Taste of Research
Gough Yumu LUI
Engineer's Log Book

Week 2

- Monday 22nd November 2010

Saw FM-Assist to get access to 401A. At first, they said the print-out of the e-mail was insufficient. Then they relented and discovered that I already had access. They re-encoded my card and I have physical swipe access to the room now. Unfortunately, while I have a zpass account, I can't seem to login to the SOE PC's in that room. I dusted off my hx4700 PDA and installed ministumbler but found that the software was not compatible with the onboard card (ACX110 based). Thought about possibility of wireless channel having an impact on propagation, output power, sensitivity depending on the equipment – bandwidth of antennas may have an effect.

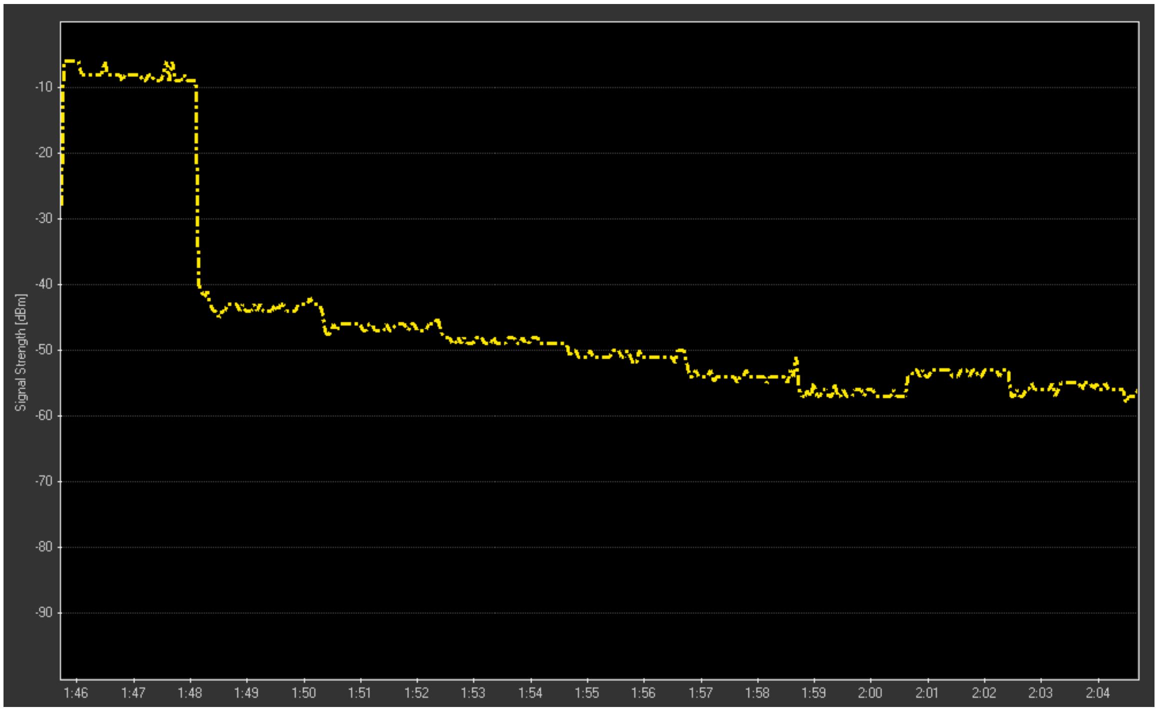
Met with Binghao and Thomas – we discussed a few things which I've read from papers and experimental design. The goal for this week is to have all my wireless devices collected, information recorded, set up with drivers and tested. Another part of my goal is to design my experiment and provide a list of equipment which I will need Binghao to arrange for me. It was also decided that, while the enterprise grade devices offered more interesting challenges to positioning, that they were too complex to use and would be too expensive to justify purchasing multiple units in the immediate future – and so a basic wireless N router was used instead as it was inexpensive and would let us begin our testing of wireless chipsets.

We went to Harvey Norman and purchased a Linksys E2000 router and the Belkin Play USB dongle – unfortunately the E2000 was only selectable dual band and was not suitable for our testing. Instead, we exchanged it for a Belkin Play Access Point which is simultaneous dual band. Enquries about the chipset took a while to resolve – according to technical support, the chipset is a Ralink. The chipset in the Play USB dongle is definitely a Broadcomm.

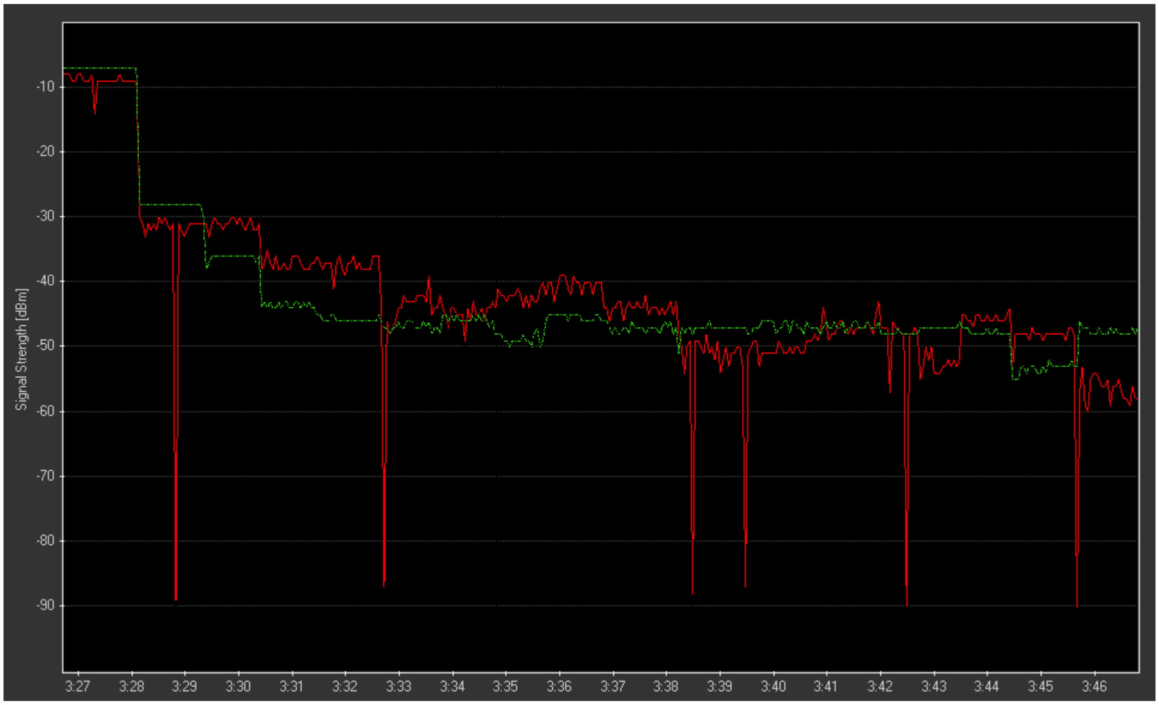
Preliminary experiments were conducted in Electrical Engineering, Level 3 on the bench in order to ascertain the functionality of the equipment. InSSIDer was used to perform scanning – logging was limited to GPX only and I will probably have to write my own GPX parser in order to extract the RSSI data. Time of each logging is not automatically recorded – a GPS is required and so I think I will connect my USB GPS (to avoid Bluetooth 2.4Ghz interference) in order to have the time logged.

Basic testing involved moving the wireless card in linear distance steps away from the AP and plotting signal strength. First, the AP had to be unpacked and be configured.

First test was performed with the Linksys E2000 in 5Ghz mode only and the Belkin Play USB dongle. The signal seemed unusually stable – this could possibly be attributed to the fact that it was broadcasting on the 5Ghz band at channel 44 where there were no other AP signals to interfere. The card was moved in increments corresponding to the intervals between wall mounted powerpoints – approximately a meter. Unusually the signal strength trend reversed toward the end – we postulate possible multipath effects enhancing strength.



That router was returned and a similar experiment was performed with the Belkin Play router – in which case, a simultaneous logging of 2.4Ghz and 5Ghz signal levels was accomplished with the Belkin Play card. Instead of dwelling at each distance for two minutes, we chose to dwell for one minute in order to save time. Distances were stepped in increments of half an A4 sheet of paper (~15cm) for the first six steps then, in increments of power point distances in order to provide more resolution in the signal trends due to the exponential decay characteristic. It is heartening to see that, for the most part, the 2.4Ghz follows the 5Ghz trend fairly well, however, strange signal strength reversals seem to be quite prevalent, and the 5Ghz signal strength seems to plateau to a flat much more than the 2.4Ghz signals do. That being said, this single test is not enough to draw any conclusions from – it is only a preliminary test to prove the feasibility of performing such a test given our present equipment and test design status. It is strange to see dips in the 2.4Ghz output – I propose that this may be due to clients or AP's on the same channel corrupting beacon frames which cause a loss of strength measurement – these readings should probably be filtered out for the final analysis.



As for the final test – it was decided to choose Channel 6 for 2.4Ghz for antenna bandwidth reasons. A channel choice for 5Ghz was not decided on, however, the equipment seems to limit us to 38, 40, 44, 46 regardless of the advertised channels on the box. It would make sense to choose any of these channels as Uniwide does not presently use these channels – instead they use 146, 148, etc which are closer to 5.9Ghz. Hopefully there should not be much difference in the signal readings for the different frequency ranges – however, this is a concern. It was also decided to minimize influence from environment, that the test be performed outdoors on the village green or on the top floor of the carpark with the wireless devices situated on top of upturned plastic bins (as they are non-metallic and will not contribute to multipath effects and cause more errors in readings). Orientation of the device was considered unimportant as long as it was held consistently – I proposed a USB extension lead would be used which keeps the USB dongles vertical as that would probably be the most consistent method. Various devices were to be tested – some of which would not be connectable to a computer – and so a fair test had to be constructed for all these platforms within the limitations of the software and hardware. So far, I am still a bit fuzzy about the software to be used on the phones – but this will be sorted out in the future.

- <u>Tuesday 23rd November 2010</u>

Today I begin to think about experimental design a bit seriously. First of all, if we conduct our experiment in outdoor space, we can expect our free space path loss rules to hold. Most AP's give out 17 to 20dBm of power, and the acceptable receive power for our tests should extend down to about -90dBm. Assuming dipoles at both ends, which would contribute 2.15dBi of gain, we can try to work out the approximate distance a wifi signal will reach travelling in free space.

Transmitter power + gain of antenna at TX – FSPL + gain of antenna at RX = RX power level. Taking the centre wifi channel at 2450mhz – and the formula below, we can substitute and solve:
FSPL = Transmitter power + gain of antenna at TX + gain of antenna at RX – RX power level
FSPL = 20 + 2.15 + 2.15 – (-90)
FSPL = 114.3dB

To find the distance, we'll solve for d:

$$\begin{aligned}\mathrm{FSPL(dB)} &= 10\log_{10}\left(\left(\frac{4\pi}{c}df\right)^2\right)\\ &= 20\log_{10}\left(\frac{4\pi}{c}df\right)\\ &= 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right)\\ &= 20\log_{10}(d) + 20\log_{10}(f) - 147.55\end{aligned}$$

114.3 = 20 log10 (d) + 20 log10 (2450x10^6) – 147.55
20 log10 (d) = 114.3 – 20 log10 (2450x10^6) + 147.55
20 log10 (d) = 74.06667…
Log10 (d) = 3.70333…
D = 5050.4946
D = 5050 m

This means that, theoretically, we will need to go out 5km in the ideal case. Unfortunately, this doesn't seem realistic. Most wifi products advertise 300m outdoor range, so I think we should restrict ourselves to this. Pulling out Google Earth to look at the village green tells me that it's about 145m across which is a bit small for testing the complete range but it is an open area. There is a nearby oval but it's only marginally better at 152m across. Astrolabe Park is 400m across – with very few trees in the way, but it's public ground and requires transport about a mile down the road. The carpark that was suggested is only 76m across. I guess we have to make do with what we've got.

It was discovered in earlier tests that the cards seemed to scan at their own rates – we would ideally want at least 100 sample points per card. In this case, I aim to set the scan rate to once per second and then instead over-collect data for post processing as advised by Binghao. In essence, the current plan is to save stopping and starting logs – we will just set on Automatic logging, and every 5 minutes we will move from one position to the next. The first 4 minutes will be data collection, the 4-5th minute is for repositioning – so say from xx:00:00 to xx:04:00, data will be collected undisturbed, then from :04:01 to :04:59 will be for repositioning and clearing the area, then xx:05:00 will be collection again. At this rate, to collect data for many points for each card will be quite costly in terms of time – we can probably do one or two devices a day.

The automatically logged GPX files will need to be post processed to retrieve the data – this will be done by a program which I will probably write in C given my familiarity with it.

Distances to be tested should ideally be logarithmically related – I propose we test at distances of 0m, 0.1m, 0.15m, 0.25m, 0.5m, 1m, 1.5m, 2.5m, 5m, 10m, 15m, 25m, 50m, 100m in order to cover the complete range of distances in an almost even logarithmic way without causing too much problems with measurement (i.e. 12.5m is not as easy to measure as 15m).

Equipment List:

- 2 x Plastic Bins of the same type to elevate AP and Client card.
- Measuring tape (100m preferable)
- Tent Pegs or something to flag positions on the ground for distances.
- Blutack or other temporary adhesive (or even adhesive tape)
- Inverter sufficient to power the AP power supply
- Battery sufficient to run AP for at least 1 hour, preferably 3 hours.
- Optionally, an inverter to run laptop.

- Barrier tape or rope and some portable poles to cordon off the area and prevent human interference with experiment.
- USB GPS receiver for time logging in GPX (I can supply)
- Laptop, power supply unit (I can supply)
- Access Point (we have just purchased)
- Wireless cards, drivers and software (I can supply some, school has bought one)
- Other devices (school should supply, I only have a few suitable devices)
- Trolley to carry all the equipment to the Village Green

- Wednesday 24th November 2010

Scrounged around in my room to collect all the wireless cards I can find. Being an avid collector of such equipment, I had quite a few cards to contribute to the mix. There's still a few more, I'm sure, but as of now, we have the following:

Manufacturer and Model	**Chipset**	**Comments**
Diamond Digital A101 (Rebadged Asus WL-600G)	Envara WiND502	Fairly old model. Precursor to the Centrino. Two samples available to test.
Netgear WG111v2	Realtek RTL8187L	Popular low cost Wireless G dongle. (Should have two samples available to test)
Netgear WPN111	Atheros (AR5523A/AR2112A)	Super G 108Mbit/s dongle. Rangemax series.
Netgear WG111U	Atheros (AR5523A/AR5112A)	Super AG dual band 108Mbit/s dongle, Rangemax series.
D-Link DWA-140	Ralink RT2870	Wireless N 2.4Ghz only.
D-Link DWL-122G	Ralink RT2570	Wireless G, low cost dongle.
Netgear MA101	Atmel AT7650x	Wireless B (old)
Billion BiPAC3011G	Zydas ZD1211	Another popular low cost Wireless G dongle. Three samples available to test. Bought by Atheros
Belkin Play	Broadcomm (BCM4323)	Expensive. Wireless N dual band, purchased by the school.
Broadcomm ABG from HP 2133 Mini Notebook	Broadcomm (BCM4312)	ABG support, mPCI-E form factor.
Intel Centrino 3945ABG in BenQ Joybook R55UV10 laptop.	Intel Centrino 3945ABG	ABG support, mPCI-E form factor.
Intel Centrino 2915ABG in HP Pavilion dv4000 series laptop.	Intel Centrino 2915ABG	mPCI form factor
Atheros Wireless G in Asus EeePC 701	Atheros (AR5006x)	BG support, mPCI-E form factor.
Texas Instruments ACX110 in HP hx4700 series PDA	Texas Instruments ACX110 (?)	Integrated in WM2003SE based PDA.
Conexant Wireless in Nokia N800 Internet Tablet	Conexant (CX3110X)	Integrated in Maemo 5 based device.
Sychip based Wireless G in HP rx5965 Travelling Companion PDA	Sychip 6100EB	Integrated in WM5 based device

I also have some cards based on the RT2500 from Ralink and the Marvell chipset (WG311v3) but they are for desktop computers and testing them would be difficult. I also have a Socket Branded CF 802.11b card whose chipset is unknown but will require further experimentation. Other devices from the school are not listed.

I have visited almost all the manufacturer's websites and have been able to download drivers for the cards above. The only exception is the Diamond Digital cards where an Asus OEM driver is used instead. Internally, the Diamond Digital cards have Asus branding and so this is an appropriate substitution.

I have also taken out a MSI Wind U100 laptop which I intend to use for testing. It will need to be formatted to do the testing so that we don't have interference from background applications which are already installed on the laptop – that will probably be done tomorrow.

- Thursday 25th November 2010

I reformatted the MSI Wind U100 as expected and installed all the drivers. Unfortunately since the Atheros based cards – i.e. the Netgear WPN111 and WG111U are both firmwareless cards based on the same baseband, the drivers cause conflict with each other. When the device is connected, the last installed driver's firmware is downloaded to the card and that causes the "other" card to fail to function.

Also, I have not been able to find anything that does long term wifi signal logging software compatible with the hx4700 – it may not be useful for our use. Furthermore, all of the software I've been able to find for the N800 does not log signal strength over time. Custom installs of airodump-ng seem to fail after five minutes of logging and requires a hard reboot.

I have also done some research and came across a concept called the Fresnel Zone – an area which contributes to the signal which should be kept free of obstruction. The equation can be used to find the largest radius of the n-th Fresnel zone.

$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}$ - the largest size of the Fresnel zone given by d1 = d2, D = d1 + d2, and $\lambda = \frac{c}{f}$.

At 100m, the radius of the first Fresnel zone is:
Fn = sqrt((1*((3*10^8)/(2450*10^6))*50*50)/100)
Fn = 1.749m … approx 1.75m.
Unfortunately, I don't think the bins may be high enough to keep the first Fresnel zone clear of the ground at 100m. This is taller than I am.

At 50m, the radius of the first Fresnel zone is:
Fn = sqrt((1*((3*10^8)/(2450*10^6))*25*25)/50)
Fn = 1.237m … approx 1.24m.
This still might not be cleared by a bin.

At 25m, the radius of the first Fresnel zone is:
Fn = sqrt((1*((3*10^8)/(2450*10^6))*12.5*12.5)/25)
Fn = 0.8748m … approx 0.875m.
This can probably be cleared by a bin.

At 10m, the radius of the first Fresnel zone is:
Fn = sqrt((1*((3*10^8)/(2450*10^6))*5*5)/10)
Fn = 0.5532m … approx 0.56m.
This might be just cleared by a tall box.

At 1m, the radius of the first Fresnel zone is:
Fn = sqrt((1*((3*10^8)/(2450*10^6))*0.5*0.5)/1)
Fn = 0.1749m … approx 0.175m.
This might be just cleared by a small box.

- Friday 26th November 2010

I decided to conduct some experiments of my own just to confirm the SiRFstar3 USB GPS and inSSIDer could work together. Also just to check that every single card can scan with inSSIDer. Unfortunately, the Atheros cards have their own problems – they seem to return scans erratically, with the same value being reported for almost 20 seconds in a row at times, then three new samples, then it reports the same value for a while longer. The rest of the cards seem to scan properly.

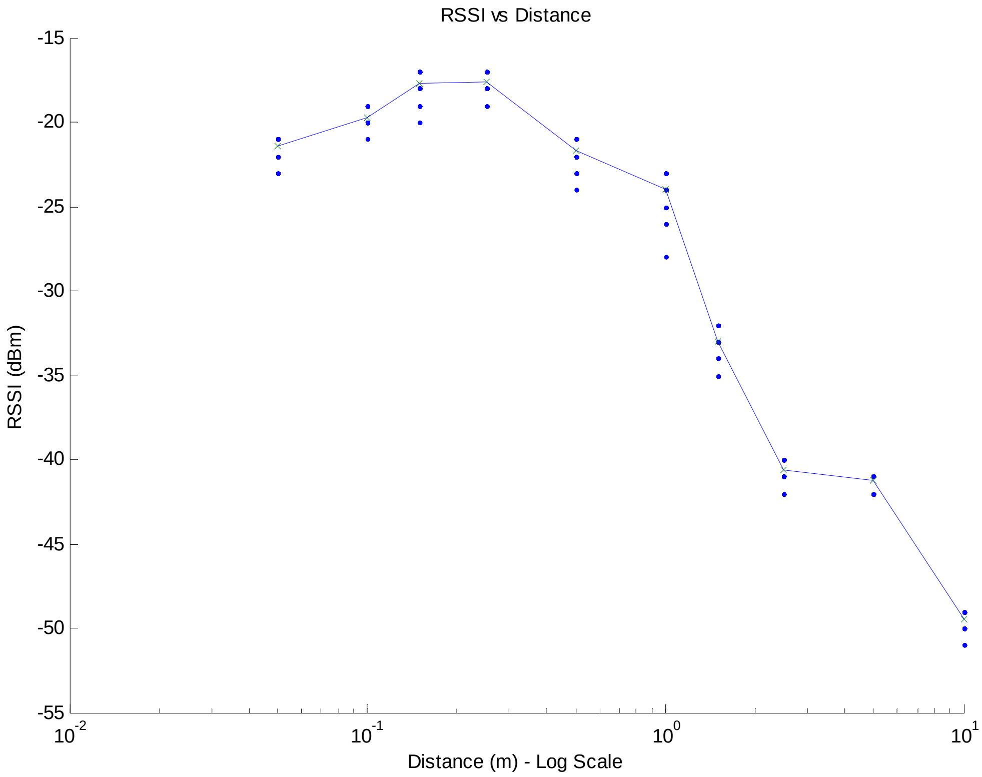
I setup my own Belkin access point at home, SSID 2G4TEST and used two boxes of about 20cm to prop them up. Unfortunately this means results for distances greater than 1m may be affected by Fresnel zone blockage. Furthermore, since I did this experiment in my house, multipath may have contributed to the strange signal trends – similar to that of earlier experiments conducted earlier in the week.

The raw data was stored as a .gpx file, and a parser was written in C in order to parse it. There is one which I wrote to parse to a csv file for excel to process, the other parses it to something similar to matlab arrays but require some manual tending. The program automatically skims out all the unnecessary tags, skips all non-matching SSID’s, and removes segments from xx:04:00 to xx:04:59, xx:09:00 to xx:09:59 etc.

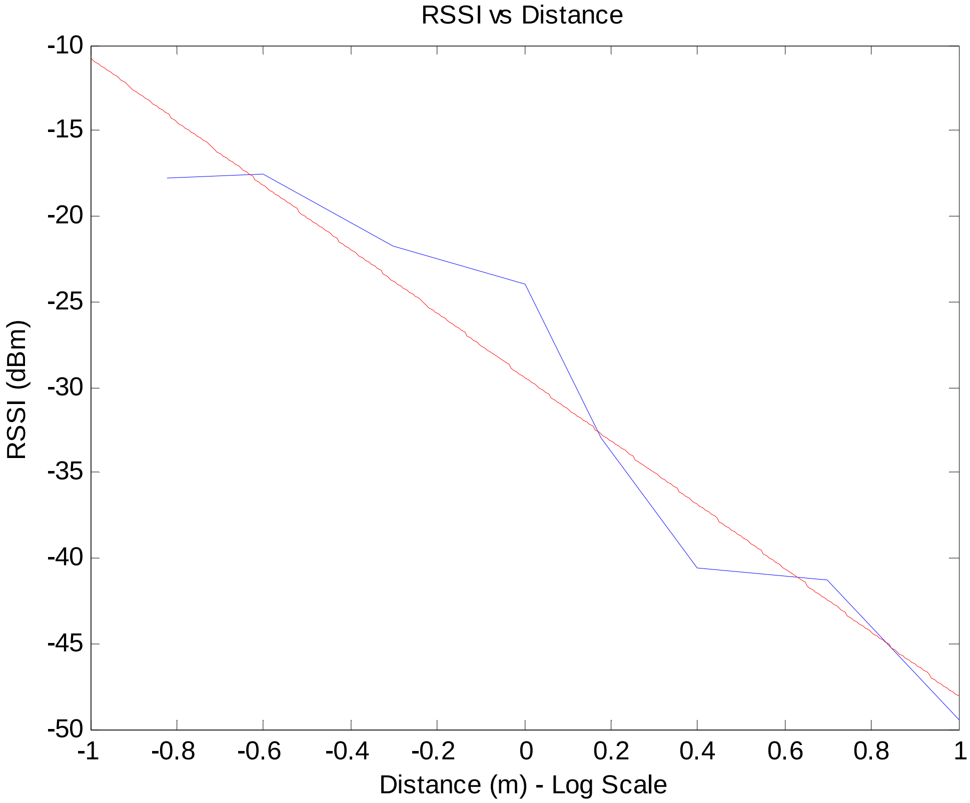
gpxbgtom.c is for SSID 2G4TEST and produces matlab array output. gpxparsebg.c is for SSID 2G4TEST and produces CSV output. All of these programs receive input from stdin and produce output to stdout – simple piping data in and out will result in the data being processed.

A strange trend was found that at higher distances, the number of samples from the same amount of time monitoring was less. This does not appear to be a bug in my program – verifying with the original GPX file seems to suggest that the card did not return any scans for certain seconds, and other times, there were two returns for a second. In all, the theory is that the longer distance allows competing signals to interfere with and destroy some beacon frames by reducing SNR, and thus we lose a sample.

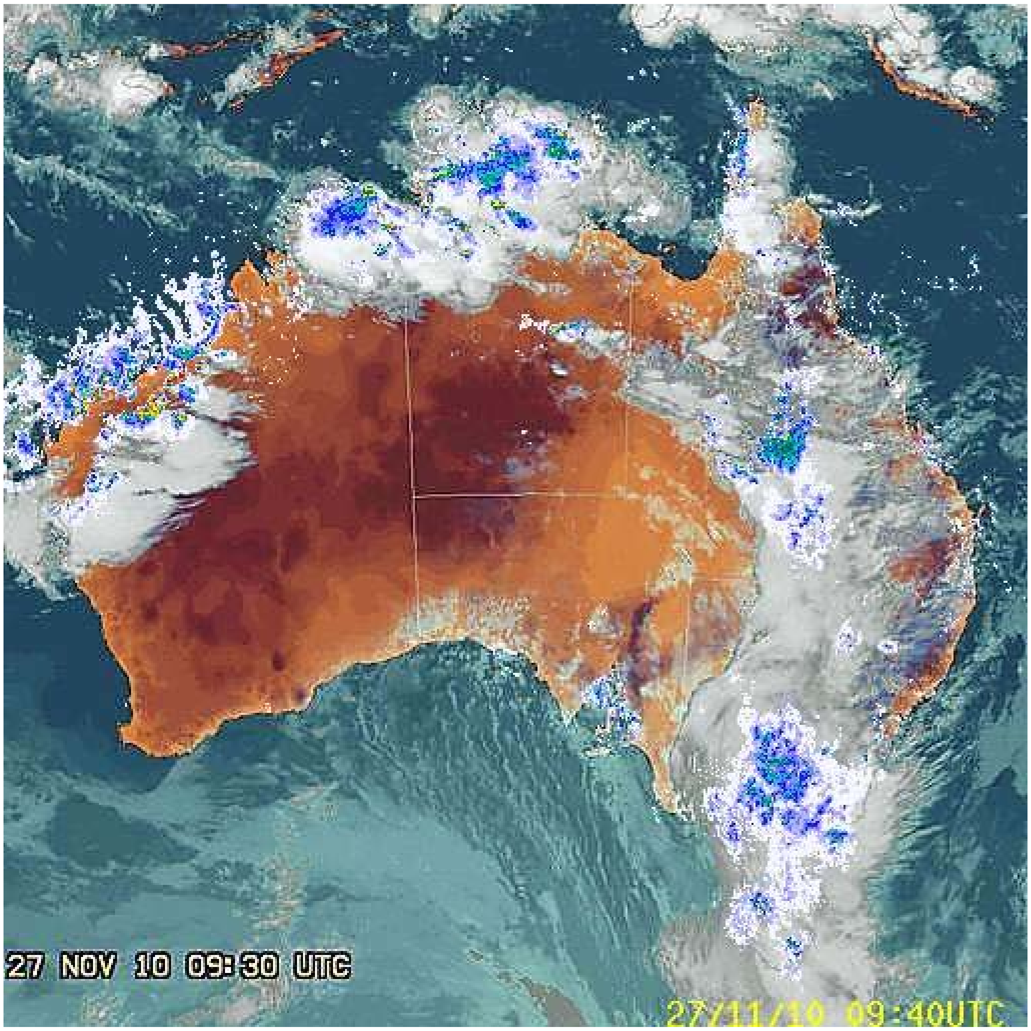
I’m not all that great at MATLAB, and so I had some difficulty, but the first plot below shows the raw data received as a scatterplot, with the means plotted on top and joined by lines. It is interesting to see that at near fields, the signal strength seems to decrease – this may be due to radiation angle limitations of the antennas and slightly different heights of transmitter and receiver. Removing these few points, we produce a plot of the means and a linear fit to the data – note the scale is logarithmic. The data appears to be of fair quality.


RSSI vs Distance
-15
-20
-25
-30
-35
-40
-45
-50
-55
RSSI (dBm)
10^{-2}
10^{-1}
10^{0}
10^{1}
Distance (m) - Log Scale

i


RSSI vs Distance
-10
-15
-20
-25
-30
-35
-40
-45
-50
RSSI (dBm)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Distance (m) - Log Scale

It can also be seen that the weather doesn’t look so good for the upcoming week. This will probably cause issues with outdoor testing.


27 NOV 10 09:30 UTC
27/11/10 09:40UTC