BLE Smart Tags and Sales in Physical Clothing Retail Markets

Yu-Hsuan Yeh
Department of Computer Science Univ. of Illinois at Urbana-Champaign
Email: yhyeh2@illinois.edu

Robin Kravets
Department of Computer Science Univ. of Illinois at Urbana-Champaign
Email: rhk@cs.uiuc.edu

***Abstract*—The rapid development of online shopping poses a major threat to physical retailers recently because of the low maintenance cost and variety of choices. But for clothing industry, it is the physical display that make retailers hold the significant advantage that attract customers to touch and try-on in store. This project aims to design an IoT sale system that benefits both customers and retailers and make the physical clothing retailers more competitive. Our methodology used Bluetooth Low Energy (BLE) wireless protocol and was implemented on Feather nRF52840 boards because BLE is battery friendly and can be received with smart phone. While customers walking in store and approach the target clothing, they will get notifications or promotions on mobile phone. For retailers, they can inventory the clothing when receiving the same BLE advertising signal. Moreover, in order to find proper transmit power and observe the influence of number of channels and the position of receivers, we simulated the high device density environment in store and evaluate with packets delivered rate and average received RSSI.**

***Keywords*—BLE, Retail, Clothing, Feather nRF52840, Internet of Things, Advertising, Passive Scanning**

I. Introduction

The development of various online shopping platforms poses a major threat to the physical retailers in recent years. Because of the low prices and transparent information for stocks, more and more customers tend to shop online instead of heading to a physical store. However, for clothing retailers, physical display and try-on are both quite significant services that online shops cannot provide, compared to grocery, electronics and other retail catagories. Thus, beside this natural advantage, how to reduce the maintenance cost and improve user experiences in store at the same time becomes a challenge for clothing retailers.

With the development of Internet of Things (IoT) technology, large physical retailers have introduced variety technologies like Radio Frequency IDentification (RFID) for inventory and checkout automation, Bluetooth Low Energy (BLE) for in-store localization and promotion, Computer Vision (CV) for misplacement detection and grab-and-go services. Some of these cases worked, but some of them failed. Because all of them have their own pros and cons, and most of them benefit either customers or retailers but not both. Therefore, the cost of build up the system end up became relatively higher than the revenue it brought, and only large retailers can afford this risk.

The proposed methodology in this project aims to figure out the win-win solution for both customers and retailers. It provides clothing information to both customers and retailers with one integrated system, so not only the user experiences can be improved but the labor cost for retailers can be reduced. We adapted the advantages of above wireless technologies and built a BLE advertising and passive scanning communication system. The concept is based on the idea of BLE Beacon because it can advertise actively and be received with smartphone. And the system further combined with the function of the inventory that RFID system provided.

In section III, we will introduce the communication architecture with three layers and how we utilize BLE advertising and passive scanning communication. Also, we described the functions the system would like to achieve, for example what kinds of the information have to send to customers and retailers in the limited packet size or what customer and retailer portal should response when they receive an advertising packet. More details and codes about the implementation on Feather nRF52840 boards are provided in section IV.

Considering the influence of high device density on packet transmission, we tried different combinations of transmission parameters, like the number of channels, transmit power, the position of receiver and the number of interference devices, to observe the influence. The analysis and figure of result are shown in section V. And the overall discussion and conclusion are places later in section VI.

II. Motivation and Background

In recent year, online shopping grows rapidly because it provides customer a convenient channel to buy whatever they want in a cheaper and more efficient way. This new retail ecology makes physical retailers facing a giant challenge because even they offer the physical display of goods, customer can still leverage this advantage and then order online when they back home. However, in clothing market, physical display and try-on are both quite significant services, compared to grocery and electronics. Even Amazon has promoted the Wardrobe services [1] that user can try-on at home and return those they don't like, but it also causes the cost of delivery, so I still think the physical store in clothing retail market is irreplaceable. Thus, how to persuade customer to make decision and purchase right in store turns out to be an important issue for retailers.

The main reasons that stop customers purchasing in-store could be: first, the price is higher than in the Internet; second, the information of items is not clear, including price, discount, stock availability, customer review, recommendation and so on. For the first problem, most of the physical retailers have come up with tons of promotion strategies to attract customers. This may increase the sale amount, but the revenue they get also decrease by offering discounts. Furthermore, online store can also cut down the price for this kind of competition. Actually, they do it all the time. Therefore, merely lower the price just hurts both physical and online retailers. Since online retail is born with low cost virtual stores, they can always provide better deal than physical stores. So, we have to figure out how to diminish the maintenance cost for physical retailers to make them competitive. For the second issue, it is the clear information that make online virtual shop persuade customers to buy successfully. The more information the customers can get, the higher possibility they pay. If we can build a more reliable system for revealing complete facts of items, it will also benefit physical retailers.

Nowadays, some leading retailers, for example, Decathlon, one of the largest sporting retailers from French, have tried using Radio Frequency IDentification(RFID) and robotic technologies [2] to help processing inventory works, which are usually done by human employees manually. RFID inventory system contains identification tags on items and readers on robots. When walking along aisles, the reader on the robot can identify the sequence number stored in each tag on an item from a distance. Unlike traditional barcode, it does not require line of sight. Another advantage which RFID is famous for is that the tag is battery-free and low-cost, so it can be sold with items to customers. However, the circuit inside the tag makes it hard to recycle, and the reader is relatively expensive compared to Wi-Fi router or Bluetooth devices. After the querying process, the robot will know whether the items are misplaced or out of stock and later notify the retailer. With this system, retailers can indeed cut down the cost of labors. But for customers, they still cannot access the data of items queried by RFID reader directly.

While other grocery retailers, like Walmart, have developed mobile application [3] to provide more information for items they have. Users can either search the items they want to buy, then get the aisle number to find the location in-store or they can scan the barcode on the items in-store to get detail information once the items are mislabeled or misplaced. This service requires retailers to maintain a database to manage all item information, passively waiting users' requests and then making responses.

On the contrary, some retailers deployed the Bluetooth Low Energy (BLE) Beacon advertising systems in-store [4][5] within a few years after Apple announce their iBeacon protocol in 2013. They installed palm-sized small Beacon devices on racks, advertising promotions actively toward customers' mobile phones nearby and navigate customers in-store by using wireless localization technology [6]. In the Generic Access Profile (GAP) layer in BLE protocol [7][8], there are 3 states: Idle, Discovery Mode and Connection Mode.

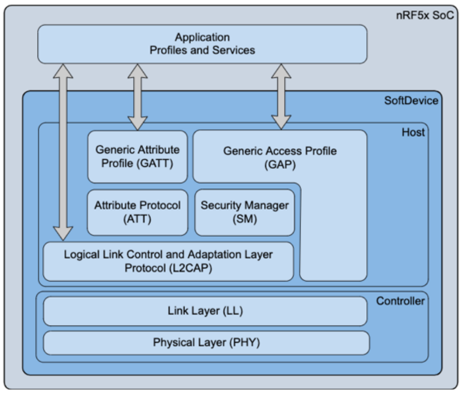

Figure 1. BLE Protocol Stack [7]

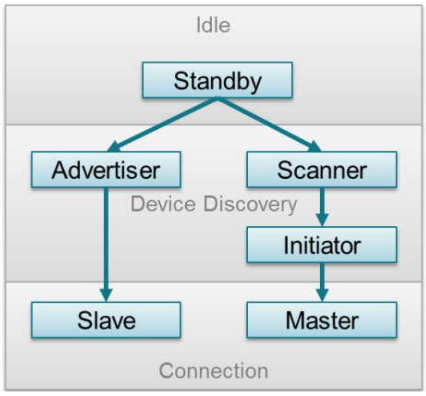

Figure 2. GAP State Diagram [8]

Beacon only works on BLE discovery mode, which is battery friendly and could be received directly by mobile phones without additional agent systems in between. The advertising packet format mainly contains Beacon prefix, UUID, Major, Minor, calibrated power. Therefore, when customers received the packet, the application can parse it by referring the format define in prefix, figure out the identification of the beacon devices using UUID, Major, Minor and finally estimate the proximity comparing the pre-calibrated power value with actual measured signal strength indicator. Mostly in retail application, UUID, Major, Minor, these three entries are encoded with the company name, branch location, aisle ID and rack ID. After receiving these IDs, the App still have to send request to central database to get corresponding information as the data length is limited to 31 bytes for advertising packet.

This project is inspired by the actively advertising functionality, low power consumption and mobile phone accessibility of BLE. Based on above technologies, we design a 3-layer decentralized BLE advertising system and encode clothing information within advertising packets. Not only customers can search for what they want, receive promotion and information in-store with cellphone, retailers can also inventory through the same system without using additional readers and servers. However, we use lots of advertising devices as tags attaching to all clothes, which will cause interference because the density of devices is pretty high in store [12]. So, we tried different combinations of transmission parameters, like the number of channels, transmit power, the position of receiver and the number of interference devices, to observe the influence after the system established.

III. Approach

The communications in the architecture are divided into 3 layers: user, controller and clothing, as figure 3 shown.

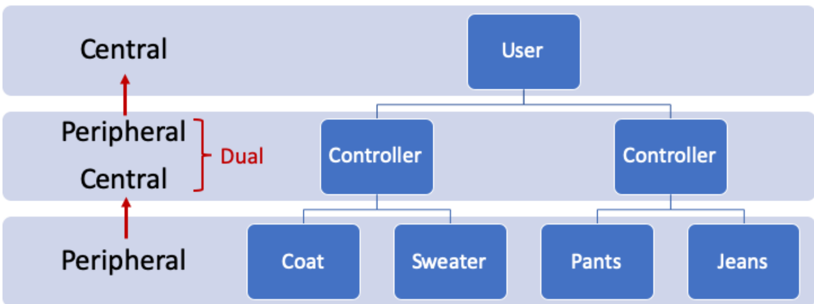

Figure 3. Communication Architecture

From the bottom, each clothing item is tagged with one advertiser, working in peripheral mode, advertising the information of that clothing. Therefore, we need to encode the information inside the packet, such as:

- sender and receiver (clothing/controller)
- target customer group(male/female/youth/kid)
- season

- category (coat/sweater/pants/jeans…)
- size
- color
- promotion type (70%off/buy one get one…)
- price
- item ID (distinguish item with same above characteristics)
- instance ID (distinguish item with same item ID)

In the middle layer, each controller will be in charge of a set of clothing, for example, a clothing rack or a showcase. They are similar to an agent working in dual mode in GAP, that is, to clothing they act as a scanner in central mode, but to users, they act as advertiser in peripheral mode. What controller do is collecting the advertising packets received from clothing, integrating these packets from same type and delivering only unique information to users. Basically, the advertising content is similar to the one from clothing, but since we integrate the same type of clothing, the instance ID column would be replaced with total quantity of instances under that controller. If we ignore the controller layer and connect the users and clothing directly, the users would be overwhelmed with duplicate packets, wasting computation sources and battery to filter packets out, since there will be a stack of same clothing on rack in most of the time.

For users in the top layer of architecture, they can be customers or employees in store. Both of them only have to hold a smart phone, walk around the store and receive the signals from controllers as a scanner in central mode. But the difference is the functions inside the App. For customers, the App can record the preferences of clothing style and size for the customers when they register or the App can analyze the browsing and purchasing history automatically, then predict the clothing that customers would like. Once the customers walk in store, pass through a rack and receive the advertising packets from controller, the App can compare the incoming clothing information with a list of preferred clothing stored in App. If the data match, the App can send a notification to customers and say "Hey, here is a skirt you might like! BOGO only today!". For employees, the App aims to inventory all the stocks, thus the App has to store which controller is responsible for which items. While the employees walk along aisles and receive the advertising from controllers, the App can get the remaining quantity of stocks and then match which item needs to reorder.

IV. Implementation

The BLE devices we use is Feather circuit board [9] constructed by Adafruit and using nRF52840 SoC developed by Nordic as shown in Figure 4. It supports BLE protocol and Arduino IDE with the library Bluefruit [11], which will be utilized in this project.

Figure 4. Feather nRF52840

But actually, Adafruit provides another programming language CircuitPython [10], based on traditional Python language but added more hardware supports and worked much faster than Arduino IDE. Compared to Arduino IDE, which compiles the C/C++ code first then uploaded to Feather, CircuitPython works more like storing the python script in the SRAM of Feather with native USB connected to our computers and then interpret directly. However, the Bluefruit library for CircuitPython is still under construction, so this project will be implemented in C/C++ and complied by Arduino IDE.

For 3 roles in the architecture, we have to specify the mode in setup function using:

```
Bluefruit.begin(1, 0); // for peripheral
Bluefruit.begin(0, 1); // for central
Bluefruit.begin(1, 1); // for dual
```

No matter in which mode the devices are, we can set address, name, TX power before advertising or scanning using:

```
Bluefruit.setAddr(myAddress);
Bluefruit.setName("BFItem......");
Bluefruit.setTxPower(8);
// Options: -40, -20, -16, -12, -8, -4, 0, +2, +3, +4, +5, +6, +7, +8 dBm
```

Notice that in the project, all device names are set to “BFItem” in the beginning to distinguish with other Bluetooth devices in the same environment.

For peripheral clothing devices, we encode the required information as the 6 bytes address of each devices, which is shown in figure 6. And then replace the default address in the very beginning of the code since it cannot be modified dynamically during runtime. We also found that the first byte of the address seems to be fixed after we verified by modifying each byte one by one. Therefore, the length we set is only 5 bytes. In addition, the address is in little endian, so it must be store and print in reverse order.

Bit	4	2	2	2	2	4	4	4	2	6	8	8
Address	C	1		7		6	3	0	8	F	0 B	0 3
Content	cannot modify	receiver	sender	group	season	category	size	color	promo	price(x3+5)	item ID	instance ID
Example	-	ctrl 0	item	female	winter	coat	M	red	XMAS	50	item No.11	the third coat in stock

Figure 5. Clothing Address Format

```
09:18:23.585 -> seq_num: 0
09:18:24.552 -> seq_num: 1
09:18:25.547 -> seq_num: 2
09:18:26.543 -> seq_num: 3
09:18:27.556 -> seq_num: 4
09:18:28.577 -> seq_num: 5
09:18:29.572 -> seq_num: 6
09:18:30.574 -> seq_num: 7
09:18:31.576 -> seq_num: 8
09:18:32.607 -> seq_num: 9
09:18:33.608 -> seq_num: 10
09:18:34.623 -> seq_num: 11
09:18:35.614 -> seq_num: 12
09:18:36.609 -> seq_num: 13
09:18:37.621 -> seq_num: 14
09:18:38.654 -> seq_num: 15
09:18:39.645 -> seq_num: 16
09:18:40.654 -> seq_num: 17
09:18:41.672 -> seq_num: 18
09:18:42.688 -> seq_num: 19
09:18:43.671 -> seq_num: 20
```

Figure 6. Send One Packet per Second

In order to observe the influence of device density, channel number and transmit power toward advertising processes, we have to add a sequence number to each transmitted advertising packet and advertise one packet per second. This means to change the advertising content in the period of advertising, which is prohibited in the library code. Thus, we forced the maximum advertising packet in one advertising period to be one and the result is shown in figure 5. Beside the sequence number that we use to monitor the packet loss, we have to configure which channel to advertise in library code. Notice that the channel mask is a 5-byte array set in reverse order and one bit represent the status of one channel (0 means on, 1 means off). For example, if we only transmit on channel 37, channel_mask[4] would be 0xC0. After the packet was sent, the adv_stop_callback function will be invoked. Inside the callback function, the process is delayed for one second, the advertising data will be cleared and reset to new content. The implementation details are shown in the following code:

```
// In Bluefruit lib: BLEAdvertising.cpp
ble_gap_adv_params_t adv_para = {
    .max_adv_evts = 1,
    .channel_mask = { 0, 0, 0, 0, 0 } // {7-0,15-8,23-16,31-24,39-32}
};

if (evt -> evt.gap_evt.params.adv_set_terminated.reason ==
BLE_GAP_EVT_ADV_SET_TERMINATED_REASON_LIMIT_REACHED){
    if (_stop_cb) ada_callback(NULL, 0, _stop_cb); // invoke stop callback
}

// In main function
Bluefruit.Advertising.setStopCallback(adv_stop_callback);

void adv_stop_callback(void) {
    Bluefruit.Advertising.clearData();
    // attach seq num in name string
    sprintf(newName, "BFItem%7d", ++seq_num); // string length must be fixed
    Bluefruit.setName((char const *) newName);
    Bluefruit.Advertising.addName();
    delay(1000); // millisec
    Bluefruit.Advertising.start();
```

For controllers, the structure of the code is similar to the combination of peripheral and central devices. In the central part, we also need to setup a callback function. When the scanner received a packet, the function will be invoked, process the incoming packet and do the inventory job as shown in pseudo code. Furthermore, we only use passive scanning mode, that is, the scanner only listens on what advertiser said and has no interaction with advertiser. Since currently the payload in passive mode is sufficient, we don't need the three-way handshake provided by active mode, which costs more time and energy.

```
Bluefruit.Scanner.setRxCallback(scan_callback);
Bluefruit.Scanner.useActiveScan(false);

// pseudo code
void scan_callback(ble_gap_evt_adv_report_t* report) {
    Filter packet by name starting with "BFItem";
    Filter packet by the receiver slot in address;
    Parse the name and get sequence number;
    clothing_data = address & 00:11:11:11:11:00; // mask irrelevant data
    instance_idx  = address & 00:00:00:00:00:11; // address is in readable order
    Count and list unique clothing_data with unique instace_idx as catalog.
}
```

In the advertising part, we forward the items in catalog sequentially and periodically through advertiser to users. Compared to clothing advertisers, which encode the data in their address, we encode the data in the similar format but send it through name string as the sequence number because the address cannot be modify dynamically even the advertising restarts. The only two differences in the address shown in figure 7 are: first, it will specify the controller to send so that the controller won't get the items next to it but not under its control; second, the instance ID will be change to the number of instances in-stock. The procedure in code is similar to clothing advertiser as well.

Bit	4	2	2	2	2	4	4	4	2	6	8	8
Address	C	0		7		6	3	0	8	F	0 B	0 A
Content	cannot modify	receiver	sender	group	season	category	size	color	promo	price(x3+5)	item ID	# instance in stock
Example	-	user	ctrl 0	female	winter	coat	M	red	XMAS	50	item No.11	total 10 coats in stock

Figure 7. Controller Advertising Format (Yellow part is different from clothing.)

For users, we implement the basic function prototype on Feather instead of developing a real App on the phone. The scanning process is the same as we mentioned in controller part. But the slightly difference in the scan_callback function is that we compare the received clothing data with predefined target clothing data and print notification once match.

To observe the packet transmission and determine the proper parameter to be used, we tried several combination of channel numbers, receiver position, TX power and the density of devices in the experiment. Imagine the scenario in clothing store as figure 8 shown, there will be clothing hanging on the rod but also stacks of folded clothing on the table. To simulate the deployment, we plan to do both at first. But actually, no matter which way the clothes are arrange, it's just the vertical and horizontal version correspondingly and won't affect the result too much. Moreover, when we put devices in the stack of clothing layer by layer, is not easy to see the LED and monitor the advertising status of devices as figure 9 recorded.

Figure 8. Scenario in Clothing Store

Figure 9. Stack of Clothes

Figure 10. Clothes hung on rod

Therefore, we decided to do experiments horizontal on a platform, put devices with a 3-inch interval in between to mimic the clothing hanging on a 30-inch long rod as shown in figure 10. From no interference device shown in figure 11, we increase the interference devices to 8 sequentially as in figure 13 and put the advertiser we want to observe at another end of the sequence of devices. For the receiver position, it could be place at the end point of the rod or be lifted over the platform for about 7 inches as shown in figure 12.

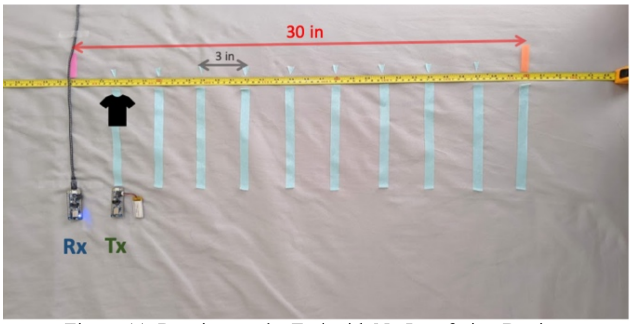

Figure 11. Receiver at the End with No Interfering Device

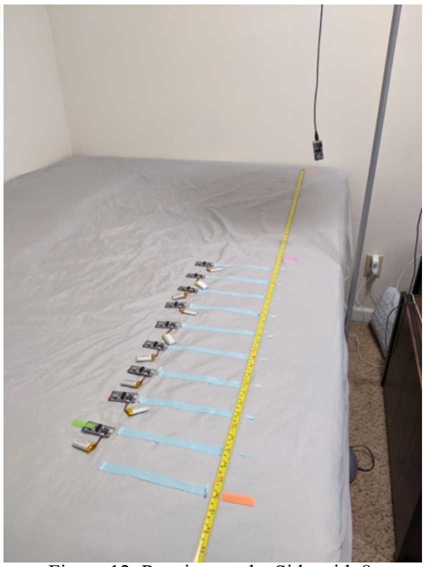
Figure 12. Receiver at the Side with 8 Interference Devices

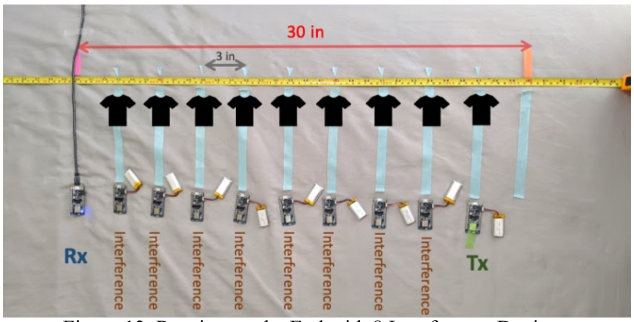

Figure 13. Receiver at the End with 8 Interference Devices

After the setup of environment, we extract the number of packets that scanner received within one minute and the average Received Signal Strength Indicator (RSSI) to do further analysis.

V. Evaluation

To analyze the data collected in the experiments, we plot the data rate (average number of received packets/minute) and average received RSSI corresponding to number of channels that used in transmission, receiver position and transmit power. The parameters we display here is mostly the boundary of the options provided on Feather board. For the number of channels, we tried advertising in default 3 Bluetooth channel 37, 38, 39, and only channel 37 as well. For the TX power, we tried the maximum 8 dBm and the minimum -40 dBm.

First, the channel is all opened and the higher TX power, the more advertising we sent successfully as shown in figure 14 and 15. If we use 8 dBm TX power, almost half of the packets can be transmitted. However, when reducing the number of channels from 3 to 1, as plotted in figure 16 and 17, we can see that the data rates decrease proportionally, regardless of the TX power and the position of the receiver. Next, we consider the position of the receiver. It is not clear that which side benefit the system most because when we place the receiver at the end, the performance for closer devices is pretty well but for those far away from the receiver, the performance will drop. But if we put the receiver at side, the performance is quite even, but not better than at the end. This might because the actual distance between TX and RX increased slightly since the shortest path is in the line of sight. Furthermore, for higher TX power, the influence for the position of the receiver become relatively small. And with higher TX power, even the number of interferences increased, the data rate doesn’t drop too much.

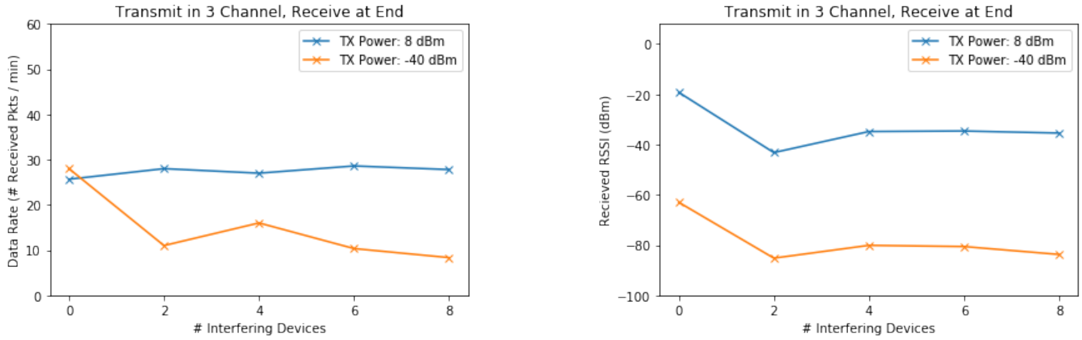

Figure 14. Transmit in 3 Channels, Receive at End (left: Data Rate, right: RSSI)

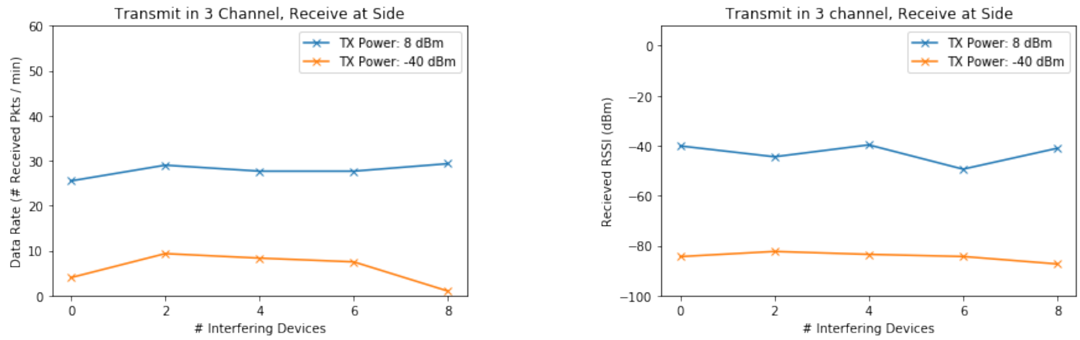

Figure 15. Transmit in 3 Channels, Receive at Side (left: Data Rate, right: RSSI)

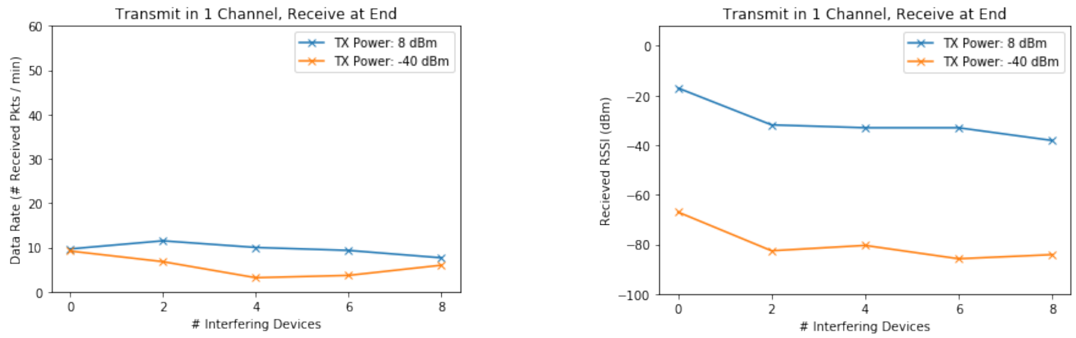

Figure 16. Transmit in 1 Channels, Receive at End (left: Data Rate, right: RSSI)

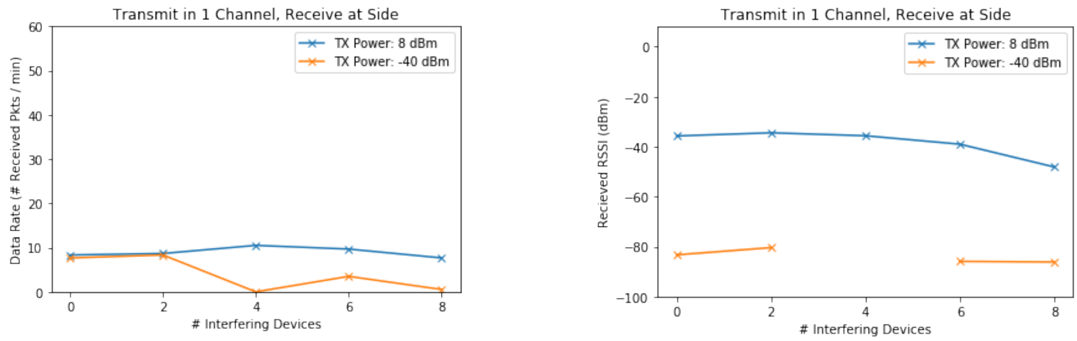

Figure 17. Transmit in 1 Channels, Receive at Side (left: Data Rate, right: RSSI)

VI. Conclusions and future directions

From the design prototype to the experiment that proof the influence of channel and density interference, I get to know more about the protocol of BLE and the pros and cons of it compared to other wireless technology. During the implementation of the system, I also learned how to trace and modify the firmware code for embedded Feather board and the library code developed by Adafruit. After the analysis and evaluation of experiment result, we realized the theorem is valuable but not enough for realistic cases. Because we know the density of the deployed devices will affect the performance, but in the result, we see that with enough TX power, the data rates are still pretty good despite the more power consumption. At first, I am not that familiar with the embedded system and wondering if I can achieve the functionality designed in my protocol on my own. But after several meetings and attempts, I found the way to do the experiment gradually. Even the presented result is just small part of the BLE domain, the experience of all these trials is more valuable.

In the ideal design, I believe this BLE system can improve the disadvantages of other existing wireless IoT systems to some extent. For example, our BLE clothing system benefits both customers and retailers, but existing RFID inventory system benefits only retailers and Beacon promotion system only benefits customers. The cost of each Feather board is not that cheap as an RFID tag while it is cheaper than a RFID reader and can be reused. But it is still not perfect because: First, we only used the passive scanning mode for controller and user, in other word, the interaction between users and system is only one-way. When the status of items change, like one shirt is taken away under the controller, but the shirt will not be aware of the fact that it is taken and notify its controller. The simplest way to deal this problem is to require the controller rescan all items periodically within short interval. But this might not be a good solution since the polling process will occupy part of the transmission channels. Therefore, we may try to use active scanning mode to deal this problem in the future. Second, even the users and retailers can receive the advertising packets directly using smart phone, the interface and the functions in App are also important. Especially for the customer portal, if the App can provide accurate recommendation and notify customers with proper frequency, it would help the whole system works better on user experience. Nevertheless, if the App fail to satisfy the demand of customers then it could ruin the whole system since on one would like to make good use of it. As a consequence, we have to treasure the feedback of users in the future as serious as the technical performance of our system.

This project is designed to help clothing retailers originally, but with adequate modifications, the system can also be utilized in most of the retail scenarios. I hope this can not only bring customers back to store but boost the surrounding economy as well in the future.

References

[1] Amazon Wardrobe, https://www.amazon.com/b/?_encoding=UTF8&bbn=14807110011&node=16122413011&ref_=sv_tbyb_l0_1, June 2017

[2] Simbe Robotics, https://www.simberobotics.com/

[3] Walmart Mobile App, https://www.walmart.com/cp/walmart-mobile-app/1087865

[4] Nikita Adkar, Akshata Talele, Chinmayee Mundhe, Anita Gunja, "Bluetooth Beacon Applications in Retail Market", International Conference On Advances in Communication and Computing Technology (ICACCT), 2018

[5] Ravi Ramakrishnan, Loveleen Gaur, Gurinder Singh, "Feasibility and Efficacy of BLE Beacon IoT Devices in Inventory Management at the Shop Floor", International Journal of Electrical and Computer Engineering (IJECE), Vol. 6, No. 5, pp. 2362~2368, 2016

[6] Sudarshan S. Chawathe, "Beacon Placement for Indoor Localization using Bluetooth", 11th International IEEE Conference on Intelligent Transportation Systems, 2018

[7] Nordic Semiconductor, Bluetooth Low Energy protocol stack, https://infocenter.nordicsemi.com/index.jsp?topic=%2Fsds_s140%2FSDS%2Fs1xx%2Fble_protocol_stack%2Fble_protocol_stack.html

[8] Texas Instruments, BLE Generic Access Profile, http://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_40_00_45/docs/blestack/ble_user_guide/html/ble-stack-3.x/gap.html, 2016

[9] Adafruit Feather nRF52840, https://learn.adafruit.com/introducing-the-adafruit-nrf52840-feather/overview

[10] CircuitPython, https://learn.adafruit.com/welcome-to-circuitpython/overview

[11] Bluefruit52Lib, Adafruit, https://github.com/adafruit/Adafruit_nRF52_Arduino/tree/master/libraries/Bluefruit52Lib

[12] Albert F. Harris III, Vansh Khanna, Guliz Tuncay, Roy Want, Robin Kravets, "Bluetooth Low Energy in Dense IoT Environments", IEEE Communications Magazine, Vol.54, pp.30-36, 2016