Logo, company name

Description automatically generated

**COMP9336 – Mobile Data Networking**

**Lab 4 - WIFI API**

**T2 2022**

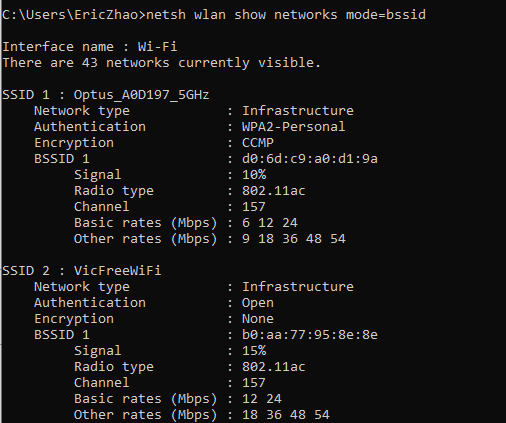
**Name:** Yuhua Zhao – **ZID:** z5404443

**Date**: 23/06/2022

**Disclaimer:**

* Due to a hardware issue, my laptop cannot obtain the WIFI Frequency (2.4 & 5GHz) from the WIFI API. So, I use Channel to determine whether it's 2.5 or 5GHz. (More discussion is shown below)
* My laptop cannot obtain the signal strength directly, I used the Signal in Percentage to dBm function to do the conversion provided by Tutor Rui.
* Juypter Notebook will be submitted with all the code and output that I performed. (Feel free to have a look)

**Task 1 – WIFI API Information extraction:**

****

This WIFI API can be extract the neighbour SSID information include:

* + SSID name
  + Authentication Method of Specific SSID
  + Encryption method
  + BSSID Information (Signal, Radio Type and so on)
    - Within the same network, if there are multiple Access Points exist, there will be multiple BSSID existing under the same SSID.

**Note:**

* This scan perform after the coding is completed (same location), some of the SSID may not be shown on the table that I created for task 2.

**Task 2 – Counting Surrounding Devices and distance estimation:**

**Program design:**

1. Run the following command to get the Neighbour SSID information.

(Same as running “netsh wlan show networks mode=BSSID” on Window 10 OS)

****

1. After retrieving the data, I split the String base on SSID. For example, if there are two SSID discovered, each SSID will be separated as String, basically listing the data structure with two strings.

Text

Description automatically generated

1. After splitting the SSIDs, the string of each SSID will feed into the function of “get\_SSID\_Info()” to store the SSID, BSSID, Signal, Channel information, and so on according. In my case, I use Dictionary to store the SSID information such as SSID name, Authentication method. As mentioned above, if there are multiple APs for the same SSID, the BSSID will be under the SSID section, since I use List to store each BSSID with the corresponding signal, channel, Radio Type, and so on.

Text

Description automatically generated**In each BSSID Array**

Index 0: BSSID

Index 1: Signal

Index 2: Radio Type

Index 3: Channel

Index 4: Basic Rates (Mbps)

Index 5: Other Rates (Mbps)

Text

Description automatically generated

1. A couple Function is created for different purposes:

|  |  |
| --- | --- |
|  | Use the Channel number to determine which frequency is used.  **Note:** The 2.4 and 5GHz channels are not overlapped according to the Australia Wireless Frequency usage standard. |
|  | This algorithm is provided by Tutor (Rui) to convert Signal in Percentage to dBm. |
|  | The FT\_2\_4 and the FT\_5 show the exact frequency usage for each channel for 2.4GHz and 5GHz.  (Link: <https://en.wikipedia.org/wiki/List_of_WLAN_channels> )  The estimated Distance calculation will be using “Free Space Loss Path equation” |

1. Text, letter

   Description automatically generatedPerform calculations and output as a Table.

**Result:**

Table

Description automatically generated

(This Screenshot only shows part of the table, more info can be obtain on the Jupyter Notebook)

1. Same SSID shows in a row with a Different BSSIDs, which means that for the same SSID, there are a number of WIFI Access Points.
2. If the SSID is blank, it means that the SSID is hidden but can be able to be discovered by the WIFI API.
3. The Estimated Distance is calculated by using the “Free Space Loss Path equation”. As we can see some of the estimated Distances are very large (over 300m). There are two reasons cause that. Firstly, it’s because the obstacle blocks the signal transmission, or the Receiver only receives the reflected signal that causes the estimated Distance larger than the actual distance. The secondary is because the Signal Percentage converted to dBm is a rough calculation that may have deviation.