THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

MEASURING WIRELESS DATA THROUGHPUT TO EMULATE REALISTIC NETWORK CAPACITY

by

Samuel Bryce Teed

School of Information Technology and Electrical Engineering

University of Queensland

Submitted for the degree of Bachelor of Engineering

in the division of Software Engineering

12/06/2017

Samuel Bryce Teed
samuel.teed@uqconnect.edu.au

12/06/2017

Prof Michael Brünig
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Brünig

In accordance with the requirements of the Degree of Bachelor of Engineering in the division of Software Engineering in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled "Measuring Wireless Data Throughput to Emulate Realistic Network Capacity".

The thesis was performed under the supervision of Dr Marius Portmann. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely

____

Samuel Bryce Teed

Acknowledgments

This project involved several new experiences for me and introduced me to many new fields of study, so I would to thank those who have helped me achieve what I have. I would like to thank my supervisor Dr Marius Portmann for all the support he has provided over the past year. Without his guidance I believe I would not have been able to complete as much work as I did and may have lost sight as to what this project is trying to achieve.

I would also like to thank Mr Anees Mohsin Hadi Al-Najjar for his help with devising and running the tests for this project. It was his past work which inspired this projects conception and allowed me to undertake this endeavour. I would also like to thank both Marius and Anees for allowing me to use some of their department's equipment, without which this project would have been near impossible.

Abstract

This project explored methods for recording network speeds and the ability to emulate that data on virtual networks. Using several machines and software tools a tester moved around the University of Queensland’s campus recording the WiFi and LTE maximum throughput along with the GPS location simultaneously. From these test a small data set of the mentioned variables was created.

Using the newly created data set a method for replaying the data on a virtual system was devised. This emulation was achieved over a single link with an error of approximately 3 mega bit per second. The data set was also shared with other University of Queensland scholars to assist with testing their new network load balancing technology, which also proved to be the inspiration for this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The world of wireless devices is ever growing larger and with this increase in technology there will certainly be problems which will arise. Currently it is almost standard for any public building to provide a WiFi connection and it is expected to also have a mobile connection to the internet (LTE/4G). Even though having a connection to one or both of these networks is an expected luxury in the developed world there is a lack of recorded data for both of these networks capabilities simultaneously.

This project looked into different ways of recording different aspects of wireless networks functionality. Details like wireless signal strength, different network access points and types of connections were explored. However the final results which this project endeavoured to produce was a data set containing information relating to the maximum data throughput of both WiFi and LTE networks along with the location at which these throughputs were measured. And all these measurements would be made simultaneously so that if the data is replayed in a virtual environment it would emulate what a device at the recorded location would expect to see in terms of network throughputs for WiFi and LTE.

The inspiration for this project came from work currently being conducted at the University of Queensland. This work involves sharing data flows over multiple links which would otherwise be sent over single links [1]. This is known as load balancing and is one case which could take advantage of the data which this project is gathering. This project was started to try and create a data set which could be used to test how well this load balancing system functions under a real world scenario.

This project also found ways of replaying the data found over virtual networks. Being able to accurately emulate the data could help any new technology confidently know their capabilities on real networks. Currently there are no other data sets like the one this project is creating. So the unique data set of GPS location along with WiFi and LTE maximum throughput recorded at the same time could be used to help improve arising technologies like load balancing. Any new or old technology which can work with WiFi and or LTE networks could take advantage of the information this project finds.

Chapter 2

Background

In this section of the report different areas which were explored for this project will be explained. The areas explored helped to gain inspiration for the kind of tests this project should be running and also to gain an understanding as to how the tests and results should be constructed.

2.1 Load Balancing

The concept for this project came about due to the current work being done at the University of Queensland related to SDN and network load balancing. In the paper '*Pushing SDN to the End-Host, Network Load Balancing using OpenFlow*' the idea of sharing data load over multiple networks is explored [1]. The paper suggests creating an architecture similar to that seen in figure 2.1.Using this architecture an existing application could continue to behave as it would using a single data path, where in reality the application is in fact receiving information via multiple paths. The paper also gives the scenario of a smartphone with WiFi and 4G capabilities as a potential user of this system.

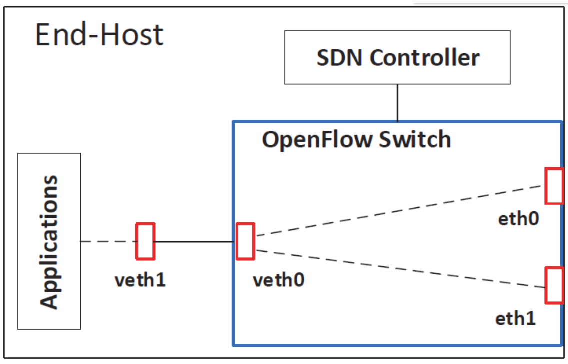

Figure 2.1 SDN System Architecture [1]

Other forms of load balancing have been created. Multipath TCP (MPTCP) is another form of load balancing which has been attempted. MPTCP is an evolution of the core internet protocol TCP [2]. TCP is the standard for creating secure connections between two hosts and has been for some time. MPTCP builds on this by allowing for multiple TCP connection to work in tandem with each other. An MPTCP session will start with the usual TCP connection and then grow by adding *subflows* which are additional TCP links across different network paths [3].

While these load balancing techniques exist they have only been shown as proof of concept architectures and have not reached far into the public market. A reason for this being the difficulty in testing these systems against realistic scenarios. This project aimed to help alleviate this difficulty by creating a data set of appropriate data.

2.2 Software Defined Networks

SDN is a new networking architecture which aims at improving a lot of the short comings that current networks are facing. Modern networks are reaching sizes which were not anticipated when computer networks were first being designed. Major issues facing modern networks are adapting to the changing and growing modern market. "Traditional network architectures are ill-suited to meet the requirements of today's enterprises, carriers, and end users" [4]. To combat this the SDN architecture aims to abstract a lot of the policies and protocols usually handled by individual nodes, like routers and switches, to a more programmable level which can be shared among a network much more efficiently [5].

The generally accepted definition as to what a SDN network should be comes from the Open Networking Foundation (ONF). ONF released a White Paper [4] on SDN, in which SDN is defined to be structured with 3 layers. Figure 2.2 shows these three layers and how they are linked to one another.

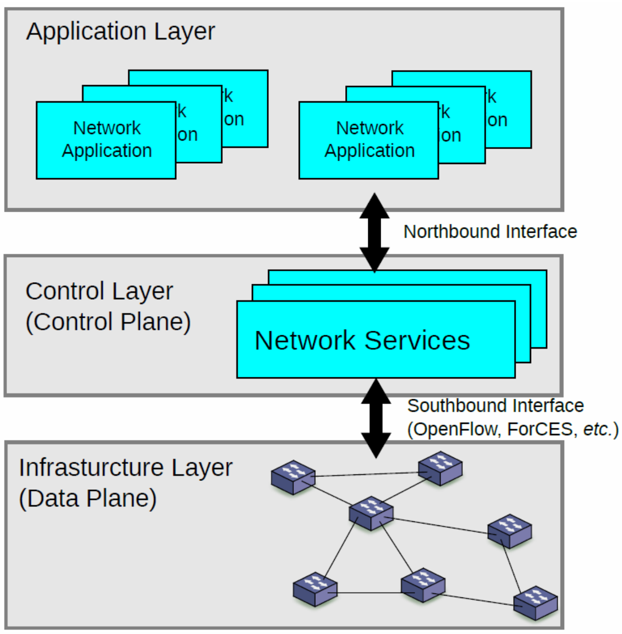

Figure 2.2 Three-layer SDN Architecture [6]

The lowest lay is the *Data Plane*. This layer is just concerned with the nodes and passing data between them. Using SDN this layer becomes quite simple. The middle layer is known as the *Control Plane* and as the name suggest this layer is in control of the data plane and tells the data plane how and where data should be sent. The final layer is the *Application Layer* which consists of network applications which can view and control how the other layers below it behave [6].

This brief introduction into SDN helps to appreciate the work being conducted at the University of Queensland and gain a better grasp as to how SDN is able to implement load balancing. The work at UQ mostly focuses on the *Southbound Interface* that is communication between the control plane and data plane.

2.3 Prior Art

Similar projects to this do exist, a project conducted out of MIT looked into network connections and compared MPTCP's performance against these connections [7]. The MIT paper however differs to the goals of this project as it did not record maximum throughput of a connection for long periods of time, but rather tested for a short amount of time in many locations. The MIT paper was also more concerned with the type of data being transferred. The method for collection was drastically different as the MIT paper employed using a smartphone application which was given to the public to use and test with. The application would test the user connection for both WiFi and LTE, and share the information with the user and the MIT researchers. After several months the data collected was then used to emulate network traffic and test a MPTCP architecture.

There are also existing network datasets which the general public can gain access to. For example the *Mahimahi Toolkit* contains several examples of web traffic which can be used to emulate networks [8]. Mahimahi is a toolkit which can both record and replay web traffic which is useful when trying to test how system handles internet traffic. This is a somewhat different project to what this project aims to do, however tools like Mahimahi could be used together with the results of this project to produce more realistic results. As this project is solely concerned with gathering information on networks maximum data throughput, information like what type of data a network is typically handling is not known.

While projects like those given previously exist none were found to contain the type of results this project aims to produce. A data set of WiFi, LTE and location data recorded simultaneously is the perfect set of information needed to test systems like load balancing. Apple is also implementing systems to take advantage of the fact that smartphones are now more likely to be within WiFi and LTE range using there WiFi assist technology [9]. Apples WiFi assist technology is an integrated functionality in Apple smartphones which tries to switch the phone between WiFi and LTE at times when WiFi connection begins to weaken. The results of this project could be used to test and improve systems such as WiFi assist.

Chapter 3

Methodology

This section of the report will explain the steps taken to collect the WiFi and LTE data as well as what was needed to be done for the collected data to be emulated virtually. Several pieces of hardware were used in the collection of the data along with multiple software items which were used in both stages of the project. Along with the hardware and software necessary for the project this section will detail how these components worked together.

3.1 Data Collection

The way in which data collection was undertaken involved several pieces of hardware and software. There were 3 main hardware components that were needed to conduct tests. One GPS device to collect location data and two machines which can connect to the university network and send data between each other. Of the two machines one would be hard wired to the universities network and behave as a host. The second machine is a laptop with wireless capabilities and behaves as a client who initiates the tests.

For this project 2 laptops were used as clients, 1 laptop used to collect WiFi results, and a second laptop equipped with LTE capabilities to collect the 4G results. To give a laptop LTE capabilities a USB 4G modem was employed. For this project a Telstra MF823 Prepaid 4G USB device was used.

Along with collecting wireless data collection location data was also collected. To do this a separate GPS device was used along with the other pieces of hardware. For this project a mobile smart phone was used along with an application to record the location data during tests.

There are 2 main software items used to collect data during tests. The first is runkeeper, a smart phone fitness application, the other piece of software is a custom python script. The smart phone application is used to record GPS data. It works like a stopwatch watch, with simple start and stop buttons to begin and end recording GPS location. Once told to stop recording the application will save the completed route to both the smart phone as well as to the runkeeper servers which can be accessed at any time via the runkeeper website [10].

The python script is used to record several different details about the current network being testing. It does this by using 3 different software tools. The first and most simple is the *ping*[16] function which sends small packets of data between the host and client machines and measures the time delay between sending and receiving the packet every seconds, this is known as a networks latency. The second tool is *iw* [11], which is a networking tool capable of retrieving information about the currently connected network. This project was mostly interested in viewing information about the networks wireless signal strength and currently connected router devices, so as to see when the machine loses signal and when it changes router connections.

The final tool used in the python script is the most important tool and it is call *iperf* [12]. Iperf is a standard tool for measuring a networks throughput. To use iperf the tool must be installed on two machines. One machine must first run iperf and declare that it will be acting as a host. Then the second machine can run iperf and declare it wishes to connect to the first machine by giving iperf the location of the first machine, usually the IP address of the machine is used for this. Once this is done the two machines will send and receive as much data between one another that the network can handle. This is done in given intervals and by viewing how much data was sent in an interval iperf can return an accurate estimate of what the throughput of the network connecting the two machines currently provides.

The following code is the command used to initiate an iperf client side session.

```
iperf3 -c 10.82.31.86 -i 1 -t 600 -p 5202 -l 4K
```

The command specifies the IP location of the desired host to connect to first. Then next number is '1' which is specifying the interval in which to measure, in this case one second. The 600 is stating how long the session should run in seconds, therefore this session will run for 10 minutes. The '5202' is the port number the session should work on. Because 2 iperf session will be running during test there needs to be two ports open on the host machine to run each client session from. The final value '4K' is the size of the read/write buffer in kilobits. This value is set smaller than default so that smaller throughput readings can be made.

The python script functions by creating 3 separate threads, one for each of the above tools. The script will then wait for each thread to be ready before starting each tool. It does this to ensure all tools are running simultaneously and that they all begin at the same time. Iperf is initialised to measure the throughput each second, to match this ping is given the same interval of one second. The iw tool does not run off an interval and so needs to be manually executed every second to match up with the other tools. This done in the python script with a for loop and a wait function.

One other minor software application is used when running tests. To connect to the iperf host machine which is running on the universities network the client machine must also be connected to the network. This causes no problems for the machine testing WiFi as it will be connect to the universities WiFi network. However the machine testing LTE will not be connected to the network. To allow the LTE testing machine to connect to the universities network the application Cisco AnyConnect [13] is used, which allows a machine to connect to the universities network from any internet connection.

To conduct a data collection test the host machine, first a computer connected to the universities network via cables starts an iperf session as a host. Then the tester will take both laptops to the test location, somewhere on campus with wireless connection. The LTE machine will start Cisco AnyConnect and connect to the universities network. Then almost simultaneously the GPS application runkeeper will be started along with the python script on both laptops. The test runs for 10 minutes, in which time the tester will move to several locations to allow for a variety in results. Once the 10 minutes is up the python script saves all results to 3 different text files, one file per software tool. The runkeeper application is stopped manually and the GPS data is collected at a later time from the applications website. All files are collected together, relevant information is extracted and results are interpreted.

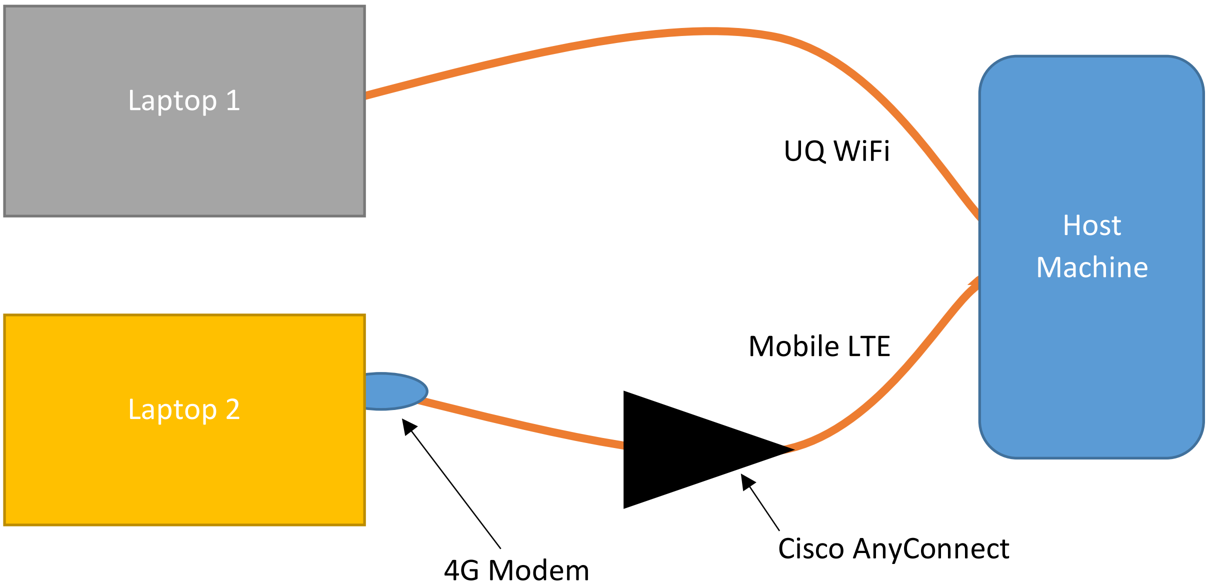

Figure 3.1 Test Setup

3.2 Data Emulation

To emulate the results found in the data collection stage of the project only a single machine is needed. The application *Mininet* [14] is used to create a virtual network running on the machine. The virtual network will contain 2 hosts connected via a single switch, which can be seen in figure 3.2. One of the hosts will, like with the data collection stage, run iperf as a host. Once one of the virtual hosts is hosting an iperf session a python script is executed which will start the client side of the iperf session from the second virtual host. This means there is an iperf session running between the two virtual hosts. The python script also changes the max throughput of one of the hosts so the connection between the two hosts closely matches that of the recorded data collection results.

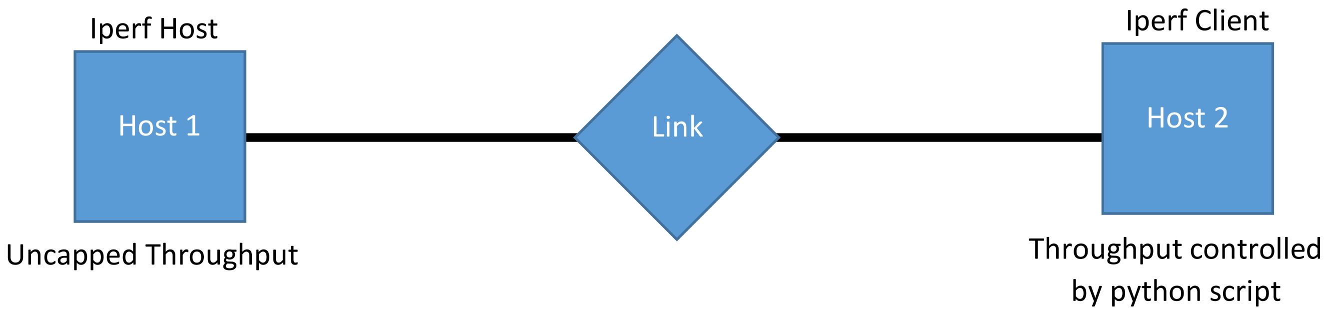

Figure 3.2 Emulation Topology

Like with the data collection python script, the script used for the data emulation takes advantage of separate threads to ensure the iperf client begins at the same time as the changes to the virtual networks throughput starts. The way in which the python script controls the throughput of the virtual network is through the following line of code.

```
Sudo tc qdisc replace dev int_name root tbf rate x latency 1000ms burst 10K
```

Using the *tc* [15] software tool the python script is able to set the maximum throughput a host allows. The above line of code is within a for loop which also contains a wait function which means the throughput will only be updated every second. In the above code 'int_name' is the host which is having its throughput changed and 'x' is the amount the hosts maximum throughput is being changed to.

The emulation session run for 10 minutes, just as the data collection sessions did. Once the session is complete the collected iperf data is saved to a text file. The throughputs measured from the virtual emulation test are then compared to those found during the data collection stage to find differences between the two data sets and determine the accuracy of the emulation.

Chapter 4

Results

In this section of the report results from all complete test will be shown. The results will be displayed as line graphs showing the measured throughput over time. Average throughput over the entirety of the tests will also be given as bar graphs with the standard error between the real measured values and the measured emulated values. Figure 4.1 shows the four route used for each of the tests.

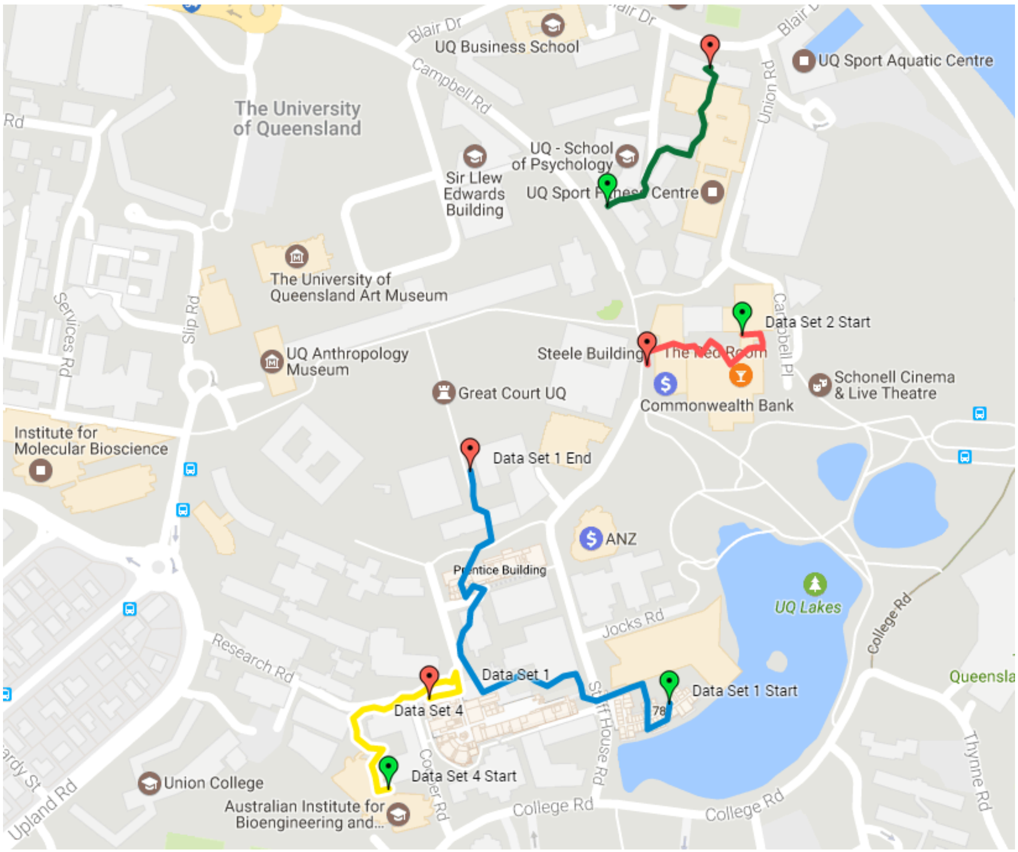

Figure 4.1 Generated Map of all routes used for testing.
Data Set 1 coloured Blue; Data Set 2 coloured Red; Data Set 3 coloured Green; Data Set 4 coloured Yellow

4.1 Data Set 1

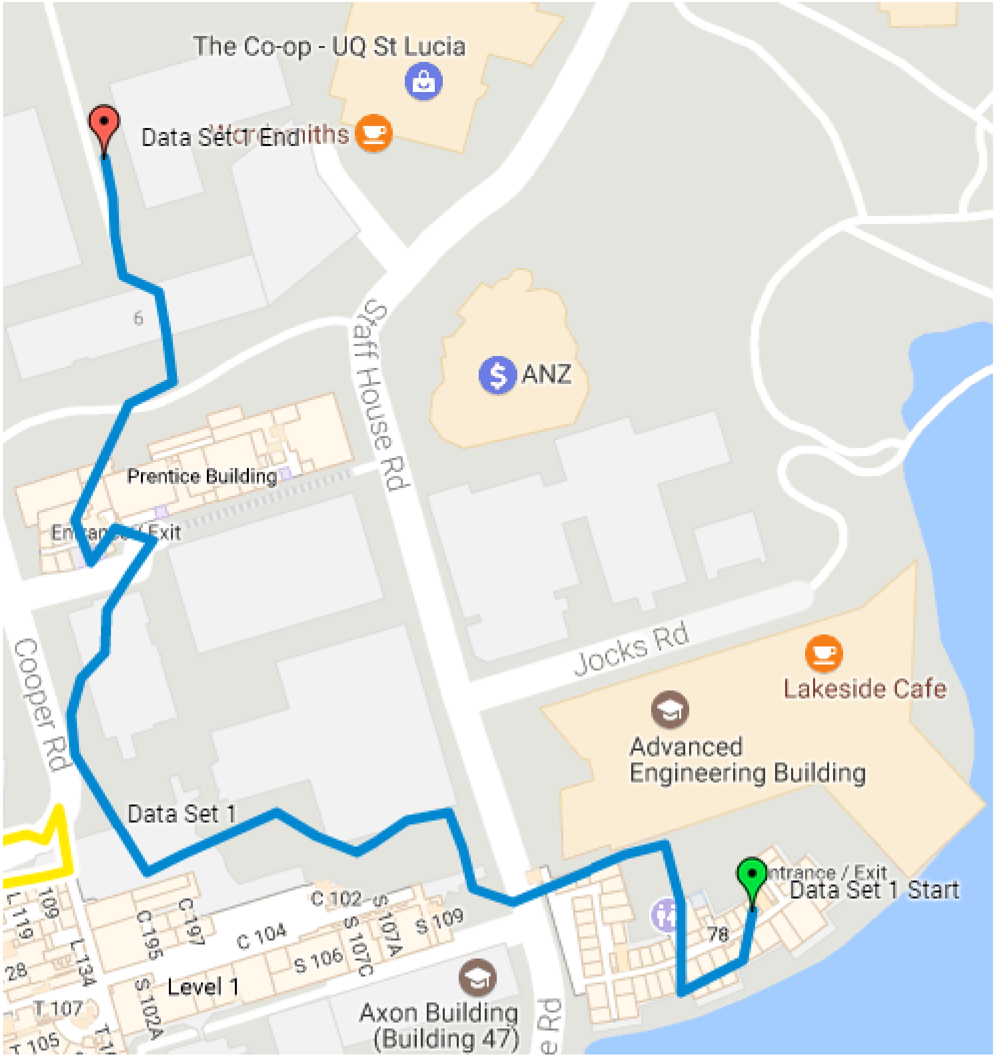

Figure 4.2 Generated Map of route used for Data Set 1

Table 4.1 Data Set 1 Test Details

Test Details	
Date:	20/03/2017
Time:	01:23 PM – 01:33 PM
Location:	UQ Engineering Department East
Route Description:	The route began inside and downstairs of the General Purpose South building. The tester moved to level 2 before stepping outside and crossing a road to the Hawken building. The tester moved through this building before exit and walking up Cooper Rd to the Prentice Building. The tester then left this building moving towards The Great Court and finishing the test at the entrance. This route contained entering and leaving several buildings as well as being outside for extended periods of time.

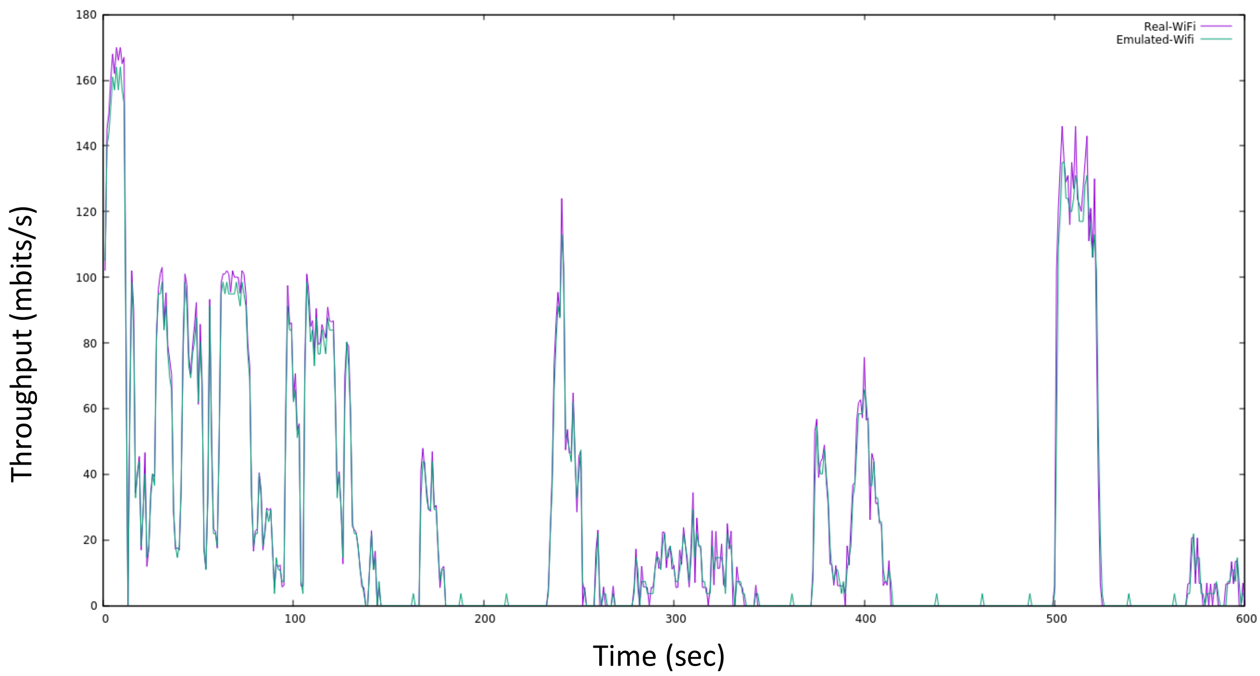

Figure 4.3 WiFi Measured Values for Data Set 1

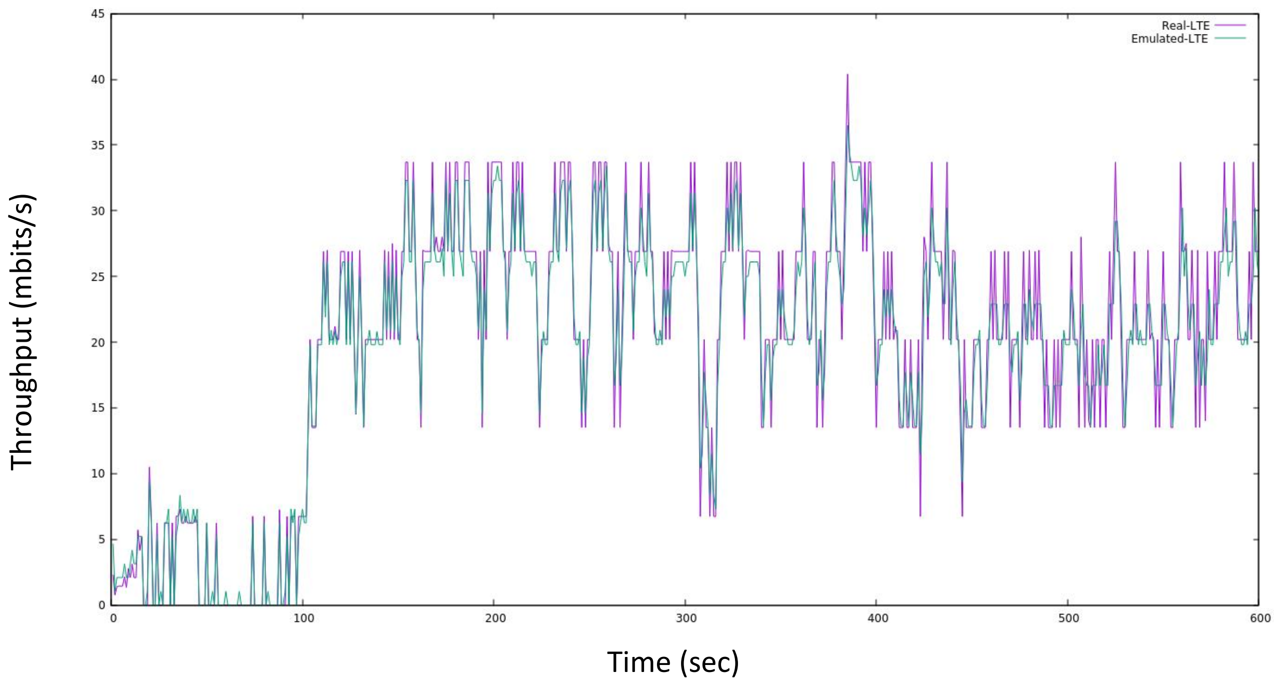

Figure 4.4 LTE Measured Values for Data Set 1

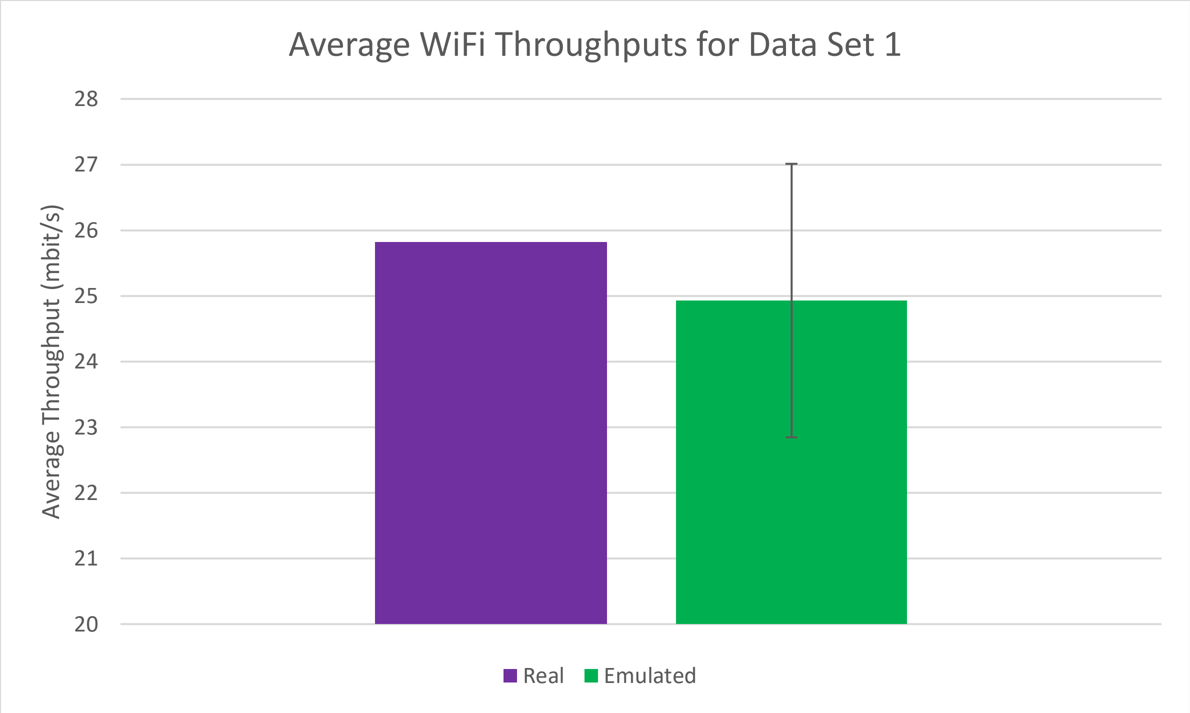

Figure 4.5 WiFi Average Values for Data Set 1

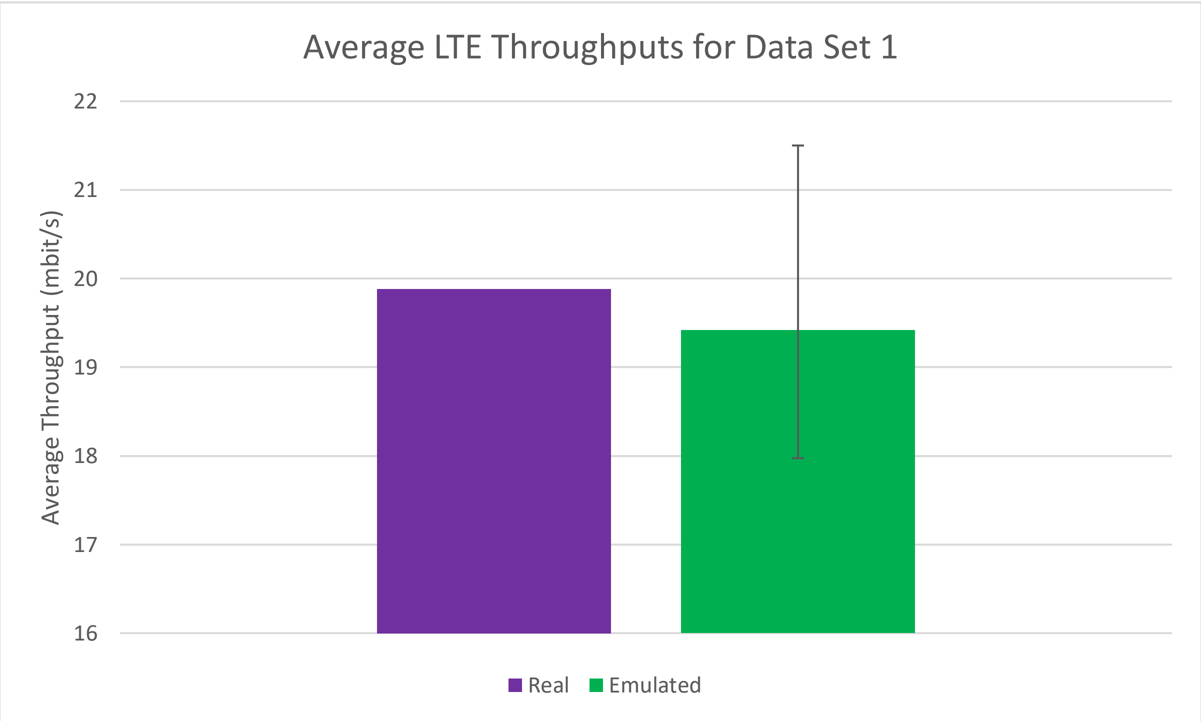

Figure 4.6 LTE Average Values for Data Set 1

4.2 Data Set 2

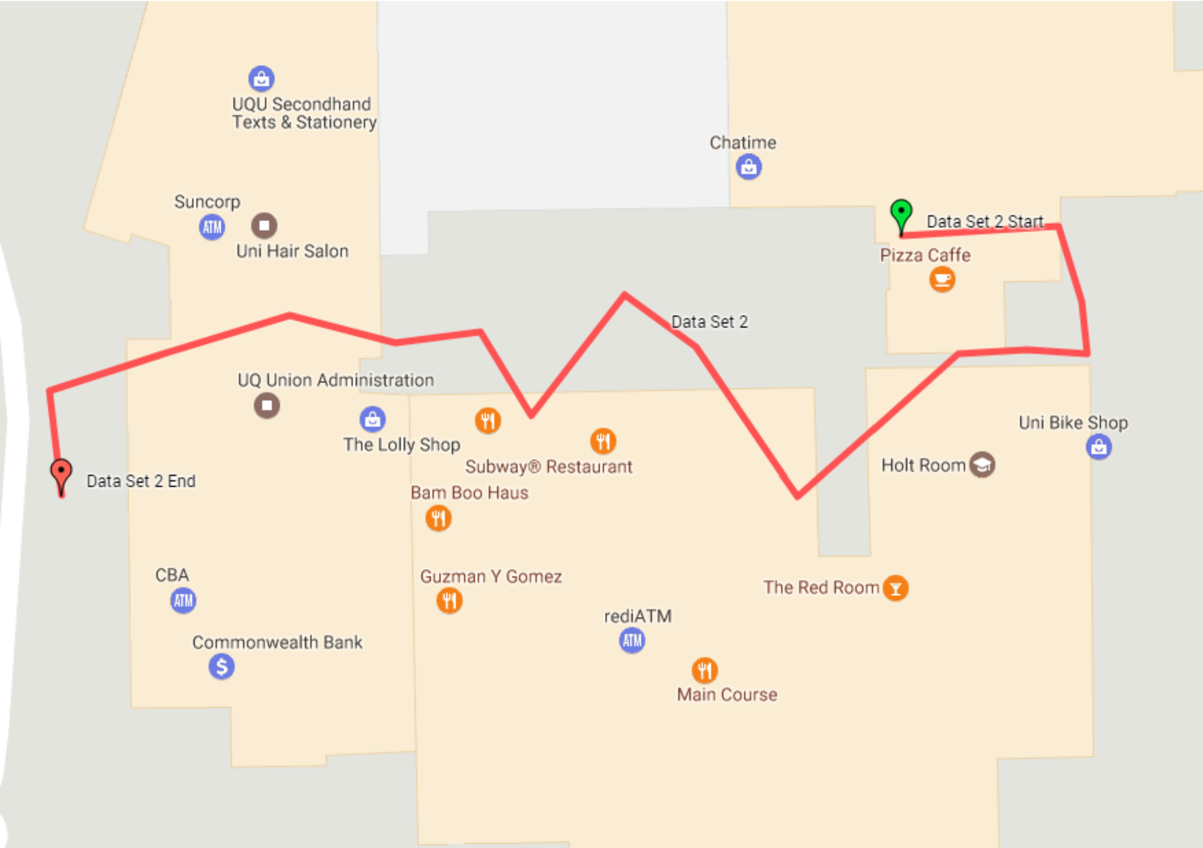

Figure 4.7 Generated Map of route used for Data Set 2

Table 4.2 Data Set 2 Test Details

Test Details	
Date:	17/04/2017
Time:	05:16 PM – 5:26 PM
Location:	UQ Food Court
Route Description:	The route began inside the Schonell Theatre lobby. The tester then moved outside through the Pizza Café before entering the main hall. After moving around the main hall the tester left heading towards The Lolly Shop. Finally the tester moved around to the Commonwealth Bank before ending the test. This route maintained close proximity to buildings for the entire route. With slight potential to obstruct connection to cell towers.

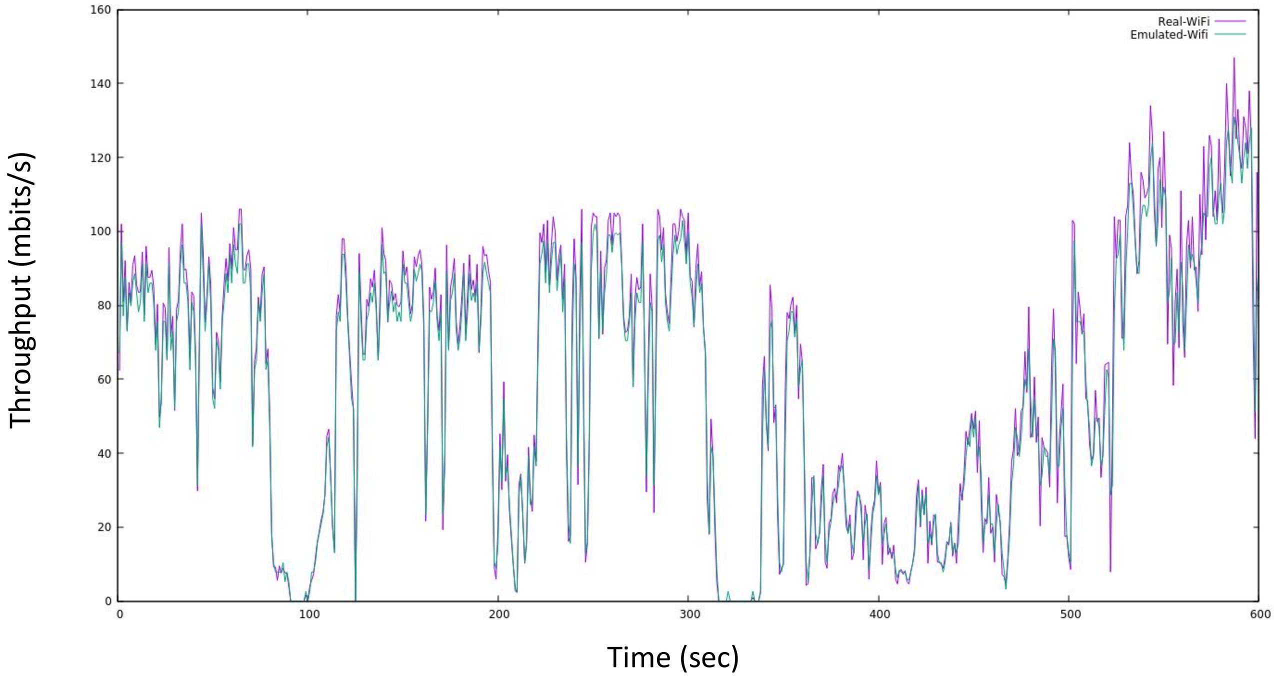

Figure 4.8 WiFi Measured Values for Data Set 2

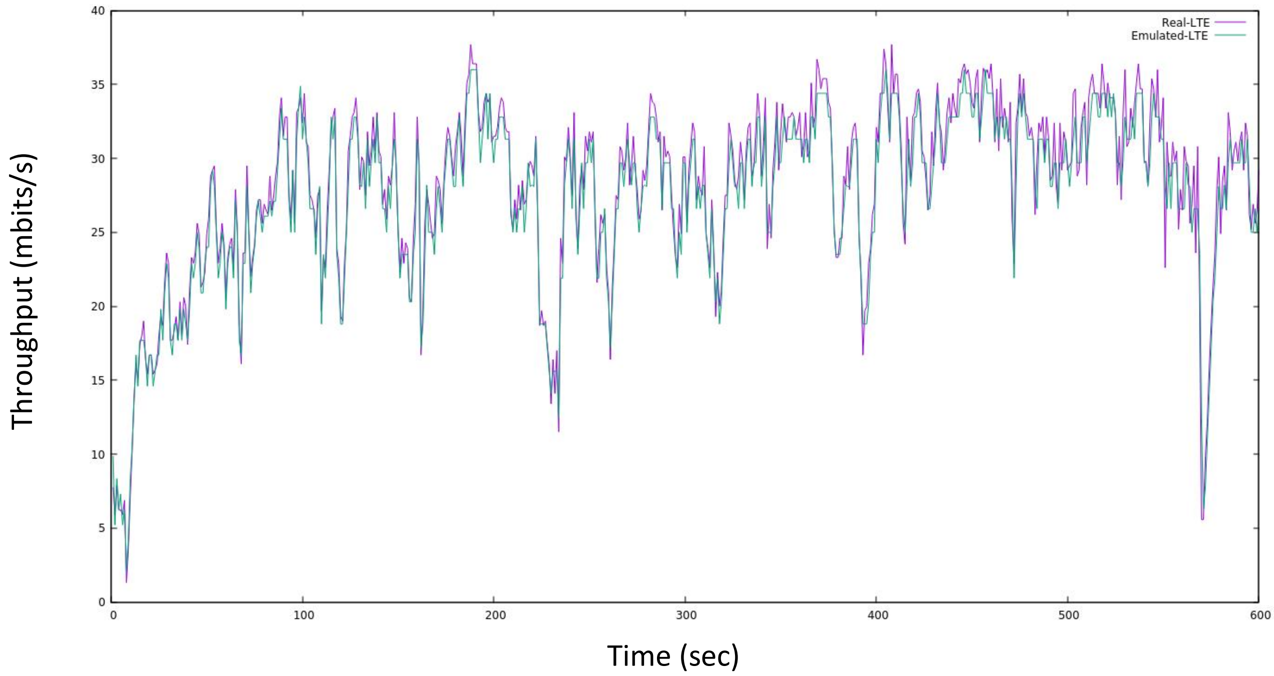

Figure 4.9 LTE Measured Values for Data Set 2

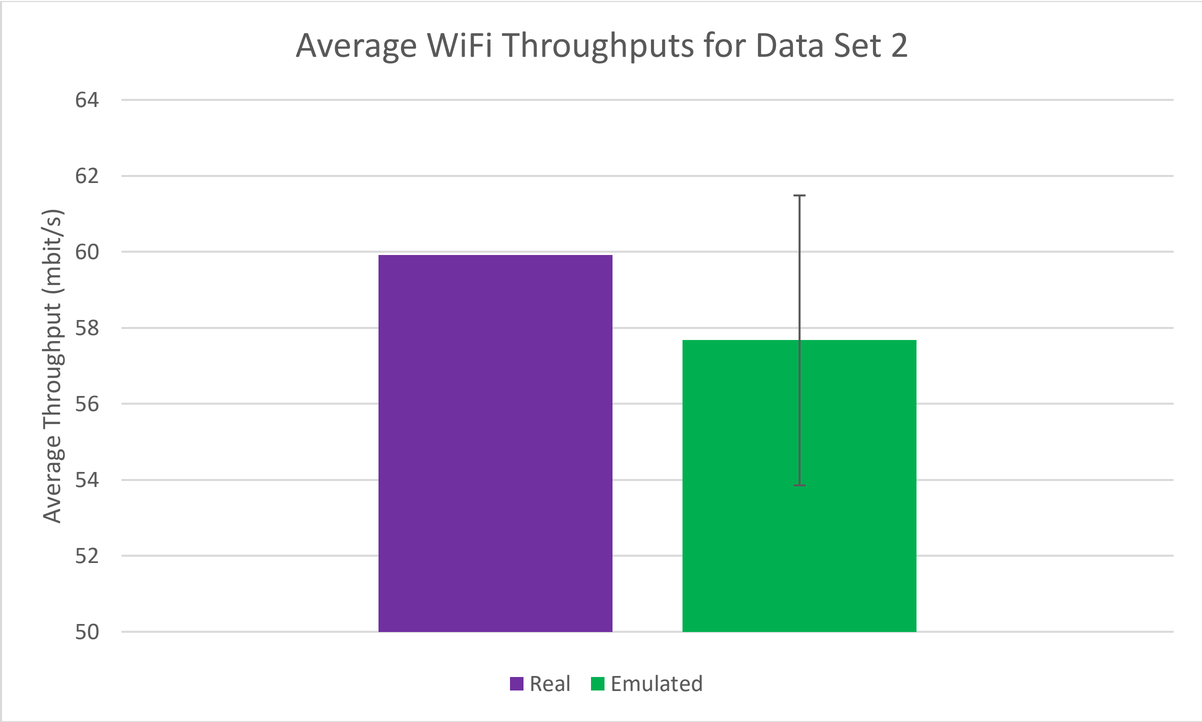

Figure 4.10 WiFi Average Values for Data Set 2

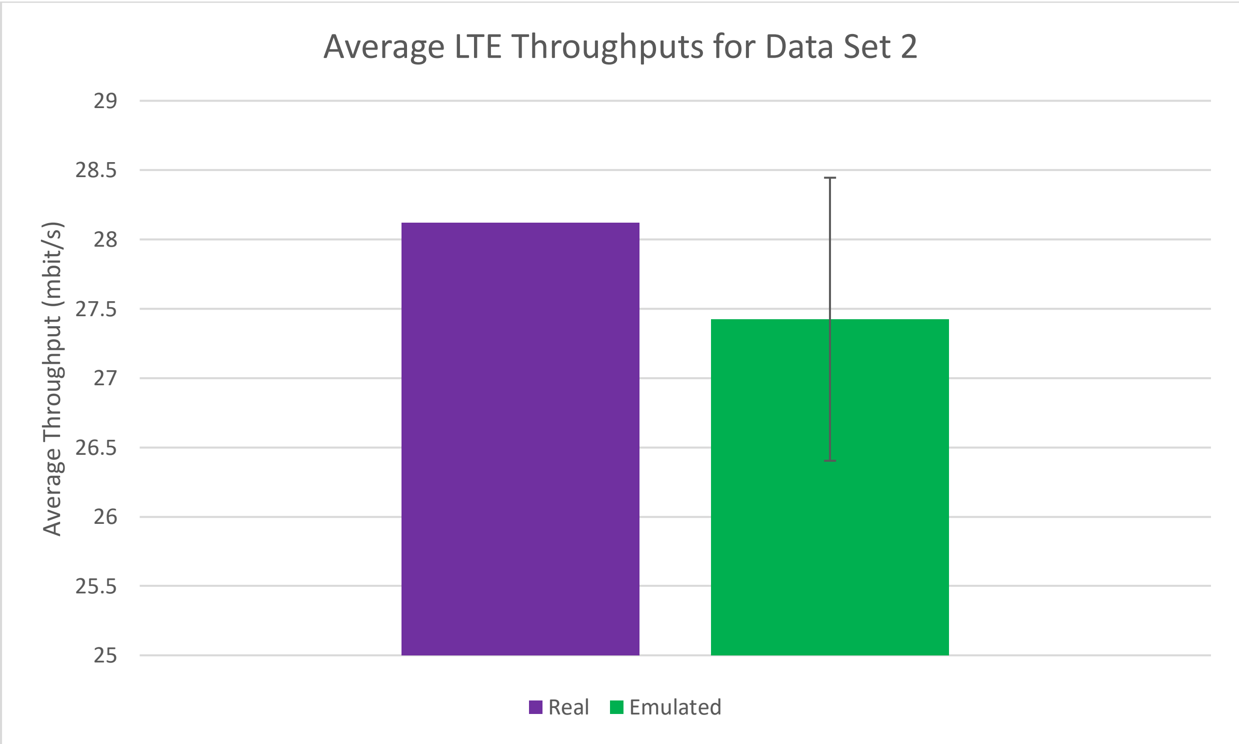

Figure 4.11 LTE Average Values for Data Set 2

4.3 Data Set 3

Figure 4.12 Generated Map of route used for Data Set 3

Table 4.3 Data Set 3 Test Details

Test Details	
Date:	18/04/2017
Time:	01:36 PM – 01:46 PM
Location:	UQ Fitness Centre
Route Description:	The test began inside the Social Science Building lobby. The tester then moved down stairs before exiting the building on ground level. The tester then walked around the Social Sciences Annexe building heading towards the Fitness Centre entrance. The test then moved to the Connell Building and entered into the lobby. The tester then moved down stairs inside the Connell Building before ending the test. This test involved moving between several buildings while remaining outside. As surrounding building were quite tall there was high chance of connection to cell towers to be obstructed.

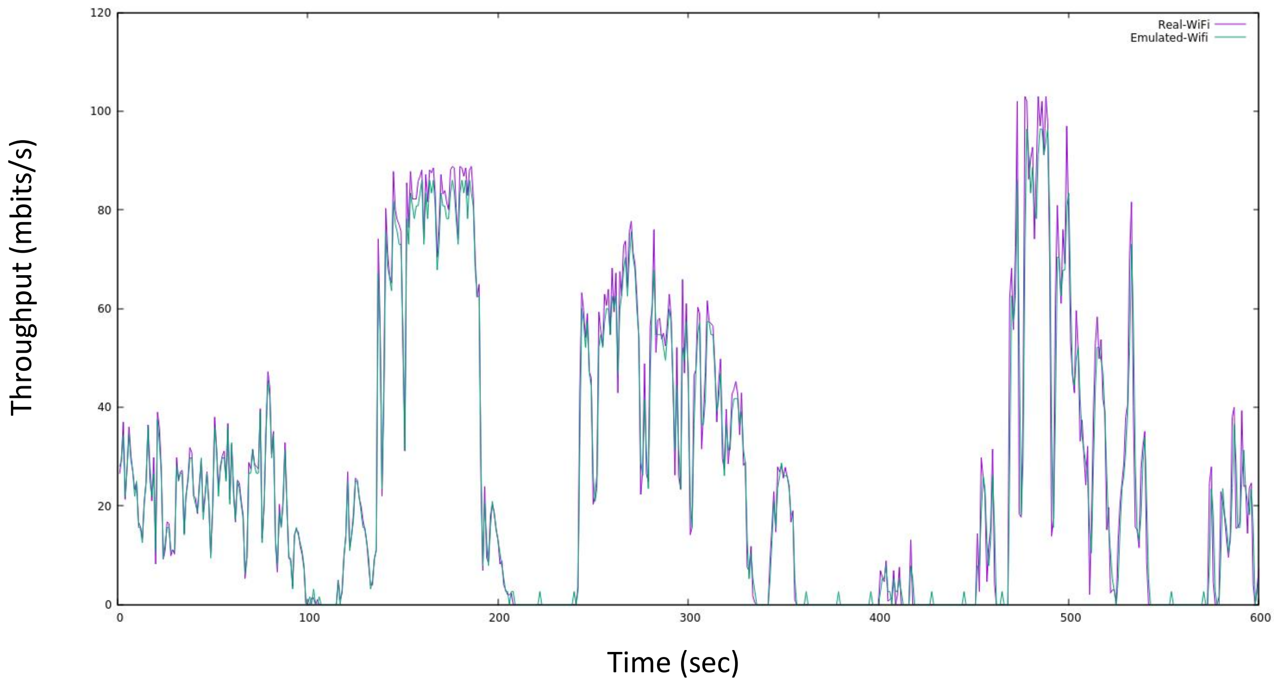

Figure 4.13 WiFi Measured Values for Data Set 3

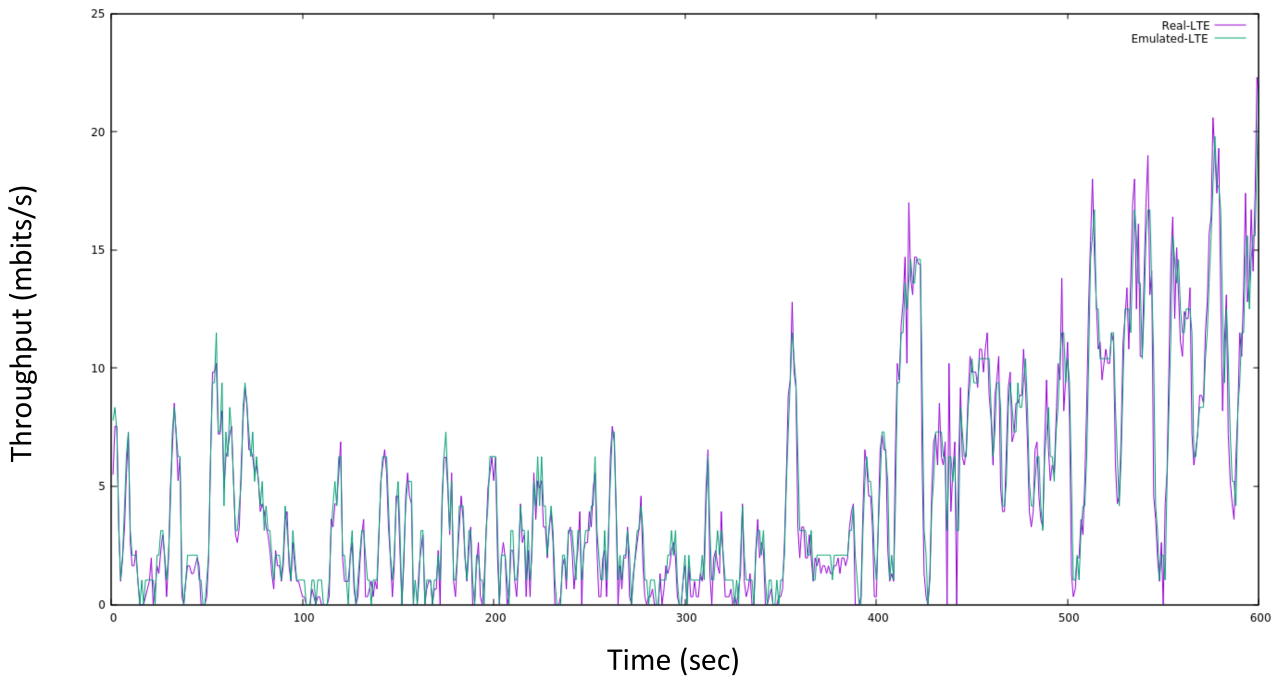

Figure 4.14 LTE Measured Values for Data Set 3

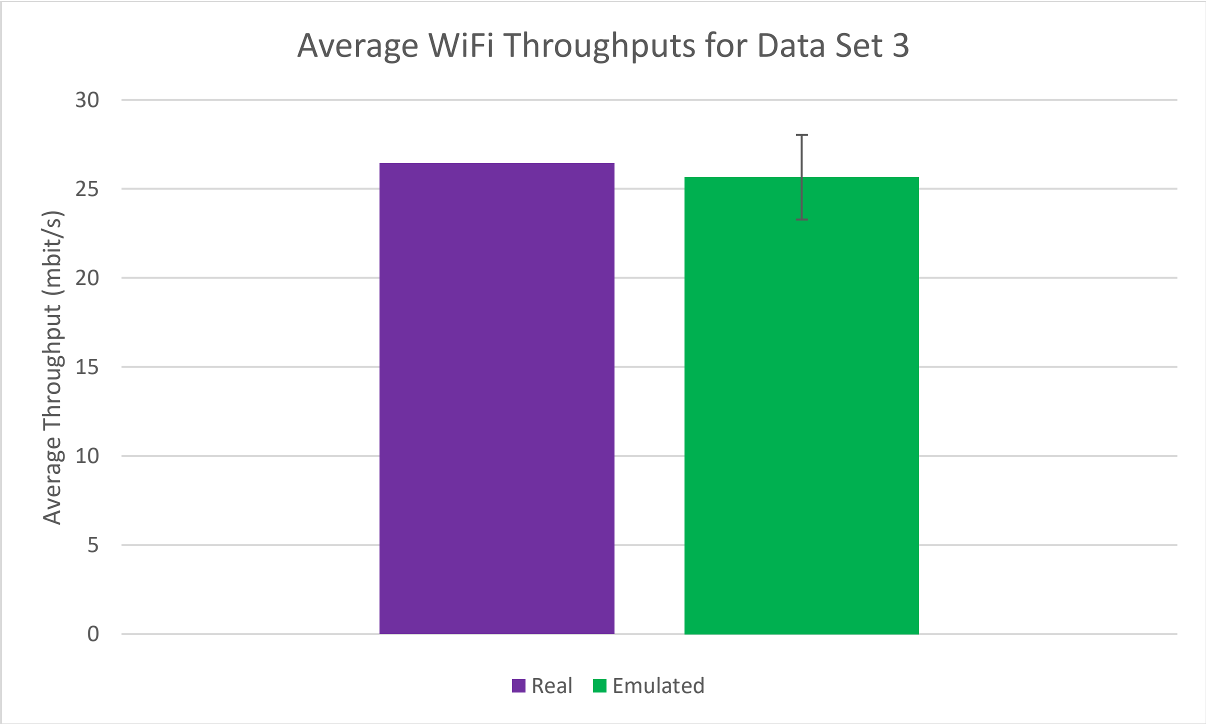

Figure 4.15 WiFi Average Values for Data Set 3

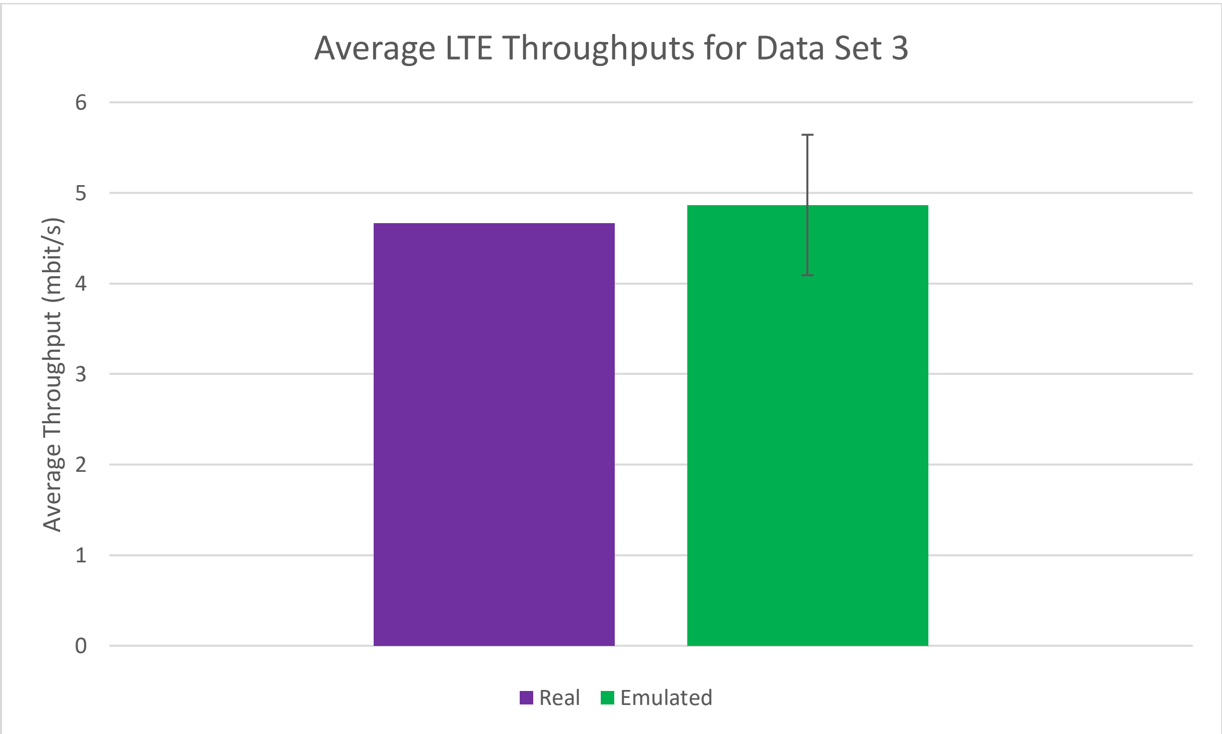

Figure 4.16 LTE Average Values for Data Set 3

4.4 Data Set 4

Figure 4.17 Generated Map of route used for Data Set 4

Table 4.4 Data Set 4 Test Details

Test Details	
Date:	18/04/2017
Time:	05:10 PM – 05:20 PM
Location:	UQ Engineering Department West
Route Description:	The test began outside near Café Nano. The tester then moved towards Research Rd before stopping at the Cooper Rd intersection. The tester then crossed the road and entered the Hawken Building before then entering the Dorothy Hill Library and ending the test. This route involved being outside for the majority of the test. With little to obstruct connection to cell towers before entering the library at the end of the test.

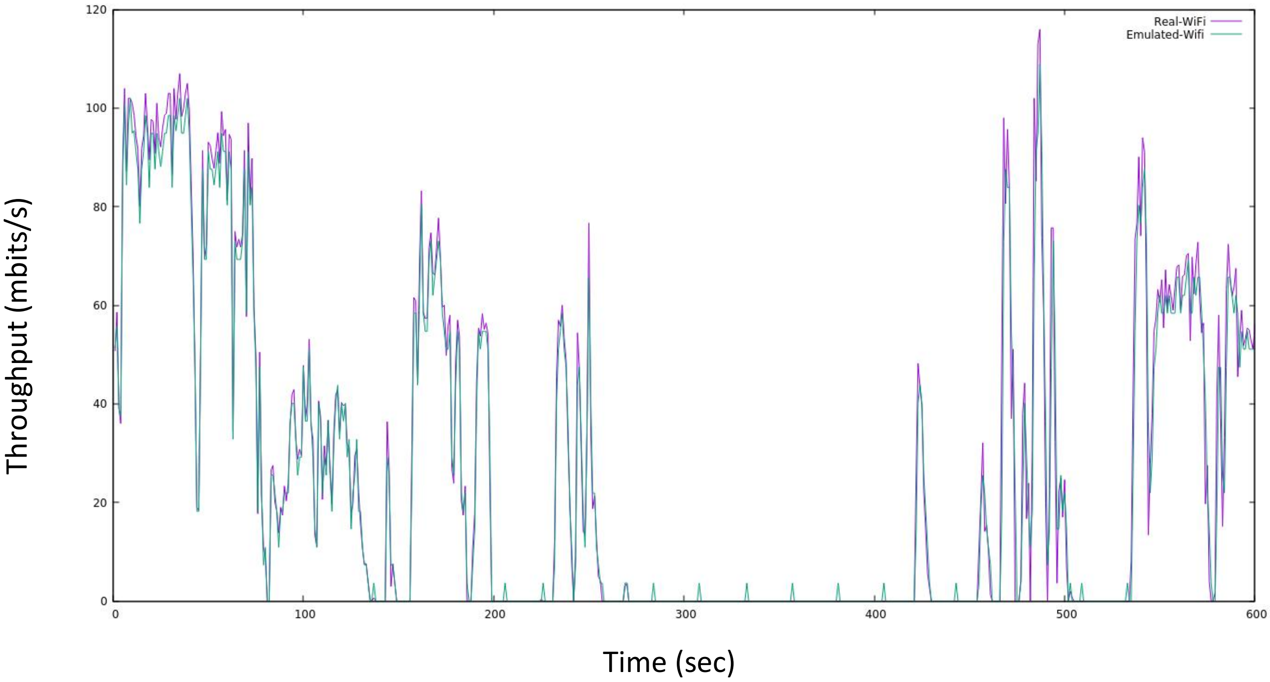

Figure 4.18 WiFi Measured Values for Data Set 4

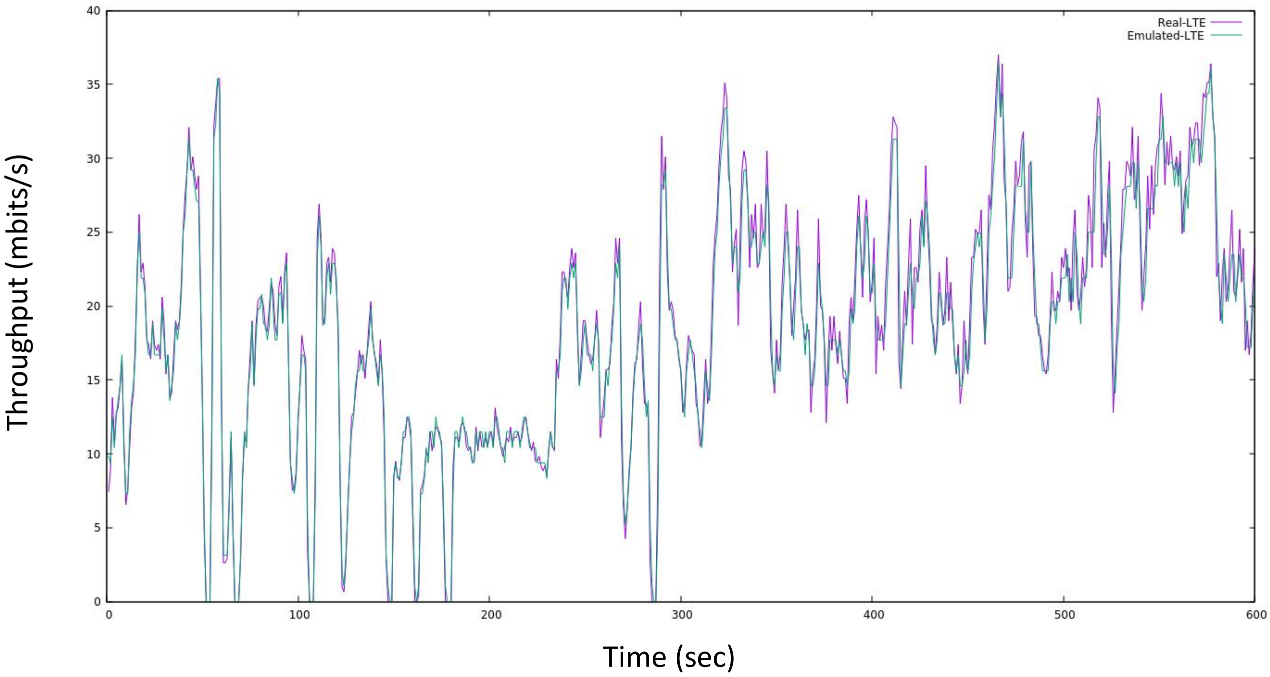

Figure 4.19 LTE Measured Values for Data Set 4

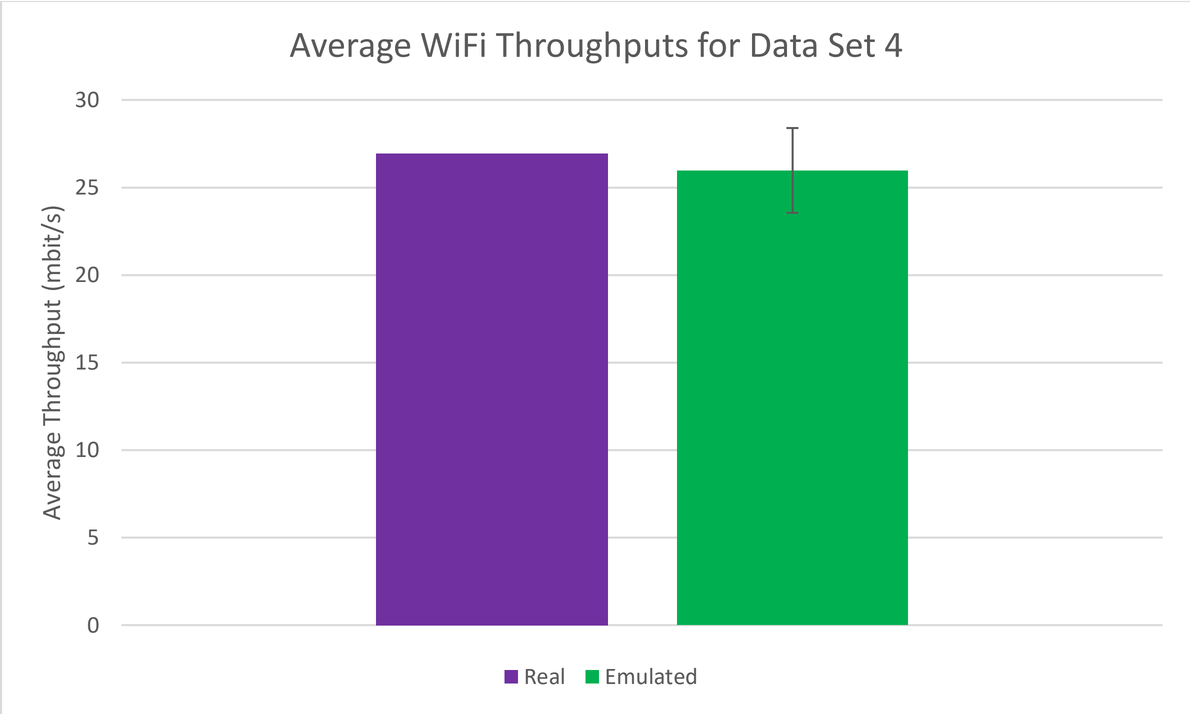

Figure 4.20 WiFi Average Values for Data Set 4

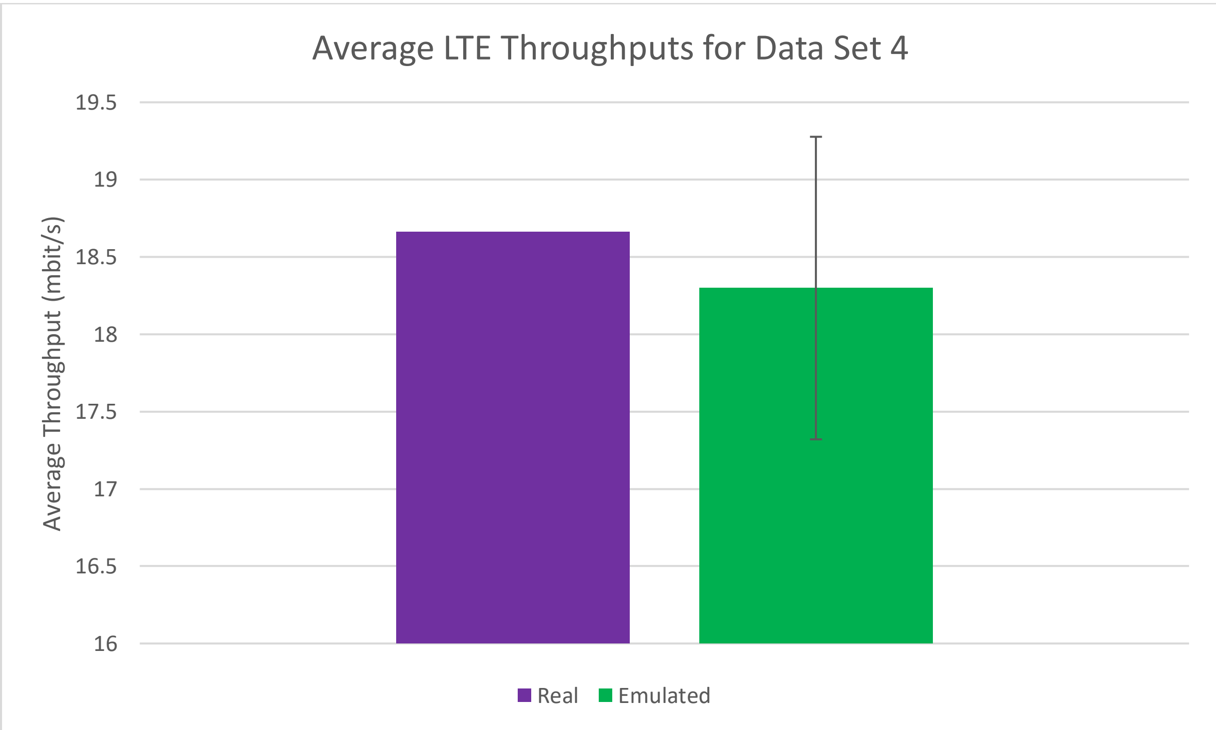

Figure 4.21 LTE Average Values for Data Set 4

4.5 Summary

Overall there were 4 tests completed which resulted in a decent amount of throughput for both WiFi and LTE. Across all tests a variety of locations were explored. These locations allowed for different levels LTE connection which helped to vary the LTE throughput. Several building were traversed through to ensure changes in network access points during tests. Below in tables 4.5 and 4.6 is a summary of all the average throughput values measured across all tests.

Table 4.5 Average WiFi Values Across all Tests

Test #	Average Real WiFi (mbit/s)	Average Emulated WiFi (mbit/s)	Average Error (mbit/s)	Percentage Error (%)
1	25.82	24.93	2.08	8.07
2	59.92	57.67	3.82	6.37
3	26.46	25.66	2.38	8.99
4	26.93	25.97	2.42	8.99

Table 4.6 Average LTE Values Across all Tests

Test #	Average Real LTE (mbit/s)	Average Emulated LTE (mbit/s)	Average Error (mbit/s)	Percentage Error (%)
1	19.88	19.42	1.44	7.27
2	28.12	27.42	1.02	3.63
3	4.66	4.87	0.78	16.63
4	18.66	18.30	0.98	5.24

Chapter 5

Discussion

For this section of the report the results from the previous sections will be discussed and compared to the expected out comes for the project. Improvements to the methodology as well as possible future work which could be conducted will also be explored.

5.1 Data Collection

The main focus of the collection stage of the project was to ensure the quality of the data gathered was high. To this end several tests were run before the results found in the previous section were achieved. When searching for a test area which would produce high quality results the major factor looked for was as much connection time between the client machines and host machine as possible. This is because if the client can't connect to the host and iperf is just recording zero values this will lead to disappointing results since emulating 0 throughput is a trivial matter.

As can be seen in the line graphs of the results section a majority of the time during the tests both the WiFi client and the LTE client are reading values other than 0. This was achieved by finding locations that had strong WiFi connections. From the line graphs it can be seen that WiFi connections were quite sensitive which made finding adequate tests quite difficult.

During initial tests it was quickly noticed that LTE throughputs were while sporadic, stayed within certain bands no matter where the test was taking place. Only locations which contained a heavy amount of obstructions to cell tower connections resulted in different throughput values, as can be seen at the start of test 1 and during most of test 3. Because of this test were mostly modified to ensure a WiFi connection was maintained for the majority.

Overall the data collected during the final tests were of a sufficiently high quality. However due to the difficulty of finding location which allowed for large amounts of movement without long section of no connection the quantity of data collected was not as high as anticipated.

5.2 Data Emulation

The purpose of the emulation stage of the project was to show that the data collected could be accurately replayed using virtual networks. This part of the project was met with little resistance as the emulations worked without too much trouble. From the line graphs of the results section it can be seen that the emulations were able to mimic the measured results quite closely.

While the emulations weren't always returning the exact measured values, the emulation was able to increase and decrease the throughput by similar amount to that of the real tests results at appropriate times. As can be seen in the bar graphs and the summary tables in the results section all emulations were able to result in a similar average throughput for most test.

One stand out results however is that of the LTE results for test 3 which had a much higher percentage difference then the rest of the tests. The reason for this is because that tests values were all quite low. This meant the usual 1 mega bit average difference seen in other LTE tests resulted in a much higher percentage difference.

5.3 Project Performance Review

There were two main goals of this project; collected realistic data throughput for WiFi and LTE networks, and the accurate emulation of these throughputs on virtual networks. While both goals were achieved the quantity of results were not as high as expected due to several problems which were experienced during the project.

The main issue which occurred during the project was locating area on campus that would present high quality results for tests. This proved more difficult than first thought as in many location it did not take much movement before WiFi connection was lost.

Another issue which made the first problem even more critical was that the software tool iperf seemed to have difficultly continuing a session once connection had been lost. It was noticed on several occasions that while running an iperf session if a connection to WiFi was lost and then regained, iperf would not notice the reconnection for an extended period of time during which iperf would continue to read a throughput of 0 even though the machine had regained connection to the network.

While it never caused any problem a constant concern was the over use of LTE data. A Telstra 4G modem was being used which came with a cost of approximately $10 per gigabyte of data used. A large chunk of data was bought early on in the project which was almost entirely used by the end of the project. To ensure as little data was wasted test routes were always trialled with just WiFi running as LTE results rarely changed depending on location. This sometimes meant spending more time in a location or walking a test route more times than necessary.

5.4 Future Works

If continued to be worked on more areas for test would need to be conducted. Growing the set of data would be the main goal as it has already been shown that the collection and emulations stages are possible. However if this project was to be continued there are also some improvements to the methodology which could be implemented.

During this project only throughput over a TCP connection was measured. If further work was to be constructed the testing of UDP and other connection types would be considered to ensure a wider range of results. This project was also only concerned with knowing the maximum throughput. Some effort could be put into recording the type of data being typically transferred through a network to then better emulate network behaviour.

The current set up of using two laptops to measure data throughput and a phone to track location can be overly cumbersome for the tester. If the project was to be continued a point of interests would be to explore how to minimize the equipment needed for a tester to conduct tests. The idea of placing all aspect of the test into smart phones through the use of applications were briefly considered however such work would require considerable amount of time to complete.

Another possible change to the test set up would be moving the current iperf host out of the UQ network and to a public web server. Doing so would mean the use of the software application Cisco AnyConnect would no longer be needed for the LTE machine as the iperf host would be accessible to anyone with an internet connection. This would also mean tests could be conducted anywhere, even off campus. While this change would bring some improvements there would be a large issue of cost in hosting a server like this. Since iperf is testing maximum throughput, a large amount of data is being transferred and so the cost of using a web server would also be large.

The final change which would be considered if this project was to be continued would be looking into alternatives to the software tool iperf. While the tool did prove itself useful during the project there were some issues found with the program, as mentioned above. While alternatives were explored during this project not much time was spent searching as iperf was the clear favourite. However after running into issues several times with iperf, if the project was continued other alternatives or even the creation of a custom tools may be worth exploring.

5.5 User Case

As mentioned previously this project was inspired by the work being done on SDN and load balancing at the University of Queensland [1]. This work involved sharing network traffic over multiple links to try and achieve a greater maximum throughput. This is an example of a use case for the data collected during this project.

Figure 5.1 shows the scenario which this research was inspired by. A device, like a smart phone, would use a software tool, like an SDN switch, to share data across multiple networks, like LTE and WiFi. Below are results found when using the data found during test 2.

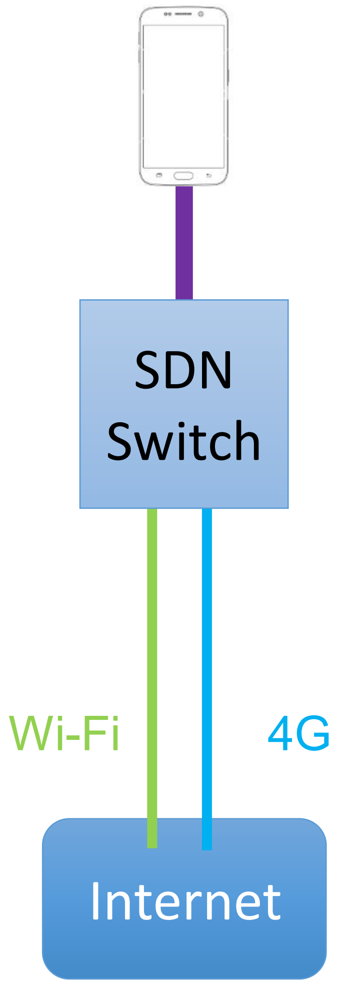

Figure 5.1 Load Balancing Scenario Depiction

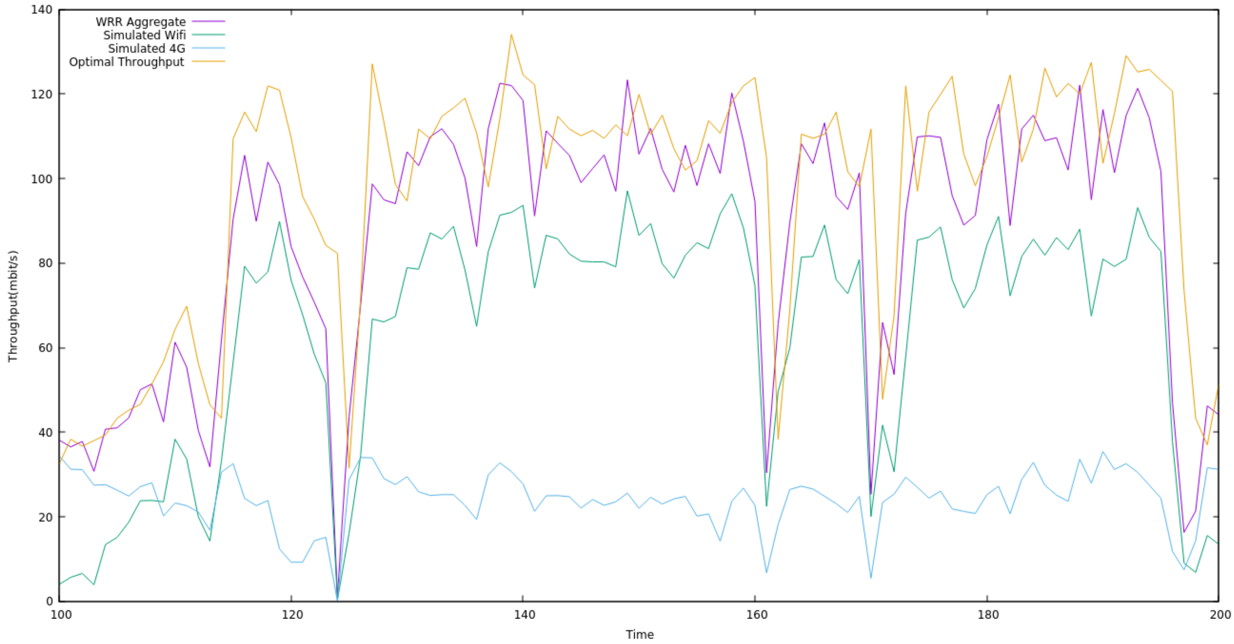

Figure 5.2 Throughput Values for Load Balancing Test.
LTE coloured Blue; WiFi coloured Green; WRR coloured Purple; Optimal coloured Yellow

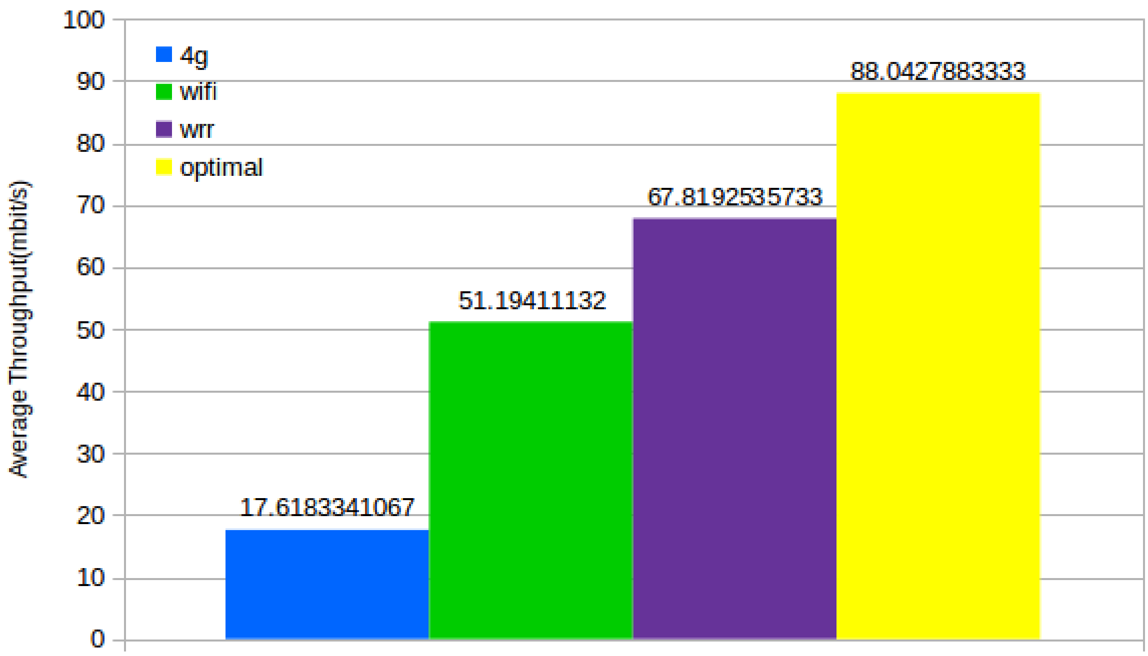

Figure 5.3 Average Throughput Values for Load Balancing Test

For the above results the data from test 2 were emulated across several virtual machines connected in a way similar to figure 5.1. The values shown are for the LTE and WiFi emulated on this virtual network. WRR (weighted round robin) was the load balancing method implemented for this network. Using WRR means that data was shared between the emulated LTE and WiFi networks in proportion to what those networks throughputs currently are. The optimal value is the addition of the originally measured WiFi and LTE values from test 2.

This test shows that load balancing across two networks with an SDN switch is possible. It also shows how well this SDN switch handle realistic LTE and WiFi throughput values.

Chapter 6

Conclusion

The project was able to devise a system which is able to collect network data for both WiFi and LTE networks as well as GPS location. Using this data collection system four complete data sets where recorded. Among these data sets there is a variety of performance for both WiFi and LTE. The locations of the tests were also varied to ensure there would be a good amount of change in throughput of both networks during tests.

The data gathered during the project has been shared with other University of Queensland scholars and is being used in their projects. The initial results from a load balancing test involving this data shown is an example of how the results of the project can be used to test systems which look to take advantage of current and future networking technology.

Not all aspects of the project were completed to the standard wanted. The overall quantity of data collected fell short of initial goals. However the quality of the data gathered was of an acceptable standard. While the project progressed several changes were made to the methodology and some ideas were left as possible future works. If work was to be continued the most important thing to do would be to expand the data set with more measurement as well as measurements of different connection types.

Appendices

Appendix A

Data Collection Python Script

```
#!/usr/bin/python

import time
from datetime import datetime
import threading
import os

class perpetualTimer():
  def __init__(self,t,hFunction):
    self.t=t
    self.hFunction = hFunction
    self.thread = threading.Timer(self.t,self.handle_function)
  def handle_function(self):
    self.hFunction()
    self.thread = threading.Timer(self.t,self.handle_function)
    self.thread.start()
  def start(self):
    self.thread.start()
  def cancel(self):
    self.thread.cancel()

def perf(i_ready, n_ready, p_ready):
   print "start perf"
   os.system("echo 'time' > /home/sam/Documents/per.txt")
   i_ready.set()
   n_ready.wait()
   p_ready.wait()
   with open('/home/sam/Documents/per.txt', 'a') as f:
      f.write('began at %s\n' %datetime.now().time())
   os.system("iperf3 -c 10.82.31.86 -i 1 -t 600 -p 5202 -l 4K >> /home/sam/Documents/per.txt")
   with open('/home/sam/Documents/per.txt', 'a') as f:
      f.write('finished at %s\n' %datetime.now().time())
```

```
def nets(i_ready, n_ready, p_ready):
   print "start nmcli"
   os.system('echo "TIME, SSID, BSSID, SIGNAL, ACTIVE" > /home/sam/Documents/nd.txt')
   n_ready.set()
   i_ready.wait()
   p_ready.wait()
   #timer - loop as many seconds needed
   global count
   starttime=time.time()
   while (count<600):
      count+=1
      nmlook()
      time.sleep(1.0-((time.time()-starttime)%1.0))

def nmlook():
   with open('/home/sam/Documents/nd.txt', 'a') as f:
      f.write('%s  ' %datetime.now().time())
   os.system("iw dev wlo1 link | awk '{printf \"%s \",$0} END {print \"\"}' >> /home/sam/Documents/nd.txt")
   with open('/home/sam/Documents/nd.txt', 'a') as f:
      f.write('\n')

def pingD(i_ready, n_ready, p_ready):
   os.system('echo "TIME # BYTES FROM IP SEQ# TTL DELAY MS" > /home/sam/Documents/pings.txt')
   p_ready.set()
   n_ready.wait()
   i_ready.wait()
   os.system("ping 10.82.31.86 -c 600 -D >> /home/sam/Documents/pings.txt")

def main():
   print(datetime.now().time())
   i_ready = threading.Event()
   p_ready = threading.Event()
   n_ready = threading.Event()
    print "starting"
   ithread = threading.Thread(target=perf,
       args = (i_ready, n_ready, p_ready))
   ithread.start()
   pthread = threading.Thread(target=pingD,
       args = (i_ready, n_ready, p_ready))
   pthread.start()
   nets(i_ready, n_ready, p_ready)

count = 0
main()
```

Appendix B

Emulation Throughput Control Python Script

```
#!/usr/bin/python

import time
from datetime import datetime
import threading
import os
import subprocess
import math
import sys

def control(i_ready, c_ready):
    wait_time = int(1)
    int_name = str('h1-eth0')
    with open("/home/sam/Documents/Thesis/network/test.txt", "r") as in_file:
        c_ready.set()
        i_ready.wait()
        for line in in_file:
            st = time.time()
            x = "1"
            try:
                bw = int(math.ceil(float(line)))
                if bw == 0:
                    bw = 1
                    x = str(bw)+"Kbit"
                else:
                    x = str(bw)+"Mbit"
            except ValueError:
                return
            print x

subprocess.call(['sudo','tc','qdisc','replace','dev',int_name,'root','tbf','rate',x,'latency','1000ms','burst','10K'])
            print "next"
            time.sleep(wait_time-(time.time()-st))

def perf(i_ready, c_ready):
    print "start perf"
    os.system("echo 'time' > /home/sam/Documents/Thesis/network/per.txt")
    i_ready.set()
    c_ready.wait()
    with open('/home/sam/Documents/Thesis/network/per.txt', 'a') as f:
        f.write('began at %s\n' %datetime.now().time())
    os.system("iperf3 -c 10.0.0.2 -i 1 -t 600>> /home/sam/Documents/Thesis/network/per.txt")
    with open('/home/sam/Documents/Thesis/network/per.txt', 'a') as f:
        f.write('finished at %s\n' %datetime.now().time())
```

```
def main():
   print(datetime.now().time())
   c_ready = threading.Event()
   i_ready = threading.Event()
   print "starting"
   ithread = threading.Thread(target=perf,
       args = (i_ready, c_ready))
   ithread.start()
   cthread = threading.Thread(target=control,
       args = (i_ready, c_ready))
   cthread.start()

main()
```

Appendix C

Accompanying ZIP File Format

SamuelTeed_43211915 (root folder)

- DataSet1 (subfolder of data set 1 files)
    - LTE (subfolder of LTE file for data set 1)
        - nd.txt (text file with network details)
        - per.tx (text file with iperf output)
        - pings.txt (text file with ping output)
    - WiFi (subfolder of LTE file for data set 1, with same internal structure as LTE)
    - Details.txt (text file with test details)
    - RK_gpx _2017-03-20_1323.gpx (gpx file containing gps details of test 1)
- DataSet2 (subfolder of data set 2 files, with same internal structure as DataSet1)
- DataSet3 (subfolder of data set 3 files, with same internal structure as DataSet1)
- DataSet4 (subfolder of data set 4 files, with same internal structure as DataSet1)
- emulate.py (Emulation Throughput Control Python Script)
- gather.py (Data Collection Python Script)
- Teed,Samuel_43211915_ENGG4802.pdf (this document)

Bibliography

[1] Al-Najjar, Anees, Siamak Layeghy, and Marius Portmann. "Pushing SDN to the end-host, network load balancing using OpenFlow." *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2016. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7457129&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7457129 [Accessed 08/2016]

[2] Costin Raiciu and Christoph Paasch. "Multipath TCP". (2012). [online] Available: https://www.youtube.com/watch?v=02nBaaloFWU [Accessed 08/2016]

[3] Goeff Huston. "Multipath TCP". (2015). [online] Available: http://www.potaroo.net/ispcol/2015-06/mptcp.html [Accessed 08/2016]

[4] Open Networking Fundation. "Software-defined networking: The new norm for networks." *ONF White Paper* (2012) Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/whitepapers/wp-sdn-newnorm.pdf [Accessed 08/2016]

[5] Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." *IEEE Communications Magazine* 51.7 (2013): 36-43. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6553676&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D6553676 [Accessed 08/2016]

[6] Braun, Wolfgang, and Michael Menth. "Software-Defined Networking using OpenFlow: Protocols, applications and architectural design choices." *Future Internet* 6.2 (2014): 302-336. Available: www.mdpi.com/1999-5903/6/2/302/pdf [Accessed 08/2016]

[7] Deng, Shuo, et al. "Wifi, lte, or both?: Measuring multi-homed wireless internet performance." *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014. Available: http://web.mit.edu/ravinet/www/imc_submission.pdf [Accessed 08/2016]

[8] Ravi Netravali. "Mahimahi: A Lightweight Toolkit for Reproducible Web Measurement." Available: http://web.mit.edu/ravinet/www/mahimahidemo.pdf [Accessed 05/2017]

[9] "About Wi-Fi Assist." (2017). Available: https://support.apple.com/en-au/HT205296 [Accessed 05/2017]

[10] "Runkeeper-Track your runs, walks and more with your iPhone or Android phone." (2017). Available: https://runkeeper.com [Accessed 03/2017]

[11] "iw(8) - Linux man page." Available: https://linux.die.net/man/8/iw [Accessed 03/2017]

[12] "What is iPerf / iPerf3?" Available: https://iperf.fr/ [Accessed 03/2017]

[13] "Cisco AnyConnect Secure Mobility Client" Available: http://www.cisco.com/c/en/us/products/security/anyconnect-secure-mobility-client/index.html [Accessed 03/2017]

[14] "Mininet An Instant Virtual Network on your Laptop (or other PC)." Available: http://mininet.org/ [Accessed 05/2017]

[15] Bert Hubert. "tc(8) - Linux man page." Available: https://linux.die.net/man/8/tc [Accessed 05/2017]

[16] "ping(8) - Linux man page." Available: https://linux.die.net/man/8/ping [Accessed 03/2017]