COMP9336 – Mobile Data Networking

Lab 5 - Observation of Bluetooth Low Energy Frequency Hopping

T2 2022

Name: Yuhua Zhao – **ZID:** z5404443

Date: 29/06/2022

Procedure

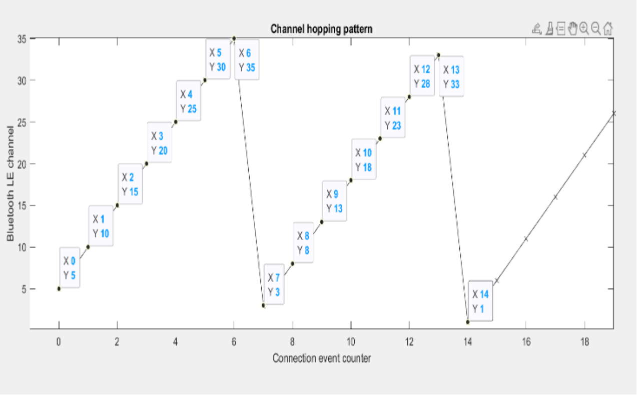
- Create account by using the following link
  - https://au.mathworks.com/login
- After finish creation by using Uni Email, you will be able to download the Mathlab
- Add-on (Mathlab)
  - Communication toolbox
  - Bluetooth Box
- Use the following command to use the Bluetooth API

```
openExample('bluetooth/BLEChannelHoppingExample')
```

Task 1:

Explain Algorithm #1’s calculations and equation when *unmappedChannel* is a good channel. (1 mark)

$$\boxed{f_{k+1} = (f_k + h)\ mode\ 37}$$



- Initial Parameters

○ Algorithm: Channel Selection Algorithm #1

○ Hop increment: 5

○ Channel Map: unmapped Channel is good channel. (All good channel)

○ Number of channel hops to generate: 20

The process of calculation:

Counter	f_k(the Last hop value)	h(Hop)	Detail	Calculation Result
0	0	5	(0+5) mod 37	5
1	5	5	(5+5) mod 37	10
2	10	5	(10+5) mod 37	15
3	15	5	(15+5) mod 37	20
4	20	5	(20+5) mod 37	25
5	25	5	(25+5) mod 37	30
6	30	5	(30+5) mod 37	35
7	35	5	(35+5) mode 37	3
…	…	…	…	…
19	18	5	(21+5) mod 37	26

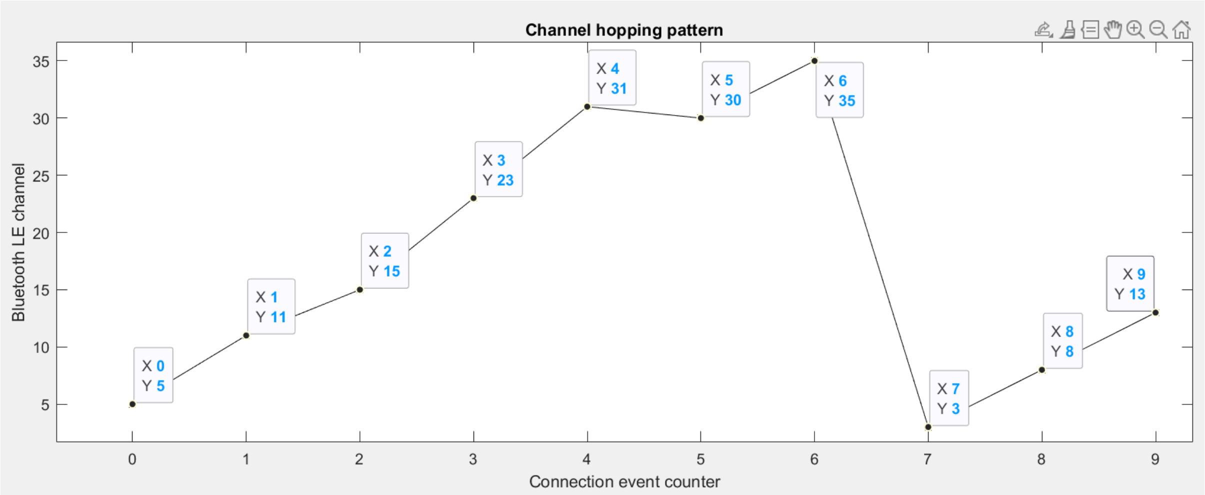
The calculation is simply using the previous calculation result to add on the hop value and mod 37.

For Good Channel, since it won’t encounter any bad channel, it is simply use the Algorithm #1 Algorithm, the h is the Hop Increment (Fix value) and the f_k is the last hop value (Initial f_k is 0).

Task 2:

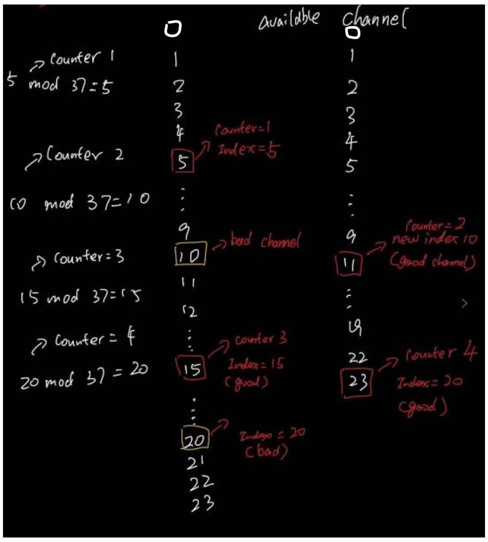
Explain algorithm #1’s calculations and equation when *unmappedChannel* is a bad channel.

$$\boxed{f_{k+1} = (f_k + h)\ mode\ 37}$$



- Initial Parameters
    - Algorithm: Channel Selection Algorithm #1
    - Hop increment: 5
    - Channel Map: 10, 20, 21, 25, 26, 27 are bad channels.
    - Number of channel hops to generate: 10 (Task 1 is 20, but 10 is enough for explanation)

Counter	f_k	Detail	Calculation Result
0	0	(0+5) mod 37	5
1	5	(5+5) mod 37 + 1	10 (Bad channel -> search from available channel) -> 11
2	10	(10+5) mod 37	15
3	15	(15+5) mod 37	20 (Bad channel -> search from available channel) -> 23
4	20	(20+5) mod 37	25 (Bad channel -> search from available channel) -> 31
5	25	(25+5) mod 37	30
6	30	(30+5) mod 37	35
7	35	(35+5) mode 37	3



As shown above, when counter = 1, the channel hopping should be channel 10, but if channel 10 is a bad channel, it will search from the available channel list (index 10) which becomes channel 11. When counter = 3, it will search normally. When index 4, Channels 20 and 21 are bad channels then it will again search from the available list Index 15 -> 23. (As shown on the left).

This is hard to descript in words, but I draw a list of the table that shows the process. The same Logic shows apply to the rest of the calculation.

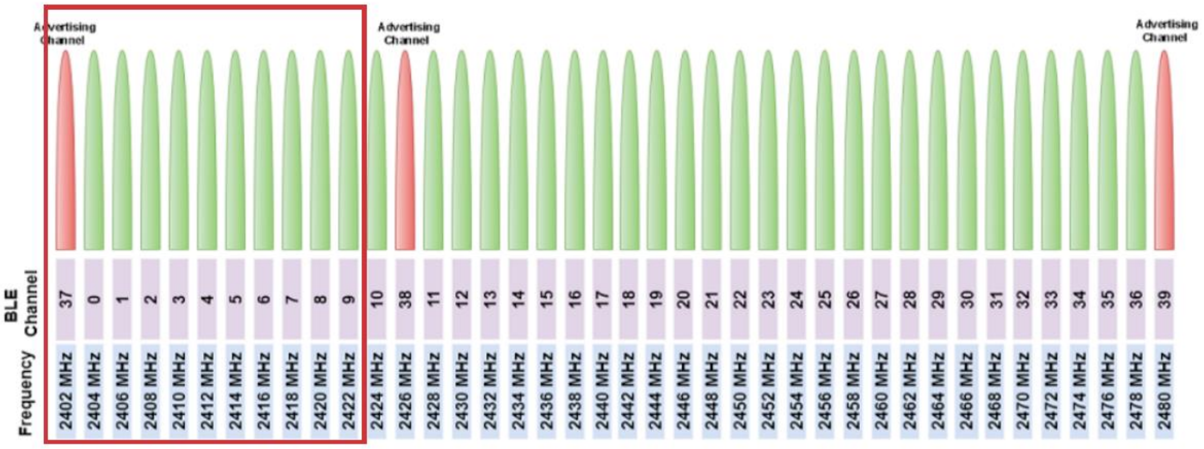
Note: Minimum of available Bluetooth channel is 20 (according to the lecture)

Yuhua Zhao
Z5404443

Task 3:

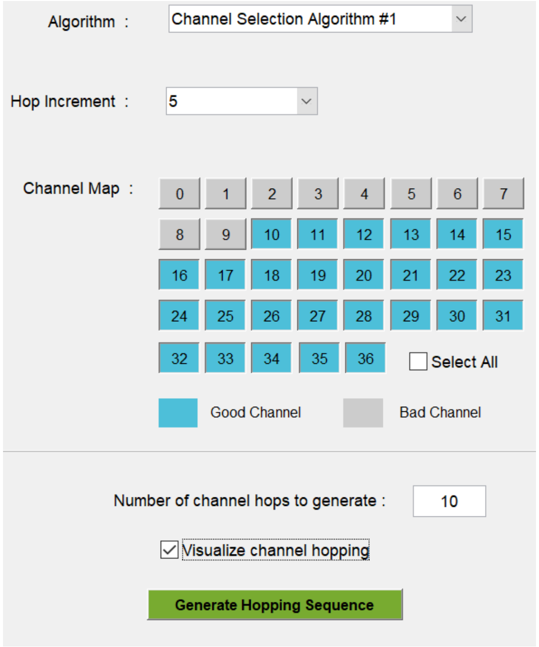
Use the “graphical user interface” in the example to run algorithm #1. Now consider that, in the vicinity of the Bluetooth network, a 2.4GHz WLAN is operating on WiFi Channel 1. Work out and select the good and bad channels for this scenario. Select a hop increment of your choice, set the number of channel hops to 10, tick the Visualization and generate the hopping sequence for these 10 hops. Add the resulting figures to your report and explain the figures in one paragraph. (2 marks)

Since WiFi and Bluetooth are using the same 2.4GHz frequency, which means that the overlapping channel interferes with each other. According to my research [Reference 2], The 2.4 GHz WIFI Channel 1’s frequency is between 2401 MHZ to 2423 MHZ. This means it overlaps with the Bluetooth Channel 0 to 9 and Channel 37 (), the overlaps channels consider Bad Channels.


Advertising Channel
Advertising Channel
Advertising Channel
BLE Channel
Frequency
37 0 1 2 3 4 5 6 7 8 9 10 38 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 39
2402 MHz 2404 MHz 2406 MHz 2408 MHz 2410 MHz 2412 MHz 2414 MHz 2416 MHz 2418 MHz 2420 MHz 2422 MHz 2424 MHz 2426 MHz 2428 MHz 2430 MHz 2432 MHz 2434 MHz 2436 MHz 2438 MHz 2440 MHz 2442 MHz 2444 MHz 2446 MHz 2448 MHz 2450 MHz 2452 MHz 2454 MHz 2456 MHz 2458 MHz 2460 MHz 2462 MHz 2464 MHz 2466 MHz 2468 MHz 2470 MHz 2472 MHz 2474 MHz 2476 MHz 2478 MHz 2480 MHz

As shown above, the red rectangle labelled the bad channels.

- Initial Parameters
  - Algorithm: Channel Selection Algorithm #1
  - Hop increment: 5 (random)
  - Channel Map: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 are bad channels.
  - Number of channel hops to generate: 10 (As Required)



Algorithm : Channel Selection Algorithm #1

Hop Increment : 5

Channel Map :

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	□ Select All		

Good Channel   Bad Channel

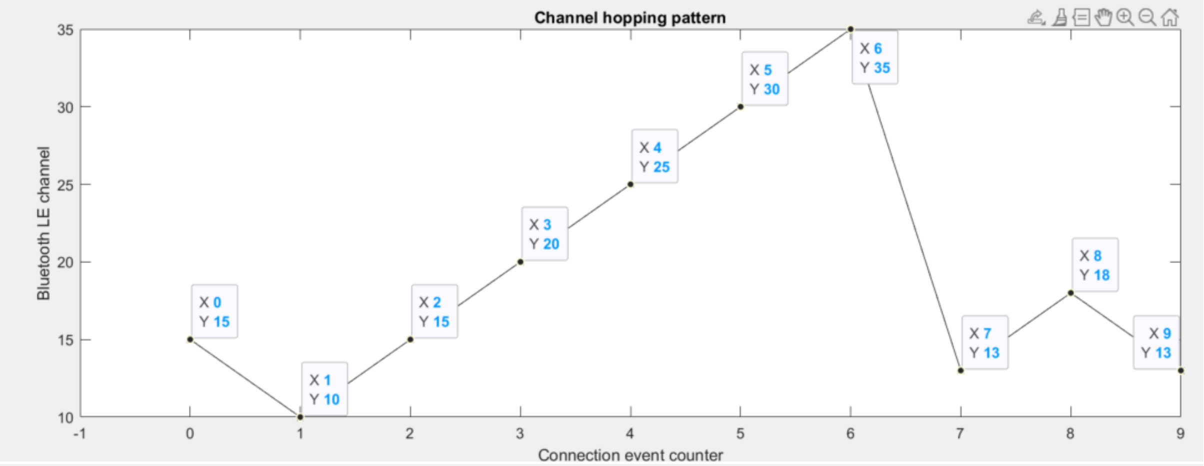
Number of channel hops to generate : 10

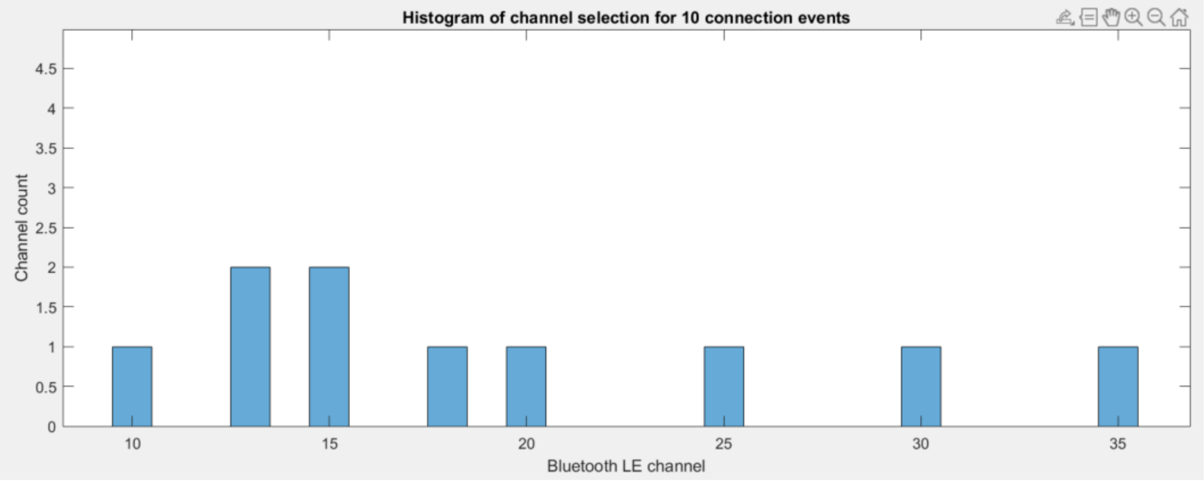
☑ Visualize channel hopping

According to the Task 2 Explanation, when X = 0, it looks for Index 5 due to hop increment is 5 which support to be Channel 5, but channel 5 is a bad channel which search from the available channel and Index 5 is 15. When X = 1, the index 10 of channel is 10 is a good channel, that’s why y = 10. Same Logic apply to the rest of it. (Table below shows the steps)

Counter	f_k	Detail	Calculation Result
0	0	(0+5) mod 37	5 (Bad channel, search from available channel list) -> 15
1	5	(5+5) mod 37 + 1	10 (Good Channel)
2	10	(10+5) mod 37	15
3	15	(15+5) mod 37	20
4	20	(20+5) mod 37	25
5	25	(25+5) mod 37	30
6	30	(30+5) mod 37	35
7	35	(35+5) mode 37	3 (Bad channel, search from available channel list) -> 13
8	3	(3+5) mode 37	8 (Bad channel, search from available channel list) -> 18
9	8	(8+5) mode 37	13 (Good Channel)

When Counter = 7, 40 mod 37 = 3, so we are looking for Index 4 from the list, but Channel 3 is occupied, so search from the available (used) list which is 13. When Counter is 8, 3 + 5 = 8 but the channel 9 is also occupied, so its Y become 18. When X = 9, 8 + 3 = 12 and the index 12 is available so back to become 13.





Reference (For my own):

- (Reference 1): Bluetooth Frequency Hopping:
  - https://blog.csdn.net/weixin_42583147/article/details/82623805

- (Reference 2): 2.4GHz WIFI frequency
  - https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/channels-frequencies-bands-bandwidth.php