Passively mapping wifi device positions in to a virtual world

Chris Holman[*]
Centre for Advanced Internet Architectures, Technical Report 121031A
Swinburne University of Technology
Melbourne, Australia
6964320@student.swin.edu.au

***Abstract*—We investigate how to use wireless signals to locate devices by implementing our own system in our office, then display their location in a 3D game environment. We create a simple system to track wireless devices using cheap consumer-grade access points. This is achieved by taking calibration measurements from around our room and applying a shortest vector algorithm, then sending the location data to L3DGEWorld. Accuracy levels depended on the device being used, but were all acceptable. Our work extends on previous work by adding the virtual 3D world representation component.**

***Index Terms*—CAIA, wireless, wifi, location**

I. Introduction

802.11 'wireless' networks are found everywhere in our everyday lives. Devices communicating through wireless protocols emit signals. Properties of this emitted signal can be, and are, used to locate devices - be it assisted GPS on your phone to guide you or a retail store tracking customer movement patterns.

In this report, we implement a passive wireless positioning system within our office and map device locations in to a 3D virtual world. The system will not require any modifications to be made to client devices, using only information acquired from our own wireless access points. Once the algorithm has a probable location, it will then push the information in to L3DGEWorld, where a map of our office is used to show where devices are.

Our work is based on [1], and differs in that we use a smaller area, a different set of software and export our location results to a 3D environment. As our wireless access points are closer to each other than in [1], we achieve greater accuracy.

The remainder of the report is structured as follows:

Section II covers the background, including theory, algorithms and L3DGEWorld.

Section III details our implementation and describes our environment, including the layout of our room and the hardware used.

Section IV presents some results from our testing.

Future work is outlined in section V and the report concludes in section VI.

II. Background

Wireless Access Points (APs) are common throughout most environments we frequent - businesses, shopping centres, homes, public spaces. Every client with wireless enabled emits signals that can be, and are, used to determine their approximate location. Typical applications range from asset tracking in an office [2], [3] to mapping shopper traffic patterns in a busy shopping centre [4].

Some wireless location systems rely on the client running a program [5] which works together with an AP or similar monitoring device to determine the position. In this report, our goal is a passive wireless location system. This means that we will not modify clients in any manner, and all attempts to locate a device will be made using data obtained only from our monitoring APs.

A. Wireless signals

When an 802.11 frame is transmitted, any AP listening on that channel can see the frame. Even if the transmitting device is not connected to that particular AP, the AP will still be able to see the frame. If the frame is encrypted, it cannot see the contents of the frame, but it can determine which devices it was between.

APs have a valuable view in to how far away devices are from them. The signal strength of a given transmission dissipates as it travels away from its source, according to the inverse square law. The AP does not know how much power was transmitted by the client, so the received signal strength does not tell the AP exactly how far away the client is. Instead, we can use multiple

[*] Author is currently an undergraduate engineering student doing an IBL placement at CAIA

APs, each seeing a relative signal strength, to determine where the device is located.

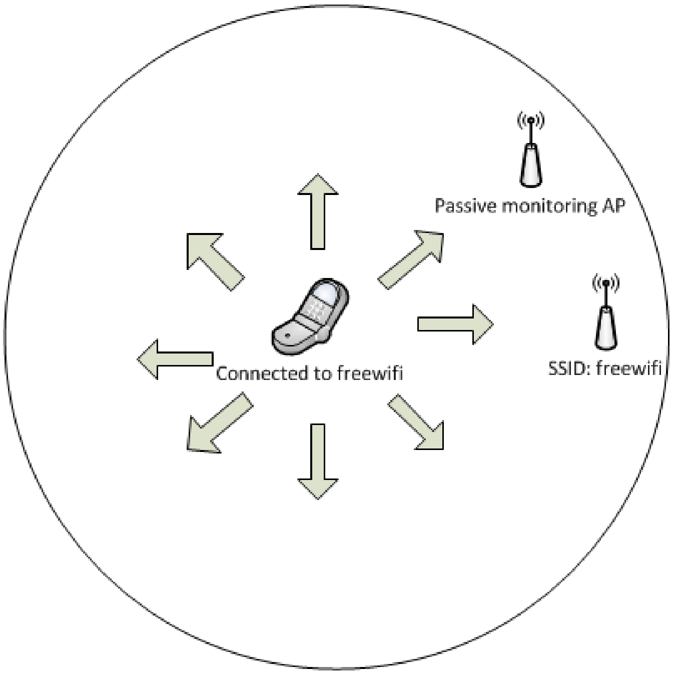

Figure 1. Traffic sent by a wireless client is seen by every other device and AP in range too

B. Algorithms

There are two general approaches to utilising signal strength data from APs : deducing the location through an algorithm or equation, and comparing values against a table of know calibration values. We only attempt to locate the device in 2 dimensions and, but these algorithms can be used for 3 dimensional location services as well. The vertical position of devices in our 3D environment is static for a given sector.

1) Mathematical algorithm: When locating a device through using a mathematical algorithm, the signal strengths from a number of reference points are used to calculate a likely position. This is typically based on the assumption that signal strength decreases in a predictable manner, such that all reference points receive a signal at a strength proportional to their distance from the client. In reality, obstructions such as building structure and office furniture (eg, filing cabinets) interfere with the data, negatively impacting the ability of the algorithm to determine a location.

In Fig. 2, the three APs have seen a frame from a client at the shown signal levels. The signal strength provides a relative distance from the client to each AP. The algorithm then finds a point where the distance from the point to each AP matches the ratio that the signal strength provided.

If an obstruction is between AP3 and the client, the signal strength will be lower, skewing the ratios. The algorithm is unaware of the obstruction, so assumes that the client is further away from AP3 than it actually is, distorting the results.

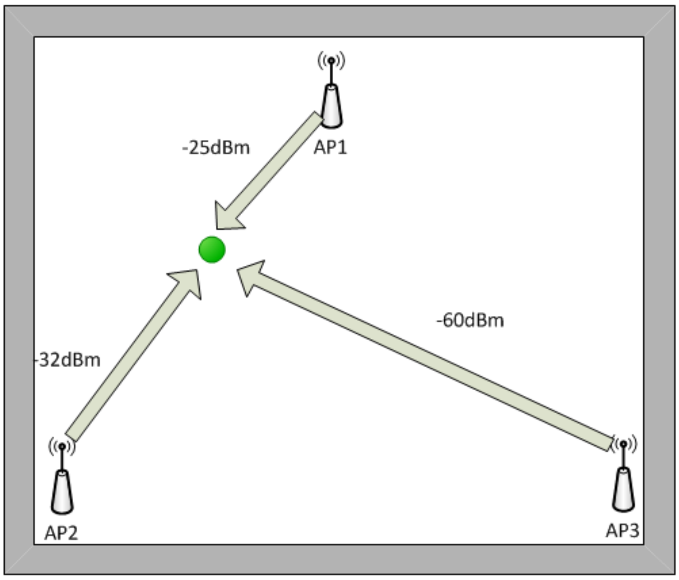

Figure 2. Deriving the ratios for a mathematical algorithm. The ratio between signal strengths correlates to the ratio of distances from the APs to the device.

2) Calibrated algorithm: Using the calibrated values method, the space is mapped out in to sectors, and a signal strength measurement is taken for each sector. Since these calibration measurements are taken within the environment, they take in to account any obstructions. In a static environment, such as an office, this works well as the obstructions should never move. In a more dynamic environment, such as a warehouse, the movement of stock to and from shelves could be enough to interfere with the location service.

[6] demonstrates this method. In their test, they used an entire floor as their test area. Our test will differ from theirs in that we are testing within a single room.

To calibrate the area, first the area must be divided in to sectors. We chose to make sectors for each desk, room and major piece of furnishing (eg, filing cabinet and fridge). There is no size limits on sectors, but the smaller they are, the less accurate the location service will be. For each sector, a test device is placed roughly in the centre, and a set of signal strength measurements are taken from the APs - we took about a minute of measurements for each sector. The 80th percentile figure for each AP are then stored as the calibrated values for that sector.

In Fig. 3, there are four calibrated reference points around a room, each with a known reference signal strength from each AP. The room has four sectors (not shown), each taking a quarter of the room surrounding each reference point. A client is at the position labelled “measured values”.

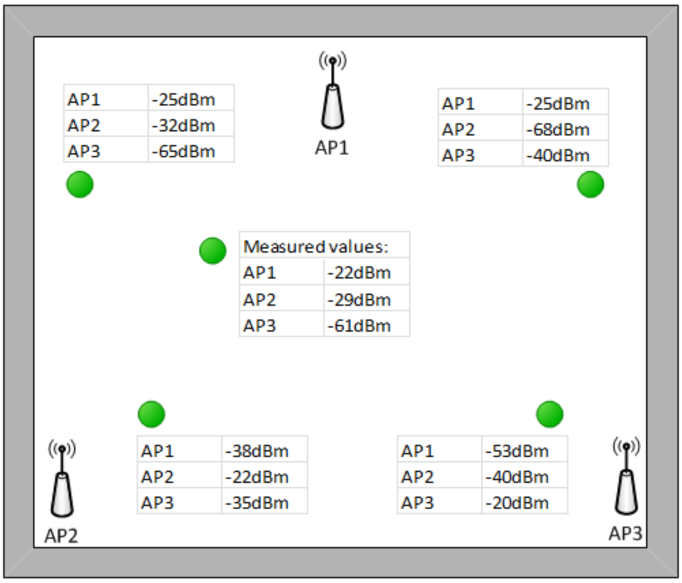

Figure 3. The calibrated method - four calibration points around the room, and one measured value of an unknown location

To determine the location of a device, a vector is created between each of the reference points and the measured signal strength values of the clients, then finding the shortest vector. Each AP forms one dimension of each vector. In the Algorithm 1, all values are signal strength values in dBm, the ’c’ values are the calibrated values of a given reference point, and the ’m’ values are the measured client signal strength values.

Algorithm 1 The generalised algorithm to determine distance from a point.

```
For each AP:
$\Delta = (AP_c - AP_m)$
Then,
$d^2 = \sum_{i=1}^{n} \Delta_i^2$
```

For example, in Fig. 3, the ’distance’ between the client and the top left reference point is as follows:

$\Delta_1 = (-25 - -22) = 3$
$\Delta_2 = (-32 - -29) = 3$
$\Delta_3 = (-65 - -61) = 4$
$d^2 = \sum_{i=1}^{n} \Delta_i^2$
$d^2 = \Delta_1^2 + \Delta_2^2 + \Delta_3^2 = 9 + 9 + 16$
$d = \sqrt{34} =\sim 5.83$

This is done for each of the four reference points in Fig. 3. Table I shows the end result of performing this calculation for the four reference points. The top left position is the clear winner.

In practice, the square root step does not need to be done, and is only included to make it clearer that we are using Pythagoras’ theorem. Performing the square root will not alter the order in which the reference points are ordered.

Table I
DISTANCE VALUES FOR THE HYPOTHETICAL MEASUREMENT IN FIG. 3

Reference point	Distance
Top Left	5.83
Top Right	44.40
Bottom Left	31.22
Bottom Right	52.56

In [7], the two approaches are tested against each other. Both are capable of achieving similar levels of accuracy, although some of the algorithms are significantly less accurate than others.

As our office is fairly static and the calibration method is programmatically simpler, we chose to use the calibrated values method. We use [1] as a basis for our work.

C. L3DGEWorld

Traditional data monitoring tools, such as Cacti and Nagios, represent data through tables and 2D graphs. L3DGEWorld [8] takes a different approach, using the freely available Quake 3 engine to provide an interactive 3D environment. Inside the environment, sets of data are presented as objects which change various visual properties according to the input data. For instance, the object may spin according to the packets per second, and it grows in size according to memory usage.

Data sources for L3DGEWorld can be anything that can be IP connected - from PCs, to embedded hardware, to arduinos. l3dgecomm [9], the protocol used for communication between L3DGEWorld applications, operates over UDP/IP and is otherwise platform agnostic.

One of the features of L3DGEWorld allows us to arbitrarily move objects around the in-game world. We will use this to display the location of a wireless client.

III. Design

A. Hardware

The design of the system is simple: a number of 'monitoring' APs spread around the environment to listen to local wireless activity; a dummy AP broadcasting an SSID for our test devices to connect to; a PC to process the data; and test devices.

We have three TP Link APs, which will serve as the monitoring APs. These can run OpenWRT (a minimal GNU/Linux OS with a small array of command line utilities), including tools to gather the data we need.

A Cisco Aironet AP will be our dummy AP. Our office has an existing wireless network, but using our own is advantageous. When running our own, we can control which channel the clients will be transmitting on. This is useful, since our monitoring APs can only listen to a single wireless channel at a time.

- 3 x TP-Link TL-WR1043N/ND v1. Each is configured with OpenWRT.
- Cisco Aironet AP1220B unit with the channel forced to 2.4GHz channel 1
- A spare PC
- A BenQ S42 laptop, with an Intel AGN 5300 802.11n wireless card (test device)
- A Motorola Atrix mobile phone (test device)

B. Software

```
05:46:17.066902 18827321us tsft
54.0 Mb/s 2412 MHz 11g -59dB
signal antenna 1 [bit 14] CF
+QoS BSSID:00:18:74:c4:ff:01
SA:e4:ce:8f:02:92:50
DA:01:00:5e:00:00:fb Data IV:2238 Pad
20 KeyID 0
```

Figure 4. An example line of tcpdump output. The signal strength is provided for every frame. In this example, -59dB.

1) Location service: OpenWRT on the monitoring APs provides tcpdump. tcpdump provides all of the information we need - time stamp, source MAC address and signal strength. The data from tcpdump is sent to the central computer using netcat - a simple command line utility to pipe data over a TCP network.

The computer runs a series of scripts to collect and process the data. One to calibrate reference points, one to determine the location of a client and a couple to generate graphs and analyse data.

The calibration script takes data from each AP and listens for signal strength information for a specific MAC address until stopped. The data is then processed to produce a single figure for each AP and stored in a database for later use. The script is run once for each reference point.

The algorithm in Section II-B2 is then used to locate a client.

2) Mapping : Using L3DGEWorld, it is simple to move objects arbitrarily around a 3D virtual environment. Objects, or 'entities', are positioned by reading a set of files, each containing the location of an entity. To change the position of an entity, the file for the entity is updated, then using l3dgecomm, an update trigger is sent to the L3DGEWorld server.

The in-game objects used to represent a wireless device are l3dgehost entities. l3dgehost entities are added to the map before it is compiled, so enough entities to cover the expected maximum number of wireless clients must be provisioned from within the map editor. To avoid visual clutter, these can be created outside of the viewable area and moved in and out as needed.

A user is able to fly around the environment and observe the environment from any point.

3) Overall program flow: Fig. 5 shows the structure of our software. The system is not limited to three APs - it can support as many as needed to cover the area. More APs should provide a higher degree of accuracy.

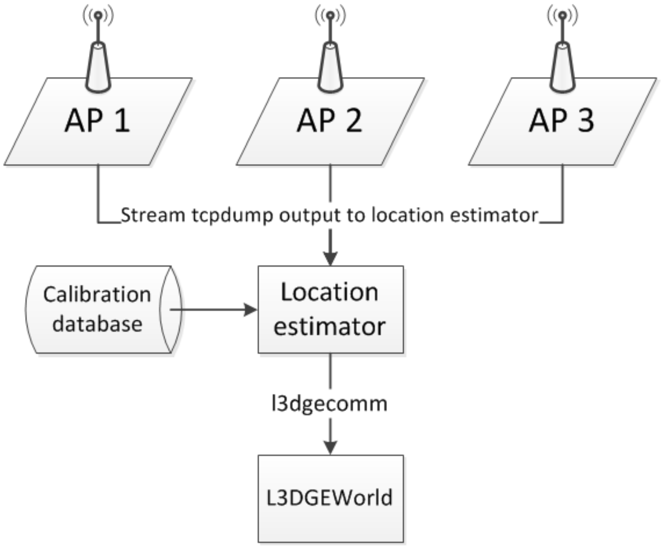

Figure 5. Program flow in our system

IV. Results

A. Calibration

We created 13 sectors in our office, and calibrated each of these reference points. We used the BenQ laptop test device to calibrate. A section of the calibration table is provided in Table II as an example.

Our sectors were defined as per Fig. 6. For calibration, the test device was placed at the circle and frames collected for around a minute per location, to provide an adequate sample of the signal from that location.

Table II
A SAMPLE OF OUR CALIBRATION TABLE (VALUES IN DBM)

Sector	1	2	3	4	5	6
AP1	-25	-32	-31	-22	-21	-31
AP2	-45	-41	-39	-42	-46	-39
AP3	-41	-35	-40	-43	-35-	-37

B. Locating devices

We tested three cases: the laptop with its transmit power at the same setting as when calibration happened; the laptop with low transmit power; and the mobile phone. In all instances, the devices were placed approximately where the test device was placed during the calibration phase, with varying degrees of accuracy.

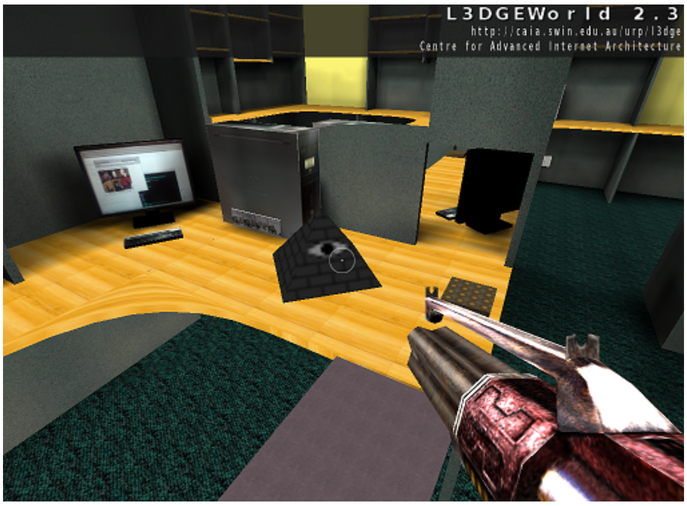
L3DGEWorld 2.3
http://caia.swin.edu.au/urp/l3dge
Centre for Advanced Internet Architecture

Figure 7. The pyramid represents the estimated location of a wireless device.

Laptop, full (original) TX power: We tested at locations 0,1,5 and 11. In all four cases, the estimations were reliable.

Mobile phone : At sector 2, the correct location was estimated, however it was regularly a tie between the correct sector and a nearby sector. The correct sector was estimated only for the fact that the location ID 2 is lower than 6.

Estimating the correct sector was not regularly achieved, but all estimates were at least in the vicinity of the correct sector - there were no estimates at the opposite end of the room.

Table III
LAPTOP LOCATION TEST RESULTS

Sector	Sector estimate
0	0
1	1
5	5
11	11

Table IV
MOBILE PHONE LOCATION TEST RESULTS

Sector	Sector estimates
0	1,12,13
1	5,0
2	2
4	0
5	9,5
11	9

Laptop, low TX power: Results for the laptop with low transmit power were near identical to the mobile phone tests. When the transmit power is lowered, the signal strength received by the APs will also be lower. The greater disparity between measured and calibrated values makes the calculated distance value greater. The end result of this is that it is harder for the program to correctly choose the sector.

C. Data

The signals sent from a stationary device will not always be received by another device at a constant signal strength. Especially in situations where there is no line of sight, the efficiency and diversity of paths the signal takes between devices can change. A person walking past, slight movements in the device's position, device power saving features and interference are a small subset of things that can impact the signal. Fig. 8 shows a CDF of the signal strength a stationary device over a short period of time, demonstrating this effect. Fig. 9 shows the same dataset in a signal strength vs time graph for comparison.

Comparing Table II and Fig. 10, most of the calibration values lie above the 99th percentile, demonstrating that most traffic is transmitted further from further away than the bounds of this office.

V. FUTURE WORK

Our developed software is fairly simple. For a public release, we would develop a small application utilising

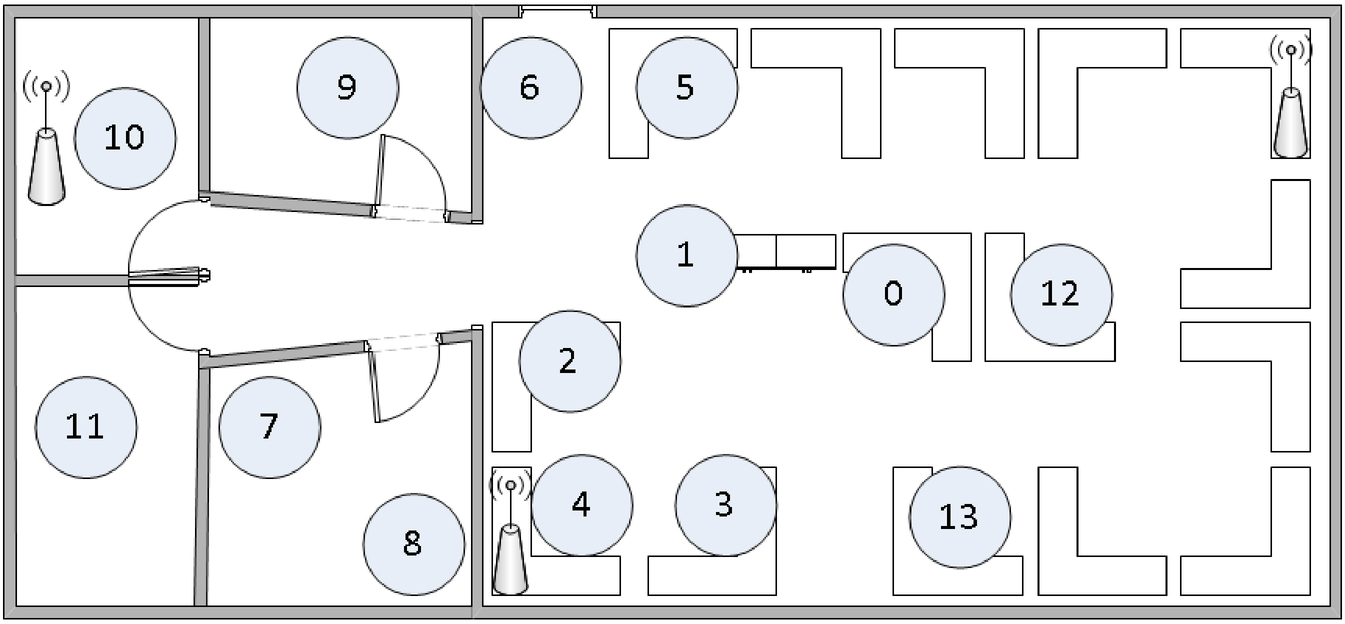

Figure 6. The layout of our office and sectors, with reference points drawn in

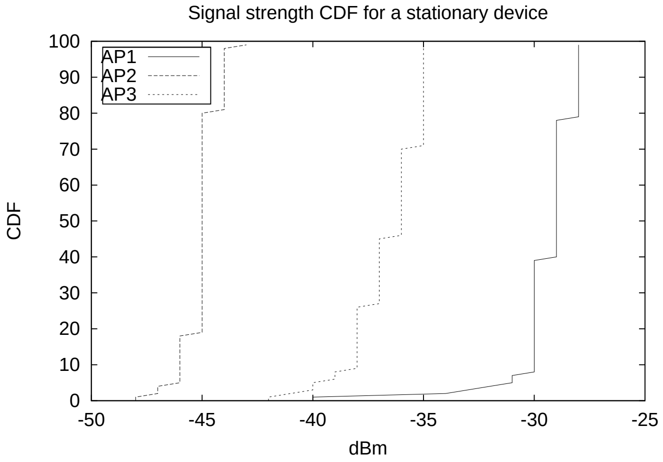

Figure 8. Variation in signal strength of a stationary client (CDF)

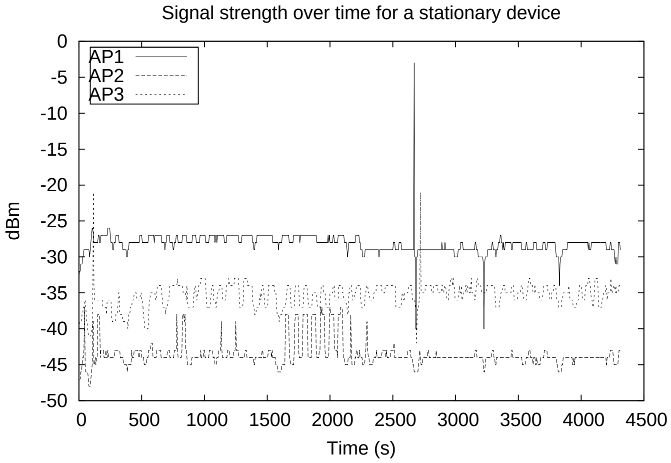

Figure 9. Variation in signal strength of a stationary client over time

the PCAP library directly, with the L3DGEWorld application already integrated in to the scripts, requiring little to no manual configuration.

Our sample area is fairly small. Further tests could expand to the entire floor, or multiple floors, studying the impact of varying range or number of monitoring devices used.

Currently each MAC address to be tracked must be pre-determined, as a lot of wireless traffic comes from outside of our office or is broadcast from APs on our ceiling. An algorithm to detect when either the transmitting client is an AP or far outside our office would allow devices to automatically be tracked as soon as they enter the room.

We have a couple of useful pieces of data available to us that we could convey through L3DGEWorld, such as variation from calibrated signal strengths, bandwidth used and strongest signal. This could augment the wireless positioning system to provide a richer insight in to the wireless environment. Additionally, a way to represent two wireless devices in the same location.

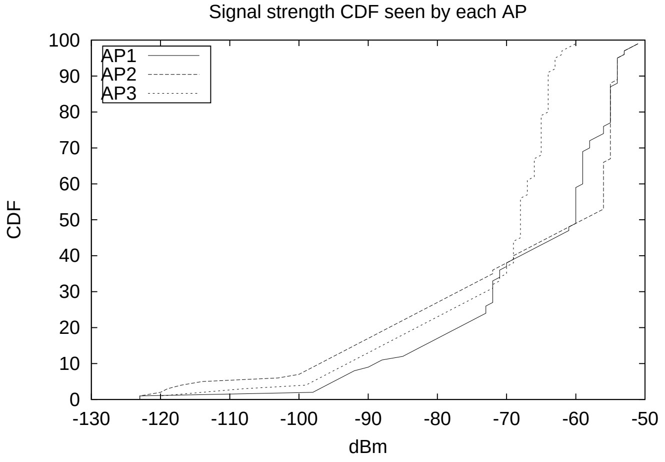

Figure 10. Signal strength of every 802.11 frame seen over a three day sampling period

VI. Conclusions

In this report, we documented our development of a wireless device location tracking system. Our system was able to accurately determine the location of the test device which was used to calibrate the system, and was able to approximately locate our other test device, which was not used for calibration. The calibration and tracking was all done passively, such that this system can be used for any 802.11 device.

We then mapped the wireless locations in to a 3D virtual map of our office, using our L3DGEWorld tool.

References

[1] K.-F. Kao, I.-E. Liao, and J.-S. Lyu, “An indoor location-based service using access points as signal strength data collectors,” in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pp. 1 –6, sept. 2010.

[2] Ekahau, “Real time location system (rtls) overview | ekahau.” http://www.ekahau.com/products/real-time-location-system/overview.html. Accessed 23rd Oct, 2012.

[3] AeroScout, “Real Time Location/Locating Systems | WIFI Real Time Location | AeroScout.” http://www.aeroscout.com/technology. Accessed 23rd Oct, 2012.

[4] A. Censky, “Malls track shoppers’ cell phones on Black Friday.” http://money.cnn.com/2011/11/22/technology/malls_track_cell_phones_black_friday/index.htm. Accessed 23rd Oct, 2012.

[5] F. Aloul, A. Sagahyroon, A. Al-Shami, I. Al-Midfa, and R. Moutassem, “Using Mobiles for On Campus Location Tracking.” http://www.aloul.net/Papers/faloul_momm09.pdf. Accessed 26th Oct, 2012.

[6] Y. Gwon and R. Jain, “Error characteristics and calibration-free techniques for wireless lan-based location estimation,” in *Proceedings of the second international workshop on Mobility management & wireless access protocols*, MobiWac ’04, (New York, NY, USA), pp. 2–9, ACM, 2004.

[7] D. Turner, S. Savage, and A. Snoeren, “On the empirical performance of self-calibrating wifi location systems,” in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pp. 76 –84, oct. 2011.

[8] CAIA, “L3DGEWorld 2.3.” http://caia.swin.edu.au/urp/l3dge/. Accessed 26th Oct 2012.

[9] L. Parry, “L3DGEWorld 2.3 Input and Output Specifications.” http://caia.swin.edu.au/reports/080222C/CAIA-TR-080222C.pdf. Access 26th Oct, 2012.