Survey and Simulation of Cross Technology Communication

CHANG LIU, Michigen State University, USA
JIANZHI LOU, Michigen State University, USA

In this paper, we studied the state-of-the-art cross technology communication (CTC) approaches, and we focus on explaining a specific CTC paper, WEBee[6], which focuses on physical layer emulation to achieve the direct communication from WiFi to ZigBee. We will explain the mechanism behind that paper, as well as using GNURadio[2] to simulate their work.

Additional Key Words and Phrases: Wireless communication, Cross technology Communication

ACM Reference Format:
Chang Liu and Jianzhi Lou. 2021. Survey and Simulation of Cross Technology Communication. 1, 1 (March 2021), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Cross technology communication (CTC) is an emerging technique which allows different type of wireless devices to communicating directly. In practice, CTC is beneficial to users with multiple types of wireless devices. Nowadays, almost everyone have a WiFi devices in the house, and meanwhile, it is a growing trend to deploy more smart-home devices at home. A lot of energy-efficient device can not receive WiFi signals since WiFi costs a lot of energy. As a result, some popular smart devices, such as smart light bulb, use ZigBee as the communication mechanism. Traditionally, as figure 1 has illustrated, the control signal should pass through a gateway to be converted into ZigBee signal to send commands to IoT devices, which is very inconvenient. Consequently, Cross technology communication (CTC), which allows different types of wireless devices to communication directly, is emerging from the huge demand of smart-home device management.

The performance is always a major concern of CTC approaches, since the huge difference between WiFi, ZigBee, Bluetooth and other wireless standards. The efficiency will decrease dramatically during the conversion between standards. Therefore, the major focus of CTC research is to improve the throughput.

Currently, the physical level CTC reaches the state-of-the-art performance. One example of physical level CTC is WEBee[6], which achieves CTC from WiFi to ZigBee. WEBee will elaborately choose wifi packet payload, so that the transmitted signal becomes similar to ZigBee signal, and can be received directly by a ZigBee receiver. WEBee doesn't require any modification to the current standards used by devices. Meanwhile, the its physical layer emulation can reaches the highest throughput comparing with other approaches such as packet level CTC or gateways.

In this paper, we will firstly illustrate the mechanism of WEBee, and then we will show our simulation through GNUradio[2]. Section 2 will be the related work, which talks about several types of CTC approaches and there advantages and disadvantages; in Section 3 we will briefly compare the difference between WiFi and ZigBee standard; Section 4 is the illustration of how WEBee works. Section 5 will show our simulation process and results; Section 6 is our conclusion about both the WEBee paper and our thinking about the future work.

Authors' addresses: Chang Liu, Michigan State University, East Lansing, MI, 48912, USA, liucha39@msu.edu; Jianzhi Lou, Michigan State University, East Lansing, MI, 48912, USA, loujianz@msu.edu.

XXXX-XXXX/2021/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

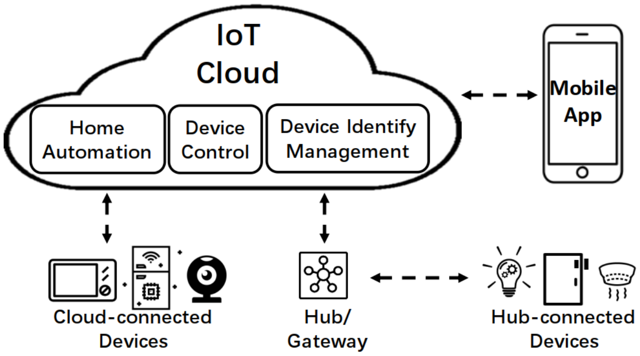


Fig. 1. The Hub/Gateway for smart home control.[13]

2 RELATED WORK

Traditional wireless network before the wide application of IoT deployment only allow the same kind of devices communicating with each other. Since different standard have totally different modulation methods on physical layer, transmitting data among different wireless devices seems to be unreachable.

Gateway is the most primitive solution for the issue above. However, Gateway have some limitation such as high hardware costs. Also, the hardware deployment will affects the performance a lot. Cross technology communication provides new methods to address such wireless communication obstacles. There are several benefits and advantages if we can transmit data directly rather than using cloud [1] or local gateways [9] [10]. For example, the cost of deployment could be saved, and also, security could be ensured without uploading data. Recently, FreeBee[4] changes internal timing to embed symbols, but the data rate is restricted by beacon rates of commercial WIFI (about 102ms/beacon).

Other attempts for CTC want to improve bit rates of CTC by physical operations like: use absence of packets to transmit data [8]; modulate predefined packets size from alphabet to convey information [12]; sequence pattern [11]. However, these attempts will sacrifice efficiency of normal data transmission between peer devices. For example, the bandwidth of WIFI is 10 times of Bluetooth and ZigBee, physical operations above are a waste of bandwidth.

Most of the works above, can be classified into packet level CTC, or side-channel encoding. Packet level CTC approaches doesn't not transmitting ZigBee or Bluetooth signal directly, but they use the presence/absence, or the signal strength of the transmitting packets (usually WiFi packets) to represent “0” and “1” of receiving signal. WiZig [3] is the state-of-art work among these side-channel CTC approaches. As Figure 2 have illustrated, WiZig use the presence/absence to encode the receving ZigBee bits. A major disadvantage of packet level CTC is the throughput, since the duration of a packet is relatively long, which decreases the efficiency of side channel dramatically.

Meanwhile, physical level CTC is a novel approach which mitigates the disadvantage of packet level CTC. In physical level CTC, the received symbols are encoded in bit level, which can easily reach a higher throughput than packet level CTC. One of the examples of physical level CTC is WEBee, which elaborately choose the payload of a WiFi packet to make it readable by ZigBee devices after the modulation of WiFi. In the following sections, we will briefly introduce how WEBee works.

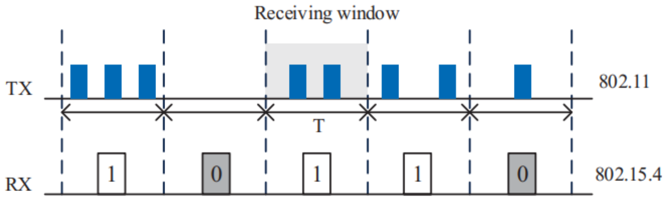


Fig. 2. The side channel encoding of WiZig

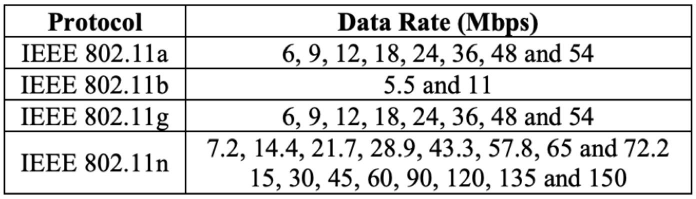
3 DIFFERENCE BETWEEN WIFI

We will start with introducing the difference between ZigBee and Wifi.

3.1 Protocols and Standards

As we know, WiFi and Zigbee are based on different standards: 802.11 and 802.15.4 respectively. We should carefully look over these two standards before analyzing and compare Wifi and Zigbee.

The 802.11 standards are one of the IEEE 802 set of standards, which occupies 2.4 and 5 GHz and is widely deployed on Wifi devices. The 802.11 standards specifies the MAC (data link) and physical layers in wireless communication while don't involve with Transport layer and above. The 802.11 evolved from 802.11a to 802.11be during about 20years, yet OFDM remains to be one of the most important ideas in 802.11 big family.[5]Although the core ideas in the 802.11 big family didn't change much, there are remarkable progress in improving data rate as Figure 3 shown.



Protocol	Data Rate (Mbps)
IEEE 802.11a	6, 9, 12, 18, 24, 36, 48 and 54
IEEE 802.11b	5.5 and 11
IEEE 802.11g	6, 9, 12, 18, 24, 36, 48 and 54
IEEE 802.11n	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65 and 72.2 15, 30, 45, 60, 90, 120, 135 and 150

Fig. 3. Data Rate Evolution

The 802.15.4 standards are one of the WPAN standards (802.15). There are 3 different types of 802.15 WPAN standards which could be distinguished by data rate. And 802.15.4 represents low data rate WPAN, especially for Zigbee devices.[7]

3.2 Modulation

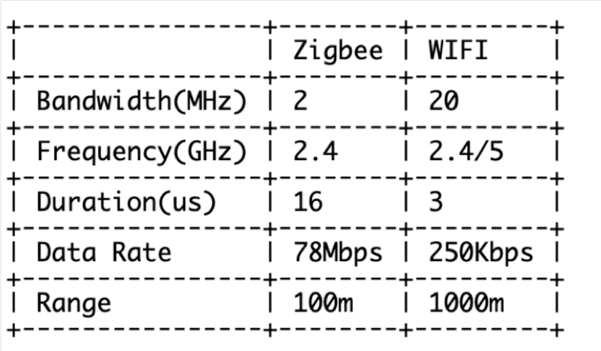
Wifi and Zigbee devices apply different modulation: QAM and OQPSK for various reasons.

As we know, OFDM is widely applied in Wifi transmission which enables Wifi devices not sensible to environment with noise. So QAM can make a single Wifi symbol carry more bits than BPSK or QPSK and thus most Wifi devices use 16-QAM or 64-QAM to modulate data.

However, Zigbee devices are much sensible to noise because its has low power, narrow bandwidth and without OFDM. And OQPSK doesn't require high SNR and its offset modulation could decrease BER(bit error rate).

3.3 Frequency, Bandwidth and Symbol Duration

Wifi and Zigbee devices have different physical features but if we want to implement Wifi to Zigbee communication , it's possible.



```
+----------------+--------+---------+
|                | Zigbee | WIFI    |
+----------------+--------+---------+
| Bandwidth(MHz) | 2      | 20      |
+----------------+--------+---------+
| Frequency(GHz) | 2.4    | 2.4/5   |
+----------------+--------+---------+
| Duration(us)   | 16     | 3       |
+----------------+--------+---------+
| Data Rate      | 78Mbps | 250Kbps |
+----------------+--------+---------+
| Range          | 100m   | 1000m   |
+----------------+--------+---------+
```

Fig. 4. Different features of Wifi and Zigbee

4 WEBEE

WEBee is a physical level CTC implementation which allows WiFi device transmit ZigBee frames directly. It's major idea is to elaborately select WiFi frame payload. It can reach 126 Kbps throughput, which is 16,000x faster than other state-of-the-art CTC approaches[6].

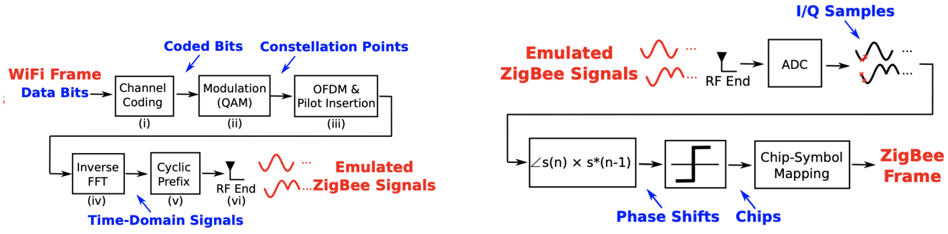
4.1 QAM Emulation

As mentioned previously, WiFi and ZigBee are different standards. Figure 5a and **??** illustrates the mechanisms of WiFi transmitter and ZigBee receiver respectively. WEBee aimed at connecting this two different parts. Their approach is physical layer emulation, as Figure 6 shows.

Firstly, for the desired ZigBee frame, it is possible to generating the corresponding ZigBee time domain signals (Using GNURadio or a ZigBee sender). After that, the generated signal is passed into some signal processing blocks which is similar to the reverse of WiFi transmitting part. As figure Figure 6 has displayed, this blocks includes FFT (which is the reverse of IFFT in WiFi transmitter) and QAM demodulation (Which is the reverse of QAM modulation in WiFi transmitter). After that, the generated "0"s and "1"s becomes the elabrately selected payload. After the processing of WiFi transmitter, the payload will be recovered (approximately) into the original desired ZigBee signal, and it can be received by ZigBee receiver directly.

4.2 Channel Coding

The QAM emulation is the core idea of WEBee, but there are some other challenges. One is channel coding. Because the redundancy of channel coding, the source bits are always less than channel bits. For example, as Figure 7 shows, the source bit number are 216 and channel bit number are 288, which means the matrix is inevitable. In this case, not every desired ZigBee frames can be emulated by WiFi frames, since the equations(288 equations with 216 unknowns) may not have a solution.



(a) How WiFi transmitter works[6]

(b) How ZigBee receiver works [6]

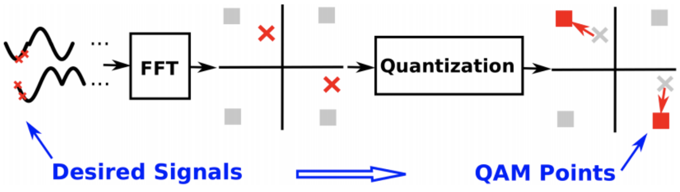


Fig. 6. Physical layer emulation[6]

Luckily, ZigBee uses only 7 subcarriers, which is much less than WiFi (48 subcarriers). Even when WiFi frame to simulate 2 parallel ZigBee frames, the channel coding bit will be only $7 \times 2 \times 6 = 84$ bits. That will solve the no-solution problem in channel coding, since 84 is less than 216.

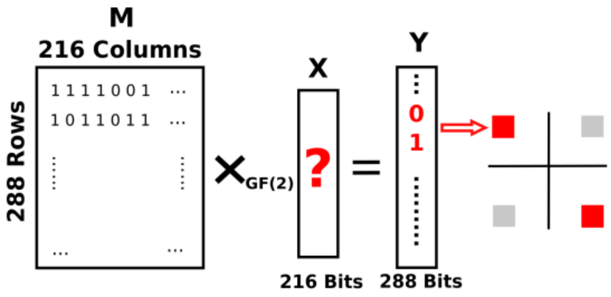


Fig. 7. Channel Coding[6]

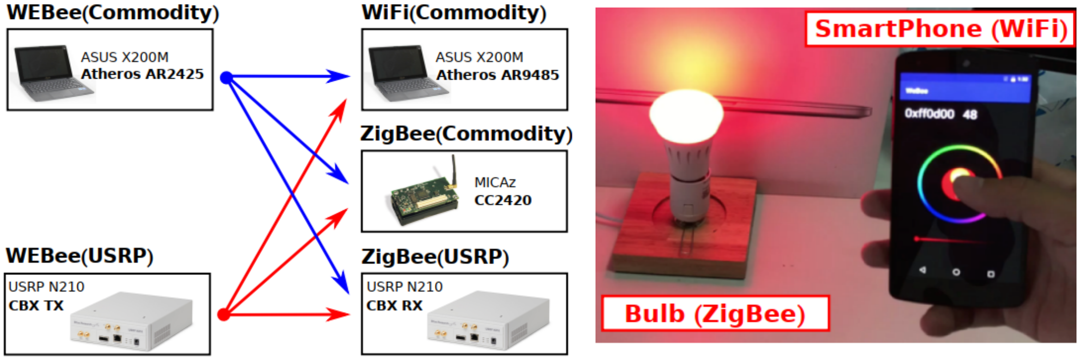
4.3 Post QAM emulation

Another challenge is caused by symbol segmentation and adding CPs. Firstly, as Figure 9 shows, because the duration of a ZigBee symbol is 16us, which is 4x of a WiFi symbol, then a ZigBee symbol have to be split into 4 pieces and then FFT can be performed. Secondly, during the WiFi transmitting, CPs will be added into each packet to prevent the noise, which is shown in Figure 10a. Both of the processing will cause the non-continuity of emulated signal. However, since WEBee aimed on achieving CTC without modifying any firmware or hardware, this issue cannot be removed totally. The mediation is to flip half of the chips in the boundary, as Figure 10b shows.

4.4 Evaluation Result

After overcoming the challenges in channel coding and post0-QAM emulation, WEBee can achieve CTC between WiFi and ZigBee, with throughput 126 Kbps, and 99% reliability.

Figure 8a is the experiment setting up of WEBee. Since there is no modification in firmware, commodity devices are enough for the implementation. The usage of USRP is to measure the performance, such as the symbol error rate. Figure 8b is a proof-of-concept implementation which allows smart phone (WiFi) to control smart light (ZigBee) directly.



(a) The experiment setting up of WEBee[6]

(b) Controlling the smart light (ZigBee) using phone (WiFi) [6]

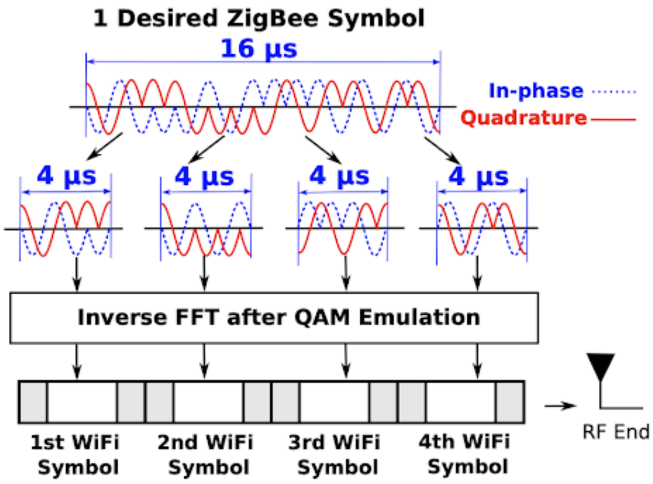
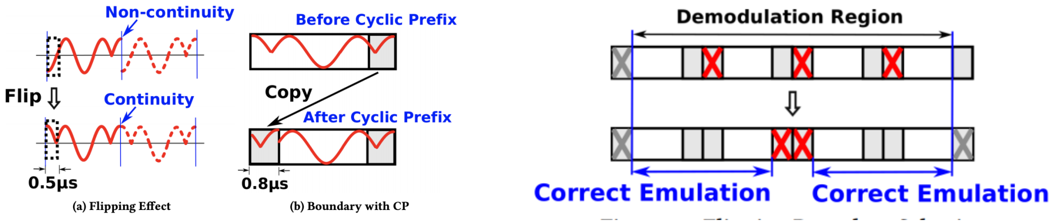


Fig. 9. The difference between ZigBee and WiFi symbols[6]



(a) Adding CP[6]

(b) Selective bondary flipping[6]

5 SIMULATION

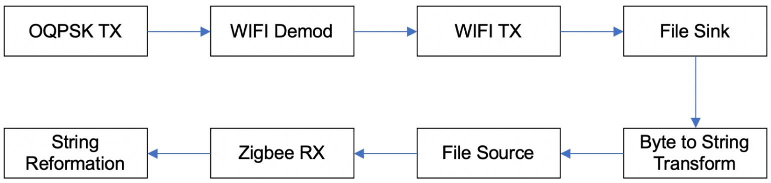


Fig. 11. Work Flow for simulation

The work flow could be divided into 2 parts by Byte to String Transform because all WiFi symbols are cached into file and then act as file source for Zigbee to receive (Simulation don't involve with hardware devices such as USRP). In the simulation, sine no hardware involved, we assume the Wifi transmitter and Zigbee receiver share the same bandwidth 20MHz, but it may not work in implementation.

5.1 OQPSK Transmitter and WiFi Demodulation

The most important step in the work flow is Wifi Demodulation (QAM Emulation). In the simulation, OQPSK symbols are transmitted by Zigbee source and this desired symbol represent certain words such as "helloworld", "callmydog" and so on. And these symbols could be interpreted as 64-QAM (All Wifi modulation in the simulation is 64-QAM) to be intermediate symbols which can not be read directly.

5.2 Byte to String Transform

In the transform, we use the standard ASCII table to assign a char for each 8-bits Wifi symbol, and those abnormal symbols will be interpreted as blank char.

Meanwhile, the Byte to String Transform assumes that all preambles and CP of Wifi TX symbols will be involved into the transform, which however, could be meaningless and these inevitable overheads affect the efficiency of the transform.

ALGORITHM 1: Byte to String Transform Function

```
Result: Wifi TX symbols
Input  : s(x), a byte sequence
Output: s(y), a string
temp, a list of 8 element
iter = 0
C, a char in ASCII Form
P, an integer of decimal
while $iter \leq s(x).length$ do
    if $iter\%8 = 0$ then
        $temp.reverse()$
        $P \leftarrow temp$
        $C \leftarrow ASCIITable(P)$
        $s(y).append(C)$
        $temp.clear()$
    else
        $temp.append(s(x)[iter])$
    end
end
return $s(y)$
```

5.3 Zigbee Receiver

The 802.15.4 standards for Zigbee set a threshold which can filter some of the symbols, but in this simulation, because of the environment with no noise, the threshold can be 0. If we compare the two time domain symbols, we could validate the Wifi TX symbols and Zigbee RX symbols.

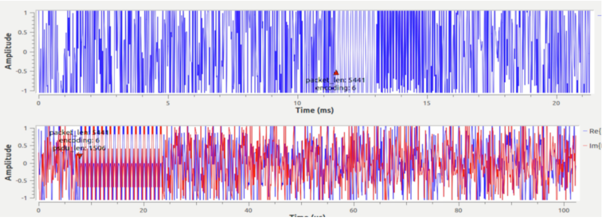


Fig. 12. Wifi TX and Zigbee RX symbols

5.4 String Reformation

The String Reformation is quite similar to Byte to String Transform Function, but in this step, we should remove all blank chars which are recognized as noise (some of them are Preambles and CP).

One problem during the simulation is that the length of desired string and reformatted string are not the same, so it's difficult to get accuracy or BER. And the accuracy we get is calculate the exact same char, which is much lower than the symbol accuracy.

ALGORITHM 2: String Reformation Function

```
Result: Reformatted String
Input  : s(x), a byte sequence
Output : s(y), a string
temp, a list of 8 element
$iter = 0$
C, a char in ASCII Form
P, an integer of decimal
while $iter \leq s(x).length$ do
    if $iter\%8 = 0$ then
        $P \leftarrow temp$
        $C \leftarrow ASCIITable(P)$
        $s(y).append(C)$
        $temp.clear()$
    else
        $temp.append(s(x)[iter])$
    end
end
Remove the blank of s(y)
return $s(y)$
```

6 SIMULATION RESULT

Figure 13 is the our simulation result. We selected some original strings and compared then with the symbols which is finally received by ZigBee receiver.

As we can see from the figure, the result is not very promising, and only very few symbols match. That may be caused by our limitations. For example, we only simulated the QAM emulation, without the detailed exploration in channel coding and post-QAM emulation. Also, without involving hardware will cause some difficulties to our project.

7 CONCLUSION & FUTURE WORK

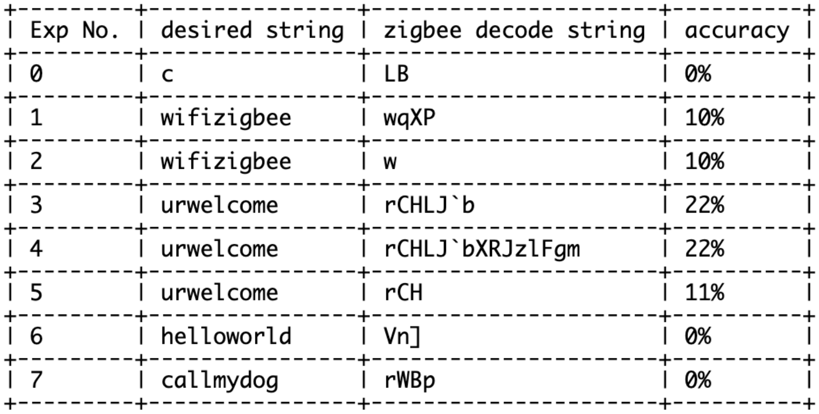
According to the result of our simulation, we found that our work still have some limitations:

1. Firstly, we only simulated the QAM emulation part without the channel coding and post-QAM emulation. Our simulation can basically make the ZigBee device receive the signal. However, the accuracy, or BER, is not ensured by only QAM emulation. If time permits, we will finish the simulation entirely and improve its accuracy.

2. Secondly, after generating the desired signal and demodulating it, we use ASCII to encode "0"s and "1"s into ASCII symbols. However, a lot of the ASCII symbols are invisible and unprintable, which causes a lot of difficulty to our simulation, since we will use the converted ASCII symbol as the WiFi payload. Maybe a solution is feed the bit level payload into physical layer directly.

3. Last but not least, we only focused on simulation but not real world implementation. Which means we didn't consider some other factors such as noise. Using USRP to implement WEBee with consideration of environment will a better choice.

Meanwhile, there are some future directions we realized during this project:



```
+---------+----------------+----------------------+----------+
| Exp No. | desired string | zigbee decode string | accuracy |
+---------+----------------+----------------------+----------+
| 0       | c              | LB                   | 0%       |
+---------+----------------+----------------------+----------+
| 1       | wifizigbee     | wqXP                 | 10%      |
+---------+----------------+----------------------+----------+
| 2       | wifizigbee     | w                    | 10%      |
+---------+----------------+----------------------+----------+
| 3       | urwelcome      | rCHLJ`b              | 22%      |
+---------+----------------+----------------------+----------+
| 4       | urwelcome      | rCHLJ`bXRJzlFgm      | 22%      |
+---------+----------------+----------------------+----------+
| 5       | urwelcome      | rCH                  | 11%      |
+---------+----------------+----------------------+----------+
| 6       | helloworld     | Vn]                  | 0%       |
+---------+----------------+----------------------+----------+
| 7       | callmydog      | rWBp                 | 0%       |
+---------+----------------+----------------------+----------+
```

Fig. 13. Simulation result

1. Firstly, the reliability is always a big problem in CTC, especially when WiFi device are communicating with ZigBee device. Sin ZigBee device is more sensitive to noise, any tiny noise will affect the reliability of CTC. In future, it is a good direction to work on CTC in noisy environment.

2. Secondly, since CTC provides more chance of communication between different types of devices, it will increase the security problem. For example, an adversary is able to attack ZigBee devices through WiFi transmitter. That would be a long term topic that we and other researchers can work on.

3. The overhead of CTC can also affect the throughput. In this project, the preambles and CPs in WiFi cannot be removed, since WEBee doesn't change any hardware of WiFi transmitter end. However, this preambles and CPs are useless in CTC, since they are noise to ZigBee receiver. In future, there would be possible solution to remove preambles and CPs during transmitting ZigBee signals.

REFERENCES

[1] D. Culler and S. Chakrabarti. 2009. 6lowpan: Incorporating ieee 802.15. 4 into the ip architecture.

[2] Inc GNU Radio Foundation. 2019. GNURadio Ofiical Website. https://www.gnuradio.org/

[3] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. 2017. Wizig: Cross-technology energy communication over a noisy channel. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[4] S. M. Kim and T. He. 2015. Cross-technology communication via free side-channel. ACM MobiCom.

[5] Chee Kyun Ng Nor Kamariah Noordin Kok Seng Ting, Gee Keng Ee and Borhanuddin Mohd. Ali. 2011. The Performance Evaluation of IEEE 802.11 against IEEE 802.15.4 with Low Transmission Power. APCC.

[6] 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, 2–14.

[7] Md.I.Hussain N.Ahmed, H.Rahman. 2016. A comparison of 802.11ah and 802.15.4 for IoT. ICT Express.

[8] Q. Li S. Yin and O. Gnawali. 2015. Interconnecting wifi devices with ieee 802.15. 4 devices without using a gateway. DCOSS.

[9] B. Campbell J. Adkins N. Jackson T. Zachariah, N. Klugman and P. Dutta. 2015. The internet of things has a gateway problem.

[10] B. Kellogg S. Gollakota V. Iyer, V. Talla and J. R. Smith. 2016. Intertechnology backscatter: Towards internet connectivity for implanted devices. arXiv preprint arXiv:1607.04663.

[11] Song Min Kim Wenchao Jiang, Zhimeng Yin and Tian He. 2017. Transparent Cross-technology Communication over Data Traffic. INFOCOM.

[12] Y. Zhang and Q. Li. 2013. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. IEEE INFOCOM.

[13] Wei Zhou, Yan Jia, Yao Yao, Lipeng Zhu, Le Guan, Yuhang Mao, Peng Liu, and Yuqing Zhang. 2019. Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 1133–1150.