set of the time which cover all the nodes, then our algorithm is to find:

\[
\min |C|_{C \subseteq T} \quad (10)
\]

In order to describe the problem clearly, we transform the matrix into a graph as shown in figure 9. The squares stand for the time consumed and the circles stand for the ZigBee nodes. There is an edge between consuming time and one node when the node is at wake slot at this time.

Figure 9. the graph for time and ZigBee nodes

In the graph, Given a set of elements \{\( Z_{i0}, Z_{i1}, Z_{i2}, \ldots \)\} as the universe, and a collection \( S \) of \( t \) sets (for example \( t_{10}=\{ Z_{i0}, Z_{i2}, Z_{i4}, \ldots \} \), \( t_{11}=\{ Z_{i1}, Z_{i3}, Z_{i5}, \ldots \} \) whose union equals the universe, the problem is to identify the smallest sub-collection of \( S \) whose union equals the universe. It is clear that the problem equals set cover problem (SCP) which is one of Karp’s 21 NP-complete problems shown to be NP-complete in 1972. That means the complexity of the problem is non-deterministic.

Firstly, a heuristic algorithm is used for polynomial time approximation of set covering to solve the problem. We choose the sets according to the rule as follows. At each stage, the set that contains the largest number of uncovered elements was selected. The following algorithm is to show the obtaining minimum set using the heuristic method.

Initially, the sleep-wake status of all nodes is put into the collection \( Z \) (line 1). If the sleep cycles of the nodes do not equal each other, the \( C \) is used as the max value of sleep cycles in the \( Z \) (line 2). The function of line 6-14 is to find a column which has the most 1s.

Algorithm 1: Heuristic Algorithm for Obtaining Minimum Set

1. Initial collection \( Z=\{z_0, z_1, \ldots, z_m, \ldots\} \)
2. \( C=\max\{\text{the length of cycle of }/z_0, z_1, \ldots, z_m/\} \)
3. \( \text{maxValue}=0; \ Z'=\{} \)
4. while (the size of \( Z \)>0) {
5. \( \text{st}=0; \)
6. for \( i=0 \) to the size of \( C \) {
7. \( \text{sum}=0; \ Temp=\{} \)
8. for \( j=0 \) to the size of \( Z \) {
9. \( \text{if}(Z[i][j]==1) \)
10. \( \text{sum}++; \ \text{put } Z[i][j] \text{ into } Temp \)
11. \( \text{if}(\text{st}<\text{sum}) \text{ st}=\text{sum}; \)
12. \( \}
13. \( \text{if}(\text{sum}>\text{maxValue}) \text{ maxValue}=\text{sum}; \ Z'=\text{Temp} \)
14. \( \}
15. \( \text{send data at time } \text{st}; \)
16. \( \text{delete } Z' \text{ from } Z \)
17. \( \}

The variable \( \text{st} \) is the time corresponding to this column. The nodes which value is 1 are put into the connection \( Z' \) and delete the nodes of \( Z' \) in collection \( Z \) (line 16). Repeat the process till there are no nodes in the \( Z \). At this time, all the nodes will receive the data from the sink node in their wake slot.

Is the \( m \) the smallest if the sink node selects the time of the most 1s in the column to broadcast every time? Figure 10 is an example that the greedy algorithm is not optimum solution.

Figure 10 an example to prove the greedy algorithm is not optimum solution
In figure 9, there are 6 ZigBee nodes. The z0, z1, z3, z5 are in wake slot at time t1 and z1, z3, z4 wake at t2 while z0, z2, z5 wake at t3. Using greedy algorithm, the number of wake node is most at time t1 which should be regarded as the first time point to be sent out. And then z2 and z4 are left. The sink node has to broadcast at t2 and t3 respectively. The total times of transmission are 3 using the method (namely, m=3). Actually, if the sink node only broadcasts data at time t2 and t3, the six ZigBee nodes can receive the data successfully. Therefore, the smallest value of m should be 2 on this example.

This algorithm achieves an approximation ratio of \( H(t) \), where \( t \) is the size of the set to be covered. In other words, it presents a covering that may be \( H(c) \) times as large as the minimum one, where \( H(c) \) is the \( c \)-th harmonic number:

\[
H(c) = \sum_{k=1}^{c} \frac{1}{k} \leq \ln n + 1 \quad (11)
\]

This heuristic algorithm actually achieves an approximation ratio of \( H(t') \) where \( t' \) is the maximum cardinality set of \( T \). The example of figure 9 shows that the greedy algorithm is essentially the best-possible polynomial time approximation algorithm for set cover up to lower order terms, under plausible complexity assumptions. In fact, a tighter analysis for the heuristic algorithm [28] shows that the approximation ratio is \( \ln n - \ln \ln n + O(1) \).

### IV. EVALUATION

To extensively evaluate different aspects of the MCTS under a wide range of settings, we utilized off the shelf ZigBee-compliant TelosB devices installed with TinyOS to receive the OQPSK signals from a PMC sender.

Figure 11 PMC speed testing

The PMC sender is implemented using an NI RF testbed which consists of a signal digitizer (PXIe-5622), a signal generator (PXIe-5652), down-converter (PXIe-5601)

Figure 12. The comparison of dissemination time that the number of ZigBee nodes are \( n = 52, 100, 200 \) and the number of original packets are \( m=100, 300, 500 \) respectively. We compare the dissemination time under different duty cycles.

Figure 13. In the network with 1% duty cycle, these are the graphs about the comparison of the dissemination time using the MCTS, traditional fountain code method with PMC (PMC-feedback) and flooding hop by hop.
Figure 14. In the network with 5% duty cycle, these are the graphs about the comparison of the dissemination time using the MCTS, traditional fountain code method with PMC (PMC-feedback) and flooding hop by hop.

Figure 15. In the network with 10% duty cycle, these are the graphs about the comparison of the dissemination time using the MCTS, traditional fountain code method with PMC (PMC-feedback) and flooding hop by hop.

up-converter (PXIe-5450), and streaming raid. Using this testbed, we choose 40MHz as WiFi sending bandwidth which borrows 28 subcarriers to send data to four ZigBee nodes. The highest aggregated throughput from a PMC sender to every ZigBee receiver is 121.02 Kbit/s, while the throughput of WiFi receiver reaches 233.77 Mbit/s. We use the real speed of ZigBee nodes receiving in the simulation experiment.

According to the method in IV-B, the sink node sends data to up to 4 ZigBee nodes simultaneously. We generate randomly 52, 100 and 200 ZigBee nodes in the distance from 1 to 30 meter away from the sink node. Four adjacent ZigBee nodes with the same sleep-wake cycle are a group. The length of original packets is 512 bit. We encode 100, 300 and 500 original packets ($K'$=100, 300 and 500) to encoding packets and send to the ZigBee nodes respectively. The number of packets $K_s$ which the sink node sends to different position nodes is different according to formula (9). The ZigBee nodes need to receive at least 117, 353 and 585 for decoding ($M$=117, 353 and 585) in accordance with formula (5).

We carry out 81 groups in the experiment to evaluate the dissemination time using the MCTS, traditional fountain code method with PMC (PMC-feedback) and flooding hop by hop in the simulation system 100 times and running time is from 1 second to 62 seconds. All results are averaged by getting rid of outlier data.

We use three methods (flooding, PMC-flooding, MCTS) to disseminate data in the experiments. The sink node sends $K'$ packets to its one hop neighbors and then the neighbors relay the packets until all the nodes receive it using flooding method. In contrast, PMC-feedback allows the sink node to stop sending encoding packets until it receives the ACKs from all leaf nodes. In the MCTS, the sink node broadcasts $K_s$ encoding packets in a different time slot according to the position of the farthest objects. The sending procedure will be implemented several times until all nodes are covered.

Figure 12 shows that using the MCTS method, the average value of dissemination time is 2.33 second, which 52 ZigBee nodes can resolve 100 original packets (the ZigBee nodes need to receive at least 117 encoding packets.) when duty cycle is 1%. During the process, the sink node sends 1032 packets and broadcast 8.3 times averagely. It is worth noting that the dissemination time is not increased when the number of ZigBee nodes rises to 100 and 200. The reason is that transmission capability of the PMC sender is larger than ZigBee nodes. Using the MCTS, the sender chooses the sending time which more ZigBee nodes are in wake slot of
according to their sleep-wake cycle. Therefore, it can cover all the ZigBee nodes using fewer broadcast times. The dissemination time is longer with the increase of sending packets. When the number of original packets is 500, the dissemination time reaches about 6.5s.

The second and third graph in figure 12 show that the sink node sends same data to those ZigBee nodes under 5% and 10% duty cycle. It illustrates that the dissemination time is reduced with higher duty cycle. That because broadcast times of the sink node decreases due to more wake time of ZigBee nodes.

Figure 13, 14 and 15 show the comparison of dissemination time using the MCTS, PMC-feedback and flooding under 1%, 5%, 10% duty cycle. The number of nodes, the number of sending packets and duty cycle impact the dissemination time.

In the network with 1% duty cycle, it spends more than 20s for all the nodes to obtain 100 original packets using PMC-feedback and flooding method. The dissemination time reaches 61.98 second when the sink node disseminates 500 packets to 200 nodes using flooding method, which is about 10 times larger than the dissemination time using the MCTS. The time using PMC-feedback in this condition is also more than 8 times than the MCTS, as shown in figure 13. Undoubtedly, time for feedback is the main reason for so long of a delay period, especially low duty-cycle.

Figure 14 and 15 illustrate the delay when duty cycle is 5% and 10% respectively. It can be found that the delay drops using the three methods. The decrease amplitude of PMC-feedback and flooding are larger than that of MCTS. For example, it spends 25s and 37s to disseminate 500 packets. Comparatively, using MCTS method, the time to decode these packets for all the nodes is 6.25s.

The figures illustrate that using the MCTS to disseminate data in ZigBee network can improve the performance at least 4~10 times than those methods that need feedback to keep reliable transmission. The MCTS obtains better performance especially in low-duty cycle networks.

V. CONCLUSION

Transmission hop by hop causes longer delay, especially in low duty-cycle networks. The same problem happened on return ACK from left nodes to sink nodes. To reduce dissemination time, this paper presents a data dissemination method MCTS which perform PMC technology to piggyback ZigBee data while sending WiFi packets. The MCTS explores fountain code to broadcast data to one-hop nodes. It decides the amount of sending packets according to packets loss rate and further avoids delay from feedback. By using 28 subcarriers of WiFi, the MCTS sends data up to four ZigBee nodes simultaneously in four different channels. Thus, the performance of transmission is improved greatly. Our MCTS provides a new insight to transmit data for asymmetry nodes and results from designed experiments prove that the MCTS can decrease transmission time greatly compared with the popular transmission methods.

ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation of China (NSFC) under Grant No. 61671482, 61601519 and the Fundamental Research Funds for the Central Universities under Grant No. 18CX02134A, 18CX02137A, 17CX02042A.

REFERENCES