Mapping Populations: Visualising spatial distributions of populations over time

Vincey Au
Talented Students Program, Faculty of Science
University of Sydney
viau6739@uni.sydney.edu.au

Masahiro Takatsuka
School of Information Technologies
University of Sydney
masa.takatsuka@sydney.edu.au

***Abstract*—More than 2.5 exabytes of data is generated every day, yet most has not been converted into meaningful information. In this paper, we developed a web application which visualises populations of people over space, specifically, all the instances during the period of 21 March to 12 September 2014 that University of Sydney students connected to on-campus wi-fi. Key findings regarding the behaviour of students were observed and discussed: The most central buildings on campus were identified, and wireless activity times were found to peak within 10 minutes past the hour from 12 to 2pm on weekdays. The findings clearly demonstrate not only the usefulness of data visualisation in mining information, but also issues encountered with the proposed methodology. Hence, future research into improving the efficiency and effectiveness of communication via this medium is advantageous.**

I. INTRODUCTION

New digital and mobile technology, in addition to the exponential increase in Internet traffic via social media such as Facebook and Twitter, has led to the generation of large-scale datasets [1] of more than 2.5 exabytes per day [2]. However, much of this data has not been utilised to its fullest potential; it lacks the human knowledge, context and interpretation that converts this into useful information [3].

Given the amount of data being generated, this information needs to not only be conveyed effectively, but also efficiently. As a result, the inherent human disposition to processing images should be considered. Specifically, the human perceptual system interprets images and graphics in parallel, in comparison to text analysis, which is done sequentially and hence, is much slower [4]. Additionally, up to 50% of the brain is involved in visual processing, and 70% of our sensory receptors are in our eyes [5], thus suggesting that one effective way to communicate information is through visualisations.

A. The Data

The data that was used to create the visualisation discussed in this paper was obtained from the University of Sydney. It contains all the instances and details of instances of students connecting to on campus wi-fi hotspots during the time period from 21 March 2014 to 12 September 2014.

Accordingly, the aim of this project was to create an interactive visual interface that would allow users to see the population distribution of students across campus during different times of the year. Ideally, individuals would intuitively understand this data, allowing users to discover useful patterns and trends, which could in turn be used by key stakeholders to improve student life on campus.

II. RELATED WORKS

The following works have been influential in deciding how to best represent the data, both in terms of basic layout and interaction controls.

A. Population Globe

The population globe [6] is an implementation of the WebGL globe created by the Google Data Arts Team. It uses geographic data obtained from SEDAC (Socioeconomic Data and Applications Center) to visualise the distribution of human population across the globe in three separate years: 1990, 1995, and 2000.

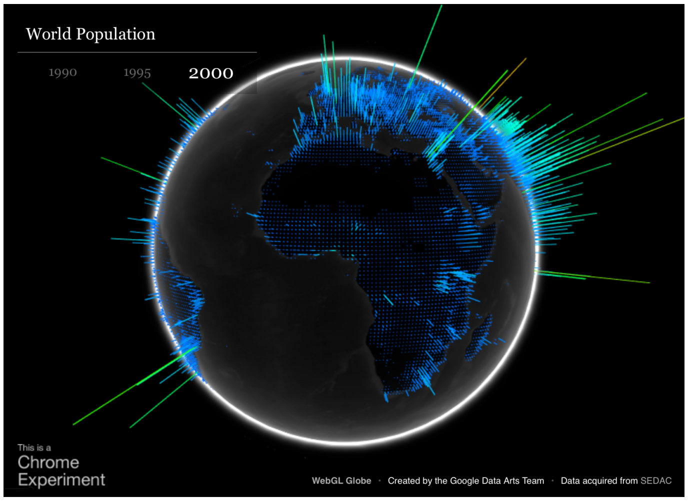

Fig. 1. The Population Globe

Structurally, this work is simple and minimalistic. It features an interactive sphere with 3D bars protruding outwards at specific latitudes and longitudes. Each bar and its colour corresponds to the population size at that location, with larger populations being represented by warmer colours such as orange, and smaller populations by cooler colours such as green and blue.

Controls were basic and intuitive. Users could zoom in and out or rotate the globe just as on any other webpage. Of particular note was the changing data over time; mousing over different years made the bars seamlessly transition to and from corresponding population data, providing an excellent user experience.

The population globe highlighted the relevance of the following technologies in creating the University of Sydney student population visualisation discussed in this paper:

1) *WebGL (Web Graphics Library)*: A low-level JavaScript API designed to render interactive 3D graphics directly in a Web browser [7]. By using WebGL, this work uses shaders and the user's own graphics processing unit (GPU) to compute 3D visualisations, thereby making smooth high frame-rate animations possible [8].
2) *Three.js*: A high-level JavaScript utility library/ API that is user-friendly and simplifies WebGL [9]. Three.js is capable of creating scenes and contains a wide variety of cameras, controls, and animation features, making it highly suitable to map populations in 3D [10].
3) *WebGL Globe*: This is an open platform for visualising geographic latitude, longitude based information. It is the globe base central to this work, and uses WebGL to include colour gradients, based on the data value or type, and interactive mouse controls to navigate the scene [11]. This highlights how many visualisations can use ready-made, animated and interactive base models to display data.

B. *Google Maps Earthquake Demo*

The Earthquake Demo is a showcase of data visualisation technologies using the Google Maps JavaScript API [12]. It plots the location and magnitude of all the earthquakes in the world during the period of 15 August 2012 to 15 September 2012.

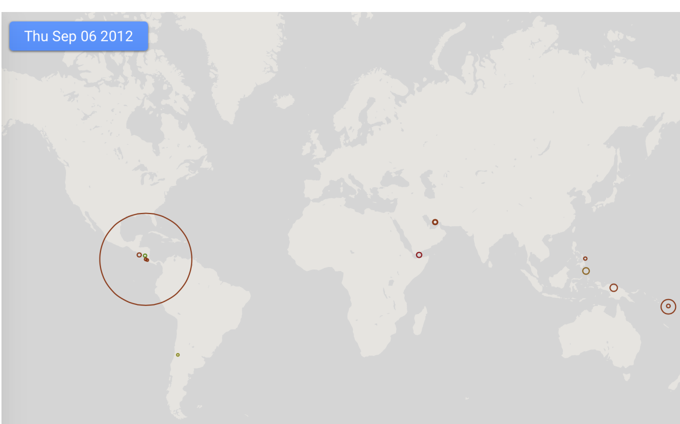

Fig. 2. The Earthquake Demo

Unlike the population globe, the demo displays a flat globe spanning the width of the webpage, with circles of varying sizes and colours changing based on magnitude. The current date is shown on the top left corner and is automatically incremented and cycled through, so the circles update accordingly.

The *Google Maps JavaScript API v3* was particularly notable: This API allows individuals to create web applications with Google's mapping platform. It includes a variety of libraries (places, geometry, visualisation) [13] for developers to create projects with ease and convenience. Specifically, the visualisation library [14] includes a HeatMap layer class that shows data intensity at geolocations.

III. IMPLEMENTATION

The visualisation was implemented as a web application, allowing users to view and interact with the data over time. The full source code for the visualisation is available from the URL in Appendix A. Overall, the approach taken to develop this application was broadly split into two parts:

1) *Back End Development*: This consisted of the original data, how it was modified and imported into a database.
2) *Front End Development*: This involved visualising the base map, then including interaction controls and aesthetic design.

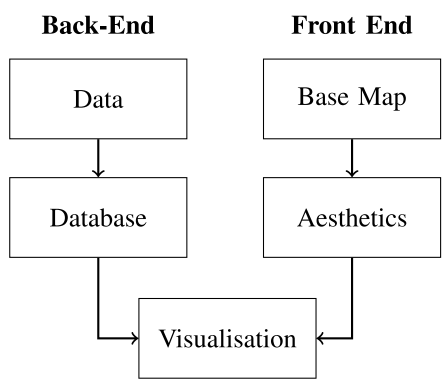

Fig. 3. The overall approach to implementation

A. *Data*

The original data was sourced as a collection of 147 .csv files, each with the following columns filled in per instance of wi-fi usage:

1) **Client IP Address:** The associated IP address
2) **Client MAC Address:** The unique device MAC address
3) **Association Time:** The exact date and time of when the connection was made, e.g. Wed Jun 04 13:18:13 EST 2014.
4) **Vendor:** The brand of the device used (e.g. Apple)
5) **AP Name:** The name of the access point the device was connected to, e.g. air-a28-L2-12. Each instance, separated by the delimiter '-', is of the form 'air', building coordinate, building level and the specific access point on that level.
6) **Device Name:** Unknown meaning, e.g. wlc-s01-3.

7) **location:** Presumably a poor implementation of which campus the connection was made on. Instances were either 'Root Area' or 'Camperdown'.
8) **SSID:** Service Set Identifier, the public name of the wireless network connected to. These were all 'UniSydney'.
9) **profileName:** Unknown meaning. Instances were all 'UniSydney-New'.
10) **Protocol:** The wireless standard and spectrum band (2.4 vs. 5 GHz) used, e.g. 802.11n(5GHz).
11) **Session Duration:** The length of time connected for. This was of the format 'x hours y mins z sec', if duration was more than one hour, or 'x mins y sec', if duration was less.
12) **Avg. Session Throughput (Kbps):** The number of requests processed per minute per server instance [15].

As such, the following steps were taken to clean up the data in all files:

1) **Split AP name column:** Microsoft Excel was used to split this column by using the delimiter '-', allowing each instance to separate into a building coordinate, building level, and AP number.
2) **Deleted unused columns:** This resulted in the following columns: Client MAC Address, Vendor, Building Coordinate, Building Level, AP number, Association Time and Session Duration
3) **Repeat for all files:** Microsoft Excel was used to record a macro doing the above steps, then a Visual Basic code was written to run this macro on all files. The code can be found in Appendix B.

Building coordinates of each instance then had to be linked to latitudes and longitudes. This data was found as an .xml file from the University of Sydney Arts Faculty website [16]. It was then copied into Microsoft Excel, and delimiters used to obtain the following columns: building coordinate, name, latitude and longitude.

B. Database

Both sets of data were then imported into PostgreSQL using the COPY command [17]. This was hosted on the University of Sydney School of Information Technologies server. SQL functions were then used to convert association times to epoch time format, and session durations to seconds. This meant that the number of students connected to the university wireless could be obtained by using a simple SQL query. The SQL updates used to change formats can be found in Appendix C.

C. Base Map

The Google Maps JavaScript API was used to load a simple road map centered at USYD. To extract the geocoordinates and count of student population at any point in time, the jQuery JavaScript library was used. Specifically, an asynchronous AJAX request calls a PHP script every time the input time changes. This allows the webpage to update with the new data obtained from the database without reloading the whole page [18]. Because PHP is a server side scripting language, its code is executed on the server before generating the HTML and being sent to the client [19]. In this visualization, the PHP script connects to the database, runs the SQL query on the University servers, and then returns the result as an XML document. Following this, the Maps API is used again to plot circles whose radius is relative to the magnitude of the population at each location [20].

D. Aesthetics

In terms of aesthetic design, the main addition was the HTML5 range slider input [21], allowing users to adjust the time input easily and frequently. A JavaScript function was also written to convert the epoch time input back into a human readable form so users could observe the exact timestamp at that point.

IV. Results and Discussion

Ultimately, the student wi-fi connection data used to test the application was a collection of historical sample points that were consistent, ordered and dense. Each had a fixed discrete time range made up of the exact start timestamp, and how long students were connected for.

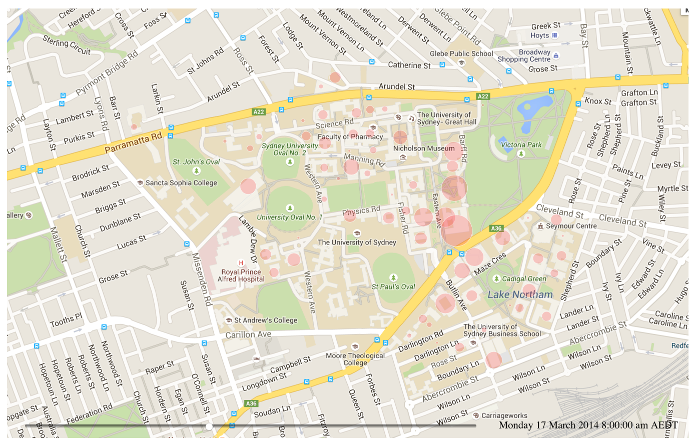

Fig. 4. The prototype with the slider input on the bottom left and the timestamp on the bottom right

As seen in Figure 4, there are four components to the visualisation:

1) The *base map*
2) The *circles* represent each population at each location. The larger the radius, the larger the population.
3) The *input slider* allows the user to control what time the data is extracted from. It is mapped to the epoch time in the database.
4) The *time display* in the bottom right hand corner allows users to know what date and time they are selecting to view the data of.

Hence, users interact with the application via the input slider, changing the time. Each time the input is changed, the visualisation queries the database, receives the new data, and plots this on the map. The application was initially tested with one weeks worth of data at the beginning of semester one 2014,

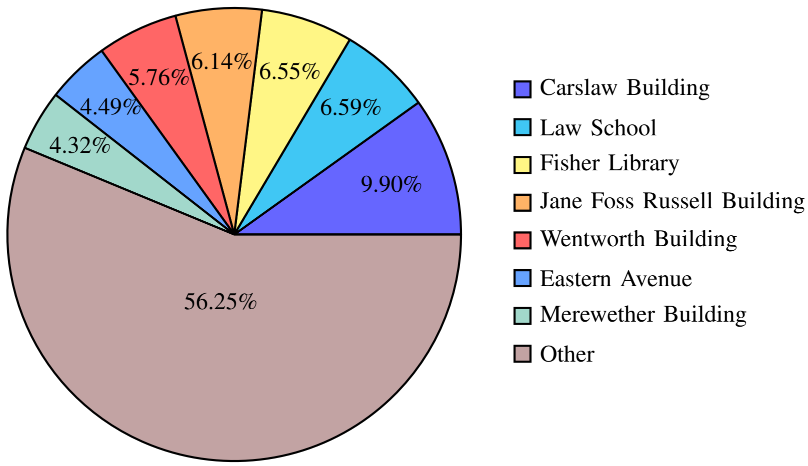

Fig. 5. Location proportions of all connections from 14 March to 21 March 2014

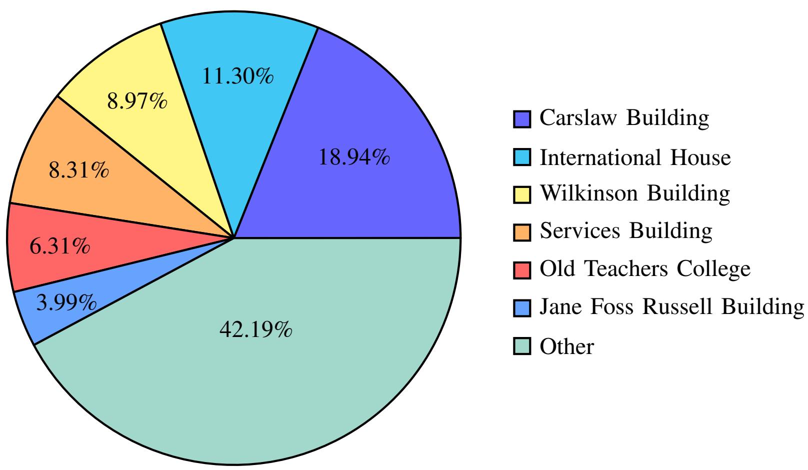

Fig. 6. Location proportions of all connections at 11pm Monday 17 March 2014

from 14 March to 21 March 2014. The following patterns and trends were of interest and give an indication of the extent of information that could potentially be extracted:

- The top 7 most central locations in the University are ranked as follows: Carslaw Building, Law School, Fisher Library, Jane Foss Russell Building, Wentworth Building, Eastern Avenue and the Merewether Building. Figure 5 displays the proportions of these, and Appendix D details the complete list of all locations used over the week of 14 March to 21 March 2014.
- The Carslaw Building has a significant 3.5% lead on connection proportion. This is reasonable given it has eight levels, but also, it has the largest night population as seen in Figure 6. This likely is due to the availability of a 24 hour access Learning Hub study space.
- With Figure 6, the International House having the second highest number of connections at night time can be explained as it is a residential college.
- There is a 565.56% increase in the overall student population from 7am to 8am on weekdays, with the largest population originally in the Services Building then Eastern Avenue Auditorium. This correlates to students arriving to their first classes at 8am.
- The Wentworth Building, which hosts a food court, experiences a significant increase in connections from approximately 12pm to 2pm, peaking at around 1:10pm on Monday 17 March 2014. The peak is a 296% increase from 131 instances at 11am to 388 instances at 1:10pm. A similar trend can be found for other weekdays, but there is no such fluctuation on the weekend.
- The overall peak of connections on any given weekday was mostly within 10 minutes past the hours of 12, 1 and 2pm.
- There was also approximately a 190% increase in connections on the hour when compared to within the hour. The last two points suggest that most students aren't on the Internet during class, as wi-fi connections decrease during classes, and instead, are using their phones between their

classes or during their lunch break.

V. Issues Encountered and Future Developments

While most issues encountered affect the aesthetic design and user experience rather than technically, both design and experience are particularly important in interactive visualization applications. This section outlines the problems and some potential solutions.

Firstly, the PHP script has been written such that every time the input time changes, the previous circles are removed before the new data is loaded. However, the circle markers do not flow seamlessly into each other and there is a noticeable visual pause. Ideally, the circles would transition smoothly from one time state to another, thus not disrupting the user experience. Additionally, the slider does not change dynamically; it has to settle on an input before the query is executed. Thus, to improve the user experience, the circles should be made to transition smoothly from one time state to another, and a dynamic slider should be included.

Secondly, the slider bar was occasionally hard to navigate with precision, despite there only being one week's worth of data. If the whole dataset was included, this would result in cumbersome interactive controls. In addition, the large amount of student activity during the day meant that the data count was constantly changing and was much interesting when compared to the night, where a lack of activity displayed stagnant data with minimal information. This resulted in lengths of time where there was little change in the data, and thus was uninteresting to the user. To maximise the efficiency of this visualization, future research should look into developing a scaling slider that changes how much the input is increased in one increment of the slider, based on the percentage change in values in the data source. For this particular visualization, this would be implemented by analysing how much the population changes at a certain time: Little change at night would compress those 7 hours into one increment of the slider, whereas high activity during the day might place 1 hour in that same increment of the slider.

Since the prototype only used one week's worth of data, the speed of marker updates was reasonable. Importing the entire dataset into the database may potentially slow down the speed of the queries, and if that is the case, then database optimization methods such as indexing need to be considered and implemented.

On a more intuitive level of understanding the visualization, it is possible for users to interpret circles incorrectly, viewing the radius to be the area used by the population, rather than proportional to the population size. The radius of the circle is not directly proportional to the population. Rather is it exponentially proportional in order to limit the circles to a reasonable size on the map. While there lacks obvious solutions to this misinterpretation in 2D, visualizing this same dataset in 3D with population bars would not only clarify the units being measured, but also allow for a more direct comparison of populations between locations.

Using unused columns from the data also has room for potential. Displaying the populations on specific building levels and APs could show patterns in how students are distributed across buildings and floors, which could determine which facilities the University should invest more into. Visualizing the average session throughput might show trends in where there poor wireless speeds. In terms of more functional improvements, including a pop-up of statistics, such as the exact number and proportion of students at that particular location, when hovering the mouse over a particular location would provide more detailed information without cluttering the view, and comparing this to the theoretical timetabled data could provide details on the actual behavior of students.

On a much more superficial level, adding a transparent layer on the map that brightens or darkens with the time of day would provide a more intuitive way of determining the time without reading the exact timestamp, and introducing an autoplay button that plays all the population changes over all times would enhance the user experience.

VI. Conclusion

The increase in social media and by extension, Internet usage, has generated data at a remarkable size and rate. Yet, we have not extracted meaningful information from most of this. This paper looked at visualising the historical data of wireless connections that University of Sydney students made during the period of 21 March 2014 to 12 September 2014, and implemented a web application by first looking at influential works and related technologies, outlining a methodology using these technologies, and then developing the visualisation. While key findings regarding the behaviour of students, such as their peak connection times, were observed and discussed, future improvements on the current implementation would increase the breadth and hence, usefulness of the information being extracted.

Acknowledgment

My deepest thanks to Jim Cook and Jai Honeybrook-Carter of the University of Sydney TechLab for their assistance in sourcing data.

References

[1] D. Price. (2015) Surprising Facts and Stats About The Big Data Industry. Accessed: 24-6-2015. [Online]. Available: http://cloudtweaks.com/2015/03/surprising-facts-and-stats-about-the-big-data-industry/

[2] P. Bradley, "Data, Data Everywhere," *Legal Information Management*, vol. 14, no. 04, pp. 249–252, 2014.

[3] T. MacPhail, "Data, Data Everywhere," *Public Culture*, vol. 14, no. 4, pp. 249–252, 2015.

[4] M. O. Ward, G. Grinstein, and D. Keim, *Interactive Data Visualization*. CRC Press, 2015.

[5] E. N. Marieb and K. Hoehn, *Anatomy & Physiology*. Pearson/Benjamin Cummings, 2008.

[6] Google Inc. (2015) WebGL Globe. Accessed: 24-6-2015. [Online]. Available: http://globe.chromeexperiments.com

[7] Khronos Group Inc. (2015) WebGL - OpenGL ES 2.0 for the Web. Accessed: 24-6-2015. [Online]. Available: https://www.khronos.org/webgl/

[8] D. Fritz. (2011) Visualizing Geographic Data with the WebGL Globe. Accessed: 24-6-2015. [Online]. Available: http://googlecode.blogspot.com/2011/05/visualizing-geographic-data-with-webgl.html
[9] Three.js Authors. (2015) Three.js - A JavaScript 3D Library which makes WebGL simpler. Accessed: 24-6-2015. [Online]. Available: http://threejs.org
[10] R. Cabello. (2015) mrdoob/three.js - JavaScript 3D library. - GitHub. Accessed: 24-6-2015. [Online]. Available: https://github.com/mrdoob/three.js/
[11] Google Data Arts Team. (2015) dataarts/webgl-globe - WebGL Globe is a platform for visualizing latitude longitude based information using WebGL. - GitHub. Accessed: 24-6-2015. [Online]. Available: https://github.com/dataarts/webgl-globe.git
[12] Google Inc. (2015) Google Maps JavaScript API Demos - Data Visualization Earthquakes. Accessed: 24-6-2015. [Online]. Available: https://embed-dot-more-than-a-map.appspot.com/demos/visualization/earthquakes
[13] Google Inc. (2015) Google Maps APIs. Accessed: 24-6-2015. [Online]. Available: https://developers.google.com/maps/
[14] Google Inc. (2015) Google Maps JavaScript API - Visualization Library. Accessed: 24-6-2015. [Online]. Available: https://developers.google.com/maps/documentation/javascript/visualization
[15] Oracle Corporation. (2015) Sun Java System Application Server Enterprise Edition 8.2 - Establishing Performance Goals. Accessed: 24-6-2015. [Online]. Available: http://docs.oracle.com/cd/E19900-01/819-4741/fygaj/index.html
[16] The University of Sydney. (2015) The University of Sydney Arts Digital Marker Data. Accessed: 24-6-2015. [Online]. Available: http://sydney.edu.au/arts/artsdigital/map/markerdata.xml
[17] PostgreSQL Global Development Group. (2015) PostgreSQL: Documentation. Accessed: 24-6-2015. [Online]. Available: http://www.postgresql.org/docs/9.2/static/sql-copy.html
[18] jQuery Team. (2015) jQuery API Documentation. Accessed: 24-6-2015. [Online]. Available: http://api.jquery.com/jquery.ajax/
[19] The PHP Group. (2015) PHP: What is PHP? - Manual. Accessed: 24-6-2015. [Online]. Available: http://php.net/manual/en/intro-whatis.php
[20] Google Inc. (2015) Visualizing data: Mapping Earthquakes with the Google Maps API. Accessed: 24-6-2015. [Online]. Available: https://developers.google.com/maps/tutorials/visualizing/earthquakes
[21] D. Storey. (2015) The HTML5 range Slider Input. Accessed: 24-6-2015. [Online]. Available: http://demosthenes.info/blog/757/Playing-With-The-HTML5-range-Slider-Input

Appendix A
Implementation Code

The implementation code can be found on GitHub:
https://github.com/invinceyble/2d-population-map

APPENDIX B
VISUAL BASIC MACRO

```
Sub CleanData()
'
' CleanData Macro
' cleans up weekly access data to just what I need

    Columns("A:A").Select
    ActiveWindow.SmallScroll Down:=1
    Range("A:A,F:F,G:G,H:H,I:I,J:J,K:K,M:M,N:N").Select
    Range("N1").Activate
    ActiveWindow.SmallScroll Down:=1
    Selection.Delete Shift:=xlToLeft
    ActiveWindow.SmallScroll Down:=-4
    Columns("D:D").Select
    ActiveWindow.SmallScroll Down:=-19
    Columns("E:E").Select
    Selection.Cut
    Columns("H:H").Select
    ActiveSheet.Paste
    Columns("D:D").Select
    Selection.TextToColumns Destination:=Range("D1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=False, Space:=False, Other:=True, OtherChar _
        :="-", FieldInfo:=Array(Array(1, 9), Array(2, 1), Array(3, 1), Array(4, 1))
    ActiveWindow.SmallScroll Down:=-191
    Range("D1").Select
    ActiveCell.FormulaR1C1 = "map"
    Range("E1").Select
    ActiveCell.FormulaR1C1 = "level"
    Range("F1").Select
    ActiveCell.FormulaR1C1 = "Apno"
    Range("G1").Select
    ActiveWindow.SmallScroll Down:=-131
    Columns("G:G").Select
    Selection.Delete Shift:=xlToLeft
End Sub
```

Appendix C
SQL Updates

```
-- (1) convert from original timestamp to useable timestamp
UPDATE data SET timestart = to_timestamp(timestart, 'XXX Mon DD HH24:MI:SS XXX YYYY');

-- (2) convert from timestamp to epoch
UPDATE data SET timestart = EXTRACT(EPOCH FROM timestart::TIMESTAMP WITH TIME ZONE);

-- (3) convert duration from hrs:mins:secs to just seconds
UPDATE data SET duration = EXTRACT(epoch FROM duration::interval);

-- (4) update type cast
ALTER TABLE data ALTER COLUMN timestart TYPE INT USING timestart::int
ALTER TABLE data ALTER COLUMN duration TYPE INT USING duration::int
```

Appendix D
List of Proportions of All Wireless Connections during 14 March to 21 March 2014

Building	**%**
Carslaw Building	9.909
Law School	6.589
Fisher Library	6.549
Jane Foss Russell Building	6.138
Wentworth Building	5.758
Eastern Ave Auditorium and Lecture Theatre	4.486
School of Economics Office, Merewether Building	4.315
Wilkinson Building	3.163
Fisher Library Stack	3.074
Chemistry Building	2.988
Education Building	2.862
Civil Engineering Building	2.295
P.N.R. Building	2.121
Services Building	1.987
Manning House	1.983
Charles Perkins Centre	1.841
Physics Building	1.615
Madsen Building	1.554
Holme Building	1.511
Biochemistry and Microbiology Building	1.508
John Woolley Building	1.484
The Quadrangle	1.422
Wallace Theatre	1.266
School of Information Technology	1.253
Electrical Engineering Building	1.123
Economics and Business Building	1.074
Institute Building	1.016
University Sports and Aquatic Centre	0.993
Edgeworth David Geology Building	0.969
Edward Ford Building	0.966
Chemical Engineering Building	0.958
Badham Building	0.923
Mechanical Engineering Building	0.873
Blackburn Building	0.826
Anderson Stuart Building	0.788
Bosch Building 1A	0.772
International House	0.766
Pharmacy and Bank Building	0.714
Brennan MacCallum Building	0.682
Sydney Dental Hospital	0.639
Margaret Telfer Building	0.540
Old Teachers College	0.538
Engineering Link Building	0.535
J.D. Stewart Building	0.515
Transient Building	0.509
The Arena Sports Centre	0.454
Macleay Building	0.454
Education Annexe	0.444
Veterinary Science Conference Centre	0.427
Seymour Centre	0.421
Bosch Building 1B	0.394
Aeronatutical Engineering Building	0.371
R.D. Watt Building	0.360
Evelyn Williams Building (Cat Hospital)	0.299
Storie Dixon Wing	0.277
Heydon-Laurence Building	0.270
Queen Elizabeth II Res. Inst. and Victor Coppleson Building	0.245
Darlington Centre	0.203
Griffith Taylor Building	0.187
Gordon Yu-Hoi Chui Building	0.161
Medical Foundation Building	0.107
Mackie Building	0.100
R.M.C. Gunn Building	0.098
R.C. Mills Building	0.086
Baxter Lodge	0.068
Sydney University Village - Building B	0.056
Boundary Lane Childcare Centre	0.046
Darlington Road Terraces	0.040
McMaster Building	0.039