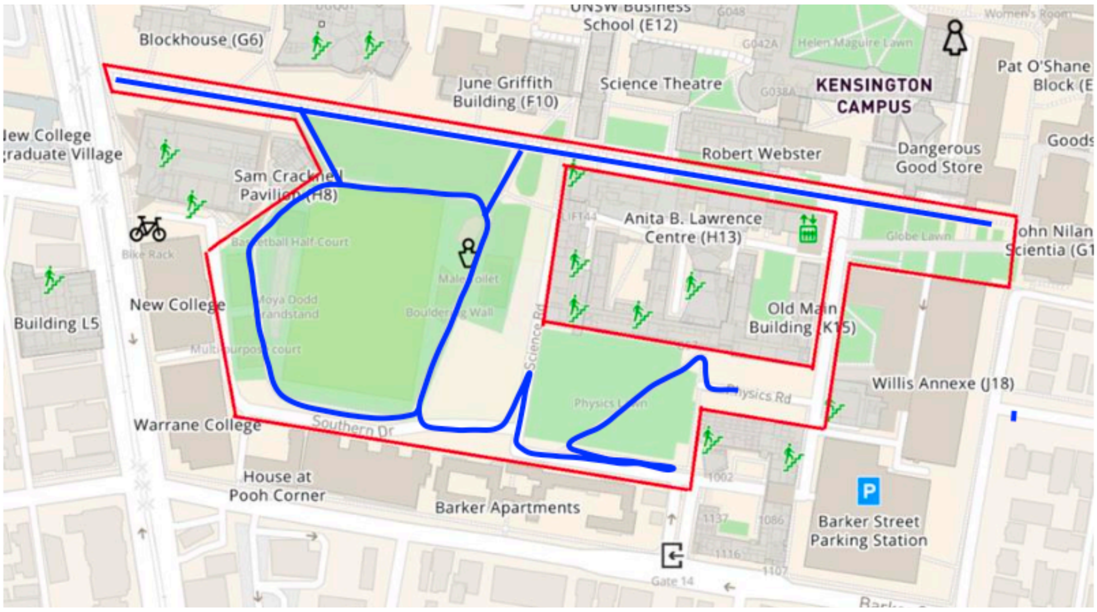
Phase 1 Report

Data Collection

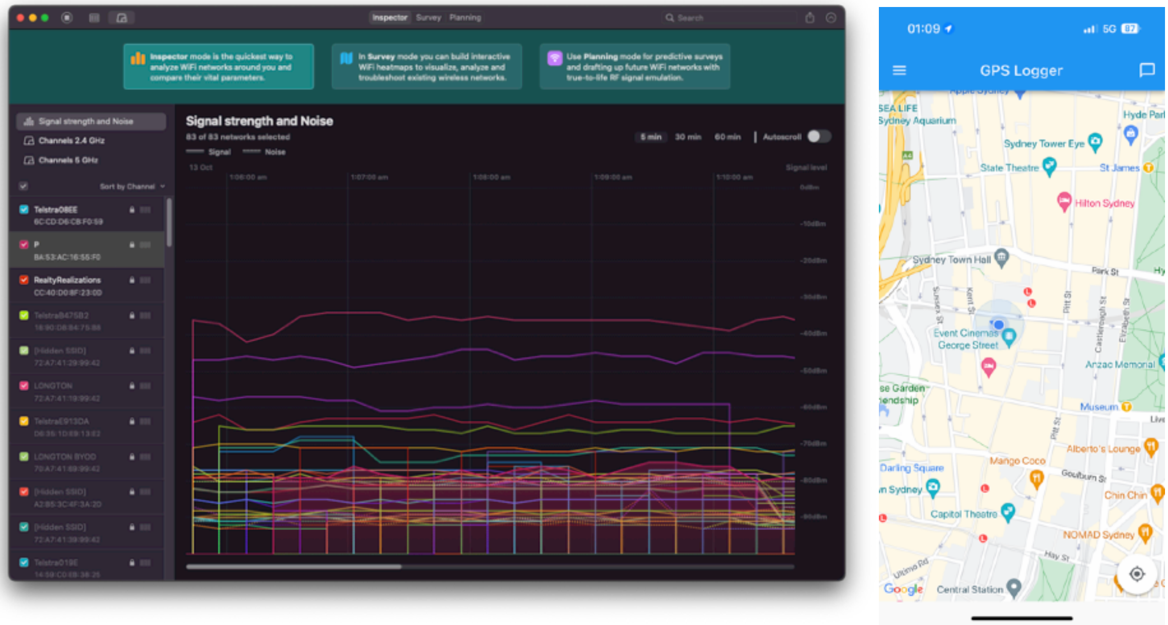
3 periods of wifi data collection, 5/10/23 13:16-14:00, 11/10/23 17:14-17:42, 11/10/23 20:05-20:38, was performed at UNSW Kensington campus with collection route marked by blue line in the figure below.

Fig 1. *Data collection map*



Wifi data was collected using inspector function in Netspot application running on MacOS laptop. GPS data was collected using GPS Logger application running on iPhone. Network latency data was collected using python script. Each application used generated csv files that need to be processed and combined

Fig 2. *Netspot application and GPS logger application used for data collection*


Inspector mode is the quickest way to analyze WiFi networks around you and compare their vital parameters.

In Survey mode you can build interactive WiFi heatmaps to visualize, analyze and troubleshoot existing wireless networks.

Use Planning mode for predictive surveys and drafting up future WiFi networks with true-to-life RF signal emulation.

Signal strength and Noise

83 of 83 networks selected

Sort by Channel

TelstraO8EE
6C:CD:D6:CB:F0:59

P
BA:53:AC:16:55:F0

RealtyRealizations
CC:40:D0:8F:23:0D

Telstra8475B2
18:90:D8:84:75:88

[Hidden SSID]
72:A7:41:29:99:42

LONGTON
72:A7:41:19:99:42

TelstraE913DA
D8:35:1D:E9:13:E2

LONGTON BYOD
70:A7:41:69:99:42

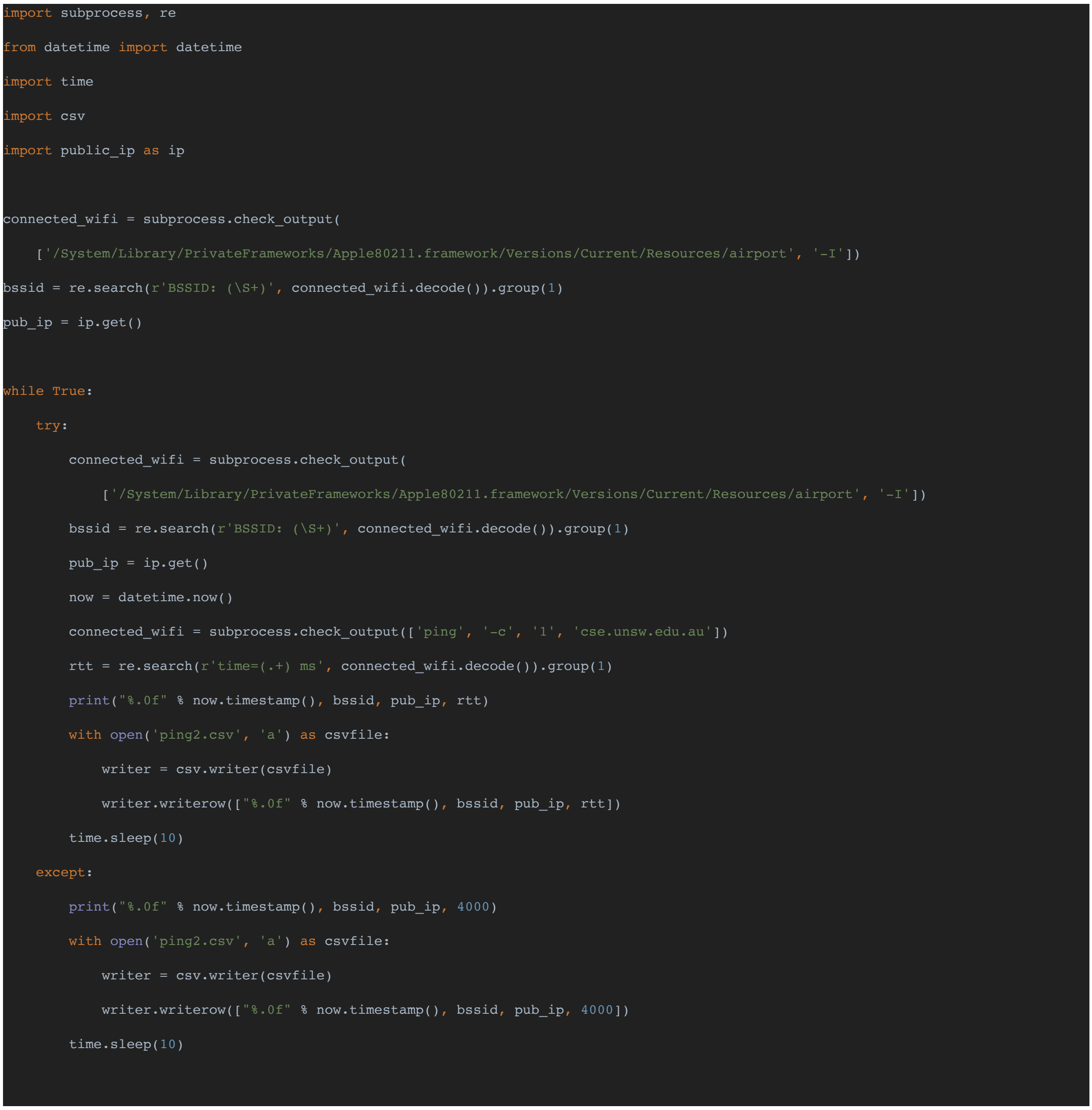
[Hidden SSID]
A2:85:3C:4F:3A:2D

[Hidden SSID]
72:A7:41:39:99:42

Telstra019E
14:59:C0:E8:38:25

GPS Logger

Fig 3. *Python script used for network latency data collection*



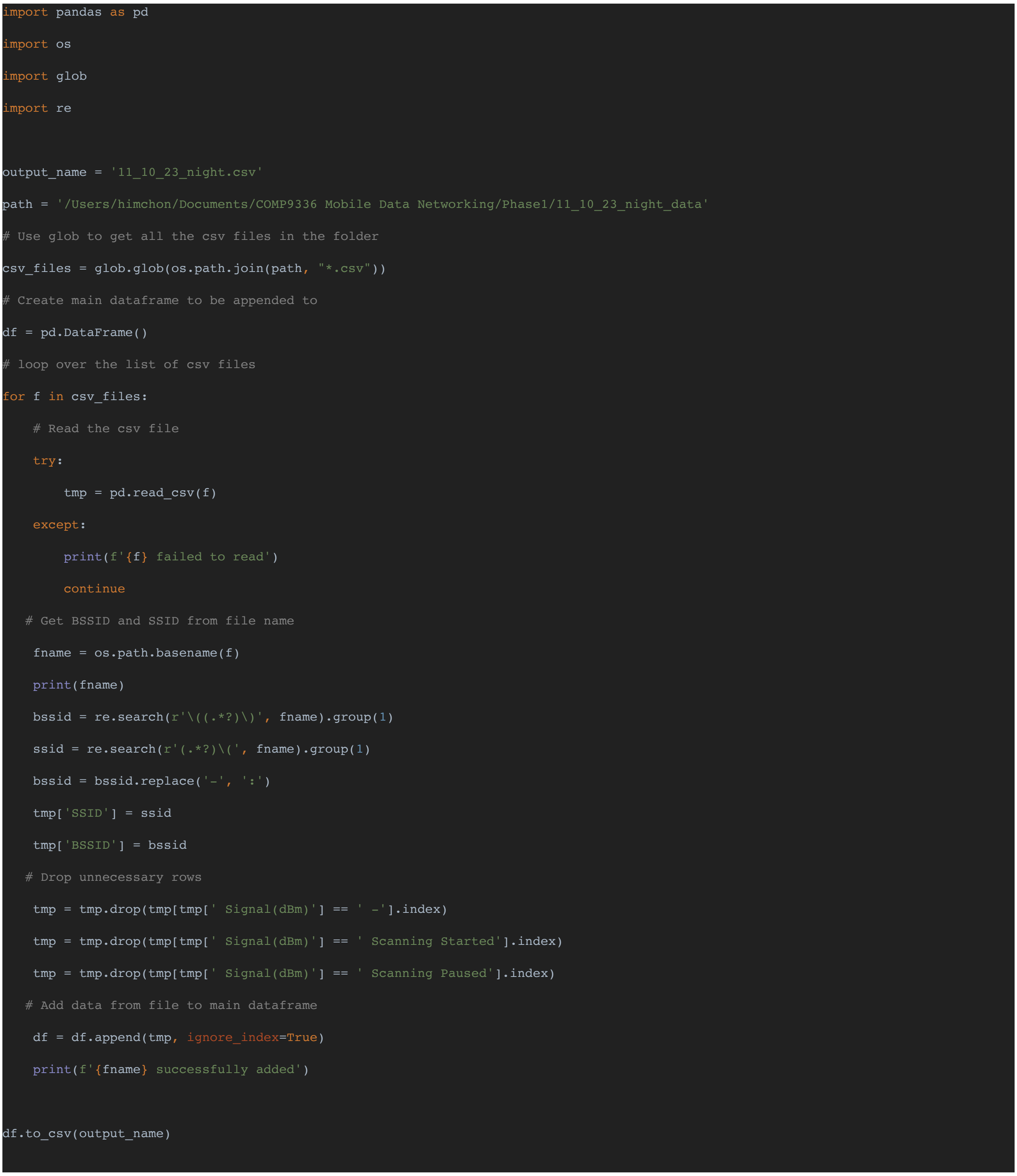
```python
import subprocess, re
from datetime import datetime
import time
import csv
import public_ip as ip

connected_wifi = subprocess.check_output(
    ['/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport', '-I'])
bssid = re.search(r'BSSID: (\S+)', connected_wifi.decode()).group(1)
pub_ip = ip.get()

while True:
    try:
        connected_wifi = subprocess.check_output(
            ['/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport', '-I'])
        bssid = re.search(r'BSSID: (\S+)', connected_wifi.decode()).group(1)
        pub_ip = ip.get()
        now = datetime.now()
        connected_wifi = subprocess.check_output(['ping', '-c', '1', 'cse.unsw.edu.au'])
        rtt = re.search(r'time=(.+) ms', connected_wifi.decode()).group(1)
        print("%.0f" % now.timestamp(), bssid, pub_ip, rtt)
        with open('ping2.csv', 'a') as csvfile:
            writer = csv.writer(csvfile)
            writer.writerow(["%.0f" % now.timestamp(), bssid, pub_ip, rtt])
        time.sleep(10)
    except:
        print("%.0f" % now.timestamp(), bssid, pub_ip, 4000)
        with open('ping2.csv', 'a') as csvfile:
            writer = csv.writer(csvfile)
            writer.writerow(["%.0f" % now.timestamp(), bssid, pub_ip, 4000])
        time.sleep(10)
```

Phase 1 Report

Data Processing

For each data collection period, wifi data is exported from Netspot application, resulting thousands of csv files separated by BSSID. A python script is used to combine and process them into a single csv file.

Fig 4. *Python script used to combine Netspot application outputs*

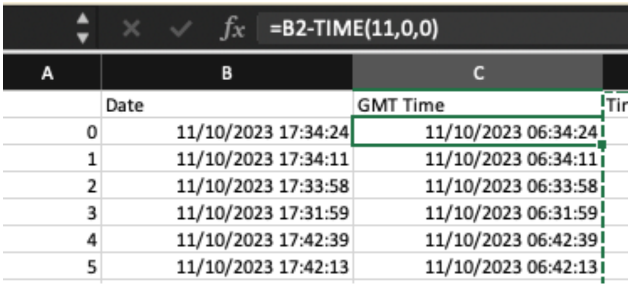


```python
import pandas as pd
import os
import glob
import re

output_name = '11_10_23_night.csv'
path = '/Users/himchon/Documents/COMP9336 Mobile Data Networking/Phase1/11_10_23_night_data'
# Use glob to get all the csv files in the folder
csv_files = glob.glob(os.path.join(path, "*.csv"))
# Create main dataframe to be appended to
df = pd.DataFrame()
# loop over the list of csv files
for f in csv_files:
    # Read the csv file
    try:
        tmp = pd.read_csv(f)
    except:
        print(f'{f} failed to read')
        continue
   # Get BSSID and SSID from file name
    fname = os.path.basename(f)
    print(fname)
    bssid = re.search(r'\((.*?)\)', fname).group(1)
    ssid = re.search(r'(.*?)\(', fname).group(1)
    bssid = bssid.replace('-', ':')
    tmp['SSID'] = ssid
    tmp['BSSID'] = bssid
   # Drop unnecessary rows
    tmp = tmp.drop(tmp[tmp[' Signal(dBm)'] == ' -'].index)
    tmp = tmp.drop(tmp[tmp[' Signal(dBm)'] == ' Scanning Started'].index)
    tmp = tmp.drop(tmp[tmp[' Signal(dBm)'] == ' Scanning Paused'].index)
   # Add data from file to main dataframe
    df = df.append(tmp, ignore_index=True)
    print(f'{fname} successfully added')

df.to_csv(output_name)
```

Phase 1 Report

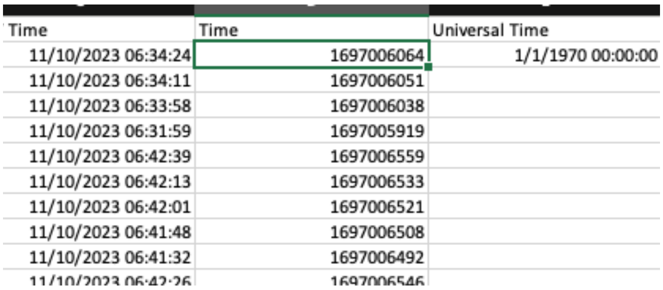
Microsoft Excel was used to process data into the required format as well as joining wifi data, GPS data, and network latency data together.

Date-time was converted to GMT date-time and, using appropriate formula, then unix timestamp.



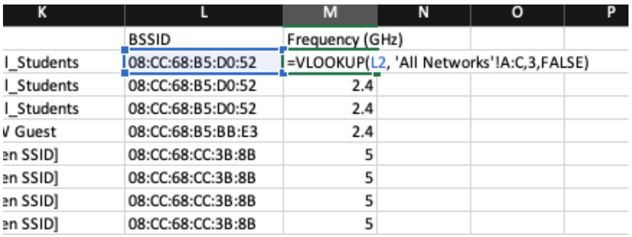
=B2-TIME(11,0,0)

A	B	C	
	Date	GMT Time	Tir
0	11/10/2023 17:34:24	11/10/2023 06:34:24	
1	11/10/2023 17:34:11	11/10/2023 06:34:11	
2	11/10/2023 17:33:58	11/10/2023 06:33:58	
3	11/10/2023 17:31:59	11/10/2023 06:31:59	
4	11/10/2023 17:42:39	11/10/2023 06:42:39	
5	11/10/2023 17:42:13	11/10/2023 06:42:13	

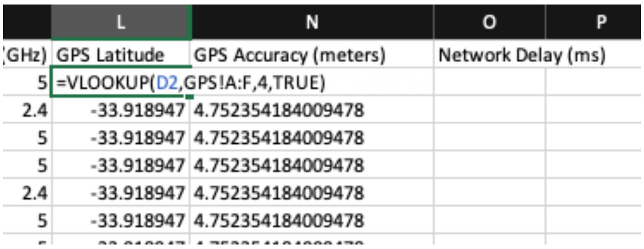


Time	Time	Universal Time
11/10/2023 06:34:24	1697006064	1/1/1970 00:00:00
11/10/2023 06:34:11	1697006051	
11/10/2023 06:33:58	1697006038	
11/10/2023 06:31:59	1697005919	
11/10/2023 06:42:39	1697006559	
11/10/2023 06:42:13	1697006533	
11/10/2023 06:42:01	1697006521	
11/10/2023 06:41:48	1697006508	
11/10/2023 06:41:32	1697006492	
11/10/2023 06:42:26	1697006546	

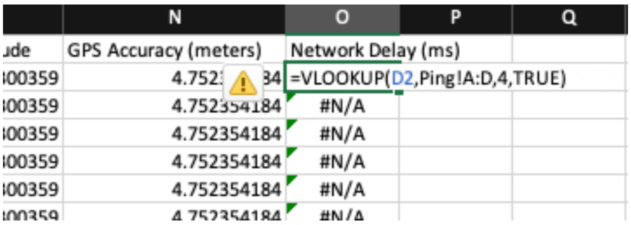
VLOOKUP function was used to join wifi data with data that exist on different table, including frequency, GPS data, and network latency.



K	L	M	N	O	P
	BSSID	Frequency (GHz)			
l_Students	08:CC:68:B5:D0:52	=VLOOKUP(L2, 'All Networks'!A:C,3,FALSE)			
l_Students	08:CC:68:B5:D0:52	2.4			
l_Students	08:CC:68:B5:D0:52	2.4			
V Guest	08:CC:68:B5:BB:E3	2.4			
en SSID]	08:CC:68:CC:3B:8B	5			
en SSID]	08:CC:68:CC:3B:8B	5			
en SSID]	08:CC:68:CC:3B:8B	5			
en SSID]	08:CC:68:CC:3B:8B	5			

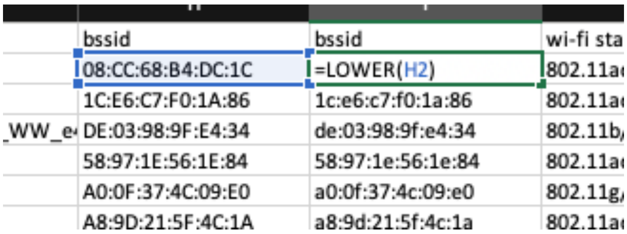


	L	N	O	P
(GHz)	GPS Latitude	GPS Accuracy (meters)	Network Delay (ms)	
5	=VLOOKUP(D2,GPS!A:F,4,TRUE)			
2.4	-33.918947	4.752354184009478		
5	-33.918947	4.752354184009478		
5	-33.918947	4.752354184009478		
2.4	-33.918947	4.752354184009478		
5	-33.918947	4.752354184009478		

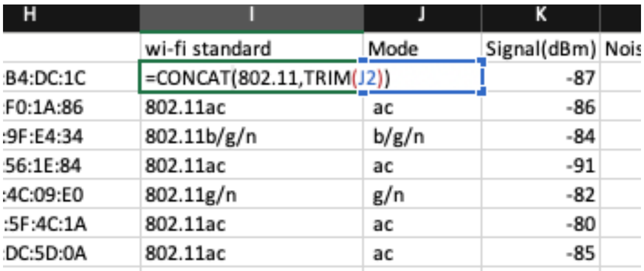


	N	O	P	Q
ude	GPS Accuracy (meters)	Network Delay (ms)		
100359	4.752⚠34	=VLOOKUP(D2,Ping!A:D,4,TRUE)		
100359	4.752354184	#N/A		
100359	4.752354184	#N/A		
100359	4.752354184	#N/A		
100359	4.752354184	#N/A		
100359	4.752354184	#N/A		

Resulting tables from different collection period are then combined. A few more columns were added and edited.



	H	I	
	bssid	bssid	wi-fi sta
	08:CC:68:B4:DC:1C	=LOWER(H2)	802.11ac
	1C:E6:C7:F0:1A:86	1c:e6:c7:f0:1a:86	802.11ac
_WW_e	DE:03:98:9F:E4:34	de:03:98:9f:e4:34	802.11b/
	58:97:1E:56:1E:84	58:97:1e:56:1e:84	802.11ac
	A0:0F:37:4C:09:E0	a0:0f:37:4c:09:e0	802.11g/
	A8:9D:21:5F:4C:1A	a8:9d:21:5f:4c:1a	802.11ac



H	I	J	K	
	wi-fi standard	Mode	Signal(dBm)	Nois
:B4:DC:1C	=CONCAT(802.11,TRIM(J2))		-87	
:F0:1A:86	802.11ac	ac	-86	
:9F:E4:34	802.11b/g/n	b/g/n	-84	
:56:1E:84	802.11ac	ac	-91	
:4C:09:E0	802.11g/n	g/n	-82	
:5F:4C:1A	802.11ac	ac	-80	
:DC:5D:0A	802.11ac	ac	-85	