

Real Time Emulation of an LTE Network Using NS-3

Tom Molloy¹, Zhenhui Yuan², *Member, IEEE*, and Gabriel-Miro Muntean³ *Member, IEEE*

¹Information & Communications Engineering
Dublin City University, Dublin 8, Ireland
Email: tom.molloy4@mail.dcu.ie

²Department of Electronic Engineering
Dublin City University, Dublin 8, Ireland
Email: zhenhui.yuan@dcu.ie

³Department of Electronic Engineering
Dublin City University, Dublin 8, Ireland
Email: gabriel.muntean@dcu.ie

Abstract— In recent years, LTE (Long Term Evolution) technology has emerged as a new standard in mobile communications. Due to barriers such as high initial costs and the need to maintain existing networks which are still widely used and profitable for Telcos, LTE network rollouts have been slow in some areas of the world. Although rollout has been slow in some areas, LTE coverage is continuing to grow and is set to become the first truly global mobile network standard. The purpose of this paper is to provide a low cost and effective test methodology, namely a real time test bed which can interact with real IP based network traffic between two computers. This paper shows how real time traffic can be sent over an emulated LTE network in NS3. While showcasing the advantages of emulation, we also present the limitations of emulations with discussion for further works and improvements.

Keywords – LTE, emulation, NS3.

I INTRODUCTION

Nowadays, the amount of data generated and transmitted is growing exponentially year on year. Cisco predicts that by 2018 there will be more mobile users, with more mobile connections, using faster mobile speeds and there will be using more data intensive applications. According to the Cisco VNI Mobile Data Traffic Forecast for 2013-2018, it is predicted that the global mobile data traffic usage per month will surpass 15 Exabytes indicating a growth of 1000% in just 5 years [1].

As the amount of data being created increases, the need for higher network speed and capacity increases. With this in mind the 3GPP (3rd Generation Partnership Project) set out to enhance the existing mobile networks. In the 3GPP Rel.8 document the 3GPP set out the guidelines of LTE. The goal of LTE was to establish a network which can provide high data rates for the expected exponential increase in network traffic, and to set foundations for future standards to build on.

However, the increased data rates and a more complex network infrastructure come at a cost. There is a balancing act for Telcos to provide satisfactory service for their customers, while trying to invest in networks to provide sufficient service in the future. LTE networks are not backwards compatible with existing 2G and 3G networks hence the options for Telcos are to either deploy a single RAN (radio access network) or to overlay new LTE equipment over existing infrastructure. The right time for a Telco to invest in an LTE

rollout must be carefully considered. It has been shown that the drivers for LTE expansion have been user behaviour (i.e. always connected via social networks, OTT messaging etc.), device adoption (i.e. latest smartphones) and content and apps (i.e. online games and video consumption [2]). Recently in Ireland, Vodafone has invested €900 million in networks and plans to spend a further €500 million in coming years [3].

With these financial barriers holding back cheap and ubiquitous LTE coverage, the following question is raised: in areas without LTE coverage how can R&D teams prototype and test applications and protocols to run over LTE networks? Network simulators such as NS3 [4] provide a way to set up large scale networks and test QoS (Quality of Service) parameters. Although NS3 is a powerful tool, real time traffic emulation is not supported over the LTE module. This paper introduces a novel test methodology to extend the NS3 simulator to allow for real time emulation over the LTE module and compare its performance to a purely simulated scenario to validate its use.

The paper is organized as follows. Section II discusses previous work and other solutions for prototyping LTE based designs. Section III explains the modifications made to NS3 and the experimental set up. Section IV validates the experimental set up through comparison of simulation data to emulation data. Section V draws conclusions and introduces discussion.

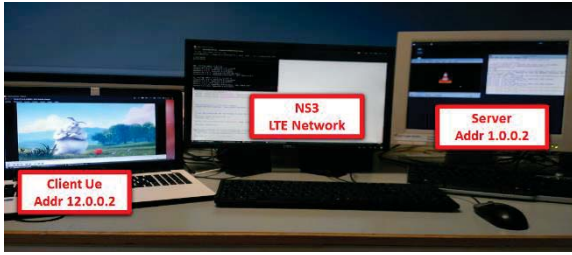


Figure 1 - Test set up

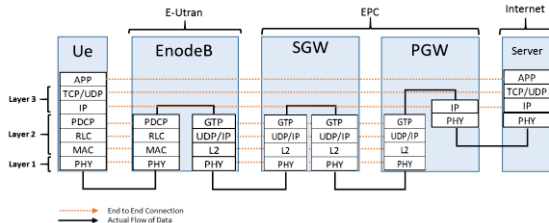


Figure 2 - Server Ue model

II RELATED WORKS

a) USRP Based Solutions

USRP (Universal Software Radio Peripheral) is a device which can act as a radio frequency front end and can be tuned using GNU Radio [5]. There are many applications developed for USRP, one of which is to use it as an RF front end for an LTE test bed. To the best of our knowledge, the unique solution is developed by Fabrice Bellard's [6]. Bellard's solution is not open sourced and is commercialised by Amarisoft [7]. Additionally, Bellard's solution is limited by the RF range of the USRP and the network configurations compared to network simulators.

b) Network Simulator-3 (NS3)

NS3 is a discrete event open source network simulator. It is primarily used in research and teaching. It contains high quality, well maintained and validated network models.

Unlike other LTE simulators such as LTE-sim, [8], NS3 supports more network modules and it is well established through the three development cycles from NS, to NS2, to NS3.

The validity of NS3 simulations are explored by Guidolin, Badia, and Zorzi in their paper on 2x2 LTE MIMO configurations [9]. In our paper we will use the LTE module but with a simpler single antenna set up.

Emulation features of NS-3 are investigated by Alvarez, Orea et. al. in their paper [10], where the limitations of emulation in NS-3 are discussed. They claim that using a distributed system, which is the approach of this paper, is more accurate than using a single machine. They give recommendations to use a more advanced hardware platform and use different network topologies to see if better performance can be achieved.

c) Lena Module

LTE in NS3 is provided in the LENA module. The LENA module was designed with the focus on providing academic and small cell vendors the opportunity to build and test novel scheduling algorithms and solutions through simulation of an LTE network [11]. There is no built-in real time emulation features available in the LENA module. This paper proposes to make minimum modifications to the model and test the validity of the results.

The work done in [12] focuses on LTE use in civil security networks. That paper explores using the LENA module in a satellite/terrestrial hybrid network. In [12] they successfully use the LENA module in emulation, but the approach is to use a simplified LENA model to conduct their experiments.

In [13] the LENA module is used in an emulation environment. The module is modified to exploit parallel programming paradigms and for use in a clustered computer environment.

The approach of this paper is to use the full version of the NS-3 LENA module but on a simple 3 computer distributed network, as opposed to a cluster, to assess its functionality..

III EXPERIMENTAL PROCEDURE

a) Experimental setup

The experimental test-bed consists of 3 PCs. The first PC is a host machine, it runs NS3-3.19 and the entire simulation is run on this machine. It is an Intel i5-3470 Quad Core, at 3.20GHz; with 8GB of DDR3 RAM; running Ubuntu 12.04 LTS.

Two PCs are connected to the host PC via the Ethernet ports and act as the Ue (user equipment) and the remote host, separately. Traffic will be transferred through the simulated network on the host machine between the two external PCs. Figure 1 shows the test set up.

The first external PC which serves as the Ue is an Intel Core i3-2365M, at 1.4GHz; 4GB of DDR3 RAM; running Ubuntu 12.04 LTS. The second external PC which will serve as the server is an Intel core i5-3230M, at 2.60GHz; 4GB of DDR3 RAM; running Ubuntu 12.04LTS.

b) Network Topology

The network topology used to test the setup is simple in nature. The scenario we are trying to emulate is shown in Figure 2. Real IP based traffic is sent between the server and the Ue across the EPC (Evolved Packet Core).

Only two Ues are used in the experiment so that the host PC will be using less resources when running the simulation in real time. Overloading of the network is provided by sending high volumes of data to each Ue. As shown in Figure 3, there are two Ues, one of which is the external PC connected into the simulation. The other Ue is simulated. Server 1 is

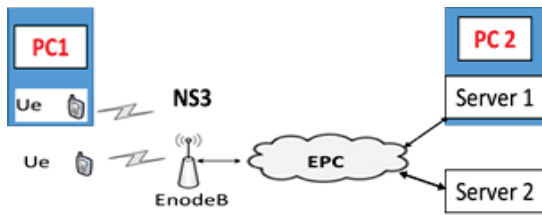


Figure 3 - Network Topology

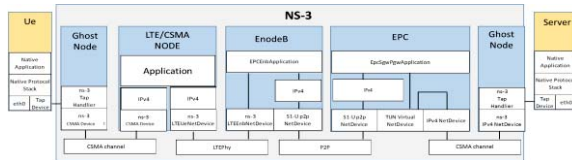


Figure 4 - Network Topology

Table 1 - Downlink test cases

Test 1
A single Ue with no background traffic. Send data in the downlink from the server to the Ue and increase until network capacity is met. Measure QoS parameters of jitter, throughput, and packet loss.
Test 2
Use 2 Ues. Send equal amounts of data to each Ue until network capacity is met. Measure QoS parameters of jitter, throughput, and packet loss.
Test 3
Use 2 Ues. Send a constant stream of traffic to one Ue. Increase background traffic to the second Ue until network capacity is met. Measure QoS parameters of jitter, throughput, and packet loss.

connected the second external PC, it sends data to the Ue connected to the first external PC only. Server 2 is fully simulated and sends data only to the simulated node.

c) Tap bridge connections

As the LTE module is not compatible with real time simulation, an extra link had to be added into the simulation to inject real traffic into the network. Only two net devices in NS3 are compatible with real time traffic, namely CSMA net device and WiFi net device. An extra CSMA link was added to connect the external PC with the Ue node. The CSMA link was overprovisioned with 500Gb/s data rate and no delay to minimise any reduction in performance.

The external PCs are connected into the simulation via a tap-bridge. Figure 4 shows a diagrammatic view of the set up in relation to the NS3 modules. The Ue and sever are deployed in PC1 and PC2, separately.

d) NS3 modifications

Certain modifications to the LTE module code were needed for this experimental set up to operate. The PC used as the Ue has no IMSI (International mobile subscriber identity) number. When a session is set up in the network, IP addresses, teid (tunnel end identifiers) and bearers are mapped to this IMSI number.

The downlink logic was introduced into the EpcSgwPgwApplication module which spoofed the IMSI number. Instead of dropping packets address to the Ue, the bearer is now changed to the bearer corresponding to the Ue node attached to the PC.

The uplink logic was introduced into the NAS (Non Access Stratum). Again this logic changed the bearer on packets sent from the PC.

In the LTE module pcap tracing is not enabled. To create QoS (Quality of Service) data from the simulated nodes the packetSink module was modified to read packet information and dump info such as delay, jitter, data rate, and dropped packets into a file for each Ue at the application layer.

e) Network Setup

The lowest level of bandwidth granularity for LTE in NS3 is a RB (resource block). One RB consists of 12 subcarriers spread over 180 kHz and lasting 0.5ms. This paper uses 6 RBs in both uplink and downlink. This corresponds to the lowest bandwidth spectrum of 1.4MHz.

To calculate our throughput according to 3GPP specifications [14] at the radio level we can compute the number of REs (resource elements) in a subframe equals 1008 according to equation (1). A subframe is of length 1ms.

$$12 \text{ subcarriers} \times 7 \text{ symbols} \times 6 \text{ RBs} \times 2 \text{ slots} \quad (1)$$

Using 64QAM we have 6 bits per RE so the theoretical throughput over 1 second can be computed using equation (2), which gives the result of 6.48Mbps

$$6 \text{ bits} \times 1008 \text{ REs} / 1 \text{ms} \quad (2)$$

This is the maximum throughput provided by eNodeB at the radio level, however, this will not be the achieved throughput in practice. A Motorola white paper on actual LTE performance shows that for the this set up we would expect an actual throughput of 4Mbps – 4.5Mbps [15].

f) Procedure

The first step in testing the emulation set up is to test the validity of using the CSMA link from the Ue to client. Throughput, jitter and packet loss are measured for a unmodified simulated Ue node. The same measurements are then taken for a modified CSMA/Ue node.

To measure emulation performance, testcases in Table 1 are performed in simulation and emulation.

IV Results Analysis

a) Preliminary test: CSMA link

Tests were carried out in a simulated environment to test the validity of using a CSMA

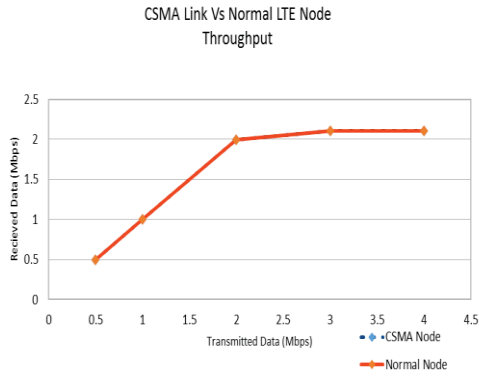


Figure 1 (a) -Throughput of modified node Vs non-modified node

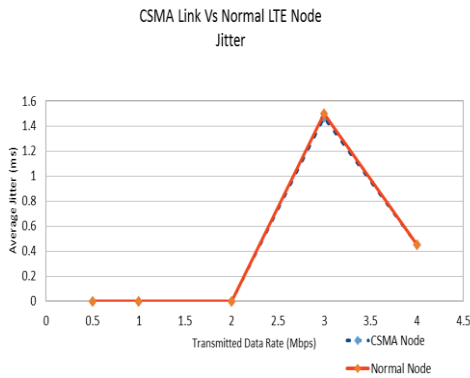


Figure 5 (b) -Jitter of modified node Vs non-modified node

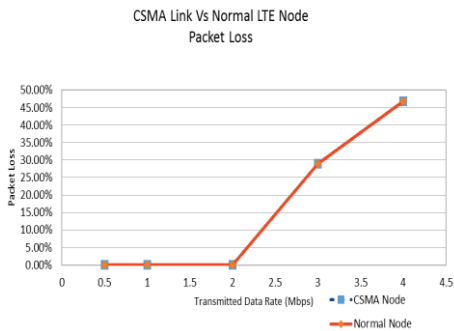


Figure 5 (c) -Packet loss of modified node Vs non-modified node

link connected to a Ue. The results show that the node with the CSMA link performed as good as the normal Ue node.

Figure 5(a) shows that the throughput of both nodes performing equally well with a only 0.10%-0.20% difference. Similarly Figure 5(b) & Figure 5(c) show the jitter performance and packet loss performance are unaffected by the addition of the CSMA link with 0% difference in each, except for jitter which recorded a 2% difference at 3Mbps. It can be concluded that an over provisioned CSMA link is suitable for injecting real time traffic into an emulated LTE environment.

b) Test 1: single Ue

Here we see how the performance of a single emulated Ue matches the performance of a simulated Ue – with the exception of a slightly worse jitter performance.

Figure 6 (a) shows the performance in terms of throughput. There is very little difference between the simulation results and the emulation results. The emulation results deviate only 0.10%-0.20% from the simulated results. These figures can be seen in Table 1.

Figure 6 (b) shows the jitter performance. Here we see unpredictable jitter performance but in general the emulated jitter is higher. Table 1 shows that at some data rates there is up to a 1.7ms difference in jitter.

Finally Figure 6(c) shows that the emulated node performs the same as the simulated node in terms of packet loss with a difference of 1%

c) Test 2: Increasing data rate with background traffic

Next we study the impact of background traffic on the performance. This corresponds to the findings in [10] where it can be seen that an increased load on the CPU correlates with a performance drop in NS-3.

Figure 7(a) shows that the emulator performs reasonably well up to 1Mbps data rate with only 0.2% deviation from simulated performance. However, when we compare the simulated nodes throughput performance to the emulated nodes throughput performance when traffic rises above 1Mbps we see a marked difference. There is a 30.33% difference in throughput levels by the time network capacity is met.

Again the jitter in this test is higher for emulation; it is worse than in test 1. Figure 7(b) shows that with more CPU resources being used we see a maximum difference of 1.85ms.

Figure 7(c) shows packet loss performs slightly worse for emulation, with up to 20.69% more packets dropped, in this test, but the magnitude of difference between emulation and simulation is not as pronounced as throughput or jitter. This suggests that packets are being processed but at a much slower rate..

d) Test 3: Constant data rate with background traffic

This test again investigates the effect of background traffic in emulation. In this scenario the data rate of the test node is constant – 500kbps, however background traffic in the network is increased from 500kbps until capacity is met.

Again we see that, for data rates below 1Mbps, the emulated node performs well with little deviation comparing to results from the simulation. However after 1Mbps threshold, the throughput begins to drop. Figure 8(a) shows throughput begins

Test 1

Single Node with Increasing Downlink Data Rate Throughput

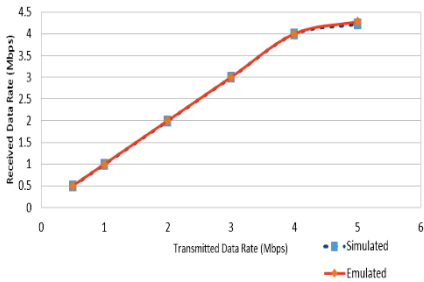


Figure 6(a) -Throughput single node

Single Node with Increasing Downlink Data Rate Jitter

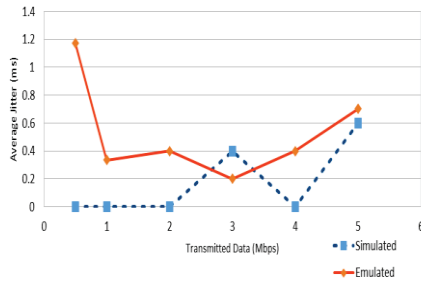


Figure 6(b) -Jitter single node

Single Node with Increasing Downlink Data Rate Packet Loss

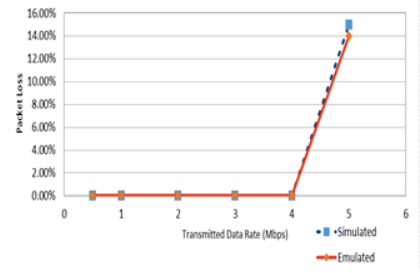


Figure 6(c) –Packet Loss single node

Test 2

Background Traffic Equal to Transmitted Data Rate Throughput

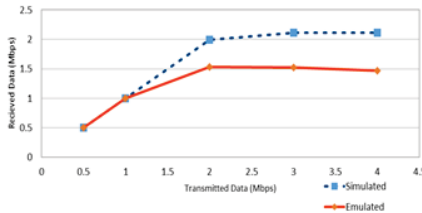


Figure 7(a) -Throughput single node with background traffic

Background Traffic Equal to Transmitted Data Rate Jitter

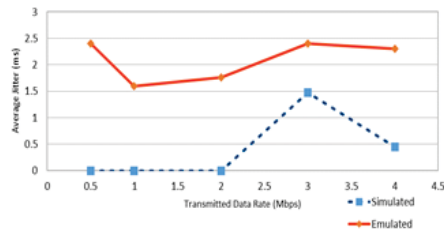


Figure 7(b) -Jitter single node with background traffic

Background Traffic Equal to Transmitted Data Rate Packet Loss

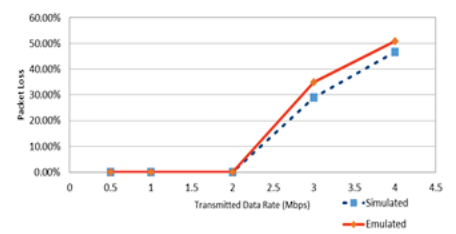


Figure 7(c) –Packet Loss single node with background traffic

Tests 3

Constant Transmit Data Rate (500kbps) with Increasing background Traffic

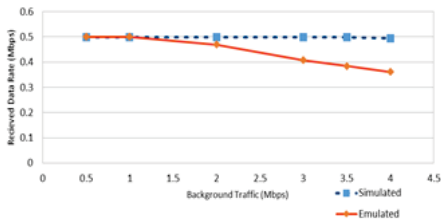


Figure 8(a) -Throughput single node with increasing background traffic

Constant Transmit Data Rate (500kbps) with Increasing background Traffic - Jitter

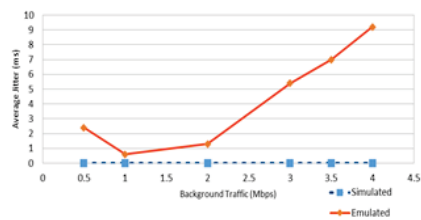


Figure 8(b) -Jitter single node with increasing background traffic

Constant Transmit Data Rate (500kbps) with Increasing background Traffic - Packet Loss

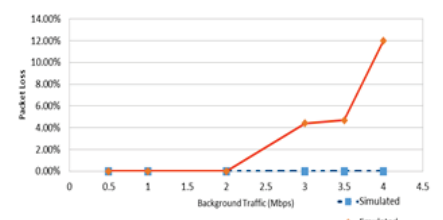


Figure 8(c) –Packet Loss single node with increasing background traffic

Table 1 – Test case statistics

Test 1 Throughput (Mbps)					Test 1 Jitter (ms)					Test 1 Packet Loss				
Tx (Mbps)	Rx Sim	Rx Em	Diff	Error	Tx (Mbps)	Sim	Em	Diff	Error	Tx (Mbps)	Sim	Em	Diff	Error
0.5	0.4995	0.5	0.0005	0.10%	0.5	0	1.174	1.174	NA	0.5	0	0	0.00%	NA
1	0.998	1	0.002	0.20%	1	0	0.336	0.336	NA	1	0	0	0.00%	NA
2	1.997	2	0.003	0.15%	2	0	0.4	0.4	NA	2	0	0	0.00%	NA
3	2.9969	3	0.0031	0.10%	3	0.4	0.2	0.2	50.00%	3	0	0	0.00%	NA
4	3.995	4	0.005	0.13%	4	0	0.4	0.4	NA	4	0	0	0.00%	NA
5	4.23	4.28	0.05	1.18%	5	0.6	1.4	0.8	133.33%	5	15.00%	14.00%	1.00%	6.67%
Test 2 Throughput (Mbps)					Test 2 Jitter (ms)					Test 2 Packet Loss				
Tx (Mbps)	Rx Sim	Rx Em	Diff	Error	Tx (Mbps)	Sim	Em	Diff	Error	Tx (Mbps)	Sim	Em	Diff	Error
0.5	0.499	0.5	0.001	0.20%	0.5	0	2.4	2.4	NA	0.5	0	0	0.00%	NA
1	0.998	1	0.002	0.20%	1	0	1.6	1.6	NA	1	0	0	0.00%	NA
2	1.996	1.53	0.466	23.35%	2	0	1.759	1.759	NA	2	0	0	0.00%	NA
3	2.11	1.52	0.59	27.96%	3	1.47	2.4	0.93	63.27%	3	29.00%	35.00%	6.00%	20.69%
4	2.11	1.47	0.64	30.33%	4	0.45	2.3	1.85	411.11%	4	46.80%	51.00%	4.20%	8.97%
Test 3 Throughput (Mbps)					Test 3 Jitter (ms)					Test 3 Packet Loss				
Tx (Mbps)	Rx Sim	Rx Em	Diff	Error	Tx (Mbps)	Sim	Em	Diff	Error	Tx (Mbps)	Sim	Em	Diff	Error
0.5	0.499	0.5	0.001	0.20%	0.5	0	2.4	2.4	NA	0.5	0	0	0.00%	NA
0.5	0.4995	0.5	0.0005	0.10%	0.5	0	0.6	0.6	NA	0.5	0	0	0.00%	NA
0.5	0.499	0.469	0.03	6.01%	0.5	0	1.3	1.3	NA	0.5	0	0	0.00%	NA
0.5	0.499	0.407	0.092	18.44%	0.5	0	5.4	5.4	NA	0.5	0.00%	4.40%	4.40%	NA
0.5	0.499	0.384	0.115	23.05%	0.5	0	7	7	NA	0.5	0.00%	4.70%	4.70%	NA
0.5	0.495	0.361	0.134	27.07%	0.5	0	9.2	9.2	NA	0.5	0.00%	12.00%	12.00%	NA

to drop after 1Mbps up to a maximum of 27.07%.

Figure 8(b) shows that jitter is worse in emulation. In fact in simulation, due to the low data rate, there was no jitter recorded. In emulation the jitter increases as high as 9.2ms when transmitted traffic levels reach network capacity.

Finally packet loss was higher in the emulation scenario. In simulation the Ue was able to keep up with traffic rates and recorded no packet loss. Once background traffic reached 3Mbps we started to see some degradation in packet loss. When transmitted traffic levels reach network capacity we see a 12% packet loss rate.

V CONCLUSIONS AND FUTURE WORK

The usage of network emulator over a simulated model has many benefits. It allows R&D teams to quickly test and deploy new applications at low cost. This paper has shown that it is possible to send real IP based traffic over the NS-3 LENA module. Taking a scenario where only one Ue node is deployed in a cell, this setup performs on the same level as simulation based nodes in terms of throughput and packet loss. We see a slightly worse jitter performance, but this setup can still be useful for many testing applications.

The limitations of NS-3 based emulations have been outlined in [10], and indeed many of these limitations were met in the evaluation of the LTE model. One of the goals of this project was for a quick and easy emulation tool. As the deployment of this tool is quite simple, the emulation setup in this paper was deployed on different hardware platforms. Immediate performance improvements were seen when deployed on a better hardware platform, so it can be said that the limitations are at least in some part due to the hardware platform used.

The work in [13] recognised the limitations of NS-3 for larger network topologies. As such, it is proposed to use a cluster of CPUs. This is certainly an area for future work for this project, but starts to deviate away from the goal of a cheap and effective tool set out at the start.

Future work on this project will focus on allowing for larger network topologies. For this we must explore two areas. Firstly investigation into the NS-3 source code will be made to see if any algorithmic optimisations can be made. Secondly moving the environment to a better hardware platform, similar to the one outlined in [13], using multiple processors and the openmpi module for NS-3 may reap better results.

ACKNOWLEDGEMENTS

This work was carried out in the Performance Engineering Laboratory, Dublin City University, Ireland.

REFERENCES

[1] Cisco, "Cisco VNI Global Mobile Data Traffic Forecast, 2013–2018," Cisco, 2014.
[2] B. Ayvazian, "LTE Deployment Strategies:

Network Overlay vs Sigle RAN," UBM Tech, 2013.

- [3] Vodafone, "Vodafone Ireland Network Expansion Plans," 2014. [Online]. Available: <http://www.vodafone.ie/network/expansion-plans/>. [Accessed 19 03 2014].
- [4] NS3, "NS3 homepage," 2014. [Online]. Available: <https://www.nsnam.org/>. [Accessed 19 03 2014].
- [5] GNURadio, "GNU Radio wiki," [Online]. Available: <http://gnuradio.org/redmine/projects/gnuradio/wiki>. [Accessed 09 04 2014].
- [6] F. Bellard, "Fabrice Bellards website," [Online]. Available: <http://bellard.org/lte/>. [Accessed 23 03 2014].
- [7] Amarisoft, "Amarisoft Website," [Online]. Available: <http://www.amarisoft.com/?p=amarilte>. [Accessed 21 02 2014].
- [8] Telematics, "LTE-sim webpage," 2014. [Online]. Available: <http://telematics.poliba.it/index.php/en/lte-sim>. [Accessed 19 03 2014].
- [9] F. Guidolin, L. Baidia and M. Zorzi, "Implementation of 2×2 MIMO in an LTE Module," in *2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*, 2012.
- [10] A. Alvarez, R. Orea, S. Cabrero, X. G. Paneda, R. Garcia and D. Melendi, "Limitations of network emulation with single-machine and distributed ns-3," in *3rd International ICST Conference on Simulation Tools and Techniques*, 2010.
- [11] CTTC, "LENA project homepage," 2014. [Online]. Available: <http://networks.cttc.es/mobile-networks/software-tools/lena/>. [Accessed 19 03 2014].
- [12] M. Crosnier, R. Dhaou, F. Planchou and A.-L. Beylot, "TCP performance optimization for handover management for LTE satellite/terrestrial hybrid networks," in *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference*, 2012.
- [13] R. D. Ford, 2012. [Online]. Available: http://eeweb.poly.edu/~srgan/testbeddocs/for_d_thesis.pdf. [Accessed 25 03 2014].
- [14] 3GPP, "3GPP TS 36.201 - Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description," 3GPP, 2012.
- [15] Motorola, "Realistic LTE Performance," Motorola, 2010.

