University of Southern Queensland

Faculty of Engineering and Surveying

Analysis and Taxonomy of Network Quality of Service (QoS) Concepts in the Long Term Evolution/System Architecture Evolution (LTE/SAE) System

A dissertation Submitted by:

Saud Khlaif K Alenazi

In a fulfilment of requirements for:

Courses ENG4111/ENG4112

Toward the Degree of :

Bachelor of Engineering

(Electrical & Electronics Major)

Date of Submission:

28th October 2010

Abstract

Modern mobile communication networks provide a variety of voice and data services. These services have different Quality of Service requirements and priorities. The latest set of mobile technology specifications by the 3rd Generation Partnership Project is referred to as Long Term Evolution/System Architecture Evolution. These 4th generation systems' features major changes in access and core network as well as service delivery. The main aim of this project is to investigate performance of the Quality of Service concept of these systems. This includes an analysis of the categorisation of the available Quality of Service mechanisms. In particular, the project investigates and evaluates the performance features of the network. These performance features include packets' delay and downstream throughput. Better performance of loaded network in the presence of Quality of Service mechanism is one of the main goals of this project. More specifically, speed-up real-time packets as they are the highest delay sensitive packets. Network simulator, NS2, is used to emulate a LTE/SAE network and to simulate traffic. Simulation results show that the throughput and delay of real-time packets are improved in the existence of Quality of Service mechanism.

University of Southern Queensland
Faculty of Engineering and Surveying

ENG4111 & ENG4112 Research Project

Limitations of Use

The council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or Completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

The dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Prof. Frank Bullen

Dean

Faculty of Engineering and Surveying

Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged. I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Saud Alenazi

Student Identification Number:

0050077753

Signature: 

Date: 28/ 10 / 2010

Acknowledgments

The author would like to acknowledge the efforts and assistance have been provided by the supervisor, Dr Alexander Kist. Dr Kist deserves thanks for continuous help and support throughout this project. Technical and Academic information have been provided by him, have guided the author to the right and proper way during the project.

The author would like to thank his wife as she deserves thanks for her patience and support. His little son must be mentioned here, as the author has not spent an adequate time caring after him during this academic year.

Table of Contents

Table of Figures

Table of Codes

List of Tables

Chapter 1 : Introduction

In this project, a simple Long Term Evolution/System Architecture Evolution (LTE/SAE) model will be implemented and simulated in accordance with practical Quality of Service (QoS) parameters set by the operator and simplified for this project. Specifically, the total throughput, packet delay of this system will be tested when QoS parameters are enabled as well as when they are disabled. Long Term Evolution (LTE) is introduced to deal with the increase in the number of users and the need for high speed communications. It represents the fourth generation and most recent digital technology created since digital communication was invented. The Quality of Service (QoS) mechanism is essential to provide reasonable identification of the Long Term Evolution/System Architecture Evolution (LTE/SAE) System performance. Network Simulator 2 (NS2) tool is the nominated tool for building and testing the system. The expected results should show an improvement in the three tested criteria.

1.1 Background

1.1.1 Mobile Systems Evolution

In the early of 1980s, mobile communication has been first introduced. Figure 1 below shows the major communication standards evolution. The first standard was an analogue system and so-called the Advanced Mobile Phone System (AMPS). In the early 1990s Global System for Mobile Communication (GSM) has been released as the first standard system for the digital mobile communication network and it represents the 2nd Generation of communication standards. General Packet Radio Service (GPRS) has been released in the mid 1990s, representing the 2.5 Generation. GPRS has enhanced the data bit rate to reach up to 114 Kbps. Enhanced Data rate for GSM evolution (EDGE) was the key for starting the 3rd Generation towards the end of the 1990s. The Enhanced 3rd Generation has been presented in 2002 and was represented by the Universal Mobile Telecommunication System (UMTS) standard. The first step to transfer the technology towards the 4th generation has been achieved once High Speed Packet Access (HSPA) was identified in the middle period of the first decade of the 21st century. HSPA has improved the data bit rate to reach up to 14Mbps and it is known as 3.75 Generation. However, Long Term Evolution (LTE) is the major key that leads the communication technology to start the 4th Generation level.

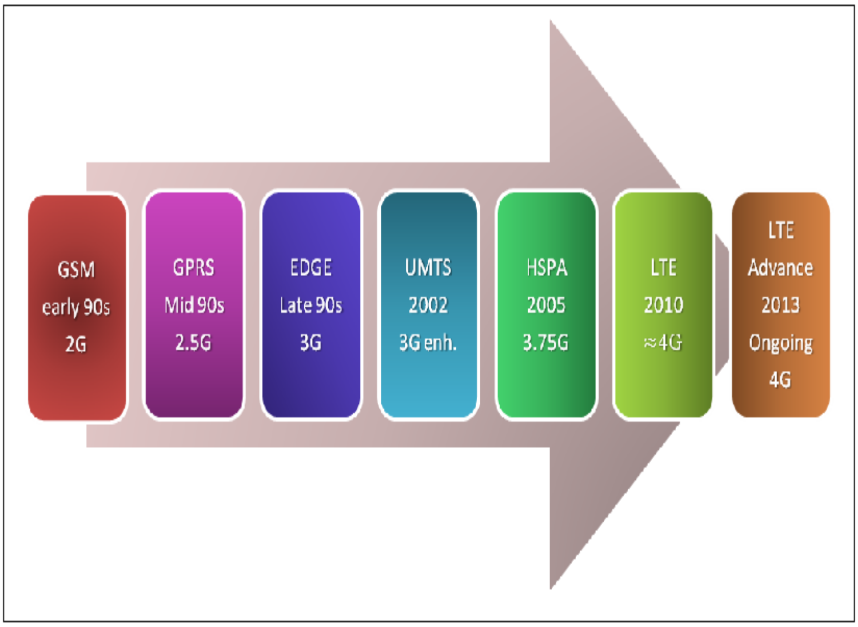

Figure 1: Digital Communication Standards System (this figure made using MS office power point)

1.1.2 Long Term Evolution (LTE)

Long Term Evolution (LTE) is the current technology that is standardized to transfer the communication from the previous standard HSPA in the 3rd Generation to 4th Generation. The reasons that motivate this transfer are the rapid increase in the number of users of technology, the need of high speed data and uplink/downlink speed. LTE philosophy is based on the 3rd Generation Partnership Project (3GPP) specifications release 8. LTE decreases the latency with a factor of 1/6 than HSPA does and it enhances the call setup delay to be 50-100ms. Therefore, LTE has an ultra low latency.

1.1.3 Long Term Evolution/System Architecture Evolution (LTE/SAE)

When these two terminologies, (LTE/SAE), come together, this results in LTE network architecture improvement. As a result, architecture complexity is eliminated due to reducing the number of nodes in the core network. Furthermore, the network becomes Flatter Network which means that the communication between stations in the network is performed without mediators such as routers. Therefore, the time taken for packets to travel is minimised which means latency is improved. Some of the nodes are redistributed in the network and/or merged to some other nodes since LTE/SAE elements have the ability to take place and substitute the user and/or control nodes. An example of this is the Radio Network Controller (RNC) that is split between the Access Gate Way (AGW) and Base Transceiver Station (BTS) or what is called (eNodeB). Core Network elements such as SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node) or PDSN (Packet Data Serving Node) are combine with the AGW. Figure 2 below shows simplified LTE/SAE network and it shows that the network includes two major parts, E-UTRAN (Evolved Universal Terrestrial Radio Access Network) and EPC (Evolved Packet Core). UE is connected to the eNodeB via Air interface. The interface that connects the eNodeBs or BTSs together is the Air interface or x2 interface. Network nodes are connected to each other via S1 interface.

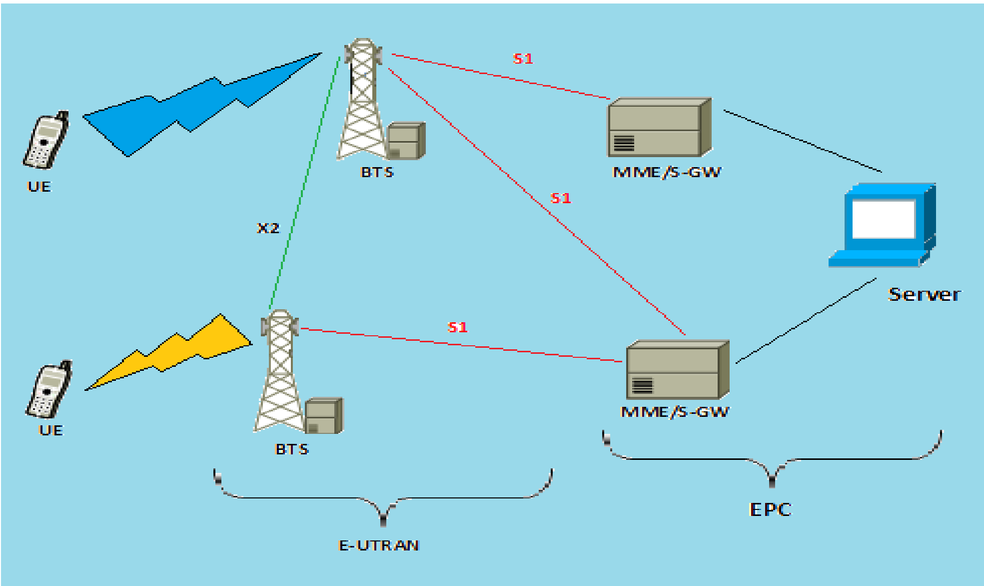

Figure 2: Simplified LTE/SAE Network (this figure made using DIA software office power point)

1.1.4 LTE Network QoS

When the service is shifted from single user to multi user service, the number of users is increased and higher traffic communication is needed. Therefore, it is important to define Quality of Service (QoS). QoS defines Policies rather than improving service's features. QoS standardizations are in 3GPP release 8. A general example of QoS is when a supermarket has a policy which states that the customer should not wait for more than 1 minute at the checkout. The QoS in the LTE/SAE network is for both the access and service network.

1.2 Project Aim

The main aim of this project paper is to develop methods and techniques as well as providing measurements and test results analysis of the Long Term Evolution/System Architecture Evolution network environment in accordance to the Quality of Service concept. One more aim of this project is to provide an analysis of the categorisation of the available Quality of Service mechanism. In particular, the paper investigates the concept of bearer, evaluates the performance of the network as well as a number of applications and last but not least, aimed at being simulated in either real-time applications or mathematical model.

1.3 Research Objectives

Number of objectives has been set in order to achieve the main goals of the projects. The following dot points summarize the objectives:

- Researching the System Architecture Evolution/Long Term Evolution system environment.
- Study and analyse of available Quality of Service mechanisms.
- Develop a mathematical models and/or simulation to evaluate the performance of Long Term Evolution/System Architecture Evolution network.
- Evaluate and analyse the performance of a number of applications

1.4 Dissertation Outline Overview

This dissertation includes six chapters. Chapters' titles and brief description of them are as following:

- **Chapter 1:** it is the introductory chapter, which includes: introduction, background information of the project, project aim description, research objectives and dissertation outlines.
- **Chapter 2:** it is the Literature Review chapter, including a research on the Quality of Service Mechanisms in regards to increasing the performance of system.
- **Chapter 3:** this chapter includes the project methodology and the simulation model description.
- **Chapter 4:** it shows the test scenarios and simulation results concluded with the discussion and analysis of results.
- **Chapter 5:** it includes the consequential effects and project resources.
- **Chapter 6:** this chapter summarizes and concludes the whole work has been achieved in this project.

1.5 Chapter Summary

To sum up this chapter, it includes the introductory information about different generation of telecommunication technologies. The project aim and main objectives have been discussed. Chapters' brief descriptions are included to give an overall overview of the dissertation parts.

Chapter 2 : Literature Review

This project covers a number of topic areas. Description of relevant subjects is introduced in this chapter. Literature includes Quality of Service and 3GPP traffic classes, packet scheduling techniques and investigation of throughput and delay calculation.

2.1 Quality of Service and 3GPP Traffic Classes

Long Term Evolution/System Architecture Evolution (LTE/SAE) network, has large variety of standardisations and requirements. These standardisations are made by different international organisations and operators of 3^{rd} Generation of telecommunication technologies. This collaborated work is concluded with number of rules listed in documents. This collaboration project called 3^{rd} Generation Partnership Project. 3GPP has number of releases; each one specific release has a start and end date. The recent updated release once this project has started is release 8. Release 8 has different number of versions. Parts of this release sections, concern of the requirements of network's Quality of Service. The scope of this project is to look in details to the Quality of Service requirements in relation to traffic. Therefore, the following subsections discuss the traffic classes, quality of service differentiation and Long Term Evolution Quality of Service mechanism.

2.1.1 Traffic Classes

It is the QoS class level that is defined by the operator and they should be considered as traffic limitations. In this model the most popular four QoS and traffic classes are significant. The first class is the high sensitive delay class or so-called Conversational Class. The second class is the Streaming Class, which is lower sensitive delay class than the first one. Both of the first two classes are used for real-time traffic flow. The third class is the Interactive Class, while the fourth one is the lowest sensitive delay class the Background class. These traffic classes are specified in *3GPP TS 23.107*

V8.0.0.Specifications. Table 1 below shows the four QoS classes and their popular application.

QoS Classes (Traffic Classes)	Popular QoS Class Application
Conversational (q0 class)	Voice and Video conferencing
Streaming (q1 class)	Streaming Video and audio
Interactive (q2 class)	Web browsing
Background (q3 class)	Telemetry and background process (email background download, web pages serving).

Table 1: QoS Classes (Traffic Classes)

2.1.2 Quality of Service Differentiation

Once the network is not loaded, the Quality of Service differentiation is not required. But once the network gets loaded, the QoS differentiation is highly recommended. The definition of QoS differentiation is depicted in Figure 3 below.

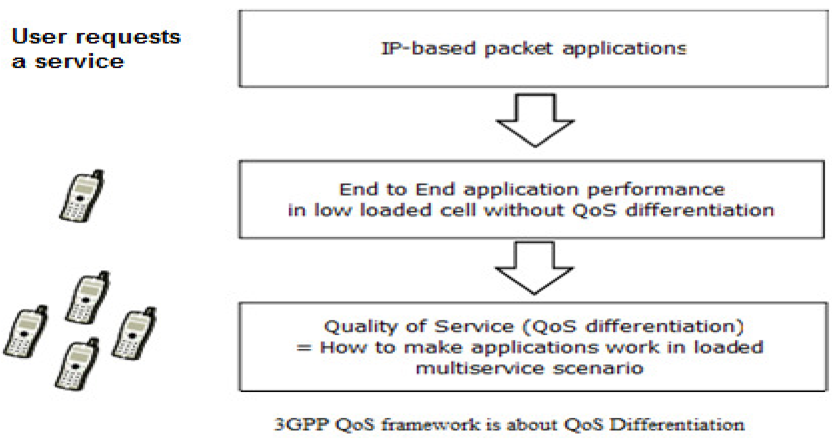

Figure 3: Definition of QoS differentiation

The real-time classes are the ones get benefits from the QoS differentiation, while the best effort classes are not much affected by the technique as they are low-delay service traffic. Ones the eNodeB node has the ability to identify different types of traffics; it can prioritise the traffic in relation to time sensitivity (Holma & Toskala 2007).

2.1.3 LTE QoS Mechanism

As it is defined in 3GPP TS 36.300, LTE/SAE Quality of service is a set of criteria and improvement parameters that have been defined to the network by the operator. These parameters would define the Maximum Bit Rate (MBR) which is the threshold value of the traffic rate, and they define the Guaranteed Bit Rate (GBR) value. QoS parameters and bearers are controlled by the signalling procedures in the Evolved Packet System (EPS), where the bearer is the mean of identifying the transfer and control of the stream of data packets and signals, which have been improved in accordance to QoS assigned for the network. The architecture of LTE bearer service is shown in Figure 4 below

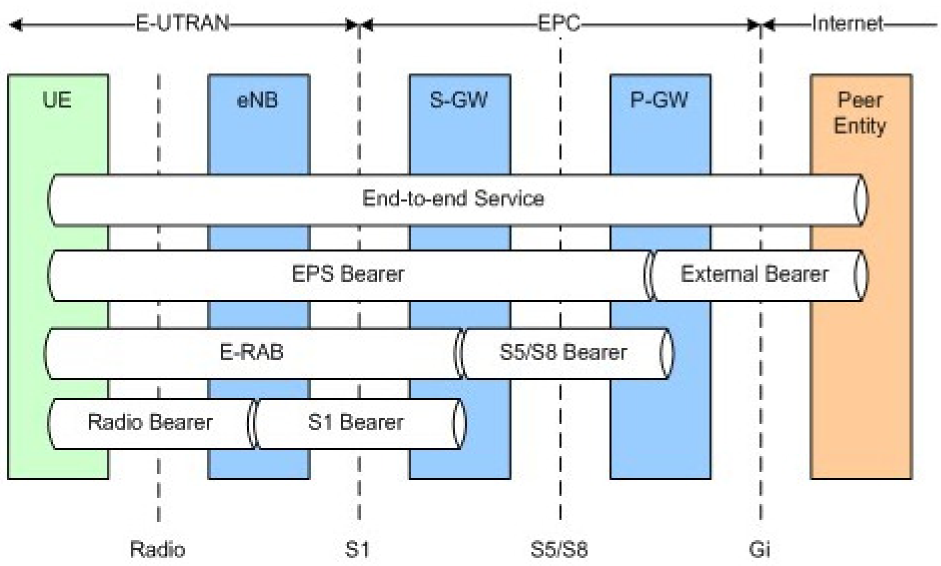

Figure 4 : Architecture of LTE bearer service (3GPP TS 36.300)

Evolved Packet System (EPS) bearer is the stream of packets in between the user equipment (UE) and Packet Data Network Gateway (PDN-GW). Between each specific mobile equipment and service, there is a stream of data called Service Data Flows (SDFs). Evolved Packet System bearer/E-UTRAN Radio Access Bearer (EPS/E-RAB) is the indicator of bearer's granularity in the radio access network. This means that a specific QoS treatment will be applied to all packets in same EPS bearer. An example of such treatment is a prioritisation scheduling. The Quality of Service Class Identifier (QCI) is to give an identity value to a specific bearer to identify the type of class included in this bearer. The QCI specifications are preset by network's operator. Once the bearer has been firstly created, the GBR value would specify the type of resources that this bearer need. Services assigned to default bearers will experience Non Guaranteed Bit Rate (Non-GBR). It is standardized that the number of QCI values are unified due to roaming between networks. Table 2 below show this standardisation (Alcatel & Lucent 2009).

QCI	Resource Type	Priority	Packet Delay (ms)	Service Example
1	GBR	2	100	Conversational call
2		4	150	Live Streaming
3		3	50	Real-time gaming
4		5	300	Buffered streaming
5	Non-GBR	1	100	IMS Signalling
6		6	300	TCP-based buffered streaming
7		7	100	Interactive
8		8	300	Background
9		9		

Table 2: QCI Standardisation (Modified from LTE World Summit paper, Berlin, 2009)

Once the call is firstly admitted to the network, it uses a parameter called Allocation and Retention Priority (ARP) to test if the nominated bearer is suitable to its type of traffic. Therefore, the main advantage of ARP parameter is to avoid any prospected risk results from the wrong assigned bearer. In layer two of the transport layers shown in Figure 4, the radio bearer is the stream of data used to transfer data in between the user equipment (UE) and eNodeB and it can be named as X1 bearer. S1 bearer is the stream of data in between eNodeB and Core Network (CN) which is used to transfer data between the Access Network (E-UTRAN) and Core Network (CN). In addition, S5 and S8 bearers are the bearers used in between Core Network elements (Alcatel & Lucent 2009).

According to Vadada (2009), to distinguish between the QoS service differentiation in LTE and QoS differentiation of WiMax, a comparison has been made in Table 3 below.

	QoS Transport Unit	Scheduling Types	QoS Parameters	QoS Handling in the Control Plane
WiMax	Service Flow	UGS rtPS nrtPS BE	MSTR≠MRTR	Network and user initiated control
LTE	Bearer	GBR Non-GBR	GBR=MBR	only network initiated QoS control

Table 3 : LTE and WiMax Contrast

The comparison made in Table 3 is generally divided in four items:

- QoS Transport Unit: the transport unit used in WiMax as specified in IEEE 802.16 is the Service flow which is the packets flow connecting the mobile station to base station while LTE uses bearer between the mobile phone and gateway.
- Scheduling Type: WiMax uses different type of schedulers, they are Unsolicited grant service (UGS) for fixed size real-time traffic, Real-time polling service (rtPS) for changeable real-time traffic, Non-real-time polling

service (nrtPS) for delay-tolerant traffic which need some rate to be reserved and Best effort (BE) service for usual services where LTE uses Guaranteed Bit Rate and non-Guaranteed Bit Rate bearers.

- Quality of Service parameters: WiMAx lets the operator to predefine traffic prioritisation and Maximum Sustained Traffic Rate (MSTR) and Minimum Reserved Traffic Rate with different values while LTE state that the operator must set Guaranteed Bit Rate and Maximum Bit Rate at same values.
- Control Plane: in WiMax the network initiated control or the user initiated control and network initiated control are both available while in LTE only network initiated control is available.

2.2 Packet Scheduling

This Section will discuss number of subsections. These subsections include packet data protocols, transport channels, and packet scheduling techniques have been investigated by number of researchers.

2.2.1 Packet Data Protocols

According to Holma & Toskala (2007), each transport protocol has different characteristics. The traffic to be transported must be suite its transport protocol. Therefore, real-time traffic such as conversational traffic uses Real-Time Protocol (RTP) via User Datagram Protocol (UDP) transport protocol. In addition, Best effort traffics are transported via Transmission Control Protocol (TCP). Figure 5 below shows the real and non-real time protocol.

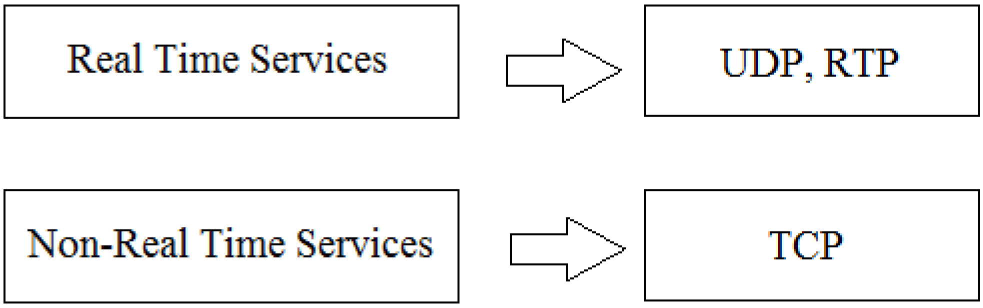

Figure 5 : Typical packet protocols for real and non-real time services.

The real-time traffics such that conversational and streaming traffic needs guaranteed bit rates where the non-real time traffics doesn't need guaranteed bit rate. Conversational real-time traffic doesn't need scheduling and it is transmitted over Dedicated Channel (DCH). Figure 6 below shows the mapping of traffic classes to scheduling and to transport channels.

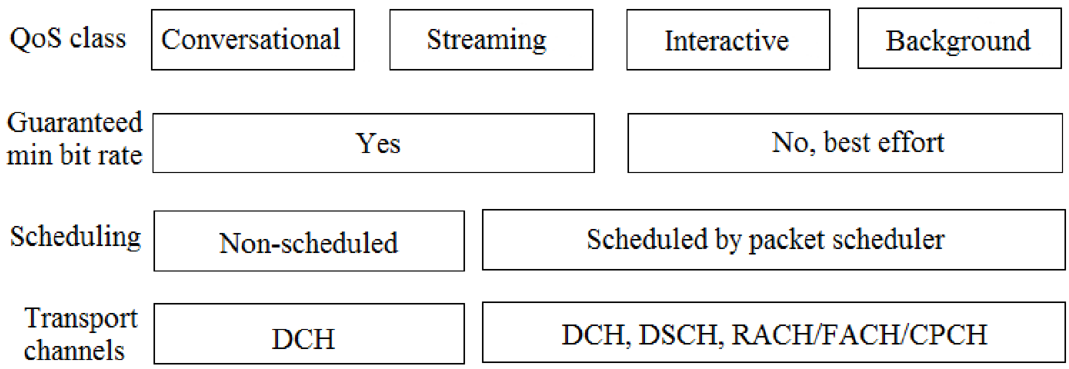

Figure 6 : Mapping of traffic classes to scheduling and to transport channels

Regarding to 3GPP TSG RAN, TS 36.413 V8.5.0, user plane is mainly provided by eNodeB entity. The user plane includes protocols to execute number of functions. This protocol stack is depicted in Figure 7 below.

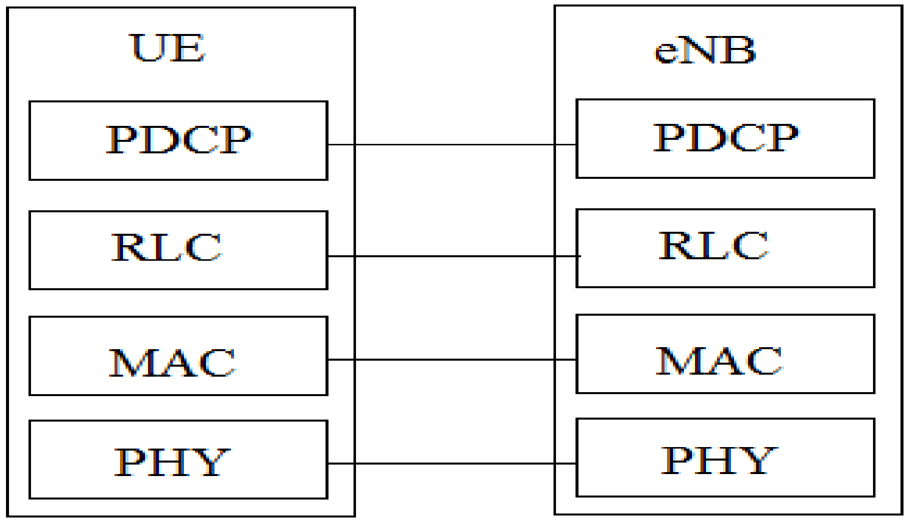

Figure 7 : User Plane Protocol Stack

Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC) layers are terminated in eNodeB. The main functions behind the user plane are scheduling, ciphering, header compression, ARQ and HARQ.

The control plane that controls the user plane is as shown in Figure 8 below.

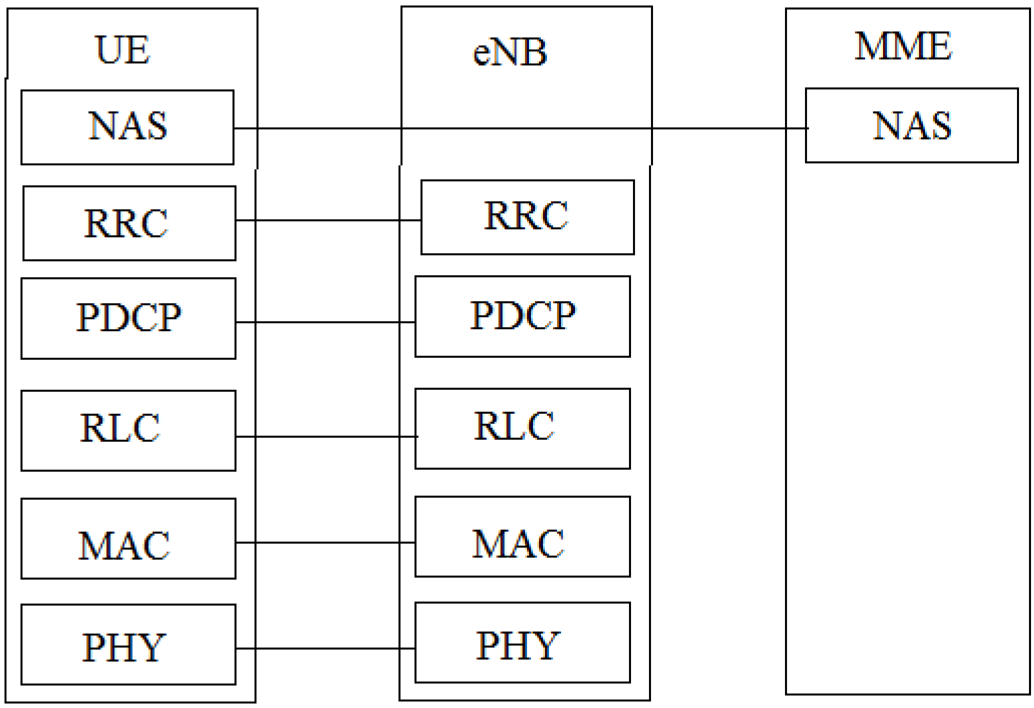

Figure 8 : Control Plane Protocol Stack

The new layers introduced in Figure 8 are the Radio Resources Control (RRC) that terminated in eNodeB from the network side, and the Non-Access Stratum (NAS) that terminates at the MME from the network side. RRC main functions are radio resource management, scheduling, compression and decompression and many others.

2.2.2 Packet Scheduling and Transport Channels

The packet scheduler in LTE network is located in eNodeB. The air interface measurements is measured by the eNodeB itself and they are forwarded to the packet scheduler. In addition, the information of size of the upstream traffic is provided by user equipment (UE). This section includes the user packet scheduler description

2.2.2.1 Transport Channels for User Scheduler

. There are number of transport channels used in LTE network in regards to packet transfer. These transport channels and their relation to scheduling is discussed in the following subsection.

2.2.2.1.1 Common Channel

Common Channels in LTE network are the Random Access Channel (RACH) as an uplink channel, where the downlink common transport channel is Forward Access Channel (FACH). These two channels are used as user data transport channels. The common channels can cause more interference than other channel types due to that they can use soft handover (Holma & Toskala 2009, pp. 271-2).

2.2.2.1.2 Dedicated Channel (DCH)

DCH channel is an efficient channel, and it has both directions uplink and downlink. In comparison to the common channels, dedicated channels has feedback channel which make it more powerful than common channels. DCH doesn't cause any significant interference in the presence of soft handover. The only disadvantage of this channel is that the delay is more than it is in the common channel due to that the accessing time of DCH is more than the common channels. According to 3GPP, the bit rate of this channel reaches to 2Mbps (Holma & Toskala 2009, pp. 272-3).

2.2.2.1.3 Downlink Shared Channel (DSCH)

This type of channels used for bursty packets transport. It is used with DCH in parallel, if DCH is a lower bit rate channel. It is not suitable in the case of soft handover. It is suitable for the slow start TCP. This channel can be assigned to another UE before the time located for DCH is up, due to this it is called shared channel (Holma & Toskala 2009, pp. 274-5).

Table 4 below shows the transport channels overview.

	DCH	DSCH	FACH	RACH
RRC state	Cell_DCH	Cell_DCH	Cell_FACH	Cell_FACH
UL/DL	Both	DL	DL	UL
Suited for	Medium and large data size	Medium and large data size	Small data	Small data
Suited for Bursty data	No	Yes	Yes	Yes
Available in first network and terminal	Yes	No	Yes	Yes

Table 4 : Transport Channels Overview (Modified from Holma & Toskala 2009, p. 274)

2.2.2.2 Cell Packet scheduler

According to Holma & Toskala (2009), the capacity of the non-real time is equally divided between the users in the cell. This is divided with the aid of packet scheduler. A periodic division has been executed by the cell packet scheduler within a range from 100ms to 1 second. Once it is overloaded, the packet scheduler helps user equipments to not losing the service by decreasing the bearer's bit rate. The idea behind the scheduling is to use the available non-real time traffic to assist the user equipments with service improvement.

The cell packet scheduler needs number of inputs to do scheduling efficiently, and these inputs are as follow:

- The power value of eNodeB
- The estimated load
- Bearers capacity in relation to throughput, especially non-real bearers
- Maximum interference level can be tolerated
- Requests of bit rate need upgrade.

Figure 1 below shows the main principle of the input information and calculations needed by the cell packet scheduler.

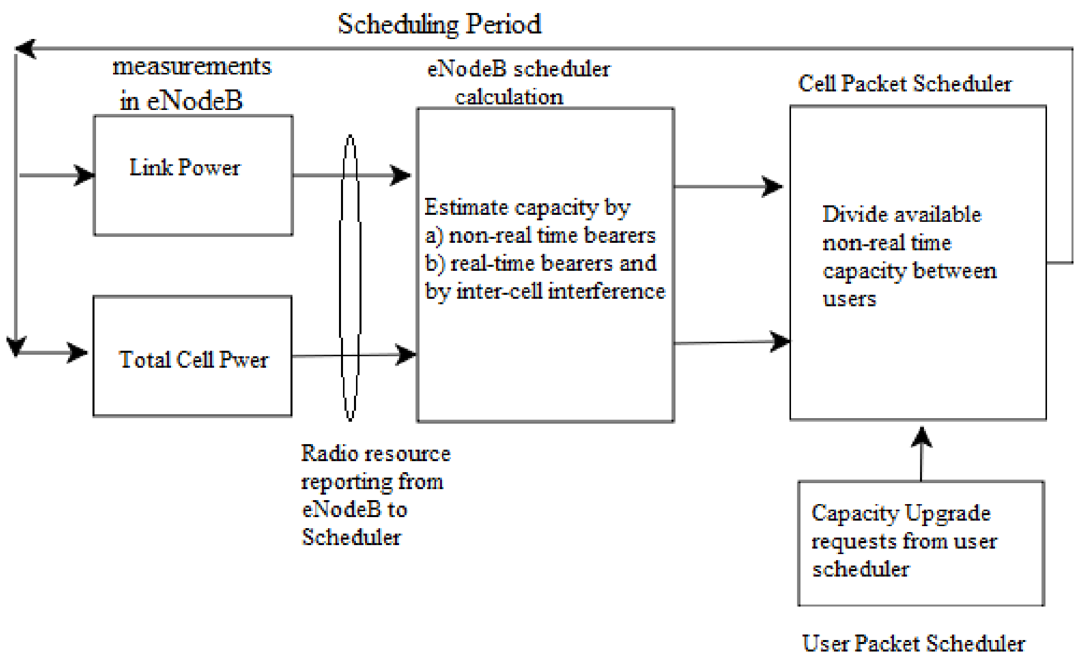

Figure 9: Principles of input information of cell packet scheduler (Modified from Holma & Toskala 2009, p. 280)

2.2.2.2.1 Priorities

The Quality of Service parameters are set by operator and it is provided to the EUTRAN network from core network (CN). The quality of service parameters includes number of important information such that allocation retention priority and the network traffic classes. This information is provided to let eNodeB distribute the radio resources between users efficiently. From these parameters, the packet scheduler gets clear picture of how the capacity to be distributed. The bearers that they are specified for high priority traffic will use the available capacity while the bearers with low priority will hold until the previous bearer is executed. Different allocation priority number is assigned to different traffic types. In one of the relevant studies (Falconio & Dini 2004) Examples of prioritisation algorithm are as follow:

- Weighted Fair Queuing (WFQ):
  In this strategy, the buffer with the highest number of data will be served first by the link.
- Early Deadline First(EDF):
  After assigning a deadline to each packets in the buffer, the lowest deadline packets are the once to be served first.
- Hybrid Algorithm:
  It is a mix between the WFQ and EDF, where the buffer length, arrival time of packets and the class each packets are belong to are known. The packet to be served is the ones with either highest length, or the lowest deadline figure.

2.2.2.2.2 Scheduling Algorithm

In regards to the provided parameters, cell packet scheduler chooses the suitable bearer to every single type of incoming traffic. As an example of this, Figure 10 below shows the DCH bit rate allocation.

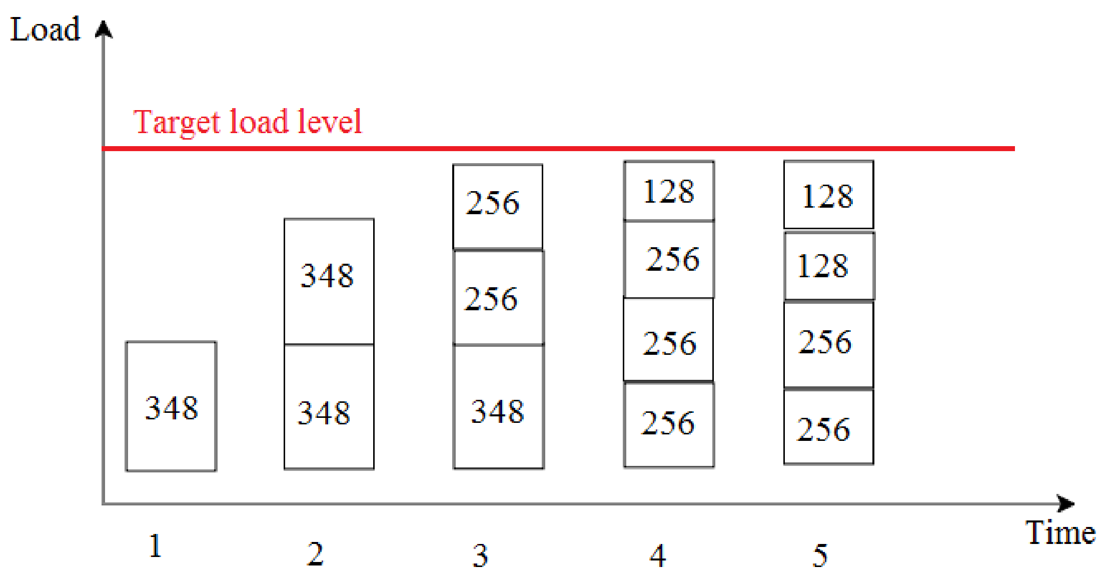

Figure 10 : DCH bit rate allocation

The capacity of cell is assumed as 900 kbps. The first user capacity will be 348 kbps; the second one will get the same value once we have only two UE. Once we have 3 users, it is not possible to have 348kbps, then the first one will maintain having 348kbps but the remaining two get 256kbps each and so forth. The downgrading operation represents the bearer bandwidth downgrade. The upgrade operation is permitted in the case that one of the bearers is free of use. This algorithm shows the distribution of resources once there is only one type of traffic classes' available (Holma and Toskala 2009, p. 282).

Different scheduling schemes can be applied in eNodeB more specifically in Radio Resource Management (RRM) as operator needs. Some of them depend on the channel behaviour and link characteristic. Another Scheduling schemes are in regards to the Traffic Behaviour. At low load system, there is no significant difference between different scheduling schemes. This difference is highly significance in high load systems. One more factor will make the difference more significance is the behaviour of the Traffic. According to Holma & Toskala (2002), number of popular algorithms is discussed in the following three sub-sections.

2.2.2.2.3 Fair Throughput Scheduling Algorithm

The scheduler receives information about the bit rate offered in the cell. This available bit rate will be distributed equally between the users. This algorithm can be used for real-time traffic as it can offer a guaranteed bit rate to the users.

2.2.2.2.4 Fair Time Scheduling

In this scheduling scheme, all user equipments are assigned at same power. The throughput is distributed in regards to the channel condition. The user equipments enabled to use higher throughput are the user equipments with high channel quality.

2.2.2.2.5 C/I Scheduling

At full buffer state(holding the packets for later use between two different telecom nodes with the maximum availability, which means the maximum availability of the buffer is used), the edge UE will have very low throughput available.

Schedulers such as the latest three types, use Time Domain Packet Scheduler (TDPS) followed by Frequency Domain Packet Scheduler (FDPS) phases.

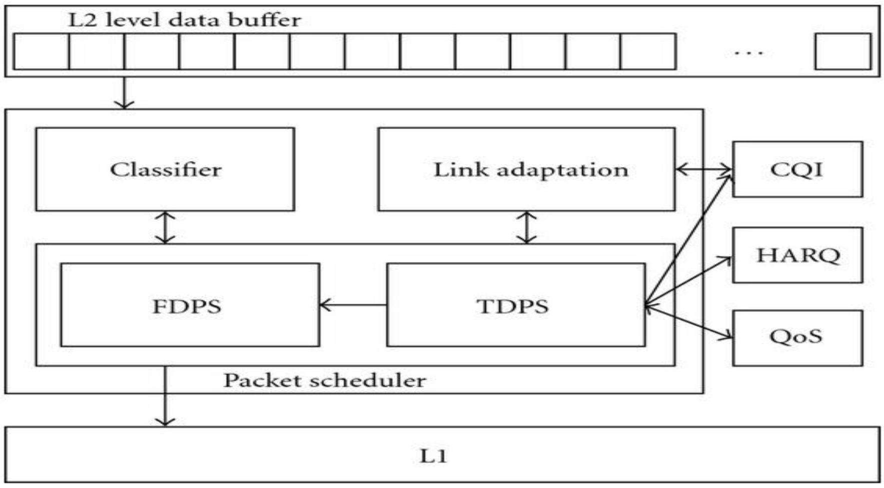

Figure 11 : the structure of RT and NRT Traffic packet scheduler in eNB (Song et al, 2010)

Once we have more than one traffic class in a system, the classifier is important for the packet scheduling. Different queues with different priorities are set by the classifier for different traffic types. It can be seen from Figure 11 above that the classifier at layer 2 buffer assigning the each stream of one type of traffic with its class (Song et al, 2010). There are different Quality of Service requirements for each class.

The degree of fairness as it is discussed above is achieved with interaction between the HARQ and Packet Scheduler (PS). This interaction is depicted in Figure 12 below.

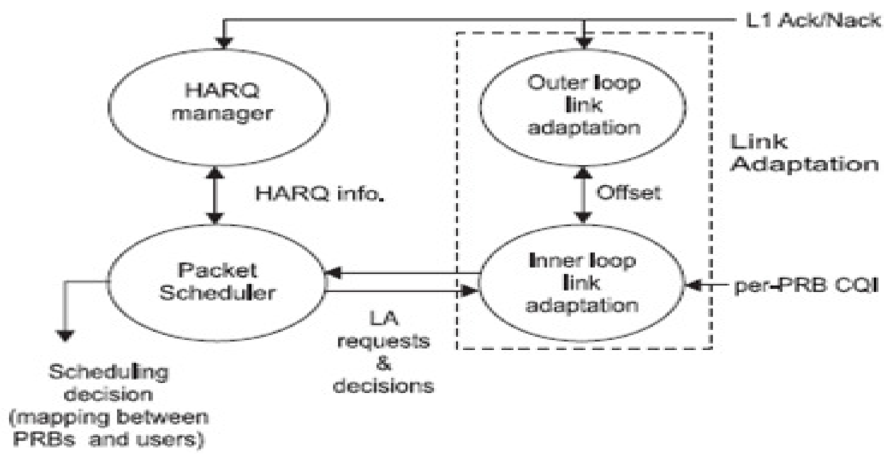

Figure 12 : Interaction between P, LA &HARQ, in Frequency Domain (Pokhariyal et all, 2007)

By looking at Figure 12, it shows the cooperation between the Packet Scheduler (PS), HARQ management and Link Adaptation (LA). These are existed in the eNodeB. Packet Scheduler is the controlling item. Physical Resource Block (PRB) is showing the resolution of the scheduling. In the frequency domain the bandwidth of PRB is 375 kHz minimum for one block and it is 24 PRBs in the 10MHz BW.

The Packet Scheduler will get a help from LA to know the data rate of different users having different PRBs. The Link Adaptation is highly dependent on the Channel Quality Indication (CQI). CQI helps LA as it obtain the users' feedback. HARQ manager has information about the buffer (Pokhariyal et al 2007).

Time Domain (TD) scheduler will identify the number of users N in the highest priority to go to Frequency Domain (FD) Scheduler. TD considers the new data users and keeps in touch with the users with retransmission requests which are not prioritized by TD. FD scheduler's function is to give M available number of the PRBs to N users. Figure 13 shows the HARQ management steps (Pokhariyal et all, 2007).

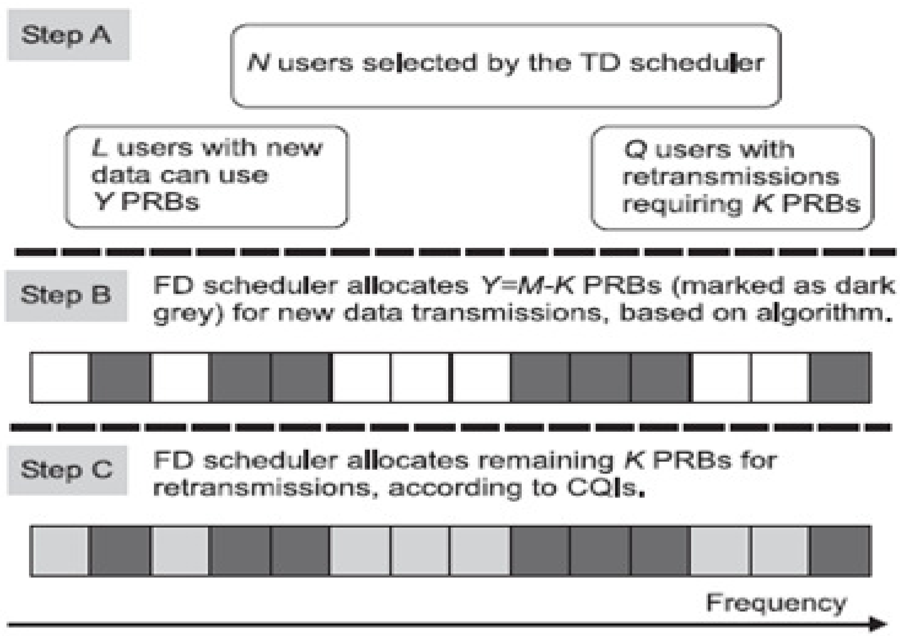

Figure 13 : Mechanism used by the FD scheduler to allocate PRB resources to 1st transmission and retransmission users(Pokhariyal et all, 2007)

2.3 Throughput and Delay Calculation

According to Oleg Berzin (2010), throughput and delay calculation method of network is provided. To understand how the packets delay and throughput are calculated, the following network example is provided.

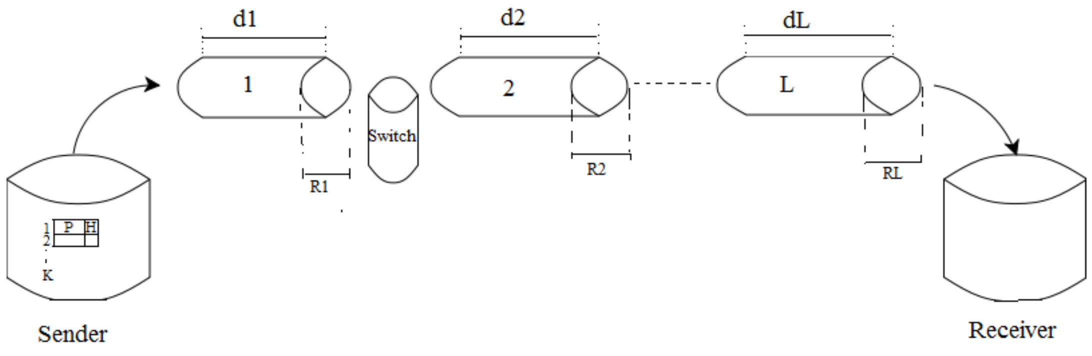

Figure 14 : Network Delay and Throughput Calculation

Number of elements and figures in Figure 14 are to be explained first.

di: is the delay of link where i=1,2,3....L

Ri: Link bit rate, where i=1, 2, 3.....L

P: Number pf payload bit per packet

H: number of Header bits per packet

K: Number of packets per window

The general equations to find the sent packets' delay is shown in equation 1 below

$$D = K\left[\sum_{i=1}^{L}\left[\frac{P+H}{Ri} + di\right]\right] \quad \text{Equation 1}$$

For the throughput calculation, it is simply found by dividing the bits by the time taken to be transported.

To find a specific link throughput is as follow:

$$Th = \frac{K(P+H)}{[K\frac{P+H}{R} + d]} \qquad \text{Equation 2}$$

2.4 Chapter Summary

To sum up, this chapter has investigated number of relevant topics. The Quality of Service of Long Term Evolution network has been discussed. The techniques and algorithms of packet scheduling have been described. The chapter concluded with basic methods of calculating network's throughput and delay.

Chapter 3 : Project Methodology and Simulation Model

3.1 Introduction

The major goal of this project is to evaluate QoS features of the LTE/SAE network, and to build a model simulating this network. The network performance will be evaluated by analysing the throughput and delay while QoS parameters are enabled as well as when they are disabled. A comparison between the two situations triggered and non-triggered QoS features will be investigated. One more criterion to be examined is the packet flow of information being transmitted edge to edge from the eNodeB to the GateWay.

3.2 Network Simulator 2 (NS2) and LTE/SAE virtual network model

An initial step to complete this project is to be familiarized with the tools required for building a LTE/SAE virtual network. Background knowledge of the LTE/SAE system itself is a supportive factor to build a model. Network Simulator 2 (NS2) is the nominated software simulator to model the system and to obtain accurate results. It is industry approved software that is widely used by communication engineers for networking research. The LTE/SAE model will include the Network Model and Traffic Model. The Network Model includes the following interfaces Air interface and S1 interface. Separating these two models would increase the reusability and flexibility of the LTE/SAE model.

The basic architecture of the Network Simulator is as depicted in Figure 15 below

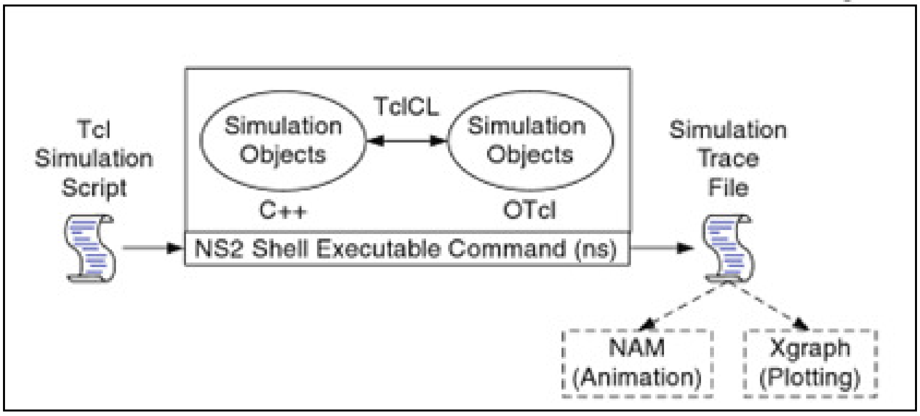

Figure 15 : Basic Architecture of Network Simulator (Issariyakul & Hossain, 2009)

As it is illustrated in the above figure, the NS2 mainly consists of two main languages, C^{++} and Object-oriented Tool Command Language (OTcl). C^{++} in NS2 is mainly existed for simulation objects' mechanism definition. The OTcl is used for simulation scheduling discrete events' setting. The main simulation script as shown in the above figure is the very far left item. It is used to set up network topology, links, setting the type of traffic between nodes and much more input arguments. It is run via an executable command ns followed by the name of the script. The name of this project main simulation script is *lte.tcl* and it is provided in the appendixes. The first right hand item is the output information file, which can be used to obtain and graph the results.

3.3 Network Model and Implementation

LTE/SAE Model and its Implementation in NS2, is a project's paper investigates Long Term Evolution network performance (Qiu et al, 2009). The results have been tabled in the paper were initially faulty. Starting from this paper, an improved project simulation model is made to test the performance of LTE/SAE network.

This project simulates one cell of LTE/SAE network. Therefore the part of network needed for simulation is as illustrated in between blue borders of LTE/SAE network in Figure 16 below.

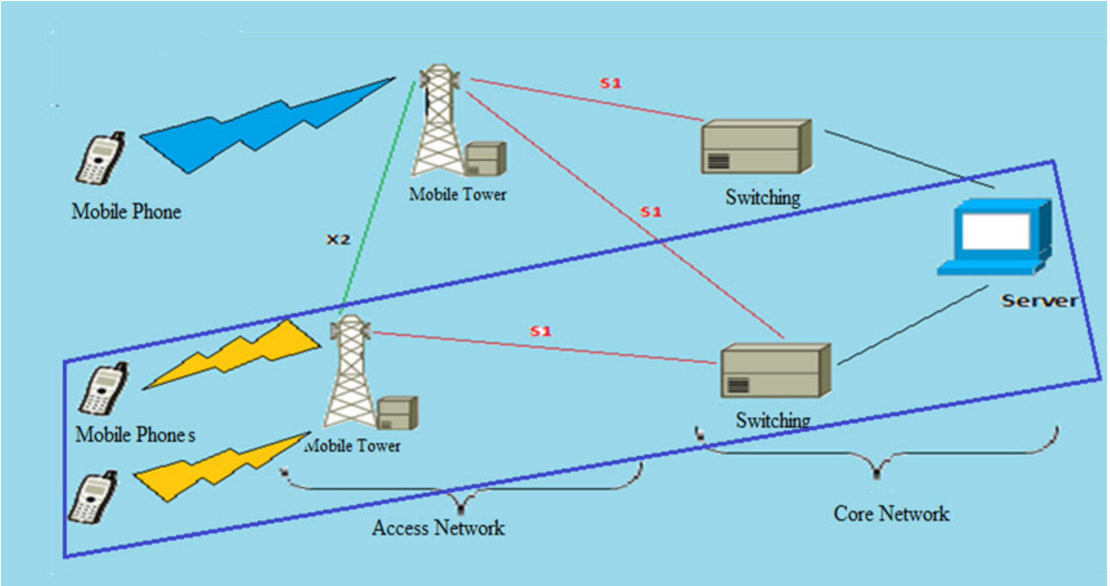

Figure 16 : The parts of LTE/SAE network to be simulated

In clearer picture, the LTE/SAE network model to be simulated is shown in Figure 17 below.

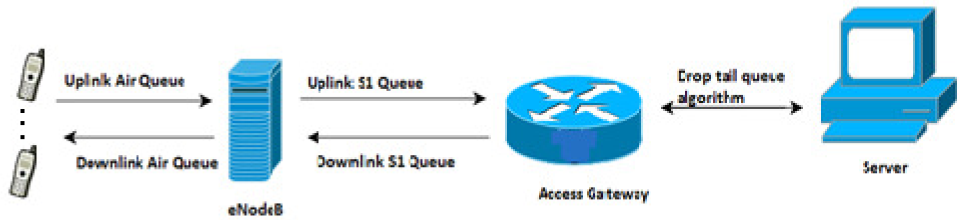

Figure 17 : LTE/SAE Network Model (this figure made using DIA software office power point)

The network model needs to be configured on demand therefore; number of elements and bandwidth parameters between them can be changed, added and/or eliminated. The only limitation to the model that is the eNodeB and Gateway which are permanent and fixed and cannot be moved or multiplied.

The model has five elements. They are: number (quantity) of User Equipment, at least 1; eNodeB, or the base station of the model, which provides the network with the needed flow control data; an Access Gateway to help the network with the flow control and caching Hypertext Transfer Protocol (HTTP) and; one main server providing signalling services, File Transfer Protocol (FTP) and HTTP.

The following figures of codes show how the network model's elements built in tcl file

```
set numberClass0 5
set numberClass1 2
set numberClass2 1
set numberClass3 1
set number [expr {$numberClass0 + $numberClass1 +
$numberClass2 + $numberClass3}]
```

Code 1 : Setting Different Classes' User Equipments Number

The first four lines of the Code 1 above are to set the number of user equipments for each class, where the last line is the total number of user equipments in the network.

The steps of defining network's elements in the script file are demonstrated in Code 2 below

```
set eNB [$ns node];#this node id is 0
set aGW [$ns node];# this node id is 1
set server [$ns node];# this node id is 2
for { set i 0} {$i<$number} {incr i} {
set UE($i) [$ns node];# node(s) id is > 2    }
```

Code 2 : Network Element Definition in tcl file

Node 0 in the simulation stands for eNB, node1 refers to Access Gate Way, the Main Server's number is 2 and any number greater than 2 refers to user equipments.

To connect the nodes to each other, instproc unidirectional and bi-directional links are used. These steps are illustrated in the Code 3 below.

```
for { set i 0} {$i<$number} {incr i} {
      $ns simplex-link $UE($i) $eNB 10Mb 2ms
LTEQueue/ULAirQueue
      $ns simplex-link $eNB $UE($i) 10Mb 2ms
LTEQueue/DLAirQueue
}


$ns simplex-link $eNB $aGW 100Mb 2ms LTEQueue/ULS1Queue
$ns simplex-link $aGW $eNB 100Mb 2ms LTEQueue/DLS1Queue
.

$ns simplex-link $aGW $server 1000Mb 2ms DropTail
$ns simplex-link $server $aGW 1000Mb 2ms DropTail
```

Code 3 : Node Connection Script

In Node Connection Script, Code 3 above, the upper part of the script is to set the number of air interface with reference to the total number of active user equipments in the network. The first instproc uni-directional link is to connect the user equipment to eNodeB as an uplink link with bandwidth equal to 10 Mbps, and it is assigned to ULAirQueue flow controller. The next line is for the downlink connection between same nodes and the flow controller is DLAirQueue. Delay of all links is 2msec. The middle part of the script is to set the connection between eNodeB and access Gate Way which is the bottleneck link, and it is assigned to the flow controllers DL and UL S1 queue. The bandwidth of this link will be downgraded from 100Mbps to 2Mbps due to the limitation of computer's memory used in this project. The general behaviour is same with both values of the bottleneck bandwidth as it is tested previously. The bottom part of the script is to create a connection between the main server and the access gateway and it is assigned to the normal Drop tail.

3.3.1 Flow Control

To avoid any packet loss due to downlink limitation, Flow Control is required. The flow control has been represented in the model with Air interface downlink Queue and S1 interface downlink Queue Script files. Air interface downlink Queue Script file provides the flow with the required information such as average data rate while S1 interface downlink Queue Script File decides whether or not to send the packet to the Air Interface down link queue in accordance to available information. The scripts of the flow control are written in C++ language and they are included in the appendixes.

3.3.2 Network Configuration

S1 and Air interfaces are to be simulated by setting and implementing queue classes in the network model. These Queue Classes are LTE main Queue, Air interface down link queue, S1 interface Uplink Queue, and S1 interface downlink Queue. In the main class, the condition of whether or not to use these optimization features are defined. Other classes define the interface implementation.

3.3.3 Packet Classification and Scheduling

The packets enter the main queue with identification of the class and sub queue where they come from. When the Quality of Service features are triggered, the q0 packets travel first to Drop Tail queue, then q1 and q2 respectively. The q3 packets will travel to the Red queue. But when QoS features are disabled all the packets travel to Drop Tail queue and the first packet in the first packet out. The order of packet transmission when QoS features are enabled, is q0 then q1 then q2 and finally q3 as it is the least priority packet, will remain in hold state until all the high priority packets have been sent. This queuing technique is depicted in Figure 18 below

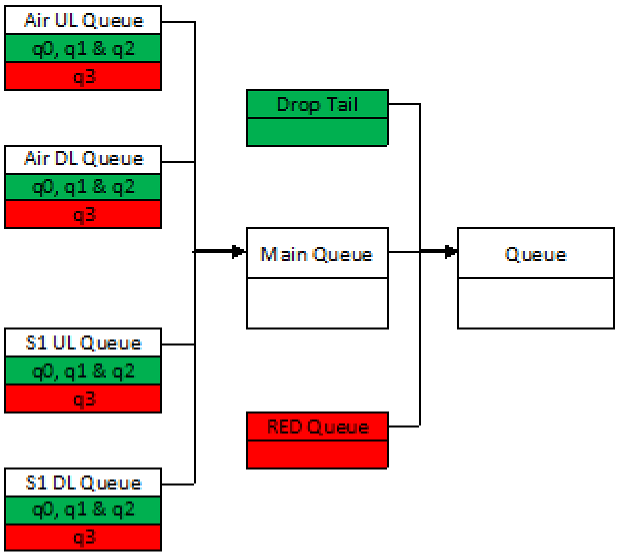

Figure 18: Queue classes (this figure made using MS office word and MS painter)

3.3.4 Traffic Model

In this model the most popular four QoS and traffic classes are simulated. The first class is the high sensitive delay class or so-called Conversational real time class, is usually simulated with Real Time Protocols. The second class is the Streaming Class, and it is usually simulated by using Constant Bit Rate protocol. The third class is the Interactive Class and it is simulated by using Transport Control Protocol Agent, while the fourth one which is the Background class and it is simulated by setting Forward Transport Protocol over TCP agent. Table 1 below shows the four QoS classes and the protocols and agents used to drive the traffic while the next subsections are to describe the traffic simulation in more details.

QoS Classes (Traffic Classes)	Popular QoS Class Application	Simulation Protocol Used
Conversational (q0 class)	Voice and Video conferencing	Session/RTP -Session/RTPAgent -Session/RTCPAgent
Streaming (q1 class)	Streaming Video and audio	CBR/UdpAgent
Interactive (q2 class)	Web browsing	HTTP/TcpAgent - HTTP/Client - HTTP/Cache - HTTP/Server
Background (q3 class)	Telemetry and background process (email background download, web pages serving).	- FTP/TcpAgent

Table 5: QoS Classes (Traffic Classes) and Simulation Protocol Used

3.3.4.1 Class0 Traffic Simulation

To simulate the conversational real-time traffic, it has been initially simulated by using rtp protocol as Code 4 : Real-time Traffic Simulation with aid of rtp protocol in tcl Code tcl coding below shows

```
for { set i 0} {$i<$number} {incr i} {
     set s0($i) [new Session/RTP]
     set s1($i) [new Session/RTP]
     set group($i) [Node allocaddr]

     $s0($i) session_bw 12.2kb/s
     $s1($i) session_bw 12.2kb/s
     $s0($i) attach-node $UE($i)
     $s1($i) attach-node $server
     }
```

Code 4 : Real-time Traffic Simulation with aid of rtp protocol in tcl Code

After simulating real-time traffic, unpredicted results for the throughput are obtained where the sent traffic is much lower than the received packets without any packet drop. After observing the traffic by running NAM network animator in NS2, it shows that this traffic is a broadcasting traffic. The following figures will show the traffic animation has been observed.

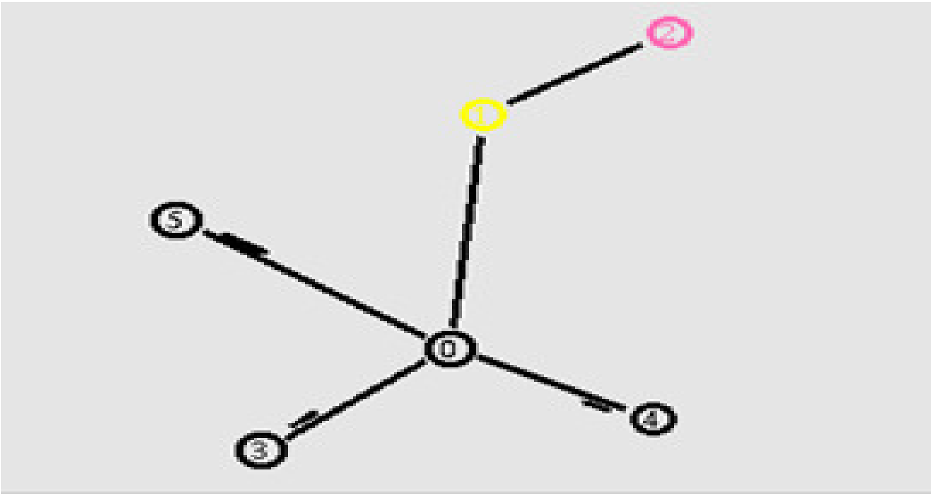

Figure 19 : RTP Traffic Travelled Initially from UEs

Figure 19 above shows the RTP traffic once it is initially travelling from UEs towards the eNB. Nodes number 3,4 and 5 are the user equipments. Node0 is the eNB. Node 1 is the Access Gate way. And the node number 2 is the main server.

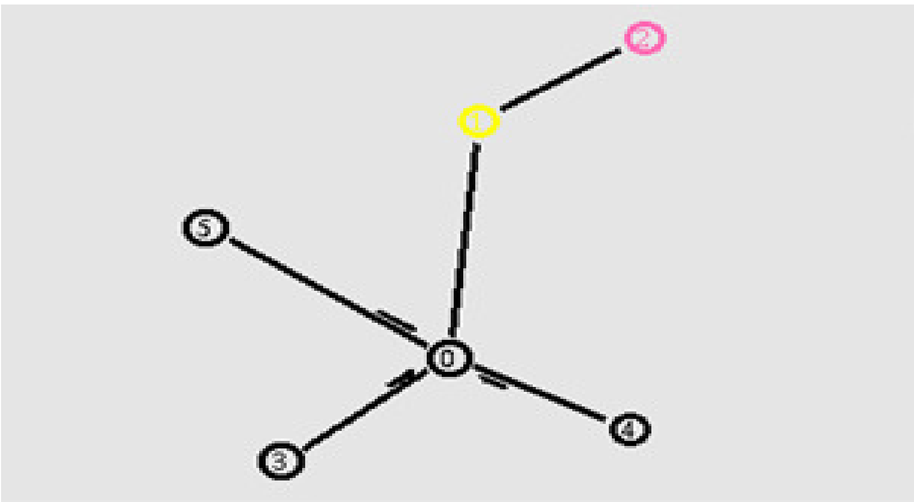

Figure 20 : RTP Traffic Arrives eNB coming from UEs

Figure 20 above shows RTP traffic about to arrive to eNB

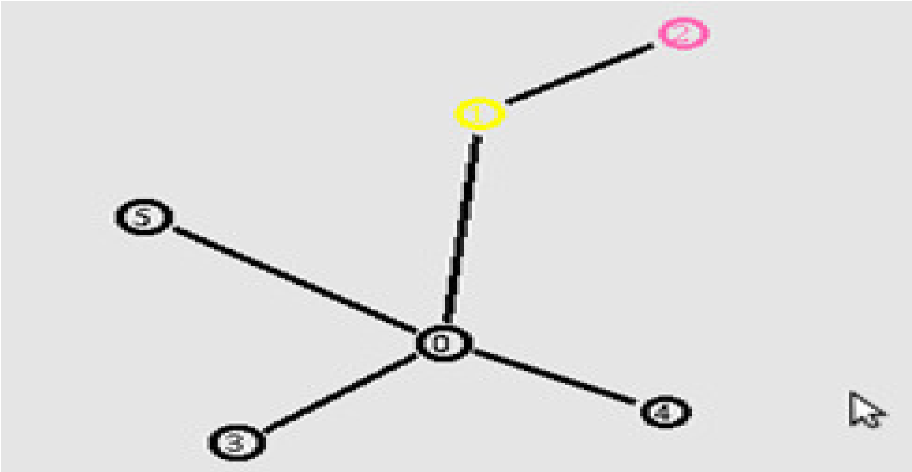

Figure 21 : All RTP Traffics enter the eNB

Figure 21 above shows the moment that all RTP traffic packets have entered the eNB node.

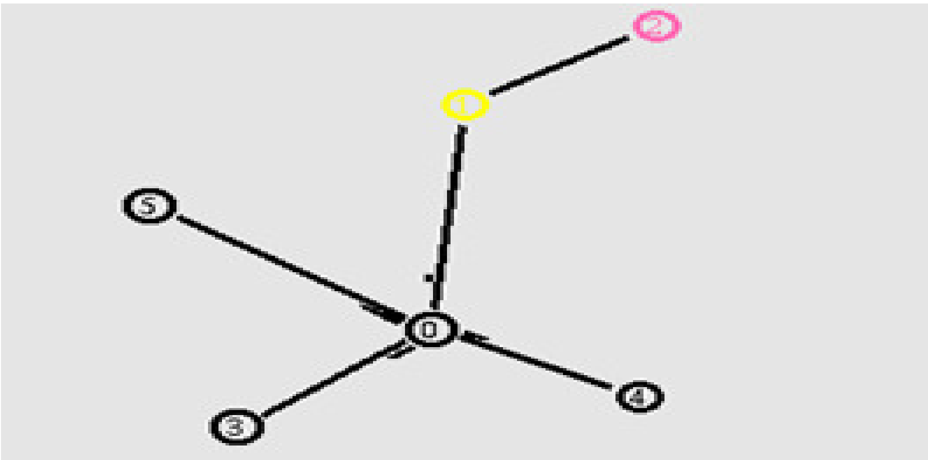

Figure 22 : RTP broadcasting

Figure 22 above shows the problem clearly where packets are broadcasting traffic, where they are broadcasted to all nodes at the same time, going to all user equipments as well as access gateway. This is not suitable for telephony service.

To solve this problem, CBR traffic to be used in both directions sends and receives direction with the aid of UDP agent.

Code 5 below shows the part of script codes to create conversational traffic

```
for { set i 0} {$i<$numberClass0} {incr i} {
     set null($i) [new Agent/Null]
     set nullS($i) [new Agent/Null]
     $ns attach-agent $UE($i) $null($i)
     $ns attach-agent $server $nullS($i)
     set udp($i) [new Agent/UDP]
     set udpUE($i) [new Agent/UDP]
     $ns attach-agent $server $udp($i)
     $ns attach-agent $UE($i) $udpUE($i)
     $ns connect $null($i) $udp($i)
     $ns connect $nullS($i) $udpUE($i)
     $udp($i) set class_ 0
     $udpUE($i) set class_ 0
     set cbr($i) [new Application/Traffic/CBR]
     set cbrS($i) [new Application/Traffic/CBR]
     $cbr($i) attach-agent $udp($i)
     $cbrS($i) attach-agent $udpUE($i)
     $ns at 0.4 "$cbr($i) start"
     $ns at 0.4 "$cbrS($i) start"
     $ns at 40.0 "$cbr($i) stop"
     $ns at 40.0 "$cbrS($i) stop"
}
```

Code 5 : Conversational Traffic codes

The conversational traffic codes in Code 5 above shows that the traffic are set in both direction where the cbr over udp agent are set at the user equipment side which terminated at the server side, and for the other direction the cbr over udp agent is set at the side of server to simulate the conversational traffic travels from server and terminated at the side of user equipments.

3.3.4.2 Class 1 Traffic Simulation

```
for { set i $numberClass0} {$i<
($numberClass0+$numberClass1)} {incr i} {
     set null($i) [new Agent/Null]
     $ns attach-agent $UE($i) $null($i)
     set udp($i) [new Agent/UDP]
     $ns attach-agent $server $udp($i)
     $ns connect $null($i) $udp($i)
     $udp($i) set class_ 1
     set cbr($i) [new Application/Traffic/CBR]
     $cbr($i) attach-agent $udp($i)
     $ns at 0.4 "$cbr($i) start"
     $ns at 40.0 "$cbr($i) stop"  }
```

Code 6 : TCL Script Streaming Codes

While the streaming traffic comes from the server to the user equipments, the traffic agent and protocol to be set in the side of server and terminated at the side of user equipments where the null agent to be set. To avoid any misdistributions of the traffic over networks the number of user equipment must not set as numbers, but it must use a range distribution to avoid distributing more than one traffic type to the same user equipment. This is clearly shown in the first line of the Code 6 above where the first user equipment using class1 is equal to the last one who is using the class0 traffic plus 1. The number assigned to last user equipment to use class1 traffic is equal to number assigned to last user equipment using class0 plus the number of user equipment will use the class1 traffic.

3.3.4.3 Class2 Traffic Simulation

Over TCP Agent, class2 simulation is done by setting HTTP/Server at the side of main server, HTTP/cache at the side of access gateway and HTTP/client at the side of user equipment. The average page size each user can see is 10K. The script of creating this traffic type is attached in the appendixes.

3.3.4.4 Class3 Traffic Simulation

Over TCP agent, FTP protocol in the side of main server is set to simulate class3. This traffic sink at the side of user equipment. Code 7 below shows the script codes of creating class3 simulation.

```
{$i<($numberClass0+$numberClass1+$numberClass2+$numberCla
ss3)} {incr i} {
      set sink($i) [new Agent/TCPSink]
      $ns attach-agent $UE($i) $sink($i)
      set tcp($i) [new Agent/TCP]
      $ns attach-agent $server $tcp($i)
      $ns connect $sink($i) $tcp($i)
      $tcp($i) set class_ 3
      set ftp($i) [new Application/FTP]
      $ftp($i) attach-agent $tcp($i)
      $ns at 0.4 "$ftp($i) start"
}
```

Code 7 : Class3 codes

3.4 Obtaining Results

This section discusses the script files used to calculate the required results. This section includes number of subsections as follow, trace file, AWK script files, and Random Number Generator Set.

3.4.1 Trace file

In the tcl script file, predefining the trace file is important, due to that; all information of NS2 based simulation is included in the trace file. This file's information sample is shown in the Figure 23 below which is a snapshot of one of trace files obtained in this project tests. The format of the trace file lines is shown in Table 6 below.

```
r 0.40416 1 0 tcp 40 ------- 3 2.0 18.0 0 0
+ 0.40416 0 18 tcp 40 ------- 3 2.0 18.0 0 0
- 0.40416 0 18 tcp 40 ------- 3 2.0 18.0 0 0
r 0.406192 0 18 tcp 40 ------- 3 2.0 18.0 0 0
+ 0.406192 18 0 ack 40 ------- 3 18.0 2.0 0 1
- 0.406192 18 0 ack 40 ------- 3 18.0 2.0 0 1
r 0.408224 18 0 ack 40 ------- 3 18.0 2.0 0 1
```

Figure 23 : Trace File Snapshot

The first symbols in the left hand side in each means as following:

(r): Received packet

(+): enqueue

(–): dequeue

(d): dropped packet

Where the format of each line is as following:

event	time	Source Node	Dest. Node	Pkt Type	Pkt Size	Flags	Flow ID	Src Addr	Dest Addr	Seq Num	Pkt ID

Table 6 : Trace File Line Format

In the main tcl file, the trace file must be predefined and it is predefined in this project main tcl file as it appears in Code 8 below.

```
set f [open out.tr w]
$ns trace-all $f
```

Code 8 : Trace File predifining code in main tcl file

From this trace file, the simulation results are obtained by reading the file. The way it is followed in this project to obtain the results is by writing a script in an AWK file.

3.4.2 AWK Script File

Two awk script files are used in this project's simulation to extract the results out from trace file and doing calculation on trace file's information. The first awk script is to calculate the throughput received, sent and dropped of the first mobile equipment use specific class traffic. The second awk script is used to calculate the time delay of this mobile phone's traffic.

3.4.2.1 AWK Script to Calculate Throughput

In the beginning of this script, number of initializations are to be set such that Code 9

```
BEGIN{
      flag=0;
      UEclass0=-1;
      UEclass1=-1;
      UEclass2=-1;
      UEclass3=-1;
     }
```

Code 9 : AWK file Initialization 1

In this part of initialization, BEGIN is the start of an awk file and the lines in between curly brackets are for main initialization that if satisfied, actions will be executed or more initialization to be set as shown in Code 10 below.

```
     event = $1;
     time = $2;
     node_s = $3;
     node_d = $4;
     trace_type = $5;
     pkt_size = $6;
     classid = $8;
     src_ = $9;
```

Code 10 : AWK Script File Initialization2

The flag=0 is satisfied in initialization 1, then in this code, Code 10, number of actions to be executed which is here giving the columns in trace file realistic names. The dollar sign with numbers refers to the number of column in the trace file's lines' format.

```
 if(event == "-" &&  (node_d == UEclass0)  || (node_d ==
UEclass1) || (node_d == UEclass2) || (node_d ==
UEclass3)) {
     if(flag==0) {
          start_time=time;
          flag=1;
     }
     end_time=time;
     ue_r_byte[classid] = ue_r_byte[classid] + pkt_size;}
}
if(event == "-" && node_d ==2 ) {
 if(src == UEclass0) {
     ue_s_byte[classid]=ue_s_byte[classid]+pkt_size;

}
}
if(event == "d") {
     #ue_d_byte[classid]=ue_d_byte[classid]+pkt_size;
```

Code 11 : Throughput Calculation in AWK file

Code 11 above shows that once the event is "-" which means dequeue, we can only calculate the received throughput by user equipments.

In the following statement once the node destination is equal to 2 where the sender is the user equipments is used to give a statement to calculate the sent throughput.

The last statement is to give a limit where the event is"d" which means the dropped packets calculation.

The resultant throughput is in byte which means that it must be divided by million to get the numbers in MByte.

3.4.2.2 AWK script to Calculate Delay

In this script file, the initialization steps are exactly similar to the previous initialization in the throughput awk file. The statements to limit the calculation are the only difference than the throughput statements.

```
if (event == "+" && (node_s ==2))
      {
            packet[pkt_id]=time;
      }

if (event == "r" &&  (node_d == UEclass0)) # || (node_d
== UEclass1) || (node_d == UEclass3))
      {
            if(packet[pkt_id]!=0){
                  delay0[classid,0] = delay0[classid,0] +
time - packet[pkt_id];
                  delay0[classid,1] = delay0[classid,1] + 1;
            }
      }
```

Code 12 : Delay Calculation Statements in AWK file for Classes 0, 1 and 3 Traffic

In this part of the delay script, the statement as it is illustrated in Code 12 above. This statement is only applicable for the traffic travelling from the server which is node number 2 to the user equipment. To find the round trip time of the class0 for example, two statements to be set once the mobile phones is the sender and the node 2 is the receiver plus the time when the sender is the node 2 and the receiver in the return trip is the same mobile phone. The last line is only a counter. The above statements are only applicable for classes 0, 1 and 3 only where the delay of class 3 traffic is illustrated in Code 13 below.

```
if (event == "+" && ( node_s==UEclass2 ))
     {

     packet[pkt_id]=time;
     }
     if (event == "r" && (node_d==1) )
     {
          if(packet[pkt_id]!=0){
               delay2[2,0] = delay2[2,0] + time -
packet[pkt_id];
               delay2[2,1] = delay2[2,1] + 1;
          }
     }
```

Code 13 : Delay Calculation Scripts for Class2 Traffic

In Code 13 above, the delay calculation is limited by a statement state that once only the sender is the mobile equipment and the receiver is the node 1 which is the aGW where HTTP/Cache is located and the class of traffic is 2. This is only applicable for Class2 traffic which is the interactive traffic.

To run these AWK files in NS2 terminal, the two commands in Code 14 are to be used directly in the terminal under the right directory where this project files located.

```
awk -f throughput.awk out.tr

awk -f delay1.awk out.tr
```

Code 14 : Commands to Run AWK Files in NS2 Terminal

Where throughput.awk and delay1.awk are the awk files used to obtain throughput and delay respectively, and out.tr is the trace file named out.

3.4.3 Random Number Generator

The Random Number Generator (RNG) is used to provide randomness in the software simulation. These random numbers are generated by selectively choosing a stream of numbers from pseudo random numbers. To provide kind of confidence to any software simulation results, testing the results at different RNG number is required. An example of this is that, a study is to be made on number of supermarket's customers on Thursday, the long shopping day. Instead of doing the study of one Thursday of single week, the more the number of weeks the more accurate the results will be. This means that the simulation is performed at different situations. RNG can be changed from 1 until 7.6×10^{22}. In this project's simulation, the results will be obtained at 10 different RNG number representing ten different situations. The confidence interval taking into account is 95% of the mean percent of the results at these ten RNGs.

The code in the main scripts used to set the RNG number is as demonstrated in Code 15 below.

```
global defaultRNG
$defaultRNG seed 10
```

Code 15 : Setting Random Number Generator Script

3.5 Expected Results

As the main aim of this project is to compare and analyse the system performance, three main criteria are important and need to be analysed. First, the throughput received and/or sent by the UE will be tested. Second, the average delay or time consumed by the travelling bits from the terminal to the gateway in the network will be evaluated. LTE/SAE model will be implemented with aid of NS2 simulator. This model will test the preceding three criteria with both QoS parameters enabled and disabled. A comparison will be made of the results from both test situations. These two criteria are expected to achieve better values when the QoS features are triggered in comparison to the values when the QoS features are not available. It is assumed that the best results will come from the high priority traffic classes.

3.6 Project Timelines

The major tasks and timelines for the project are shown in Figure 24 below

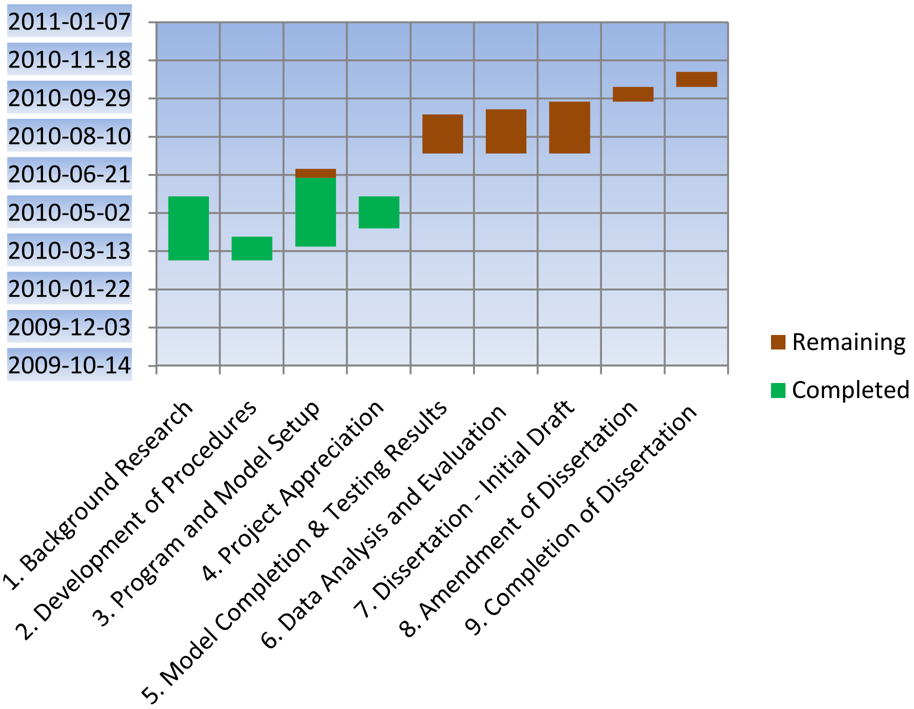

Figure 24 : Project Tasks Timelines (this figure made using MS office Excel)

3.7 Conclusion

In summary, this project aims to investigate the effect of QoS parameters on the Total Throughput and time Delay of the travelling data in the LTE/SAE system. The required improvements in the system criteria are necessary to cope with the higher communication speed, durability and efficiency of the LTE/SAE system. The Network Simulator 2 (NS2) is used to model the system. All results to be analysed to identify the advantages of the QoS features to the system's services. As a result, LTE philosophy is the key to the present and future of communication technology.

Chapter 4 : Test Scenarios and Simulation Results

4.1 Introduction

The Quality of Service scheduling mechanism used in this project is similar to the mechanism used in real Long Term Evolution Network. This mechanism has a prioritisation method and gives the highest priority to real-time conversational traffic. The main objective of this scheduling mechanism is to meet the quality of service requirements in LTE network. Lower delay of real-time packets is one of these objectives. The throughput is not ignored in the objectives and should be kept away from losing packets as possible. This project objective is to verify how good is the scheduling method in relation to the quality of service requirements has been mentioned above.

Before starting the tests, examining the bottleneck link (between eNB and aGW) is important. To find the maximum number of each class UEs to be served before any packet loss happening and this is can be mathematically calculated depending on the amount of traffic rate per UE and see how many UEs' traffic can be served by the link. An example of this is the amount of water can be delivered by main pipe once it is fed by multiple sub pipes. By knowing these threshold values, it is easy to test how much effect of other classes' traffic on the main class traffic that it almost uses the link capacity.

In this project's scenarios it would be assumed that the threshold value of traffic can be reached by using more than 5 user equipments for the class 0, class1 and class2 where it is more than 1 user equipments for the class3. Therefore, the default number of the user equipments of the class to be tested is 5 user equipments for the first three classes and one user equipment for the last one.

4.2 Conversational Class Test Scenarios and Results

This section includes scenarios and results of the conversational traffic test. The conversational traffic is high sensitive to delay and it is given the highest priority in this project traffic scheduling scheme. The main two performance features is tested and analyzed for this traffic class are the throughput and delay. These two performance indicators are tested in the presence of the traffic scheduling mechanism described in this project. The test has been made at different load amount carried by network's links. The loading traffics are from the different four classes, once at a time. A comparison is provided between the results once traffic scheduling is enabled as well as disabled. The comparison is made to proof benefits provided by the QoS scheduling mechanism to the conversational real-time traffic as well as to study the effect of other traffic's classes on class0 traffic.

4.2.1 q0, Scenario 1

This scenario objective is to test the performance of the conversational traffic (class 0) under and over load

- Once QoS parameters are OFF(no prioritisation are applied):
    - i. Sending conversational traffic only, class0 available and Classes 1,2 and 3 are not available
    - ii. Increasing the number of UEs having services with class 0 traffic between 1 and 19UEs
    - iii. Testing the class 0 Delay (the most important performance parameter in regards to the class0) and observe the limit of the link once delay increases, then testing Throughput.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
    - iv. Sending conversational traffic only, class0 available and Classes 1,2 and 3 are not available
    - v. Increasing the number of UEs, let's say between 1 and 19UEs
    - vi. Testing the Delay (the most important performance parameter in regards to the class0), then Throughput.

vii. Graph the results in comparison with the state of QoS parameters OFF(it is expected that no difference with the previous Test)

It is expected that the results will be same in both cases as we only have one type of traffic.

4.2.2 q0, Scenario 1 Results

The full results' tables are available in the Appendixes, and the graphed results only are shown here.

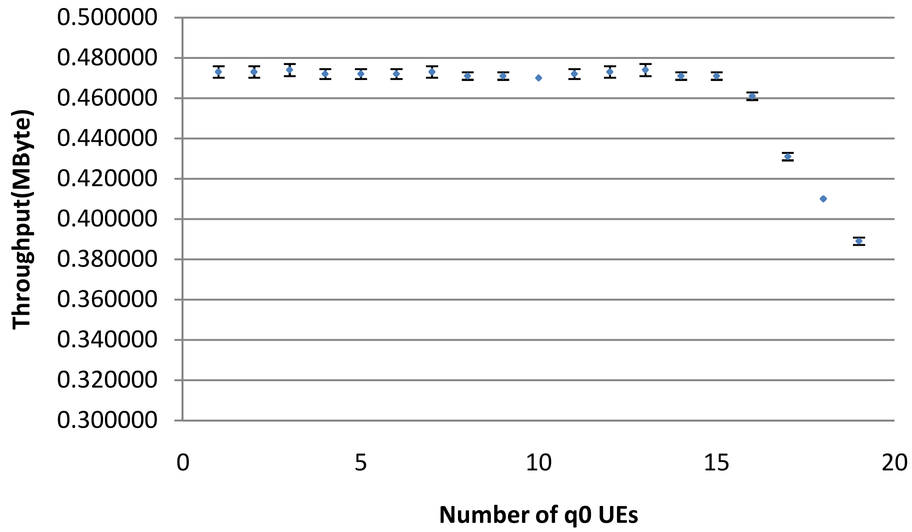

Figure 25 : q0 Received Throughput once q0 Traffic only Available

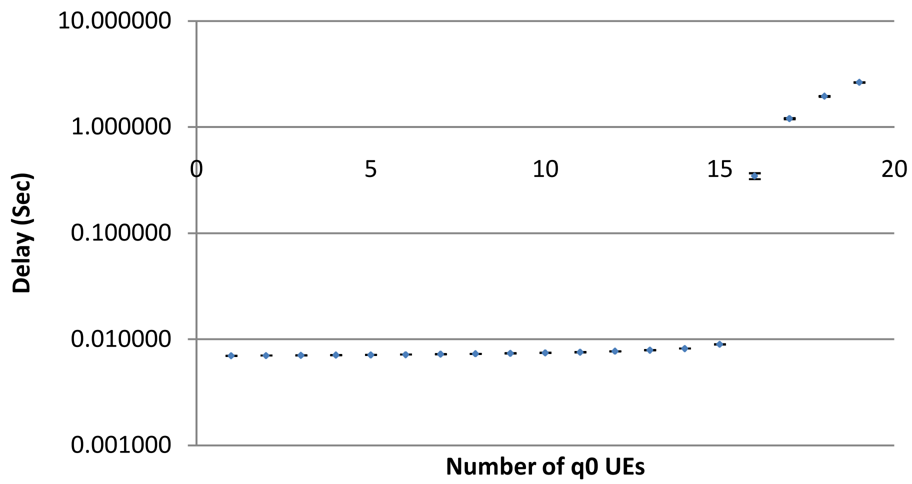

Figure 26 : q0 Delay once q0 Trafiic Only Available

It can be seen from the throughput's graph in Figure 25 that the results are reasonable. The received throughput is approximately constant. This means that the entire sent throughput is received with no drop or with low drop value. Once the link is overloaded, the dropped packets have increased gradually, which means that the received throughput has decreased once the User Equipments number has exceed certain value. The highest number of user equipments that the test shows without significant dropped packets value is 15 user equipments which is the maximum number of user equipments the link can tolerate. Once the number of user equipments exceeds the maximum number 15, the test shows decreasing in throughput of the first mobile that has used the real time traffic service. On the other hand, the delay is more important than the throughput value for the real-time traffic. It is shown in Figure 26 above, that the delay is increased after certain number of user equipments of q0 real time traffic. After this certain value of user equipments, as the number of user equipments entering the service increase, as the delay of the first user equipment is increased until it lose the service. It is standardised that the threshold value of delay that beyond this value the traffic is considered as unaccepted is 100ms as round trip time (*3GPP TS 23.107 Release 8, V8.0.0 (2008-12*). Therefore, the service is considered as unaccepted after the number of user equipments exceeds 15. The results in both situations once QoS scheduling mechanism is triggered and once it is not triggered are similar in this scenario.

4.2.3 q0, Scenario 2

The objective of this scenario is to test the performance of the conversational traffic (class 0) over load with class1

- Once QoS Parameters are OFF (no prioritisation are applied):
  - Sending conversational traffic (class0) of the default number of 5 UEs.
  - Increasing the number of UEs that they are having services with class1
  - Testing the effect of the class 1 on class 0 delay and throughput.
- Once QoS parameters are ON(Scheduling prioritisation are applied):
  - Sending conversational traffic (class0) of the default number of 5 UEs.
  - Increasing the number of UEs that they are having services with class1 between 1 and 15 user equipments.
  - Testing the effect of the class 1 on class 0 Delay and throughput.
  - Graph and compare them with the state of QoS parameters are off.

It is expected that the class 1 traffic has no effect on class 0 once strict scheduling prioritisation is applied. Significant effect is predicted for class 1 traffic on class 0 traffic once the link is overloaded, by increasing the number of dropped packets of class 0 traffic which means decreased the received throughput by the UEs, and increasing the delay.

4.2.4 q0, Scenario 2 Results

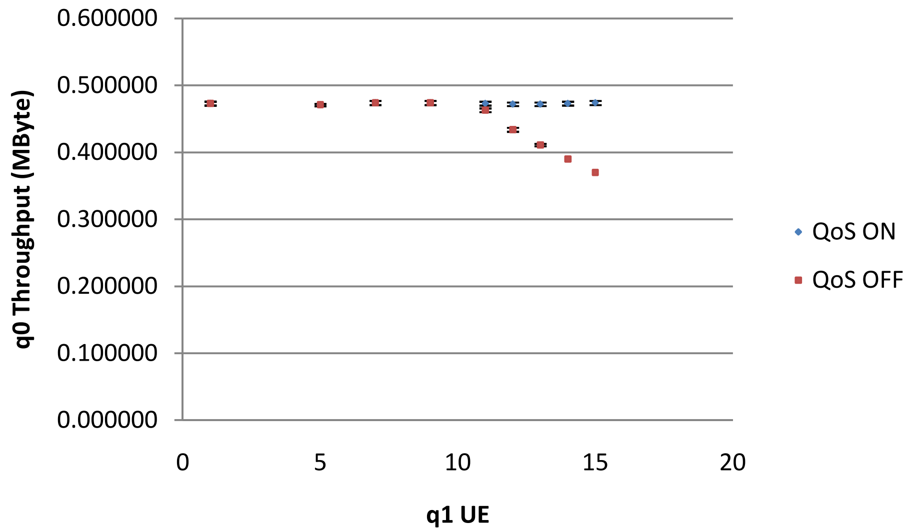

Figure 27 : q0 Received Throughput once q1 UEs has increased

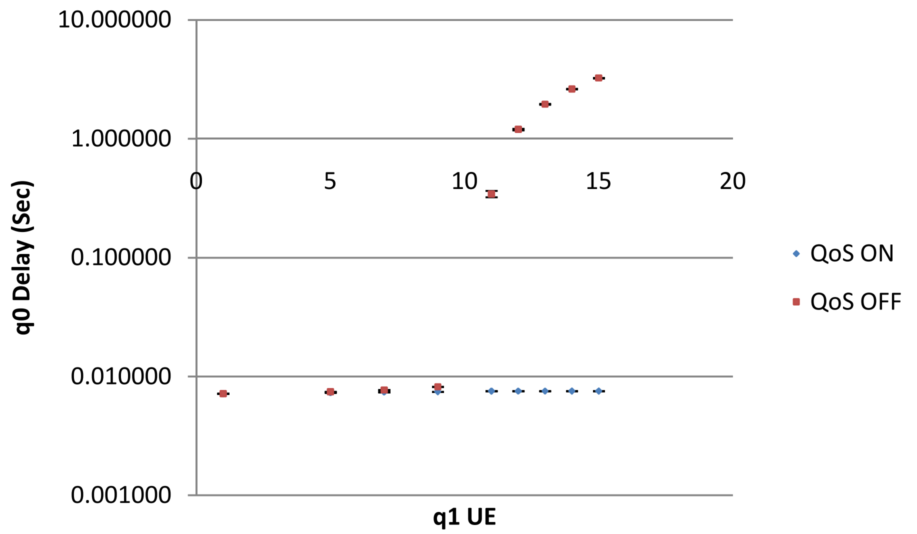

Figure 28 : q0 Delay once q1 UEs has increased

The results have been shown in this scenario's figures, Figure 27 and Figure 28 are reasonable. Once the QoS mechanism is triggered, the received throughput of the q0 user equipment is remain constant, while it decreases after certain number of user equipments using streaming traffic "11 user equipments" in the case of QoS mechanism is off. The delay of q0 user equipment has increased slowly for the QoS mechanism ON. On the other side, the delay has increased rapidly once q1 user equipments reach 11 and up.

The data range of 10 different samples in this test is still not high as shown in figures, which means that the results are more likely to be considered as accurate. The real-time traffic is considered as lost after the number of q1 user equipments is 11 and up.

4.2.5 q0, Scenario 3

Testing the performance of the conversational traffic (class 0) over load with class2

- Once QoS Parameters are OFF (no prioritisation are applied):
  - Sending conversational traffic (class0) of the default number of 5 User Equipments.
  - Increasing the number of UEs that they are having services with class2 between 1 and 50 user equipments.
  - Testing the effect of the class 2 on class 0 Delay and throughput.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending conversational traffic (class0) of the default number of 5 User Equipments.
  - Increasing the number of UEs that they are having services with class2 between 1 and 50 user equipments.
  - Testing the effect of the class 2 on class 0 delay and throughput.
  - Graph and compare them with the state of QoS parameters are off.

4.2.6 q0, Scenario 3 Results

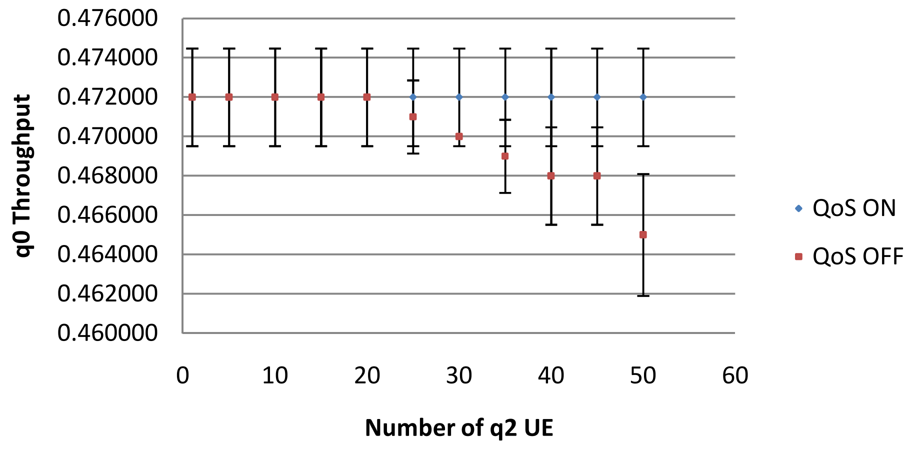

Figure 29 : q0 Received Throughput while q2 UE Increasing

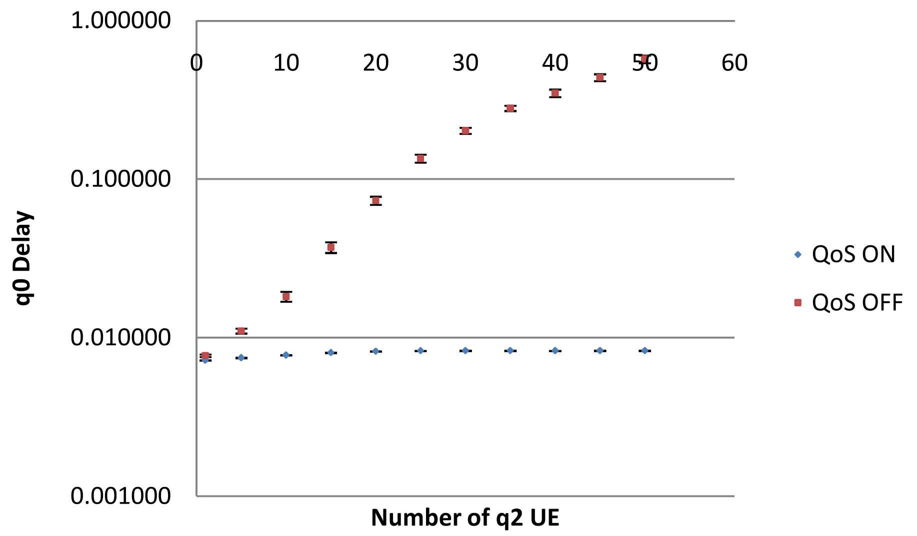

Figure 30 : q0 Delay while q2 UEs Increasing

For the throughput results as it is shown in Figure 29, it can be said that the results are good, but the range of data is not low. By looking at the general behaviour of the results it can be said that the behaviour is reasonable. Below 25 user equipments of q1 traffic, the throughput is identical for both states once QoS mechanisms is triggered and not triggered. Above 25 user equipments, different behaviour of the two situations' trends is noticed. While QoS mechanism is available, the throughput is generally constant. On the other hand, while QoS mechanism is disabled, the throughput has decreased once the number of q1 user equipments is reached. The throughput has continued decreasing with greater values after 25 user equipments of q1 traffic.

Once we have only 1 user equipment using q1 traffic, the delay values in both situations are identical. After that, as the number of q1 user equipment increases, the delay increases. In the case of QoS ON, the delay increases with very small values. In the case of QoS OFF, the delay increases with considerable values as shown in Figure 30 above. For delay figure the data range is not large, which leads to conclude that the delay results can be assumed as accurate.

4.2.7 q0, Scenario 4

The aim of this scenario is to test the performance of the conversational traffic (class 0) over load with class3 traffic.

- Once QoS Parameters are OFF (no prioritisation are applied):
    - Sending conversational traffic (class0) of the default number of 5 UEs.
    - Increasing the number of UEs that they are having services with class3 between 1 and 10 user equipments.
    - Testing the effect of the class 3 on class 0 delay and throughput.

- Once QoS parameters are ON(Scheduling prioritisation are applied):
    - Sending conversational traffic (class0) of the default number of 5 UEs.
    - Increasing the number of UEs that they are having services with class1 between 1 and 10 user equipments.
    - Testing the effect of the class 3 on class 0 delay and throughput.
    - Graph and compare them with the state of QoS parameters are off.

4.2.8 q0, Scenario 4 Results

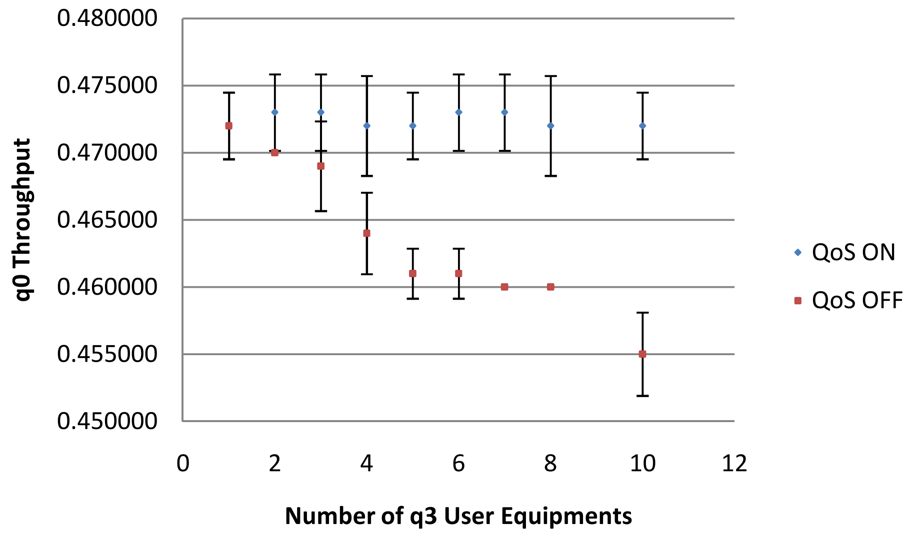

Figure 31 : q0 Received Throughput while q3 UEs increasing

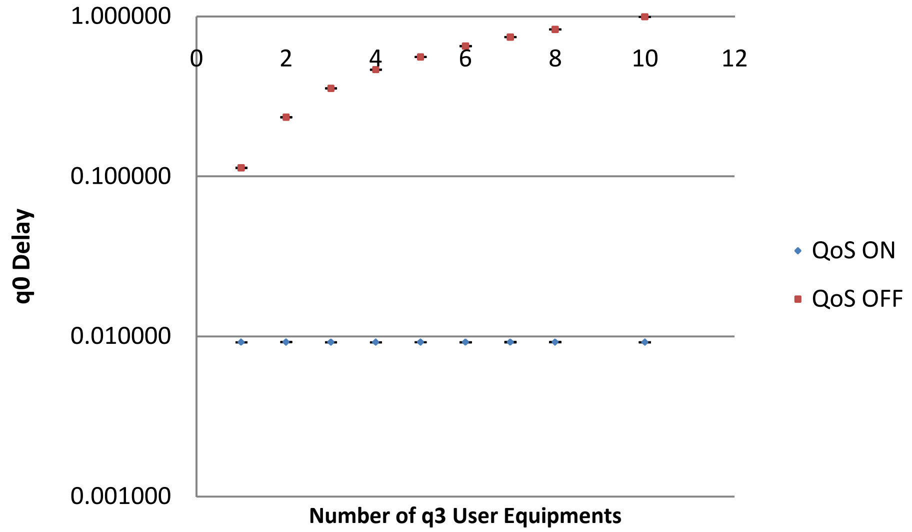

Figure 32 : q0 Delay while q3 UEs Increasing

In regards to results have been shown in Figure 31, If only one q3's user equipment is existed, the throughput in both situations is approximately identical. If q3 user equipments increases, the q0 received throughput is much better in the case of QoS mechanism is available than it is not available. It can be seen that the data range is not low due to different number of RNG use.

For the delay test as it is shown in Figure 32, it is clear from the result, that the effect of the first q3 user equipment on delay in the QoS OFF case is much higher than the QoS ON. The delay becomes greater as the number of the q3 user equipments increases.

4.3 Streaming Traffic Test Scenarios and Results

This section includes scenarios and results of the streaming traffic test. The streaming traffic is high sensitive to delay and it is given the second highest priority in this project traffic scheduling scheme. The main two performance features is tested and analyzed for this traffic class are the throughput and delay. These two performance indicators are tested in the presence of the traffic scheduling mechanism described in this project. The test has been made at different load amount carried by network's links. The loading traffics are from the different four classes, once at a time. A comparison is provided between the results once traffic scheduling is enabled as well as disabled. The comparison is made to proof benefits provided by the QoS scheduling mechanism to the streaming real-time traffic as well as to study the effect of other traffic's classes on class1 traffic.

4.3.1 q1, Scenario 1

This scenario objective is to test the performance of the streaming traffic (class 1) under and over load

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending streaming traffic only, class1 available and Classes 0,2 and 3 are not available
  - Increasing the number of UEs having services with class 1 traffic between 1 and 20 UEs
  - Testing the class 1 delay (the most important performance parameter in regards to the class1) and observe the limit of the link once delay increases, then testing Throughput.

- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending streaming traffic only, class1 available and Classes 0,2 and 3 are not available
  - Increasing the number of UEs between 1 and 20UEs
  - Testing the delay (the most important performance parameter in regards to the class1), then Throughput.
  - Graph the results in comparison with the state of QoS parameters OFF(it is expected that no difference with the previous Test)

It is expected that the results will be same in both cases as we only have one type of traffic.

4.3.2 q1, Scenario 1 Results

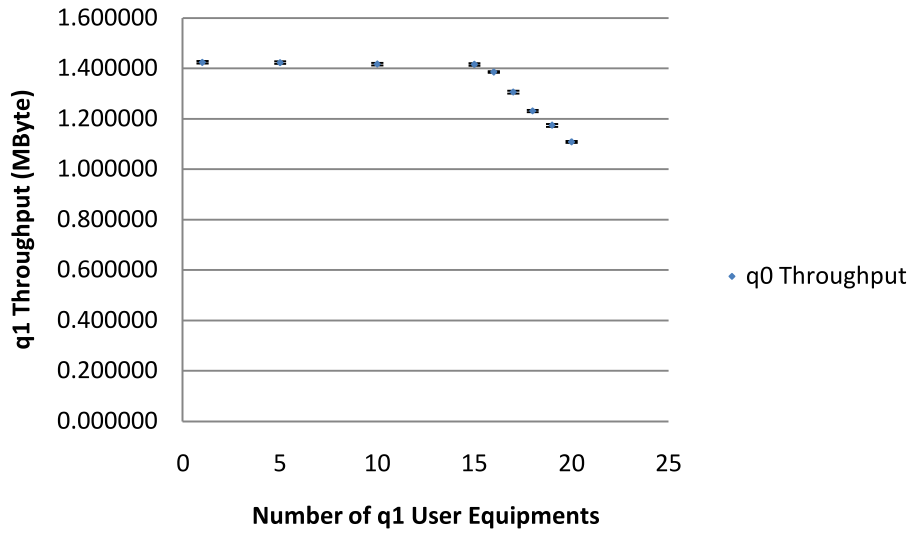

Figure 33 : q1 Received Throughput While Only Streaming Traffic is Available

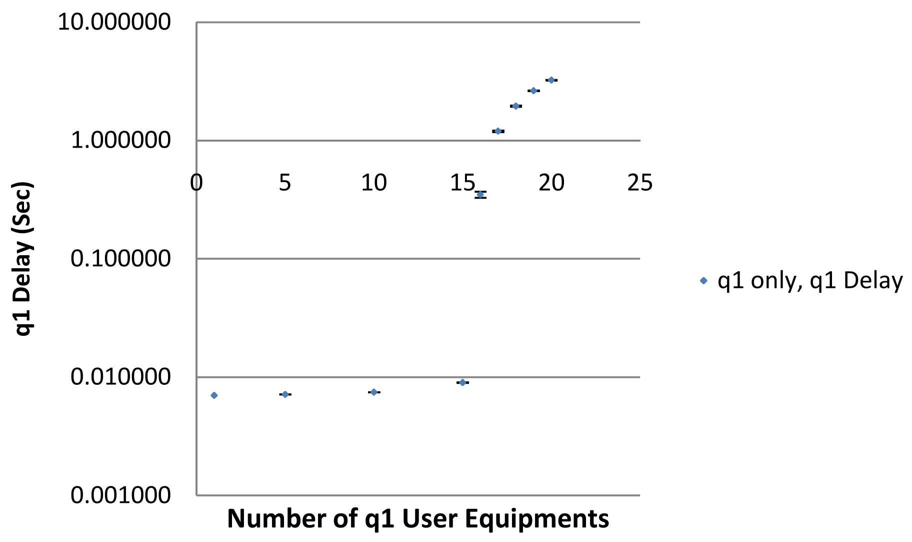

Figure 34 : q1 Delay While Only Streaming Traffic is Available

The received throughput of the first user equipment using streaming traffic remains constant until the number of user equipments reaches 15 UEs. Once the number of user equipments is 15, the dropped packet increases. As the number of user equipments exceeds 15, more traffic packets drops down. It is clearly shown in the throughput's figure, Figure 33 above that the q1 throughput decreases as the number of user equipments increases beyond 15 UEs.

It can be explained from Figure 34, the packets' delay of the first user of q1 traffic, increases gradually with low amount of time until the number of user equipments reaches 16, and then the delay is rapidly increases. Beyond 16 user equipments, the delay continues increasing to exceed 1 second. The range of data in both figures' results is small.

4.3.3 q1, Scenario 2

This scenario's objective is to test the performance of the streaming traffic (class 1) over loaded with class0

- Once QoS Parameters are OFF (no prioritisation are applied):
    - Sending streaming traffic (class1) of the default number of 5 UEs.
    - Increasing the number of UEs that they are having services with class0 (conversational traffic) between 1 and 15.
    - Testing the effect of the class 0 on class 1 delay and throughput.
- Once QoS parameters are ON(Scheduling prioritisation are applied):
    - Sending streaming traffic (class1) of the default number of 5 UEs.
    - Increasing the number of UEs that they are having services with class0 between 1 and 15 user equipments.
    - Testing the effect of the class 0 on class 1 delay and throughput.
    - Graph and compare them with the state of QoS parameters are off.

4.3.4 q1, Scenario 2 Results

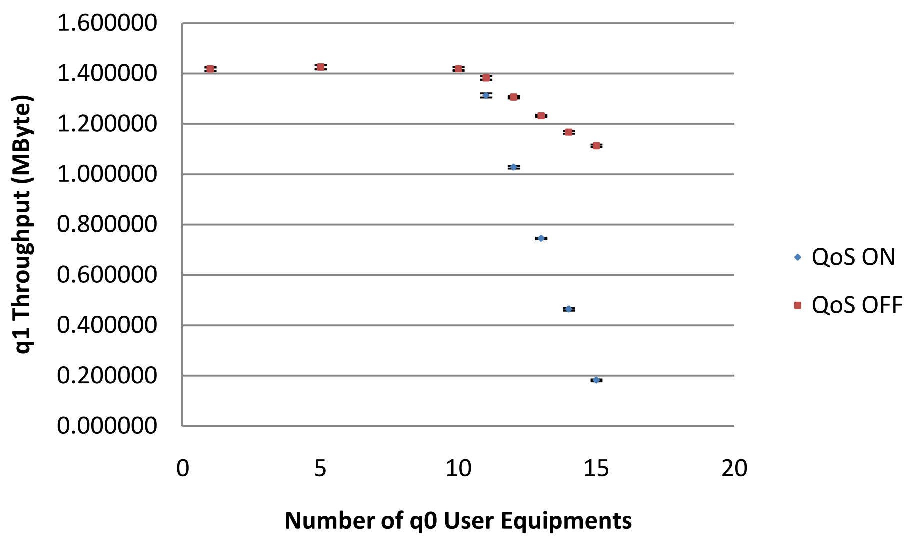

Figure 35 : q1 Received Throughput while q0 UEs increases

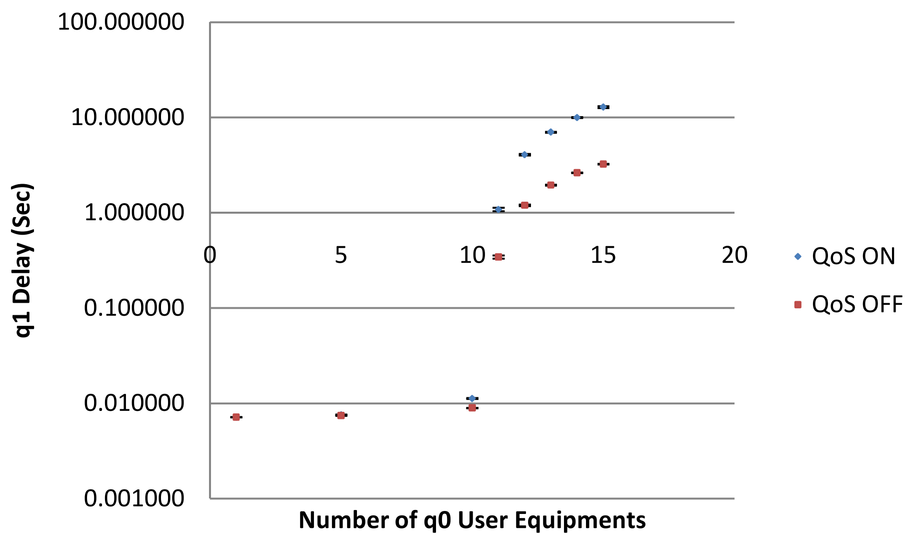

Figure 36 : q1 Delay While q0 UEs increasing

As it is shown in the q1 throughput figure, Figure 35 above, the throughput results are identical for both situations while QoS is enabled and disabled from the q0 user equipments increases from 1 until 10 UEs. At the time q0 user equipments number exceeds 10, the throughput decreases in both situations. More convergence between the two situations' results is clearly shown by the graph that the throughput of q1 user equipment decreases in the case of QoS ON much more than it decreases in the case of QoS OFF. This is what was expected before the test, that the higher priority traffic will always has the advantage in the case that the scheduling mechanism is available.

The q1 delay results' graph in Figure 36 shows that the results are identical for both cases if the number of the q0 user equipments is below 10. Once it is 10 q0 user equipments both results increase in both cases but in the case of QoS ON results shows more q1 delay than the QoS OFF results. At 11 q0 user equipments, there is a rapid increase in both cases' results and the QoS ON results still showing more delay of q1 UE. As the number of q0 user equipments continues increasing, the delay increases more and more and the results still showing better results for q1 user equipment while the QoS mechanism is disabled.

4.3.5 q1, Scenario 3

This scenario objective is to test the performance of the streaming traffic (class 1) over load with class2 (interactive traffic)

- Once QoS Parameters are OFF (no prioritisation are applied):
  - Sending streaming traffic (class1) of the default number of 5 UEs.
  - Increasing the number of UEs that they are having services with class2 traffic between 1 and 100 UEs.
  - Testing the effect of the class 2 on class 1 delay and throughput.

- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending streaming traffic (class1) of the default number of 5 UEs.
  - Increasing the number of UEs that they are having services with class2 between 1 and 100 UEs.
  - Testing the effect of the class 2 on class 1 delay and throughput.
  - Graph and compare them with the state of QoS parameters are off.

4.3.6 q1, Scenario 3 Results

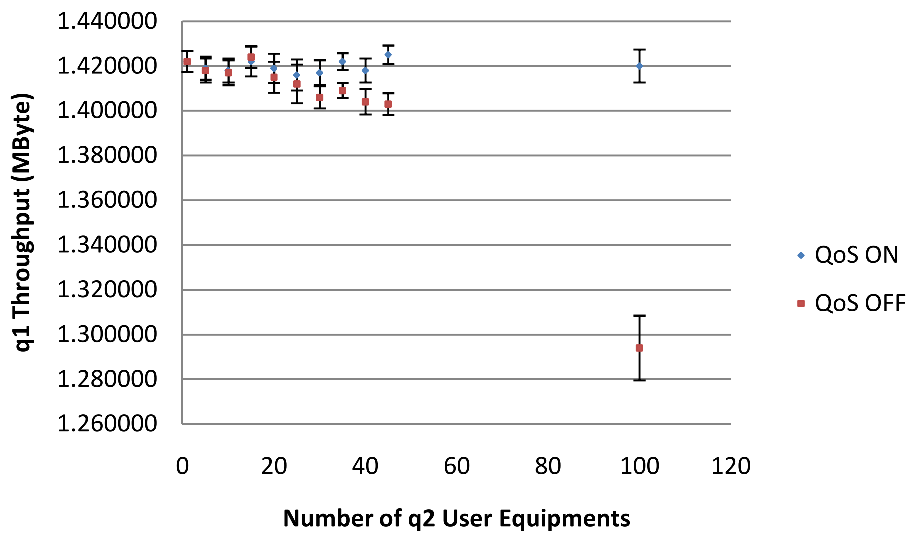

Figure 37 : q1 Received Throughput while q2 UEs Increasing

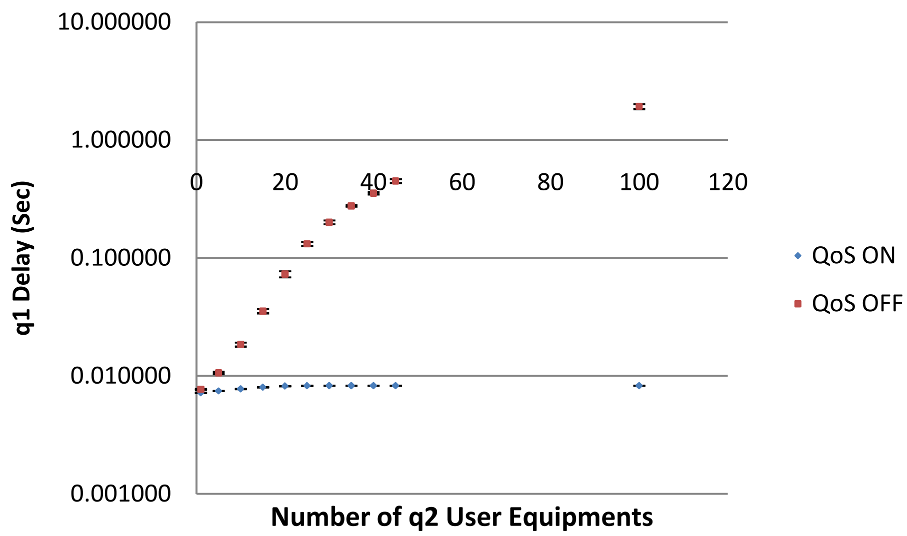

Figure 38 : q1 Delay While q2 UEs Increasing

In reference to Figure 37, the throughput of the user equipment that is first uses the streaming traffic is depicted in the q1 throughput while q2 UEs increasing figure above. While the number of q2 user equipments is between 1 and 20 UEs, the received throughput in both situations is identical. Beyond 20 user equipments of q2 traffic, the difference in throughput becomes clearer. The QoS ON case results shows more received throughput by the q1 user equipment. The difference between the two throughput values increases as the number of q2 user equipments increases. The last test sample is presented here in this test to demonstrate how large is the difference between the two values of throughput in the two cases at high number of q2 user equipments. It is clearly shown that the QoS ON mechanism is much better for the q1 user equipment in regards to throughput.

The delay results in Figure 38 show that the delay of the q1 user equipment in the case of QoS OFF is higher than it is in QoS ON. As the number of the q2 user equipment increases the q1 user equipment traffic delay in QoS OFF state becomes greater. In comparison with the state of QoS ON the delay slightly increases.

4.3.7 q1, Scenario 4

This scenario objective is to test the performance of the streaming traffic (class 1) in the existence of class3 traffic (background traffic)

- Once QoS Parameters are OFF (no prioritisation are applied):
  - Sending streaming traffic (class1) of the default number of 5 UEs.
  - Increasing the number of UEs that they are having services with class3 traffic between 1 and 10 UEs.
  - Testing the effect of the class 3 on class 1 delay and throughput.

- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending streaming traffic (class1) of the default number of 5 UEs.
  - Increasing the number of UEs that they are having services with class3 between 1 and 10 UEs.
  - Testing the effect of the class 3 on class 1 delay and throughput.
  - Graph and compare them with the state of QoS parameters are off.

4.3.8 q1, Scenario 4 Results

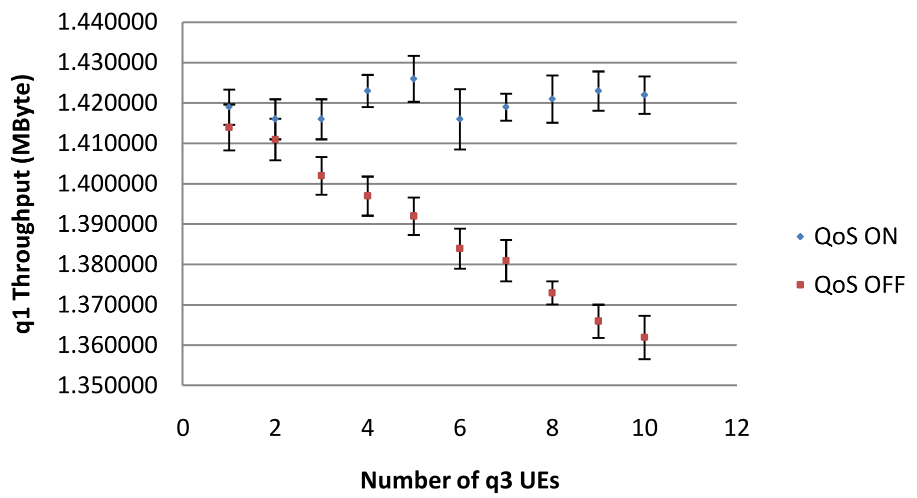

Figure 39 : q1 Received Throughput while q3 UEs Increases

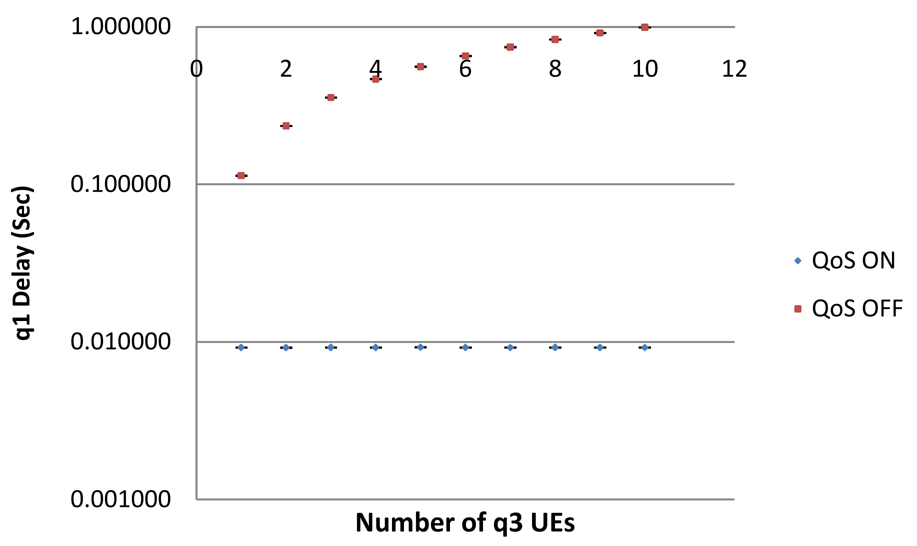

Figure 40 : q1 Delay while q3 UEs Increases

It is clearly shown in Figure 39, q1 throughput while q3 UEs Increases figure, that the results of the received throughput in QoS ON mechanism are within the range of 1.41–1.43Mbyte. While it decreases from 1.41MByte at 1 user equipment of class3 traffic until it reaches an average value of 1.36MByte in the presence of 10 user equipments of class3 traffic. The minor fluctuation of the q1 received throughput is possibly due to Random Number Generated in the software.

It can be seen from Figure 40 delay results shows that in the case of the QoS mechanism available, the q1 user equipment traffic time delay increases from 7ms in the presence of 1 user equipment of q3 traffic until it reaches around 9ms at 10 q3's user equipment. In the case of the QoS mechanism disabled, the delay increases from around 100ms in the presence of 1 UEs of class3 traffic until it reaches 1 second for 10 UEs of q3's traffic.

4.4 Interactive Traffic Test Scenarios and Results

This section includes scenarios and results of the Interactive Best Effort traffic test. The interactive traffic is high sensitive to throughput and it is given the third highest priority in this project traffic scheduling scheme. The main two performance features is tested and analyzed for this traffic class are the throughput and delay, where the throughput results is the most important to this type of traffic. The two performance indicators are tested in the presence of the traffic scheduling mechanism described in this project. The test has been made at different load amount carried by network's links. The loading traffics are from the different four classes, once at a time. A comparison is provided between the results once traffic scheduling is enabled as well as disabled. The comparison is made to proof benefits and/or limitations provided by the QoS scheduling mechanism to the interactive traffic as well as to study the effect of other traffic's classes on class2 traffic.

4.4.1 q2, Scenario 1

This scenario objective is to test the performance of the interactive traffic (class 2) under and over load

- Once QoS parameters are OFF(no prioritisation are applied):
    - Sending interactive traffic only, class2 available and Classes 0,1 and 3 are not available
    - Increasing the number of UEs having services with class 2 traffic between 1 and 90 UEs
    - Testing the class 2 throughput (the most important performance parameter in regards to the class2).
- Once QoS parameters are ON (Scheduling prioritisation are applied):
    - Sending interactive traffic only, class2 available and Classes 0,1 and 3 are not available
    - Increasing the number of UEs between 1 and 90UEs
    - Testing the throughput (the most important performance parameter in regards to the class2).

- Graph the results in comparison with the state of QoS parameters OFF(it is expected that no difference with the previous Test)

It is expected that the results will be same in both cases as we only have one type of traffic

4.4.2 q2, Scenario 1 Results

It can be seen from Figure 41 below, the q2 throughput with only class2 traffic available figure that the received throughput by the first user equipment has the service hasn't changed much once the number of mobile phones in the service between 1 and 20 user equipments. Once the number of user equipments increases to be more than 20, there are more dropped packets introduced and the throughput decreases gradually until reaching very low value if there is 90 user equipments in service.

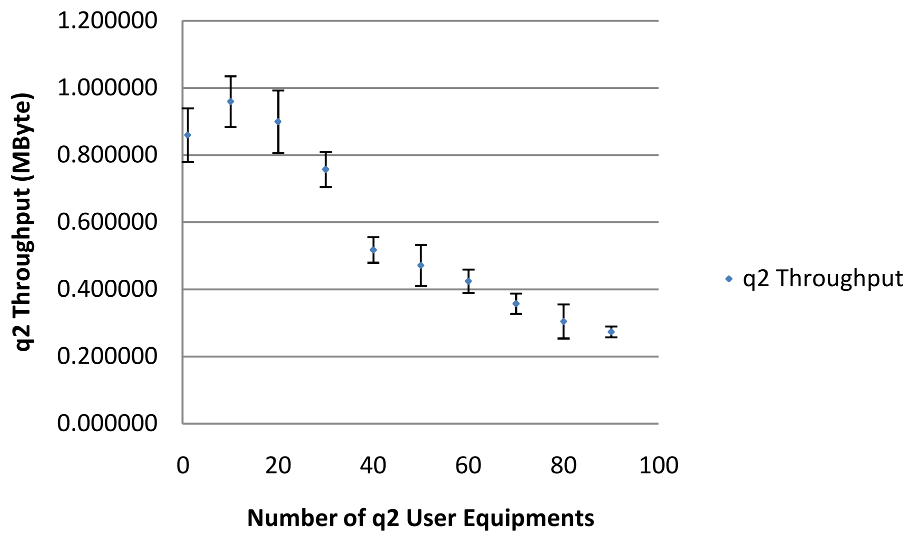

Figure 41 : q2 Received Throughput while Only Class2 Available

4.4.3 q2, Scenario 2

This scenario objective is to test the performance of the interactive traffic (class 2) under and over load in the presence of the q0 real-time traffic.

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending interactive traffic with a default number of 5 user equipments.
  - Increasing the number of UEs having services with class 0 traffic between 1 and 15 UEs, while Classes 1 and 3 are not available
  - Testing the class 2 throughput (the most important performance parameter in regards to the class2) and then the delay.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending interactive traffic with a default number of 5 user equipments.
  - Increasing the number of UEs having services with class 0 traffic between 1 and 15 UEs, while Classes 1 and 3 are not available
  - Testing the class 2 throughput (the most important performance parameter in regards to the class2) and then the delay.
  - Graph the results in comparison with the state of QoS parameters OFF

4.4.4 q2, Scenario 2 Results

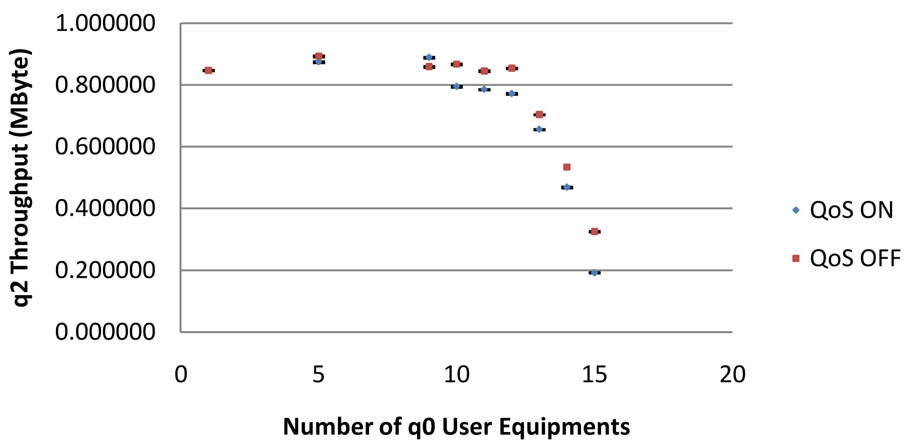

Figure 42 : q2 Received Throughput While q0 UEs Increases

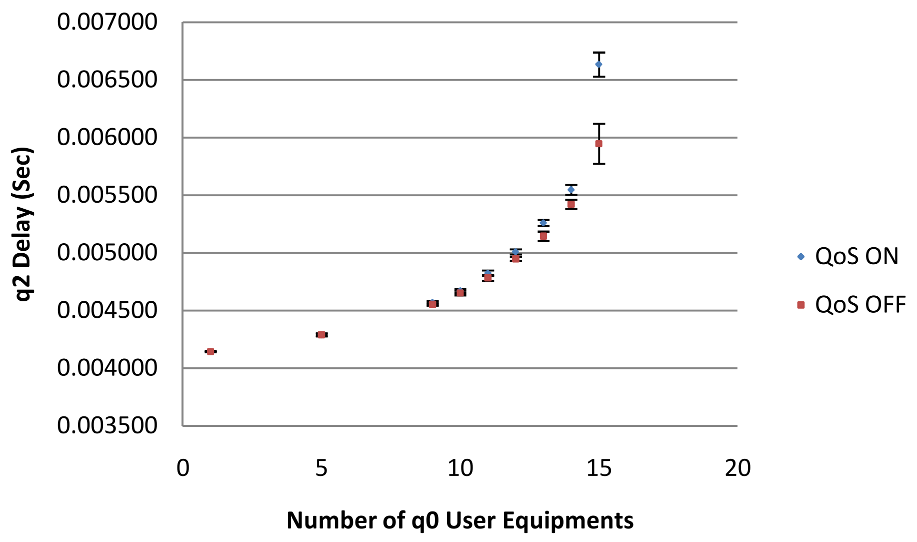

Figure 43 : q2 Delay while q0 UEs Increases

In Figure 42, the values of the received throughput of the first user equipment has a service with interactive class traffic are identical when the number of mobile phone equipments having a q0 real-time service is less than 5 UEs. When the number of q0 user equipments is 5 or more, the received throughput in the presence of QoS mechanism is less than the received throughput in the absence of QoS mechanism. This is clearly true as the high priority in the QoS mechanism is given to real-time traffic.

The delay figure, Figure 43 shows that the q2 user equipment's traffic delay values are identical in the presence of 10 user equipments of q0 traffic or less. Once the q0 UEs number is higher than 10 UEs, the delay of q2 user equipment traffic is higher with QoS mechanism is triggered than it is with QoS mechanism is off. And these results are as expected due to the packet scheduling priorities.

4.4.5 q2, Scenario 3

This scenario objective is to test the performance of the interactive traffic (class 2) under and over load in the presence of the q1 traffic.

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending interactive traffic with a default number of 5 user equipments.
  - Increasing the number of UEs having services with class 1 traffic between 1 and 15 UEs, while Classes 0 and 3 are not available
  - Testing the class 2 throughput (the most important performance parameter in regards to the class2) and then the delay.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending interactive traffic with a default number of 5 user equipments.
  - Increasing the number of UEs having services with class 1 traffic between 1 and 15 UEs, while Classes 0 and 3 are not available
  - Testing the class 2 throughput (the most important performance parameter in regards to the class2) and then the delay.
  - Graph the results in comparison with the state of QoS parameters OFF

4.4.6 q2, Scenario 3 Results

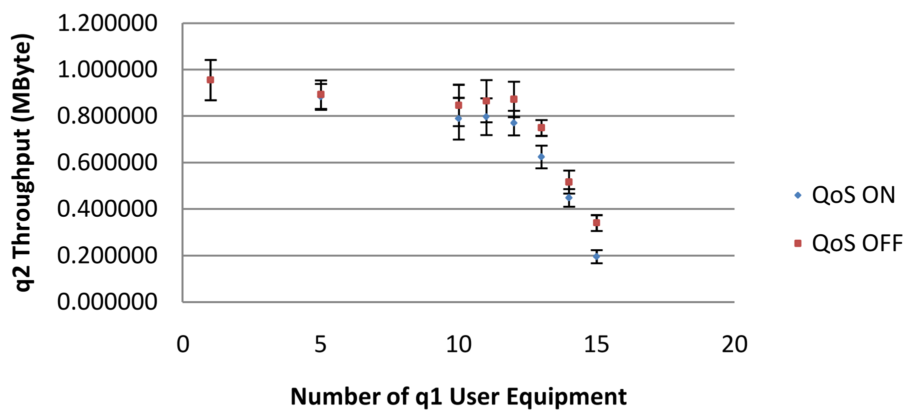

Figure 44 : q2 Received Throughput while q1 User Equipment Increases

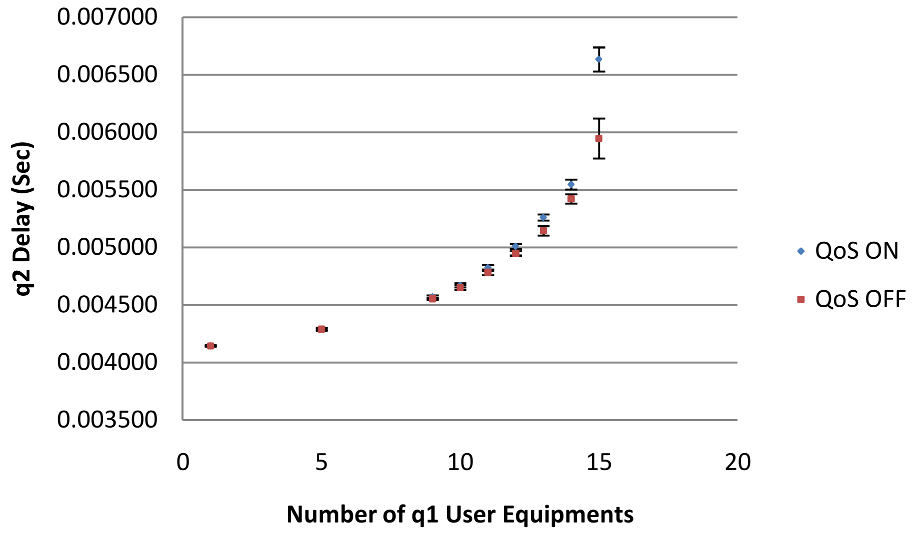

Figure 45 : q2 Delay while q1 User Equipment Increases

As the previous test once q0 increasing, similar results are obtained in this test. The values of the received throughput of the first user equipment has a service with interactive class traffic are identical when the number of mobile phone equipments having a q1 real-time service is less than 5 UEs. It can be seen in Figure 44, when the number of q1 user equipments is 5 or more, the received throughput in the presence of QoS mechanism is less than the received throughput in the absence of QoS mechanism. This is clearly true as the high priority in the QoS mechanism is given to real-time traffic.

The delay figure, Figure 45 shows that the q2 user equipment's traffic delay values are identical in the presence of 10 user equipments of q1 traffic or less. Once the q1 UEs number is higher than 10 UEs, the delay of q2 user equipment traffic is higher with QoS mechanism is triggered than it is with QoS mechanism is off. And these results are as expected due to the packet scheduling priorities.

4.4.7 q2, Scenario 4

This scenario objective is to test the performance of the interactive traffic (class 2) under and over load in the presence of the q3 traffic.

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending interactive traffic with a default number of 5 user equipments.
  - Increasing the number of UEs having services with class 3 traffic between 1 and 10 UEs, while Classes 0 and 1 are not available
  - Testing the class 2 throughput (the most important performance parameter in regards to the class2) and then the delay.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending interactive traffic with a default number of 5 user equipments.
  - Increasing the number of UEs having services with class 3 traffic between 1 and 10 UEs, while Classes 0 and 1 are not available
  - Testing the class 2 throughput (the most important performance parameter in regards to the class2) and then the delay.
  - Graph the results in comparison with the state of QoS parameters OFF

4.4.8 q2, Scenario 4 Results

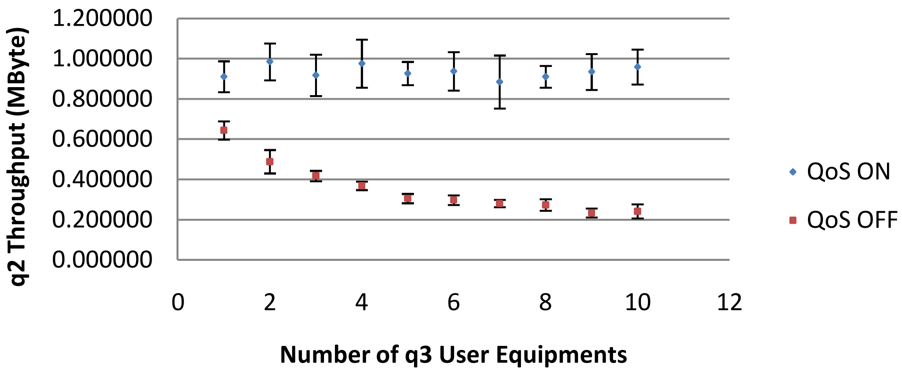

Figure 46 : q2 Received Throughput while q3 User Equipment Increases

It is clearly shown in Figure 46 that if only one q3's user equipment is existed, the throughput in of q2 user equipment in the presence of QoS mechanism is higher than the throughput in the absence of the mechanism. If q3 user equipments increases, the q1 received throughput is much better in the case of QoS mechanism is available than it is not available. The data range in the QoS ON state is not low due using different RNG's number.

4.5 Background (Best Effort) Traffic Test Scenarios and Results

This section includes scenarios and results of the background best effort traffic test. The background traffic is high sensitive to throughput and it is given the least priority in this project traffic scheduling scheme. The main two performance features is tested and analyzed for this traffic class are the throughput and delay, where the throughput results is more important to this type of traffic. The two performance indicators are tested in the presence of the traffic scheduling mechanism described in this project. The test has been made at different load amount carried by network's links. The loading traffics are from the different four classes, once at a time. A comparison is provided between the results once traffic scheduling is enabled as well as disabled. The comparison is made to proof benefits and/or limitations provided by the QoS scheduling mechanism to the background traffic as well as to study the effect of other traffic's classes on class3 traffic.

4.5.1 q3, Scenario 1

This scenario objective is to test the performance of the background (best effort) traffic (class 3) under and over load

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending background traffic only, class3 available and Classes 0,1 and 2 are not available
  - Increasing the number of UEs having services with class 3 traffic between 1 and 19 UEs
  - Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.

- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending background traffic only, class3 available and Classes 0,1 and 2 are not available
  - Increasing the number of UEs between 1 and 19UEs
  - Testing the throughput (the most important performance parameter in regards to the class3) and delay.
  - Graph the results in comparison with the state of QoS parameters OFF(it is expected that no difference with the previous Test)

It is expected that the results will be same in both cases as we only have one type of traffic

4.5.2 q3, Scenario 1 Results

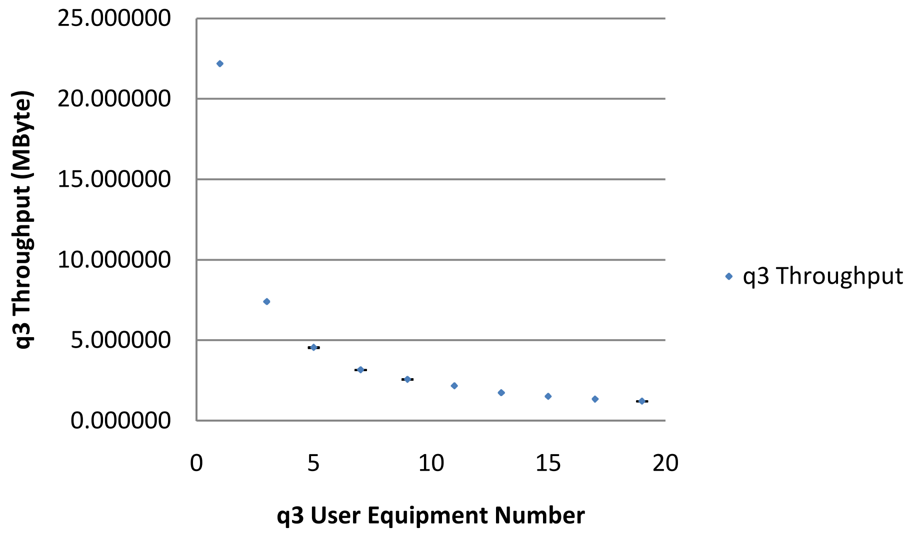

Figure 47 : q3 Received Throughput, Only q3 Traffic Available

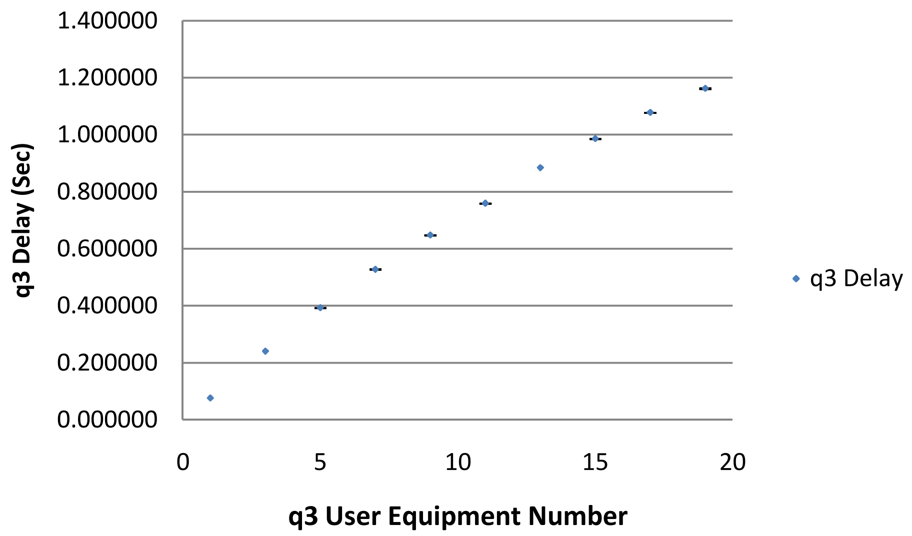

Figure 48 : q3 Delay, Only q3 Traffic Only

The received throughput by the first mobile phone equipments once only one q3 user equipment available is the full throughput available as best effort traffic which is approximately 22MByte. Once the number of q3 user equipment increases, the throughput is divided equally between the mobile phones stations available. The q3 received throughput, only q3 traffic only available figure, Figure 47 shows an exponential decay trend, which means that the results are reasonable and as expected.

As the number of q3 user equipments increases, the throughput is distributed with a factor of [1/(number of user equipments using q3 traffic)].

On the other hand, the delay of the first user equipment having the service increases as the number of user equipments sharing the throughput with it, increases.

As this type of traffic is not highly affected by the delay as soon as the service is still available, therefore the delay won't considered as big problem to certain extent.

The delay values started with approximately 7msec in a case of only one user equipment is available until it reaches 1.1sec with 19 user equipments sharing the throughput with the first mobile phone.

The results in both cases, QoS ON and QoS OFF, are identical due to that the QoS mechanism is helpful once more than one traffic type is sent via bottleneck link. This depicted in Figure 48.

4.5.3 q3, Scenario 2

This scenario's objective is to test the performance of the background traffic (class 3) under and over load in the presence of the q0 real-time traffic.

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending background traffic with a default number of 1 user equipment.
  - Increasing the number of UEs having services with class 0 traffic between 1 and 15 UEs, while Classes 1 and 2 are not available
  - Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending background traffic with a default number of 1 user equipment.
  - Increasing the number of UEs having services with class0 traffic between 1 and 15 UEs, while Classes 1 and 2 are not available.

- Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.
- Graph the results in comparison with the state of QoS parameters OFF

4.5.4 q3, Scenario 2 Results

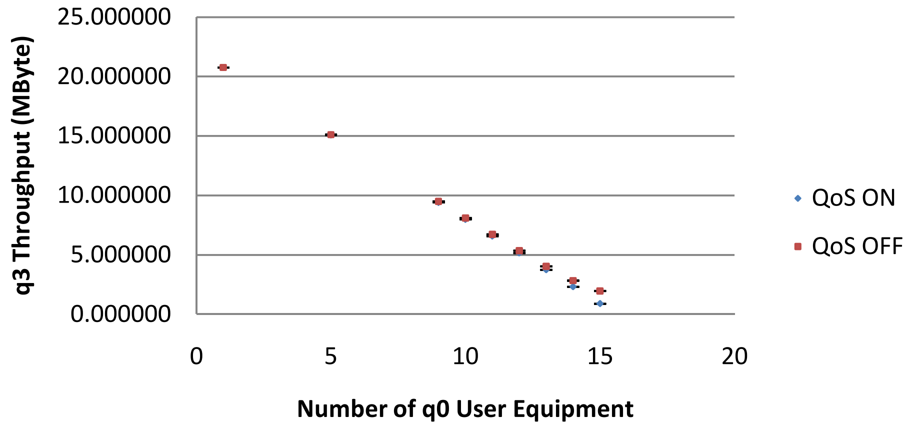

Figure 49 : q3 Received Throughput, while q0 User Equipments Increases

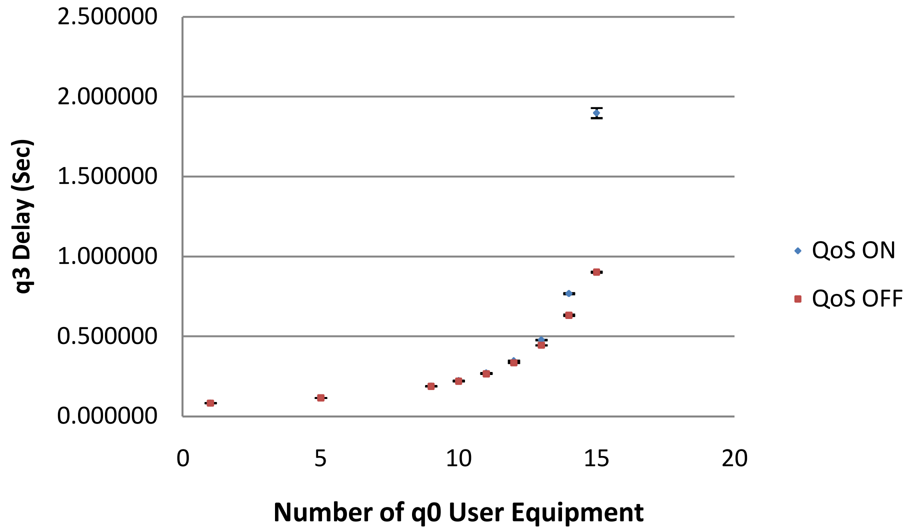

Figure 50 : q3 Delay, while q0 User Equipment Increases

It can be seen from Figure 49, the resultant values of throughput that is received by the first mobile equipment having a q3 traffic service decrease and are all identical in both QoS situations, when the number of q0 user equipment is less or equal to 10 user equipments. Once the number of q0 user equipment increases beyond 10 user equipments, the throughput in both situations continues decreasing, but it decreases more in the case of QoS ON. The divergence between the values of throughput in both cases becomes obvious with the existence of 15 user equipments using real-time traffic.

Figure 50 shows that the delay increases as the number of q0 UEs increases with identical values up to 10 q0's UEs. Beyond this, as the number of q0's UEs increases, the delay increases more in the case of QoS ON rather than in the QoS OFF. The delay becomes double in the existence of 15 user equipments using real-time traffic once QoS mechanism is applied. The results demonstrate the expectations, where the method used in this project gives higher priority to real-time traffic.

4.5.5 q3, Scenario 3

This scenario's objective is to test the performance of the background traffic (class 3) under and over load in the presence of the q1 traffic.

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending background traffic with a default number of 1 user equipment.
  - Increasing the number of UEs having services with class 1 traffic between 1 and 15 UEs, while Classes 0 and 2 are not available
  - Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending background traffic with a default number of 1 user equipment.
  - Increasing the number of UEs having services with class1 traffic between 1 and 15 UEs, while Classes 0 and 2 are not available.
  - Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.
  - Graph the results in comparison with the state of QoS parameters OFF

4.5.6 q3, Scenario 3 Results

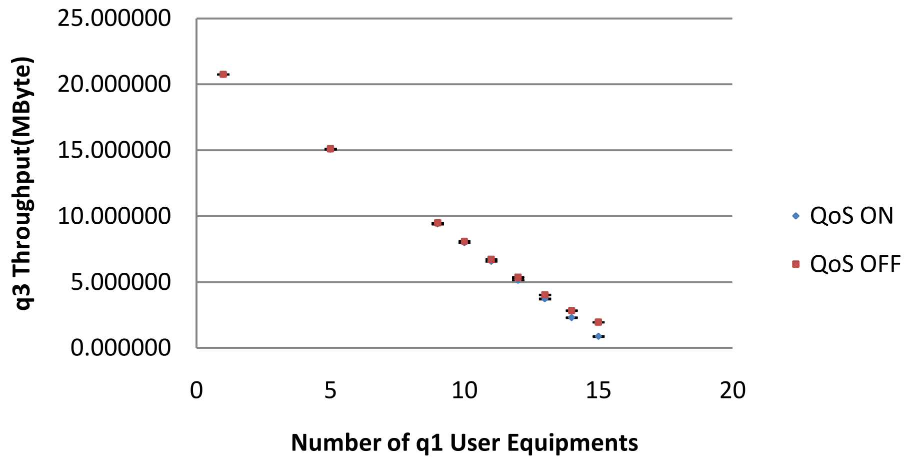

Figure 51 : q3 Received Throughput while q1 UEs Increases

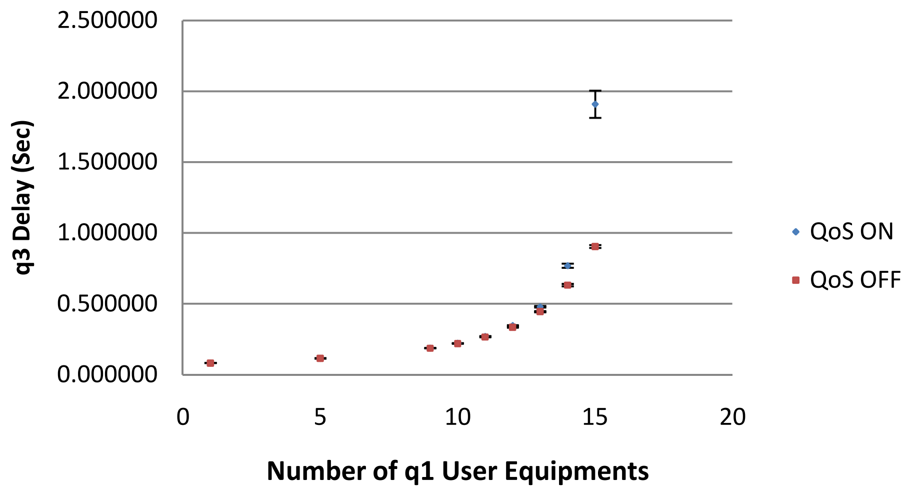

Figure 52 : q3 Delay while q1 UEs Increases

Figure 51 shows that the resultant values of throughput that is received by the first mobile equipment having a q3 traffic service decrease and are all identical in both QoS situations, when the number of q1 user equipment is less or equal to 10 user equipments. Once the number of q1 user equipment increases beyond 10 user equipments, the throughput in both situations continues decreasing, but it decreases more in the case of QoS ON. The divergence between the values of throughput in both cases becomes obvious with the existence of 15 user equipments using streaming traffic.

Figure 52 shows that the delay increases as the number of q1 UEs increases with identical values up to 10 q1's UEs. Beyond this, as the number of q1's UEs increases, the delay increases more in the case of QoS ON rather than in the QoS OFF. The delay becomes double in the existence of 15 user equipments using streaming traffic once QoS mechanism is applied. The results demonstrate the expectations, where the method used in this project gives higher priority for q1traffic than q3 traffic.

4.5.7 q3, Scenario 4

This scenario's objective is to test the performance of the background traffic (class 3) under and over load in the presence of the interactive traffic.

- Once QoS parameters are OFF(no prioritisation are applied):
  - Sending background traffic with a default number of 1 user equipment.
  - Increasing the number of UEs having services with class 2 traffic between 1 and 15 UEs, while Classes 0 and 1 are not available
  - Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.
- Once QoS parameters are ON (Scheduling prioritisation are applied):
  - Sending background traffic with a default number of 1 user equipment.
  - Increasing the number of UEs having services with class2 traffic between 1 and 15 UEs, while Classes 0 and 1 are not available.
  - Testing the class 3 throughput (the most important performance parameter in regards to the best effort traffic) and delay.
  - Graph the results in comparison with the state of QoS 9parameters OFF

4.5.8 q3, Scenario 4 Results

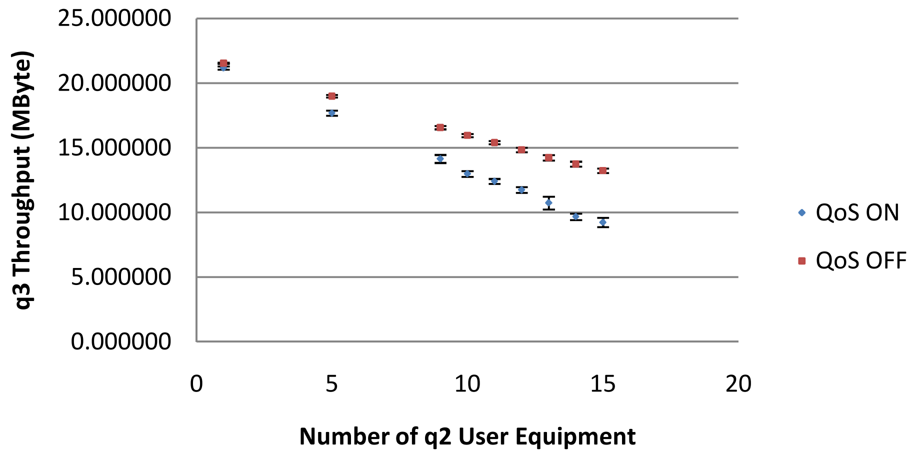

Figure 53 : q3 Received Throughput while q2 UEs Increases

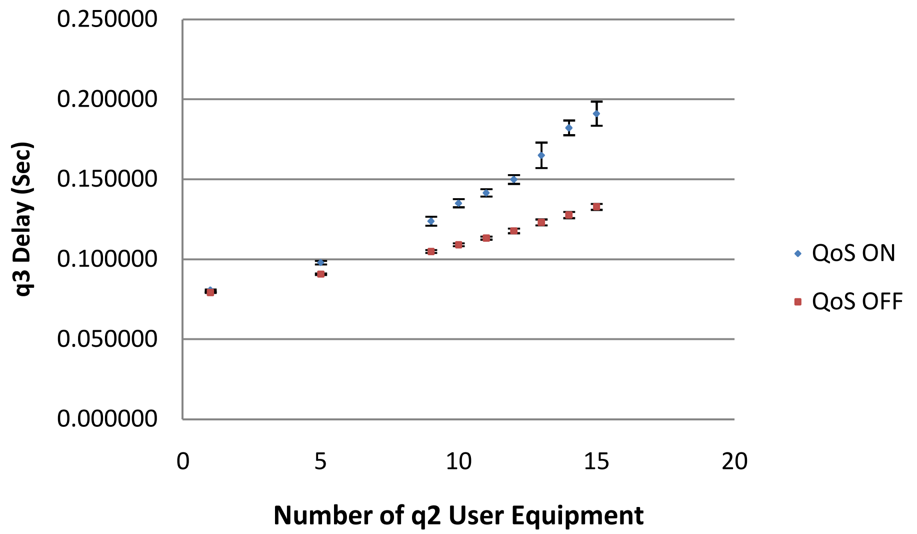

Figure 54 : q3 Delay while q2 UEs Increases

Referring to Figure 53, the difference between the results of the two cases, once QoS ON and QoS OFF is clear. The value of the q3 user equipment throughput in the existence of QoS mechanism is lower than the throughput value in the absence of QoS mechanism. As the number of q2 traffic user equipments increases, both situation throughput values decreases where the advantaged values are the ones in the case of QoS mechanism not triggered.

As the number of interactive user equipments increases, the delay increases in both situation, where the results in the case of QoS ON is much worse than the delay results in the QoS OFF state. This is clearly true in regards to the method of prioritisation used, as the q2 traffic is prioritized in higher state than the q3 traffic. The delay results are graphed in Figure 54.

4.6 Chapter Summary

From above results, the QoS scheduling mechanism has mainly achieved its primary objectives. The scheduling mechanism has assisted conversational traffic packets to travel faster from mouth to ear, while the downstream throughput is constant or even higher. The prioritisation mechanism provides advantages to real-time streaming traffic in the existence of best effort traffic as lower delay and higher throughput. Due to flexibility of the lower priority traffic classes (best effort classes) with time, there is no significant effect of the mechanism on them. The only concern is that the throughput of class2 and class3 traffic decreases in the existence of real-time traffic in loaded link. Even though, the throughput decreases with acceptable value that the traffic can still be executed.

Chapter 5 : Consequential Effects and Project Resources

5.1 Introduction

This chapter is included to discuss the consequential effects and project resources. It is highly recommended before beginning any project, to study and analyze any prospected consequential effects on everyone and /or everything. In addition the project resources should be discussed to reference the tools has been used in the project.

5.2 Assessment of Consequential Effects

5.2.1 Sustainability

As the LTE/SAE system improvement will decrease the number of nodes in the network, evidently there will be a decrease in the disposable and recyclable materials used for the infrastructure. As the LTE philosophy is compatible with previous generations, users can still use their 3rd Generation mobile equipment while still benefitting from the 4th Generation; therefore LTE will decrease the number of disposable and/or recyclable mobile phone equipments.

5.2.2 Ethical Responsibility

It is important to obtain accurate results due to that the findings of this paper would demonstrate the advantages of LTE/SAE system as well as the advantages of applying the QoS features in the network. Any limitations that would affect the results' accuracy must be described and recorded. Especially when using the university's properties and devices. Also, non-trusted softwares and products must be avoided to prevent any damage to those devices.

5.2.3 Risk Management

In general, projects would involve several numbers of risks. The type of risks involved in this project is mostly dealing with software hazards. Downloading different packages used in this project would increase the probability of virus attacks. Efficient virus detectors must be available in the computer devices used for this experiment.

One more risk related to this project especially with students who do not have any background and experience with such work, is the pattern of deviation in the research work. To fix such deviation the graph in Figure 55 shows the work strategy of identifying the deviation as early as possible.

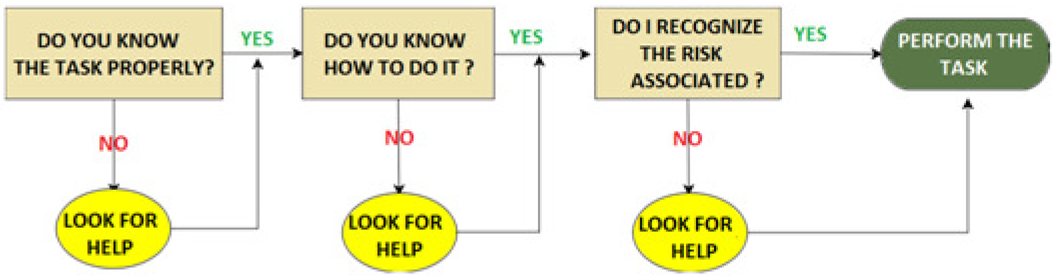

Figure 55: Task Progress and Deviation Identification Chart (this figure made using DIA software and MS office painter)

5.2.4 Safety and Health Consequences

5.2.4.1 Consequences Related to this project

Work on software modelling needs large amount of time, Due to this there are possible side effects on human being health:

- If the workstation is not well prepared for sitting, back pain and other pains can be experienced.
- Bad connection of power adapters and wires can cause a fire or electric shock.
- Bad types of Monitors and screens could harm eyes.

- Command window of Linux and Ubuntu usually has a white background which could affect the eyes with long period.
- The shiny light of screen could cause an eye harm as well as headache; therefore the light should be adjusted to comfort the eyes.
- Some other side effects can occur if a person uses equipment with screens for four hours daily without any protection could be: 'stress, headaches, irritability, insomnia, eye strain, eyesight decline, abnormal general fatigue, decrease in productivity and in the natural resistance of the immune system, decline in libido, disorders in the menstrual cycle, and hormonal disturbances.' (EMF Bioshield website, 2010).

5.2.4.2 Real Network Consequences

In relation to the human being health, there are no proven studies demonstrating that the waves and signals of the telecommunication equipments have negative impacts to the human health. The only concern with those propagated waves and signals is the interference with other electronic devices such as medical and aeroplane electronic systems.

5.3 Resources

The resources of this project are mostly open sources and available for free, as well as some software CD's that are available on-campus and have been provided on request by project supervisor.

5.3.1 Sun Virtual Box

It is a software that is used to create a virtual environment on an existing host operation system and the one have been used for this project is the version VirtualBox-3.1.8-61349-Win that is compatible with Windows Operating System.

5.3.2 Ubuntu

It is the Operating System used for the virtual environment and it is based on LINUX distribution. The version used in this project is Ubuntu 9.10 and it has been provided by the Faculty of Engineering and Surveying at the University of Southern Queensland.

5.3.3 Network Simulator 2

It is an open source simulator used to serve the user to model the network, implementing the elements, providing interfaces and simulate the network model. The version used in this model is ns-2.33 and it is installed on the foregoing Ubuntu operating system.

5.4 Chapter Summary

To sum up, the prospected effects, risks, and tools used in this project are important to be discussed. This chapter includes number of consequential effects concern to environment, human health, and risk managements. It is also includes brief discussion and description of the tools were used in the project.

Chapter 6 : Overall Summary and Conclusion

The main aim of this dissertation is to investigate and analyse the Long Term Evolution/ System Architecture Evolution Network (LTE/SAE). This network is the approximate Fourth Generation (≈4G) network of mobile network. The actual evolution is to the broadband services since the voice call can be done easily without problems by using the 2nd Generation network. LTE network has number of advantages such that it has low number of network elements which leads to low time delay. LTE network standardisations and requirements are included in 3rd Generation Partnership Project (3GPP) specifications starting from release 8. One section of Long Term Evolution network requirements concerns of network Quality of Service (QoS). Number of issues is to be addressed to improve the Quality of Service of Long Term Evolution network. To address these issues, number of requirements to be satisfied and these requirements are known as Quality of Service requirements. Parts of those requirements are delay and throughput improvement. Specifically saying, one aim of this project is to investigate the available QoS mechanism and study the benefits behind it.

The available QoS mechanism has been investigated and software model has been used to test its advantages. One advantage of this mechanism is to improve the time that the sent packet takes to arrive to its destination. More specifically, this mechanism looks after the real time information to be faster than other traffic's types. These traffic classes are mainly divided into two main classes, each main type include two subclasses. These classes are mainly the real-time traffic and best effort traffic. The real time traffic includes conversational and streaming traffic. The best effort traffic includes the interactive and background classes. The QoS mechanism also tries to improve the downstream throughput.

The model used in this project has been built in Network Simulator software. Once all software's hurdles have been overcome, number of scenarios to test the network has been listed. These scenarios' objectives are to test the performance features including packet delay and received throughput. These two features are tested for the first user equipment using a specific type of traffic in existence of the QoS mechanism and loading traffic. This loading traffic used to load the network in each time is different and it is from the four classes mentioned above. A comparison between the results obtained in the existence of the QoS mechanism is made with the results obtained in the absence of this mechanism.

The results have been obtained in this project show reasonable behaviour. The QoS mechanism helps real time traffic to be faster than it is in the case where this mechanism is not used. More specifically the results were as follow: Firstly, delay and throughput of conversational traffic are improved while using the QoS mechanism. Secondly, delay and throughput of streaming traffic are improved in comparison to best effort traffic. Thirdly, the throughput and delay of best effort are limited in the existence of real-time traffic.

To sum up, this method of traffic scheduling demonstrates that the LTE's Quality of Service features has been satisfied. As a result, the performance features of LTE network are improved. This is clearly shows that the QoS mechanism used by Long Term Evolution network is reliable and effective for real-time traffic.

References

Berzin, O 2010, *'Bandwidth, Delay, Throughput and Some Math',* viewed 17 August 2010, < http://www.ccieflyer.com/>.

Blum, N de Gouveia, FC Magedanz, T 2007,' An Open IMS Testbed for Exploring Wireless Service Evolution and Network Architecture Evolution towards SAE and LTE ', *Wireless Broadband and Ultra Wideband Communications, AusWireless2007,The 2nd International Conference,* p. 53.

Castro, JP 2004, *All IP in 3G CDMA Networks: the UMTS Infrastructure and Service Platforms*, John Wiley & Sons Ltd, England

Classifying VoIP Signalling and Media with DSCP for QoS 2010, Release 12.2 T, Cisco IOS Software, Cisco Systems, viewed 12 April 2010, <http://www.cisco.com/en/US/docs/ios/12_2t/12_2t2/feature/guide/ft_dscp.html>.

Dahlman, E Ekstrom, H Furuskar, A Jading, Y Karlsson,JLundevall,M Parkvall, S 2006, 'The 3G Long Term Evolution Radio Interface Concepts and Performance Evaluation ', *Vehicular Technology Conference,* vol. 1, no. 1, pp. 137-141.

Electromagnetic Fields Biological Shielding 2010, '*Are Computer Monitors Harmful to Your Health?',* viewed 5 June 2010, <http://www.emfbioshield.com/arecrt.html >.

Face to Face-Embrace Network Technology Development Consulting in All-IP Times 2010, Issue 12, HUAWEI Company, China, viewed 15 April 2010, <http://www.huawei.com/publications/view.do?id=6110&cid=11400&pid=2043>.

Gronbaek, I 2008,'Architecture for the Internet of Things (IoT): API and interconnect', vol. 1, no. 1, pp. 802-7.

Gronbaek, I 2000,' IP QoS bearer service elements for the converging fixed and mobile 3G.IP network ', *Personal, Indoor and Mobile Radio Communications, PIMRC 2000, The 11th IEEE International Symposium,* vol. 1, no. 1, pp. 19-23.

Holma, H & Toskala, A 2002, *WCDMA for UMTS* – Radio Access for Third Generation Mobile Communications, 2nd edn, Wiley, England.

Holma, H & Toskala, A 2004, *WCDMA for UMTS* – Radio Access for Third Generation Mobile Communications, 3rd edn, Wiley, England.

Holma, H & Toskala, A 2007, *WCDMA for UMTS – HSPA evolution and LTE*, 4th edn, Wiley, England.

Issariyakul, T & Hossain, E 2009, *Introduction to Network Simulator NS2*, Springer Science+Business Media, New York.

Krishnan, S Marchand, L, Cassel, GN 2008,' An IETF-based Evolved Packet System beyond the 3GPP Release 8', *Ericsson Research & Development*, viewed 29 April 2010, <http://www.ericsson.com/res/thecompany/docs/journal_conference_papers/packet_technologies/Beyond_R8_EPS.pdf>.

Kwan, R Leung, C 2010, 'Journal of Electrical and Computer Engineering', *A Survey of Scheduling and Interference Mitigation in LTE,*vol.1, no. 1, viewed 11 August 2010,< http://www.hindawi.com/journals/jece/2010/273486.html>.

Leading Edge –LTE Requirements for Bearer Networks 2009, Issue 50, HUAWEI Company, China, viewed 15 April 2010, <http://www.huawei.com/publications/view.do?id=6110&cid=11400&pid=2043>.

Long Term Evolution (LTE) Q&A 2010, GSM World Association, viewed 20 March 2010, <http://www.gsmworld.com/documents/05032009140431.pdf>.

Quality of Service Class Identifier 2009, Long Term Evolution World Summit, Alcatel & Lucent Company, Berlin

Long Term Evolution (LTE): The vision beyond 3G 2010, Technology Marketing Corporation, viewed 14 March 2010, <http://4g-.tmcnet.com/topics/4g-wirelessevolution/articles/Nortel%20LTE%20The%20vision%20beyond%203G.pdf>.

Long Term Evolution (LTE): Long Term Evolution Presentations with Audio 2009, Event Helix Incorporation, Maryland, US, viewed 10 April 2010, <http://www.eventhelix.com/lte/lte-tutorials.htm>.

Ludwig, R Ekstrom, H Willars, P & Lundin, N 2006,' An Evolved 3GPP QoS Concept', *Ericsson Research & Development* , vol. 1.

Nortel Technology Demo Long Term Evolution Wireless Access 2010, Nortel Networks, viewed 19 March 2010, <http://www.nortel.com/corporate/investor/events/investorconf/collateral/2_1_lte_demo_sheet.pdf>.

Olsson, M Sultana, S Rommer, S Frid, L & Mulligan, C 2009, *SAE and the Evolved Packet Core Driving the Mobile Broadband Revolution,* 1st edn, Academic Press, US, viewed 20 April 2010, < http://books.google.com.au/books?id=PUU_vr_XfV8C&printsec=frontcover&dq=System+Architecture+Evolution+(SAE):+Evolved+Packet+Core+for+LTE,+Fixed+and+Other+Wireless+Accesses&hl=en&ei=tBH5S7-TNMSecfiEgOcL&sa=X&oi=book_result&ct=result&resnum=1&ved=0CDEQ6AEwAA#v=onepage&q&f=false>.

Prashant, P 2009, *Introduction to LTE* , Issue 50, HUAWEI Company, China, viewed 15 April 2010, <http://www.huawei.com/publications/view.do?id=6110&cid=11400&pid=2043>.

Qiu, Q Chen, J Ping, L Zhang, Q Pan, X 2009, 'LTE/SAE Model and its Implementation in NS 2', *Mobile Ad-hoc and Sensor Networks the 5th International Conference*, pp. 299-303.

Rumney, M 2009, *LTE and the Evolution to 4G Wireless : Design and Measurements,* Agilent Technologies Publication, US, viewed 12 May 2010, < http://books.google.com.au/books?id=boePyryvxRAC&pg=PA149&dq=System+Architecture+Evolution+(SAE):+Evolved+Packet+Core+for+LTE,+Fixed+and+Other+Wireless+Accesses&hl=en&ei=tBH5S7-TNMSecfiEgOcL&sa=X&oi=book_result&ct=book-thumbnail&resnum=2&ved=0CDcQ6wEwAQ#v=onepage&q&f=false >.

Song, J Gil,G & Kim, D 2010, 'Journal on Wireless Communication and Networking',*Packet-Scheduling Algorithm by the Ratio of Transmit Power to the Transmission Bits in 3GPP LTE Downlink,* vol. 1, no 1, viewed 15 August 2010, < http://www.hindawi.com/journals/wcn/2010/251281.html>.

Third Generation Partnership Project Release 8 2010, Release 8, 3GPP Global Initiative, viewed 10 Mars 2010, <http://www.3gpp.org/Release-8>.

Third Generation Partnership Project, Technical Specification Group Services and System Aspects;Quality of Service (QoS) concept and architecture(Release 8), TS 23.107 V8.0.0 (2008-12), viewed 15 Sep 2010,<http://www.3gpp.org/ftp/Specs/archive/23_series/23.107/23107-800.zip>.

Tsai, Y Zhang, G Grieco, D Ozluturk, F 2007, 'Cell Search in 3GPP Long,Term Evolution Systems', vol. 2, no. 2.

Vadada, H 2009, *QoS over LTE & WiMAX Networks*, viewed 1 September 2009, < http://knol.google.com/k/harish-vadada/qos-over-lte-wimax-networks/39mvtk6o5rtho/4#>.

Appendixes

Appendix A: Project Specification

Saud Alenazi w0077753 Appendix A

University of southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG4111/4112 Research Project
PROJECT SPECIFICATIONS

FOR: **Armando APAN**

TOPIC: ANALYSYS AND TAXONOMY OF NETWORK QoS CONCEPTS IN THE SAE/LTE SYSTEM

SUPERVISOR: Dr Alexander Kist

ENROLMENT: ENG4111 – S1, D, 2010;
ENG4112 – S2, D, 2010

PROJECT AIM: To analyse SAE/LTE system's environment and the categorisation of the available QoS mechanism.

SPONSORSHIP: To Faculty of Engineering and Surveying

PROGRAMME: **(Issue A, 23rd March 2010)**

1) Researching the SAE/LTE system environment.
2) Study and analyse available QoS mechanisms.
3) Develop a mathematical models and/or simulation to evaluate the performance of real-time media application.
4) Evaluate and analyse the performance of a number of applications.

As time permits:

5) Extended data analysis of produce results.

AGREED ____________________ (student):
(supervisor)

Date: 22 / 03 /2010 Date: 22 / 03 /2010

Examiner/CO-examiner:__________________________

Appendix B: Simulation Scripts

Lte.tcl

```
# Define the multicast mechanism
set ns [new Simulator -multicast on]

# Predefine tracing
set f [open out.tr w]
$ns trace-all $f
set nf [open out.nam w]
$ns namtrace-all $nf

# Set the number of subscribers
set numberClass0 0
set numberClass1 0
set numberClass2 15
set numberClass3 1
set number [expr {$numberClass0 + $numberClass1 +
$numberClass2 + $numberClass3}]
# qos_ means whether classfication/scheduling
mechanism is used
#Queue/LTEQueue set qos_ true
Queue/LTEQueue set qos_ false
# flow_control_ is used in the model phase
Queue/LTEQueue set flow_control_ false

# Define the LTE topology
# UE(i) <--> eNB <--> aGW <--> server


# step 1: define the nodes, the order is fixed!!
set eNB [$ns node];#node id is 0
set aGW [$ns node];#node id is 1
set server [$ns node];#node id is 2
for { set i 0} {$i<$number} {incr i} {
  set UE($i) [$ns node];#node id is > 2
}

# step 2: define the links to connect the nodes
for { set i 0} {$i<$number} {incr i} {
  $ns simplex-link $UE($i) $eNB 10Mb 2ms
LTEQueue/ULAirQueue
  $ns simplex-link $eNB $UE($i) 10Mb 2ms
LTEQueue/DLAirQueue
}

$ns simplex-link $eNB $aGW 2Mb 2ms LTEQueue/ULS1Queue
$ns simplex-link $aGW $eNB 2Mb 2ms LTEQueue/DLS1Queue
```

```
# The bandwidth between aGW and server is not the
bottleneck.
$ns simplex-link $aGW $server 5000Mb 2ms DropTail
$ns simplex-link $server $aGW 5000Mb 2ms
LTEQueue/DLQueue



#--------manual set constant-------------((the best
one until 17 Augus))--
# to change the RNG manually
global defaultRNG
#  to be changed manually from 1(default) to
approximate value equal to 7.6x10^22
$defaultRNG seed 10
#----------------------------------------------

# step 3: define the traffic, based on  TR23.107 QoS
concept and architecture
#    class id    class type     simulation application

#    ------------------------------------------------
#    0:     Conversational: CBR/UdpAgent
#    1:     Streaming:      CBR/UdpAgent
#    2:     Interactive:    HTTP/TcpAgent (HTTP/Client,
HTTP/Cache, HTTP/Server)
#    3:     Background:     FTP/TcpAgent

# step 3.1 define the conversational traffic
for { set i 0} {$i<$numberClass0} {incr i} {
  set null($i) [new Agent/Null]
        set nullS($i) [new Agent/Null]
  $ns attach-agent $UE($i) $null($i)
        $ns attach-agent $server $nullS($i)
  set udp($i) [new Agent/UDP]
        set udpUE($i) [new Agent/UDP]
  $ns attach-agent $server $udp($i)
        $ns attach-agent $UE($i) $udpUE($i)
  $ns connect $null($i) $udp($i)
        $ns connect $nullS($i) $udpUE($i)
  $udp($i) set class_ 0
        $udpUE($i) set class_ 0
  set cbr($i) [new Application/Traffic/CBR]
  set cbrS($i) [new Application/Traffic/CBR]
  $cbr($i) attach-agent $udp($i)
  $cbrS($i) attach-agent $udpUE($i)
  $ns at 0.4 "$cbr($i) start"
  $ns at 0.4 "$cbrS($i) start"
  $ns at 40.0 "$cbr($i) stop"
  $ns at 40.0 "$cbrS($i) stop"
```

```
}


# step 3.2 define the streaming traffic
for { set i $numberClass0} {$i<
($numberClass0+$numberClass1)} {incr i} {
  set null($i) [new Agent/Null]
  $ns attach-agent $UE($i) $null($i)
  set udp($i) [new Agent/UDP]
  $ns attach-agent $server $udp($i)
  $ns connect $null($i) $udp($i)
  $udp($i) set class_ 1
  set cbr($i) [new Application/Traffic/CBR]
  $cbr($i) attach-agent $udp($i)
  $ns at 0.4 "$cbr($i) start"
  $ns at 40.0 "$cbr($i) stop"
}


# step 3.3 define the interactive traffic
$ns rtproto Session
set log [open "http.log" w]

# Care must be taken to make sure that every client
sees the same set of pages as the servers to which
they are attached.
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Constant] ;# Size
generator
$tmp set val_ 10240  ;# average page size
$pgp ranvar-size $tmp
set tmp [new RandomVariable/Exponential] ;# Age
generator
$tmp set avg_ 4 ;# average page age
$pgp ranvar-age $tmp

set s [new Http/Server $ns $server]
$s set-page-generator $pgp
$s log $log

set cache [new Http/Cache $ns $aGW]
$cache log $log

for { set i [expr $numberClass0+$numberClass1] }
{$i<($numberClass0+$numberClass1+$numberClass2)} {incr
i} {
  set c($i) [new Http/Client $ns $UE($i)]
  set ctmp($i) [new RandomVariable/Exponential] ;#
Poisson process
  $ctmp($i) set avg_ 1 ;# average request interval
```

```
  $c($i) set-interval-generator $ctmp($i)
  $c($i) set-page-generator $pgp
  $c($i) log $log
}

$ns at 0.4 "start-connection"
proc start-connection {} {
        global ns s cache c number numberClass0
numberClass1 numberClass2

  $cache connect $s
  for { set i [expr $numberClass0+$numberClass1]}
{$i<($numberClass0+$numberClass1+$numberClass2)} {incr
i} {
            $c($i) connect $cache
            $c($i) start-session $cache $s
  }
}


# step 3.4 define the background traffic
# no parameters to be configured by FTP
# we can configue TCP and TCPSink parameters here.
for { set i [expr
$numberClass0+$numberClass1+$numberClass2]}
{$i<($numberClass0+$numberClass1+$numberClass2+$number
Class3)} {incr i} {
  set sink($i) [new Agent/TCPSink]
  $ns attach-agent $UE($i) $sink($i)
  set tcp($i) [new Agent/TCP]
  $ns attach-agent $server $tcp($i)
  $ns connect $sink($i) $tcp($i)
  $tcp($i) set class_ 3
  set ftp($i) [new Application/FTP]
  $ftp($i) attach-agent $tcp($i)
  $ns at 0.4 "$ftp($i) start"
}

# finish tracing
$ns at 30 "finish"
proc finish {} {
  global ns f log
  $ns flush-trace
  flush $log
  close $log
  close $f
  exit 0
}

# Finally, start the simulation.
$ns run
```

Delay.AWK

```
# calculate each class delay

BEGIN{
UEclass0=-1;
UEclass1=-1;
UEclass2=-1;
UEclass3=-1;
}
{
     event = $1;
      time = $2;
      node_s = $3;
      node_d = $4;
      trace_type = $5;
      pkt_size = $6;
      flag = $7;
     classid = $8
     pkt_id = $12;
     src = $9;
split(src_,tmp,".");
src = tmp[1];

if(node_d > 2 && classid == 0 && UEclass0 == -1) {
  UEclass0 = node_d;
}
if(node_d > 2 && classid == 1 && UEclass1 == -1) {
  UEclass1 = node_d;
}
if(node_d > 2 && classid == 2 && UEclass2 == -1) {
  UEclass2 = node_d;
}
if(node_d > 2 && classid == 3 && UEclass3 == -1) {
  UEclass3 = node_d;
}

     if (event == "+" && (node_s ==2))
     {
          packet[pkt_id]=time;
     }


#classid id =2 is HTTP traffic, cache is aGW  (event ==
"r" && node_d>=2)
     if (event == "r" && (node_d>=2))
     #if (event == "r" &&  ((node_d == UEclass0)) ||
(node_d == UEclass1) || (node_d == UEclass2) || (node_d
== UEclass3))
#if (event == "r" &&  (node_d == UEclass0))
     {
```

```
          if(packet[pkt_id]!=0){
               delay[classid,0] = delay[classid,0] + time
- packet[pkt_id];
               delay[classid,1] = delay[classid,1] + 1;
          }
     }

     if (event == "+" && node_s >2)
     {

     packet[pkt_id]=time;
     }


#classid id =2 is HTTP traffic, cache is aGW  (event ==
"r" && node_d>=2)

     #if (event == "r" && (node_d==UEclass2) )
     if (event == "r" && (node_d>2 || node_d==1) )
     {
          if(packet[pkt_id]!=0){
               delayy[classid,0] = delayy[classid,0] +
time - packet[pkt_id];
               delayy[classid,1] = delayy[classid,1] + 1;
          }
     }



}
END {
     for(classid=0;classid<4;classid++) {

     av_delay[classid]=delay[classid,0]/delay[classid,1];

     av_delayy[classid]=delayy[classid,0]/delayy[classid,
1];
          total[0] = total[0] + delay[classid,0] +
delayy[2,0];
          total[1] = total[1] + delay[classid,1] +
delayy[2,1];
     }
          print "0        1         2         3
total";
     print av_delay[0]," ",av_delay[1],"
     ",av_delayy[2],"    ",av_delay[3],"
     ",total[0]/total[1];
}
```

Throughput.AWK

```
# calculate each class throughput(received, sent,
lost)

BEGIN{
flag=0;
UEclass0=-1;
UEclass1=-1;
UEclass2=-1;
UEclass3=-1;
}

{
#r 0.241408 1 0 tcp 1040 ------- 1   4.0   0.0   3   6
#$1 $2     $3 $4 $5   $6   $7    $8    $9   $10  $11
$12
event = $1;
time = $2;
node_s = $3;
node_d = $4;
trace_type = $5;
pkt_size = $6;
classid = $8;
src_ = $9;
split(src_,tmp,".");
src = tmp[1];

if(node_d > 2 && classid == 0 && UEclass0 == -1) {
  UEclass0 = node_d;
}
if(node_d > 2 && classid == 1 && UEclass1 == -1) {
  UEclass1 = node_d;
}
if(node_d > 2 && classid == 2 && UEclass2 == -1) {
  UEclass2 = node_d;
}
if(node_d > 2 && classid == 3 && UEclass3 == -1) {
  UEclass3 = node_d;
}

#if


#eNB node id is 0
#aGW node id is 1
#server node id is 2
#UE node id >2
# note that the received throughput are the ones
received by the UEs
```

```
# and the sent ones are the ones received by the
server
if(event == "-" &&  (node_d == UEclass0)  || (node_d
== UEclass1) || (node_d == UEclass2) || (node_d ==
UEclass3)) {
#if(event == "-" && node_d >2 ) {
  if(flag==0) {
       start_time=time;
       flag=1;
  }
  end_time=time;
  ue_r_byte[classid] = ue_r_byte[classid] + pkt_size;
}
if(event == "-" && node_d ==2 ) {
 if(src == UEclass0) {
  ue_s_byte[classid]=ue_s_byte[classid]+pkt_size;

}
}
if(event == "d") {
  #ue_d_byte[classid]=ue_d_byte[classid]+pkt_size;

}
}

END {
  for(i=0;i<4;i++)
  {
       ue_d_byte[classid]=ue_s_byte[classid]-
ue_r_byte[classid]
       ue_r[i]=ue_r_byte[i]/1000000;
       ue_s[i]=ue_s_byte[i]/1000000;
       ue_d[i]=ue_d_byte[i]/1000000;
       total_r=total_r+ue_r[i];
       total_s=total_s+ue_s[i];
       total_d=total_d+ue_d[i];
  }
  printf("0\t1\t2\t3\ttotal(Mbyte)\n");
  printf("%1.2f\t%1.2f\t%1.2f\t%1.2f\t%1.2f\n",ue_r[0]
,ue_r[1],ue_r[2],ue_r[3],total_r);
  printf("%1.2f\t%1.2f\t%1.2f\t%1.2f\t%1.2f\n",ue_s[0]
,ue_s[1],ue_s[2],ue_s[3],total_s);
  printf("%1.2f\t%1.2f\t%1.2f\t%1.2f\t%1.2f\n",ue_d[0]
,ue_d[1],ue_d[2],ue_d[3],total_d);
}
```

Downlink S1 Queue Script in C++ language

```
#include "dls1queue.h"

extern int flow[100];

static class DLS1QueueClass : public TclClass {
public:
	DLS1QueueClass() : TclClass("Queue/LTEQueue/DLS1Queue") {}
	TclObject* create (int, const char*const*) {
		return (new DLS1Queue);
	}
} class_dls1queue;

void DLS1Queue::enque(Packet* p)
{
	hdr_ip *iph=HDR_IP(p);
	int classid=iph->flowid();

	if(qos_) {
		//classfication
		switch(classid){
			case 0: q0->enqueue(p);break;
			case 1: q1->enqueue(p);break;
			case 2: q2->enqueue(p);break;
			case 3: q3->enqueue(p);break;
			default:
				{
					printf("invalid classid %d\n",classid);
					exit(0);
				}
		}
	} else {//no qos_, no classification
		q0->enqueue(p);
	}
}

Packet* DLS1Queue::deque()
{
	if(!flow_control_)
	{
		if(!qos_)
			return q0->dequeue();

		//if qos_ && !flow_control
		//scheduling: strict priority
		if(q0->length()>0)
		{
			return q0->dequeue();
		}
		if(q1->length()>0)
```

```
		{
			return q1->dequeue();
		}
		if(q2->length()>0)
		{
			//return q2->deque();
			return q2->dequeue();
		}
		if(q3->length()>0)
		{
			return q3->dequeue();
		}

		//all the queues are empty, no packet to be sent.
		//printf("LTEQueue::deque(), all the queues are empty, no packet to be
sent.\n");
		return NULL;
	}

	//else flow control
	//flow control only valid to classo 2 & class 3
	if(!qos_)
	{
		for(int i=0;i < q0->length();i++)
		{
			Packet *p=q0->find(i);
			if(p==NULL) return NULL;
			hdr_ip *iph=HDR_IP(p);
			hdr_cmn *cmh=HDR_CMN(p);
			int size=cmh->size();
			int classid=iph->flowid();
			//int flowid=iph->daddr();

			if(classid==0 || classid==1)
			{
				p=q0->remove(i);
				return p;
			}
			//flow control only apply to class 2 and class 3
			if(size < flow[classid])
			{
				p=q0->remove(i);
				return p;
			}

			//else continue to find next packet
		}
		//no packet can be sent
		return NULL;
	}
```

```
//with flow control and QoS
if(qos_)
{
	if(q0->length()>0) {
		return q0->dequeue();
	}
	if(q1->length()>0) {
		return q1->dequeue();
	}
	for(int i=0;i < q2->length();i++) {
		Packet *p=q2->find(i);
		if(p==NULL) {
			// no packet to send in q2
			break;
		}
		hdr_ip *iph=HDR_IP(p);
		hdr_cmn *cmh=HDR_CMN(p);
		int size=cmh->size();
		int classid=iph->flowid();
		int flowid=iph->daddr();

		if(size<flow[classid])
		{
			p=q2->remove(i);
			return p;
		}
		//else continue to find next packet
	}

	//no packet can be sent in q2, try q3
	for(int i=0;i < q3->length();i++) {
		Packet *p=q3->find(i);
		if(p==NULL) {
			// no packet to send in q3
			return NULL;
		}
		hdr_ip *iph=HDR_IP(p);
		hdr_cmn *cmh=HDR_CMN(p);
		int size=cmh->size();
		int flowid=iph->daddr();
		int classid=iph->flowid();

		if(size<flow[classid])
		{
			p=q3->remove(i);
			return p;
		}
		//else continue to find next packet
	}
```

```
                // no packet can be sent in q3
                return NULL;
        }
}
```

Downlink Air Queue Script File in C++ language

```
#include "dlairqueue.h"

//int max_buff=51200;
extern int flow[100];

static class DLAirQueueClass : public TclClass {
public:
	DLAirQueueClass() : TclClass("Queue/LTEQueue/DLAirQueue") {}
	TclObject* create (int, const char*const*) {
		return (new DLAirQueue);
	}
} class_dlairqueue;

void DLAirTimer::expire(Event*)
{
	q_->update();
}

void DLAirQueue::update()
{
	if(!qos_) {
		flow[0] = q0->limit()*q0->meanPacketSize() - q0->byteLength();
	} else {
		flow[0] = q0->limit()*q0->meanPacketSize() - q0->byteLength();
		flow[1] = q1->limit()*q1->meanPacketSize() - q1->byteLength();
		flow[2] = q2->limit()*q2->meanPacketSize() - q2->byteLength();
		flow[3] = q3->limit()*q3->meanPacketSize() - q3->byteLength();
	}

	dlairtimer.resched(1.0);
}

void DLAirQueue::enque(Packet* p)
{
	hdr_ip *iph=HDR_IP(p);
	int classid=iph->flowid();

	if(qos_) {
		//classfication
		switch(classid){
			case 0: q0->enqueue(p);break;
			case 1: q1->enqueue(p);break;
			case 2: q2->enqueue(p);break;
			case 3: q3->enqueue(p);break;
			default:
				{
					printf("invalid classid %d\n",classid);
					exit(0);
```

```
                    }
            }
        } else {//no qos_, no classification
            q0->enqueue(p);
        }
}

Packet* DLAirQueue::deque()
{
        if(!qos_)
        {
            return q0->dequeue();
        }
        //scheduling: strict priority
        if(q0->length()>0)
        {
            return q0->dequeue();
        }
        if(q1->length()>0)
        {
            return q1->dequeue();
        }
        if(q2->length()>0)
        {
            //return q2->deque();
            return q2->dequeue();
        }
        if(q3->length()>0)
        {
            return q3->dequeue();
        }

        return NULL;
}
```

LTE Queue Script in C++ language

```
#include "ltequeue.h"

//int max_buff=51200;
int flow[100];

static class LTEQueueClass : public TclClass {
public:
	LTEQueueClass() : TclClass("Queue/LTEQueue") {}
	TclObject* create (int, const char*const*) {
		return (new LTEQueue);
	}
} class_ltequeue;

void LTEQueue::enque(Packet *p)
{
}
Packet* LTEQueue::deque()
{
}
```

Uplink Air Queue Script File in C++ language

```
#include "ulairqueue.h"

static class ULAirQueueClass : public TclClass {
public:
	ULAirQueueClass() : TclClass("Queue/LTEQueue/ULAirQueue"){}
	TclObject* create (int, const char*const*){
		return (new ULAirQueue);
	}
} class_ulqirqueue;

void ULAirQueue::enque(Packet* p)
{
	hdr_ip *iph=HDR_IP(p);
	int classid=iph->flowid();

	if(!qos_)
	{
		q0->enqueue(p);
		return;
	}

	//with QoS
	switch(classid){
		case 0: q0->enqueue(p);break;
		case 1: q1->enqueue(p);break;
		case 2: q0->enqueue(p);break;
		case 3: q0->enqueue(p);break;
		defaut:
		{
			printf("invalid class id %d\n", classid);
			exit(0);
		}
	}
}

Packet* ULAirQueue::deque()
{
	if(!qos_)
		return q0->dequeue();

	if(q0->length()>0)
      {
          return q0->dequeue();
      }
      if(q1->length()>0)
      {
          return q1->dequeue();
      }
```

```
    if(q2->length()>0)
    {
        return q2->dequeue();
    }
    if(q3->length()>0)
    {
        return q3->dequeue();
    }

      return NULL;
}
```

Uplink S1 Queue Script File in C++ language

```
#include "uls1queue.h"

static class ULS1QueueClass : public TclClass {
public:
	ULS1QueueClass() : TclClass("Queue/LTEQueue/ULS1Queue"){}
	TclObject* create (int, const char*const*){
		return (new ULS1Queue);
	}
} class_uls1queue;

void ULS1Queue::enque(Packet* p)
{
	hdr_ip *iph=HDR_IP(p);
	int classid=iph->flowid();

	if(!qos_)
	{
		q0->enqueue(p);
		return;
	}

	//with QoS
	switch(classid){
		case 0: q0->enqueue(p);break;
		case 1: q1->enqueue(p);break;
		case 2: q0->enqueue(p);break;
		case 3: q0->enqueue(p);break;
		defaut:
		{
			printf("invalid class id %d\n", classid);
			exit(0);
		}
	}
}

Packet* ULS1Queue::deque()
{
	if(!qos_)
		return q0->dequeue();

	if(q0->length()>0)
      {
          return q0->dequeue();
      }
      if(q1->length()>0)
      {
          return q1->dequeue();
      }
```

```
        if(q2->length()>0)
        {
            return q2->dequeue();
        }
        if(q3->length()>0)
        {
            return q3->dequeue();
        }

          return NULL;
}
```

Appendix C: Tables of Results

		Q0 only QoS Enabled and Disabled				
UE		average	Stnd_deviation	Conf_ Interval	Avg-conf	avg+conf
1	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
2	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
3	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
4	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
5	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
6	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
7	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
8	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
9	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
10	Throughput_rq0	0.470000	0.000000	0.000000	0.470000	0.470000
11	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
12	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
13	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
14	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
15	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
16	Throughput_rq0	0.461000	0.003000	0.001859	0.459141	0.462859
17	Throughput_rq0	0.431000	0.003000	0.001859	0.429141	0.432859
18	Throughput_rq0	0.410000	0.000000	0.000000	0.410000	0.410000
19	Throughput_rq0	0.389000	0.003000	0.001859	0.387141	0.390859
1	Delay q0	0.007008	0.000000	0.000000	0.007008	0.007008
2	Delay q0	0.007035	0.000005	0.000003	0.007032	0.007038
3	Delay q0	0.007065	0.000005	0.000003	0.007062	0.007068
4	Delay q0	0.007102	0.000004	0.000003	0.007099	0.007104
5	Delay q0	0.007136	0.000004	0.000003	0.007133	0.007139
6	Delay q0	0.007181	0.000006	0.000004	0.007178	0.007185
7	Delay q0	0.007243	0.000008	0.000005	0.007238	0.007248
8	Delay q0	0.007298	0.000009	0.000005	0.007292	0.007303
9	Delay q0	0.007374	0.000014	0.000008	0.007366	0.007383
10	Delay q0	0.007460	0.000012	0.000007	0.007453	0.007467
11	Delay q0	0.007565	0.000016	0.000010	0.007555	0.007575
12	Delay q0	0.007714	0.000010	0.000006	0.007708	0.007721
13	Delay q0	0.007915	0.000027	0.000017	0.007898	0.007932
14	Delay q0	0.008195	0.000014	0.000009	0.008186	0.008204
15	Delay q0	0.008976	0.000036	0.000022	0.008954	0.008998
16	Delay q0	0.345624	0.035142	0.021781	0.323844	0.367405
17	Delay q0	1.201787	0.027065	0.016775	1.185012	1.218562
18	Delay q0	1.949797	0.029815	0.018479	1.931318	1.968276
19	Delay q0	2.635200	0.021350	0.013232	2.621968	2.648432

Appendix Table 1: Class0 only available

		QoS ON				
q1 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
5	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
7	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
9	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
11	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
12	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
13	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
14	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
15	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
1	Throughput_rq1	1.420000	0.011832	0.007334	1.412666	1.427334
5	Throughput_rq1	1.426000	0.014283	0.008852	1.417148	1.434852
7	Throughput_rq1	1.420000	0.007746	0.004801	1.415199	1.424801
9	Throughput_rq1	1.420000	0.006325	0.003920	1.416080	1.423920
11	Throughput_rq1	1.370000	0.007746	0.004801	1.365199	1.374801
12	Throughput_rq1	1.253000	0.007810	0.004841	1.248159	1.257841
13	Throughput_rq1	1.161000	0.008307	0.005148	1.155852	1.166148
14	Throughput_rq1	1.078000	0.004000	0.002479	1.075521	1.080479
15	Throughput_rq1	1.005000	0.006708	0.004158	1.000842	1.009158
1	Delay q0	0.007174	0.000005	0.000003	0.007170	0.007177
5	Delay q0	0.007313	0.000006	0.000004	0.007309	0.007316
7	Delay q0	0.007393	0.000008	0.000005	0.007388	0.007398
9	Delay q0	0.007470	0.000008	0.000005	0.007465	0.007475
11	Delay q0	0.007558	0.000006	0.000004	0.007554	0.007562
12	Delay q0	0.007556	0.000007	0.000005	0.007551	0.007560
13	Delay q0	0.007557	0.000009	0.000006	0.007551	0.007562
14	Delay q0	0.007558	0.000007	0.000004	0.007554	0.007562
15	Delay q0	0.007560	0.000008	0.000005	0.007555	0.007565
1	Delay q1	0.007244	0.000005	0.000003	0.007240	0.007247
5	Delay q1	0.007610	0.000014	0.000009	0.007601	0.007619
7	Delay q1	0.007947	0.000022	0.000014	0.007933	0.007960
9	Delay q1	0.008592	0.000029	0.000018	0.008574	0.008610
11	Delay q1	0.497596	0.050486	0.031291	0.466305	0.528887
12	Delay q1	1.692630	0.037828	0.023446	1.669184	1.716076
13	Delay q1	2.701125	0.039053	0.024205	2.676920	2.725330
14	Delay q1	3.554847	0.046354	0.028730	3.526117	3.583577
15	Delay q1	4.329807	0.039006	0.024176	4.305631	4.353983

Appendix Table 2 : q0 5 UEs, q1 increasing, QoS ON

		QoS OFF				
q1 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
5	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
7	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
9	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
11	Throughput_rq0	0.463000	0.004583	0.002840	0.460160	0.465840
12	Throughput_rq0	0.434000	0.004899	0.003036	0.430964	0.437036
13	Throughput_rq0	0.411000	0.003000	0.001859	0.409141	0.412859
14	Throughput_rq0	0.390000	0.000000	0.000000	0.390000	0.390000
15	Throughput_rq0	0.370000	0.000000	0.000000	0.370000	0.370000
1	Throughput_rq1	1.420000	0.011832	0.007334	1.412666	1.427334
5	Throughput_rq1	1.426000	0.014283	0.008852	1.417148	1.434852
7	Throughput_rq1	1.420000	0.007746	0.004801	1.415199	1.424801
9	Throughput_rq1	1.420000	0.006325	0.003920	1.416080	1.423920
11	Throughput_rq1	1.386000	0.008000	0.004958	1.381042	1.390958
12	Throughput_rq1	1.302000	0.006000	0.003719	1.298281	1.305719
13	Throughput_rq1	1.234000	0.008000	0.004958	1.229042	1.238958
14	Throughput_rq1	1.167000	0.004583	0.002840	1.164160	1.169840
15	Throughput_rq1	1.108000	0.006000	0.003719	1.104281	1.111719
1	Delay q0	0.007184	0.000005	0.000003	0.007181	0.007188
5	Delay q0	0.007457	0.000012	0.000007	0.007450	0.007464
7	Delay q0	0.007713	0.000009	0.000006	0.007708	0.007719
9	Delay q0	0.008200	0.000020	0.000013	0.008188	0.008213
11	Delay q0	0.343821	0.034217	0.021208	0.322613	0.365028
12	Delay q0	1.198299	0.026460	0.016400	1.181899	1.214699
13	Delay q0	1.954868	0.028583	0.017716	1.937152	1.972584
14	Delay q0	2.626021	0.029446	0.018250	2.607771	2.644271
15	Delay q0	3.247439	0.025799	0.015990	3.231449	3.263429
1	Delay q1	0.007191	0.000006	0.000004	0.007187	0.007194
5	Delay q1	0.007464	0.000012	0.000008	0.007456	0.007472
7	Delay q1	0.007717	0.000014	0.000009	0.007708	0.007725
9	Delay q1	0.008190	0.000025	0.000015	0.008175	0.008206
11	Delay q1	0.344303	0.034708	0.021512	0.322791	0.365815
12	Delay q1	1.196770	0.025343	0.015708	1.181062	1.212478
13	Delay q1	1.955199	0.027380	0.016970	1.938229	1.972169
14	Delay q1	2.621516	0.035940	0.022275	2.599241	2.643791
15	Delay q1	3.246570	0.026340	0.016325	3.230245	3.262895

Appendix Table 3: q0 5 UEs, q1 increasing, QoS OFF

		QoS ON				
q2 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
5	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
10	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
15	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
20	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
25	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
30	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
35	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
40	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
45	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
50	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
1	**Throughput_rq2**	0.882000	0.148445	0.092006	0.789994	0.974006
5	**Throughput_rq2**	0.874000	0.156601	0.097061	0.776939	0.971061
10	**Throughput_rq2**	0.954000	0.144444	0.089526	0.864474	1.043526
15	**Throughput_rq2**	0.837000	0.150602	0.093342	0.743658	0.930342
20	**Throughput_rq2**	0.706000	0.082849	0.051350	0.654650	0.757350
25	**Throughput_rq2**	0.661000	0.098534	0.061071	0.599929	0.722071
30	**Throughput_rq2**	0.465000	0.060042	0.037214	0.427786	0.502214
35	**Throughput_rq2**	0.415000	0.075000	0.046485	0.368515	0.461485
40	**Throughput_rq2**	0.370000	0.047117	0.029203	0.340797	0.399203
45	**Throughput_rq2**	0.354000	0.022000	0.013635	0.340365	0.367635
50	**Throughput_rq2**	0.305000	0.032016	0.019843	0.285157	0.324843
1	**Delay q0**	0.007203	0.000014	0.000009	0.007195	0.007212
5	**Delay q0**	0.007468	0.000024	0.000015	0.007453	0.007483
10	**Delay q0**	0.007773	0.000042	0.000026	0.007747	0.007799
15	**Delay q0**	0.008048	0.000034	0.000021	0.008026	0.008069
20	**Delay q0**	0.008210	0.000015	0.000009	0.008201	0.008219
25	**Delay q0**	0.008273	0.000014	0.000009	0.008264	0.008281
30	**Delay q0**	0.008281	0.000015	0.000009	0.008271	0.008290
35	**Delay q0**	0.008286	0.000015	0.000009	0.008276	0.008295
40	**Delay q0**	0.008274	0.000015	0.000010	0.008264	0.008283
45	**Delay q0**	0.008279	0.000015	0.000009	0.008270	0.008288
50	**Delay q0**	0.008281	0.000014	0.000008	0.008272	0.008289
1	**Delay q2**	0.004293	0.000022	0.000013	0.004280	0.004307
5	**Delay q2**	0.004290	0.000022	0.000014	0.004276	0.004304
10	**Delay q2**	0.004306	0.000009	0.000006	0.004300	0.004311
15	**Delay q2**	0.0042921	0.0000179	0.000011	0.004281	0.004303
20	**Delay q2**	0.0042913	0.0000137	0.000009	0.004283	0.004300
25	**Delay q2**	0.0042723	0.0000272	0.000017	0.004255	0.004289
30	**Delay q2**	**0.008281**	**0.000026**	0.000016	0.008265	0.008297
35	**Delay q2**	**0.004216**	**0.000025**	0.000015	0.004201	0.004231
40	**Delay q2**	0.004201	0.000030	0.000018	0.004182	0.004219
45	**Delay q2**	0.004195	0.000016	0.000010	0.004184	0.004205
50	**Delay q2**	0.004159	0.000020	0.000012	0.004147	0.004171

Appendix Table 4 : q0 5 UEs, q2 increasing, QoS ON

		QoS OFF				
q2 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
5	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
10	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
15	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
20	**Throughput_rq0**	0.472000	0.004000	0.002479	0.469521	0.474479
25	**Throughput_rq0**	0.471000	0.003000	0.001859	0.469141	0.472859
30	**Throughput_rq0**	0.470000	0.000000	0.000000	0.470000	0.470000
35	**Throughput_rq0**	0.469000	0.003000	0.001859	0.467141	0.470859
40	**Throughput_rq0**	0.468000	0.004000	0.002479	0.465521	0.470479
45	**Throughput_rq0**	0.468000	0.004000	0.002479	0.465521	0.470479
50	**Throughput_rq0**	0.465000	0.005000	0.003099	0.461901	0.468099
1	**Throughput_rq2**	0.898000	0.160860	0.099700	0.798300	0.997700
5	**Throughput_rq2**	0.893000	0.159000	0.098547	0.794453	0.991547
10	**Throughput_rq2**	0.965000	0.146168	0.090594	0.874406	1.055594
15	**Throughput_rq2**	0.839000	0.146932	0.091068	0.747932	0.930068
20	**Throughput_rq2**	0.703000	0.073627	0.045634	0.657366	0.748634
25	**Throughput_rq2**	0.668000	0.090089	0.055837	0.612163	0.723837
30	**Throughput_rq2**	0.492000	0.054553	0.033812	0.458188	0.525812
35	**Throughput_rq2**	0.446000	0.054626	0.033857	0.412143	0.479857
40	**Throughput_rq2**	0.399000	0.027368	0.016962	0.382038	0.415962
45	**Throughput_rq2**	0.374000	0.027368	0.016962	0.357038	0.390962
50	**Throughput_rq2**	0.342000	0.039446	0.024449	0.317551	0.366449
1	**Delay q0**	0.007722	0.000145	0.000090	0.007633	0.007812
5	**Delay q0**	0.011046	0.000393	0.000244	0.010803	0.011290
10	**Delay q0**	0.018207	0.001325	0.000821	0.017386	0.019028
15	**Delay q0**	0.037202	0.002875	0.001782	0.035420	0.038983
20	**Delay q0**	0.073372	0.004390	0.002721	0.070652	0.076093
25	**Delay q0**	0.135392	0.007946	0.004925	0.130467	0.140317
30	**Delay q0**	0.202308	0.008869	0.005497	0.196811	0.207804
35	**Delay q0**	0.280303	0.010451	0.006477	0.273826	0.286780
40	**Delay q0**	0.349385	0.019026	0.011792	0.337593	0.361177
45	**Delay q0**	0.438747	0.021664	0.013427	0.425319	0.452174
50	**Delay q0**	0.572514	0.031593	0.019581	0.552933	0.592095
1	**Delay q2**	0.004296	0.000016	0.000010	0.004286	0.004306
5	**Delay q2**	0.004290	0.000014	0.000009	0.004281	0.004298
10	**Delay q2**	0.004301	0.000012	0.000007	0.004294	0.004308
15	**Delay q2**	0.0043006	0.0000177	0.000011	0.004290	0.004312
20	**Delay q2**	0.0042817	0.0000098	0.000006	0.004276	0.004288
25	**Delay q2**	0.0042731	0.0000248	0.000015	0.004258	0.004289
30	**Delay q2**	**0.004243**	**0.000015**	0.000009	0.004233	0.004252
35	**Delay q2**	**0.004225**	**0.000017**	0.000010	0.004214	0.004235
40	**Delay q2**	0.004218	0.000019	0.000012	0.004206	0.004229
45	**Delay q2**	0.004201	0.000024	0.000015	0.004186	0.004216
50	**Delay q2**	0.004192	0.000035	0.000022	0.004170	0.004213

Appendix Table 5 : q0 5 UEs, q2 increasing, QoS OFF

		QoS ON				
q3 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
2	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
3	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
4	Throughput_rq0	0.472000	0.006000	0.003719	0.468281	0.475719
5	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
6	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
7	Throughput_rq0	0.473000	0.004583	0.002840	0.470160	0.475840
8	Throughput_rq0	0.472000	0.006000	0.003719	0.468281	0.475719
10	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
1	Throughput_rq3	15.088000	0.025219	0.015631	15.072369	15.103631
2	Throughput_rq3	7.570000	0.008944	0.005544	7.564456	7.575544
3	Throughput_rq3	5.060000	0.035496	0.022001	5.037999	5.082001
4	Throughput_rq3	3.913000	0.074572	0.046219	3.866781	3.959219
5	Throughput_rq3	3.145000	0.151872	0.094129	3.050871	3.239129
6	Throughput_rq3	2.619000	0.075160	0.046584	2.572416	2.665584
7	Throughput_rq3	2.288000	0.006000	0.003719	2.284281	2.291719
8	Throughput_rq3	2.040000	0.000000	0.000000	2.040000	2.040000
10	Throughput_rq3	1.536000	0.016852	0.010445	1.525555	1.546445
1	Delay q0	0.009211	0.000036	0.000022	0.009188	0.009233
2	Delay q0	0.009230	0.000027	0.000017	0.009213	0.009246
3	Delay q0	0.009215	0.000027	0.000017	0.009198	0.009232
4	Delay q0	0.009208	2.537E-05	0.000016	0.009192	0.009224
5	Delay q0	0.009218	0.000021	0.000013	0.009205	0.009231
6	Delay q0	0.009213	0.000016	0.000010	0.009203	0.009223
7	Delay q0	0.009224	0.000033	0.000021	0.009203	0.009244
8	Delay q0	0.009225	0.000023	0.000014	0.009211	0.009240
10	Delay q0	0.009209	0.000024	0.000015	0.009194	0.009224
1	Delay q3	0.115463	0.000197	0.000122	0.115341	0.115585
2	Delay q3	0.235877	0.000287	0.000178	0.235700	0.236055
3	Delay q3	0.354286	0.002034	0.001261	0.353025	0.355546
4	Delay q3	0.458440	0.009033	0.005599	0.452841	0.464038
5	Delay q3	0.546711	0.006037	0.003742	0.542970	0.550453
6	Delay q3	0.636818	0.004327	0.002682	0.634136	0.639500
7	Delay q3	0.7222948	0.0023242	0.001441	0.720854	0.723735
8	Delay q3	0.7998405	0.0011986	0.000743	0.799098	0.800583
9	Delay q3	0.8918513	0.0076076	0.004715	0.887136	0.896566
10	Delay q3	**0.970602**	**0.001507**	0.000934	0.969668	0.971536

Appendix Table 6 : q0 5UEs, q3 increasing, QoS ON

		QoS OFF				
q3 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_r q0	0.472000	0.004000	0.002479	0.469521	0.474479
2	Throughput_r q0	0.470000	0.000000	0.000000	0.470000	0.470000
3	Throughput_r q0	0.469000	0.005385	0.003338	0.465662	0.472338
4	Throughput_r q0	0.464000	0.004899	0.003036	0.460964	0.467036
5	Throughput_r q0	0.461000	0.003000	0.001859	0.459141	0.462859
6	Throughput_r q0	0.461000	0.003000	0.001859	0.459141	0.462859
7	Throughput_r q0	0.460000	0.000000	0.000000	0.460000	0.460000
8	Throughput_r q0	0.460000	0.000000	0.000000	0.460000	0.460000
10	Throughput_r q0	0.455000	0.005000	0.003099	0.451901	0.458099
1	Throughput_r q3	15.107000	0.026096	0.016174	15.090826	15.123174
2	Throughput_r q3	7.582000	0.006000	0.003719	7.578281	7.585719
3	Throughput_r q3	5.088000	0.006000	0.003719	5.084281	5.091719
4	Throughput_r q3	3.896000	0.012000	0.007438	3.888562	3.903438
5	Throughput_r q3	3.240000	0.120333	0.074582	3.165418	3.314582
6	Throughput_r q3	2.600000	0.000000	0.000000	2.600000	2.600000
7	Throughput_r q3	2.290000	0.000000	0.000000	2.290000	2.290000
8	Throughput_r q3	2.040000	0.000000	0.000000	2.040000	2.040000
10	Throughput_r q3	1.709000	0.017578	0.010895	1.698105	1.719895
1	Delay q0	0.113211	0.000247	0.000153	0.113057	0.113364
2	Delay q0	0.234470	0.000309	0.000191	0.234278	0.234661
3	Delay q0	0.355032	0.000659	0.000409	0.354624	0.355441
4	Delay q0	0.464624	0.001180	0.000731	0.463893	0.465356
5	Delay q0	0.557356	0.001679	0.001041	0.556315	0.558396
6	Delay q0	0.651735	0.001751	0.001085	0.650650	0.652820
7	Delay q0	0.742124	0.002205	0.001367	0.740757	0.743491
8	Delay q0	0.829047	0.001482	0.000918	0.828128	0.829965
10	Delay q0	0.993184	0.002174	0.001348	0.991837	0.994532
1	Delay q3	0.115345	0.000210	0.000130	0.115214	0.115475
2	Delay q3	0.235217	0.000302	0.000187	0.235030	0.235404
3	Delay q3	0.353455	0.000619	0.000384	0.353072	0.353839
4	Delay q3	0.458659	0.000597	0.000370	0.458289	0.459029
5	Delay q3	0.542189	0.006774	0.004198	0.537991	0.546388
6	Delay q3	0.636935	0.001072	0.000664	0.636271	0.637600
7	Delay q3	0.7200339	0.0014000	0.000868	0.719166	0.720902
8	Delay q3	0.7973371	0.0011753	0.000728	0.796609	0.798066
10	Delay q3	**0.937607**	**0.001077**	0.000668	0.936940	0.938275

Appendix Table 7: q0 5UEs, q3 increasing, QoS OFF

		QoS Enabled and Disabled				
UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q1**	1.424000	0.008000	0.004958	1.419042	1.428958
5	**Throughput_r q1**	1.423000	0.007810	0.004841	1.418159	1.427841
10	**Throughput_r q1**	1.417000	0.007810	0.004841	1.412159	1.421841
15	**Throughput_r q1**	1.416000	0.008000	0.004958	1.411042	1.420958
16	**Throughput_r q1**	1.386000	0.004899	0.003036	1.382964	1.389036
17	**Throughput_r q1**	1.306000	0.009165	0.005681	1.300319	1.311681
18	**Throughput_r q1**	1.231000	0.007000	0.004339	1.226661	1.235339
19	**Throughput_r q1**	1.174000	0.009165	0.005681	1.168319	1.179681
20	**Throughput_r q1**	1.108000	0.006000	0.003719	1.104281	1.111719
1	**Delay q1**	0.007008	0.000000	0.000000	0.007008	0.007008
5	**Delay q1**	0.007139	0.000006	0.000004	0.007135	0.007143
10	**Delay q1**	0.007457	0.000009	0.000006	0.007451	0.007463
15	**Delay q1**	0.008985	0.000081	0.000050	0.008936	0.009035
16	**Delay q1**	0.349130	0.034333	0.021279	0.327851	0.370409
17	**Delay q1**	1.198809	0.030918	0.019163	1.179646	1.217972
18	**Delay q1**	1.954065	0.033357	0.020675	1.933390	1.974740
19	**Delay q1**	2.632495	0.028572	0.017709	2.614786	2.650204
20	**Delay q1**	3.242274	0.025974	0.016099	3.226175	3.258373

Appendix Table 8: q1 only

		QoS ON				
q0 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
5	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
10	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
11	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
12	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
13	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
14	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
15	Throughput_rq0	0.470000	0.004472	0.002772	0.467228	0.472772
1	Throughput_rq1	1.418000	0.011662	0.007228	1.410772	1.425228
5	Throughput_rq1	1.426000	0.014283	0.008852	1.417148	1.434852
10	Throughput_rq1	1.419000	0.010440	0.006471	1.412529	1.425471
11	Throughput_rq1	1.313000	0.013454	0.008338	1.304662	1.321338
12	Throughput_rq1	1.028000	0.008718	0.005403	1.022597	1.033403
13	Throughput_rq1	0.745000	0.005000	0.003099	0.741901	0.748099
14	Throughput_rq1	0.464000	0.008000	0.004958	0.459042	0.468958
15	Throughput_rq1	0.182000	0.006000	0.003719	0.178281	0.185719
1	Delay q0	0.007145	0.000004	0.000002	0.007142	0.007147
5	Delay q0	0.007313	0.000006	0.000004	0.007309	0.007316
10	Delay q0	0.007798	0.000015	0.000009	0.007789	0.007808
11	Delay q0	0.007985	0.000014	0.000009	0.007977	0.007994
12	Delay q0	0.008126	0.000016	0.000010	0.008117	0.008136
13	Delay q0	0.008309	0.000013	0.000008	0.008301	0.008317
14	Delay q0	0.008621	0.000026	0.000016	0.008605	0.008637
15	Delay q0	0.009389	0.000036	0.000022	0.009367	0.009411
1	Delay q1	0.007189	0.000004	0.000003	0.007186	0.007192
5	Delay q1	0.007610	0.000014	0.000009	0.007601	0.007619
10	Delay q1	0.011295	0.000181	0.000112	0.011183	0.011407
11	Delay q1	1.084599	0.072874	0.045167	1.039432	1.129766
12	Delay q1	4.067464	0.126067	0.078136	3.989328	4.145600
13	Delay q1	7.020248	0.142134	0.088094	6.932154	7.108342
14	Delay q1	9.982780	0.195980	0.121467	9.861313	10.104247
15	Delay q1	12.859250	0.444661	0.275599	12.583651	13.134849

Appendix Table 9 : q1 5 UEs, q0 increasing, QoS ON

		QoS OFF				
q0 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_rq0	0.472000	0.004000	0.002479	0.469521	0.474479
5	Throughput_rq0	0.471000	0.003000	0.001859	0.469141	0.472859
10	Throughput_rq0	0.474000	0.004899	0.003036	0.470964	0.477036
11	Throughput_rq0	0.464000	0.004899	0.003036	0.460964	0.467036
12	Throughput_rq0	0.434000	0.004899	0.003036	0.430964	0.437036
13	Throughput_rq0	0.410000	0.000000	0.000000	0.410000	0.410000
14	Throughput_rq0	0.390000	0.000000	0.000000	0.390000	0.390000
15	Throughput_rq0	0.368000	0.004000	0.002479	0.365521	0.370479
1	Throughput_rq1	1.418000	0.011662	0.007228	1.410772	1.425228
5	Throughput_rq1	1.426000	0.014283	0.008852	1.417148	1.434852
10	Throughput_rq1	1.419000	0.010440	0.006471	1.412529	1.425471
11	Throughput_rq1	1.383000	0.011874	0.007360	1.375640	1.390360
12	Throughput_rq1	1.306000	0.006633	0.004111	1.301889	1.310111
13	Throughput_rq1	1.232000	0.006000	0.003719	1.228281	1.235719
14	Throughput_rq1	1.167000	0.009000	0.005578	1.161422	1.172578
15	Throughput_rq1	1.113000	0.007810	0.004841	1.108159	1.117841
1	Delay q0	0.007185	0.000006	0.000004	0.007182	0.007189
5	Delay q0	0.007457	0.000012	0.000007	0.007450	0.007464
10	Delay q0	0.008962	0.000058	0.000036	0.008927	0.008998
11	Delay q0	0.344138	0.021549	0.013356	0.330782	0.357493
12	Delay q0	1.199214	0.031968	0.019814	1.179400	1.219028
13	Delay q0	1.952679	0.019672	0.012193	1.940486	1.964872
14	Delay q0	2.632313	0.034385	0.021312	2.611001	2.653625
15	Delay q0	3.230338	0.036337	0.022522	3.207816	3.252860
1	Delay q1	0.007181	0.000004	0.000003	0.007178	0.007183
5	Delay q1	0.007464	0.000012	0.000008	0.007456	0.007472
10	Delay q1	0.008971	0.000057	0.000035	0.008935	0.009006
11	Delay q1	0.344633	0.021795	0.013508	0.331125	0.358141
12	Delay q1	1.201534	0.031586	0.019577	1.181957	1.221111
13	Delay q1	1.954533	0.026007	0.016119	1.938414	1.970652
14	Delay q1	2.633586	0.029530	0.018303	2.615283	2.651889
15	Delay q1	3.236889	0.027148	0.016826	3.220063	3.253715

Appendix Table 10: q1 5 UEs, q0 increasing QoS OFF

		QoS ON				
q3 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q1**	1.419000	0.007000	0.004339	1.414661	1.423339
2	**Throughput_r q1**	1.416000	0.008000	0.004958	1.411042	1.420958
3	**Throughput_r q1**	1.416000	0.008000	0.004958	1.411042	1.420958
4	**Throughput_r q1**	1.423000	0.006403	0.003969	1.419031	1.426969
5	**Throughput_r q1**	1.426000	0.009165	0.005681	1.420319	1.431681
6	**Throughput_r q1**	1.416000	0.012000	0.007438	1.408562	1.423438
7	**Throughput_r q1**	1.419000	0.005385	0.003338	1.415662	1.422338
8	**Throughput_r q1**	1.421000	0.009434	0.005847	1.415153	1.426847
9	**Throughput_r q1**	1.423000	0.007810	0.004841	1.418159	1.427841
10	**Throughput_r q1**	1.422000	0.007483	0.004638	1.417362	1.426638
1	**Throughput_r q3**	15.094000	0.018547	0.011495	15.082505	15.105495
2	**Throughput_r q3**	7.565000	0.015000	0.009297	7.555703	7.574297
3	**Throughput_r q3**	5.049000	0.021656	0.013423	5.035577	5.062423
4	**Throughput_r q3**	3.931000	0.037269	0.023099	3.907901	3.954099
5	**Throughput_r q3**	3.059000	0.077775	0.048205	3.010795	3.107205
6	**Throughput_r q3**	2.600000	0.000000	0.000000	2.600000	2.600000
7	**Throughput_r q3**	2.290000	0.000000	0.000000	2.290000	2.290000
8	**Throughput_r q3**	2.040000	0.000000	0.000000	2.040000	2.040000
9	**Throughput_r q3**	1.728000	0.041183	0.025525	1.702475	1.753525
10	**Throughput_r q3**	1.528000	0.009798	0.006073	1.521927	1.534073
1	**Delay q1**	0.009219	0.000019	0.000012	0.009207	0.009230
2	**Delay q1**	0.009202	0.000018	0.000011	0.009190	0.009213
3	**Delay q1**	0.009221	0.000029	0.000018	0.009203	0.009239
4	**Delay q1**	0.009222	0.000024	0.000015	0.009207	0.009237
5	**Delay q1**	0.009241	0.000026	0.000016	0.009225	0.009256
6	**Delay q1**	0.009224	0.000028	0.000017	0.009207	0.009242
7	**Delay q1**	0.009215	0.000033	0.000021	0.009194	0.009235
8	**Delay q1**	0.009225	0.000029	0.000018	0.009207	0.009243
9	**Delay q1**	0.009219	0.000021	0.000013	0.009206	0.009231
10	**Delay q1**	0.009213	0.000028	0.000017	0.009196	0.009231
1	**Delay q3**	0.115590	0.000154	0.000096	0.115494	0.115686
2	**Delay q3**	0.236119	0.000384	0.000238	0.235881	0.236357
3	**Delay q3**	0.3549265	0.001172028	0.000726	0.354200	0.355653
4	**Delay q3**	0.4563237	0.005617515	0.003482	0.452842	0.459805
5	**Delay q3**	0.5504047	0.003113363	0.001930	0.548475	0.552334
6	**Delay q3**	0.6371985	0.00190066	0.001178	0.636020	0.638377
7	**Delay q3**	0.7215502	0.001246445	0.000773	0.720778	0.722323
8	**Delay q3**	0.8012263	0.001991955	0.001235	0.799992	0.802461
9	**Delay q3**	0.8941130	0.006177218	0.003829	0.890284	0.897942
10	**Delay q3**	0.9699808	0.001166222	0.000723	0.969258	0.970704

Appendix Table 11 : q1 5UEs, q3 increasing, QoS OFF

		QoS OFF				
q3 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q1**	1.414000	0.009165	0.005681	1.408319	1.419681
2	**Throughput_r q1**	1.411000	0.008307	0.005148	1.405852	1.416148
3	**Throughput_r q1**	1.402000	0.007483	0.004638	1.397362	1.406638
4	**Throughput_r q1**	1.397000	0.007810	0.004841	1.392159	1.401841
5	**Throughput_r q1**	1.392000	0.007483	0.004638	1.387362	1.396638
6	**Throughput_r q1**	1.384000	0.008000	0.004958	1.379042	1.388958
7	**Throughput_r q1**	1.381000	0.008307	0.005148	1.375852	1.386148
8	**Throughput_r q1**	1.373000	0.004583	0.002840	1.370160	1.375840
9	**Throughput_r q1**	1.366000	0.006633	0.004111	1.361889	1.370111
10	**Throughput_r q1**	1.362000	0.008718	0.005403	1.356597	1.367403
1	**Throughput_r q3**	15.107000	0.018466	0.011445	15.095555	15.118445
2	**Throughput_r q3**	7.582000	0.006000	0.003719	7.578281	7.585719
3	**Throughput_r q3**	5.090000	0.000000	0.000000	5.090000	5.090000
4	**Throughput_r q3**	3.891000	0.018138	0.011242	3.879758	3.902242
5	**Throughput_r q3**	3.247000	0.111270	0.068965	3.178035	3.315965
6	**Throughput_r q3**	2.600000	0.000000	0.000000	2.600000	2.600000
7	**Throughput_r q3**	2.290000	0.000000	0.000000	2.290000	2.290000
8	**Throughput_r q3**	2.040000	0.000000	0.000000	2.040000	2.040000
9	**Throughput_r q3**	1.850000	0.000000	0.000000	1.850000	1.850000
10	**Throughput_r q3**	1.691000	0.025865	0.016031	1.674969	1.707031
1	**Delay q1**	0.113394	0.000145	0.000090	0.113305	0.113484
2	**Delay q1**	0.234648	0.000286	0.000177	0.234471	0.234825
3	**Delay q1**	0.355264	0.000462	0.000286	0.354978	0.355550
4	**Delay q1**	0.465116	0.001803	0.001117	0.463998	0.466233
5	**Delay q1**	0.557848	0.001734	0.001075	0.556774	0.558923
6	**Delay q1**	0.652021	0.001109	0.000688	0.651334	0.652709
7	**Delay q1**	0.742281	0.001091	0.000676	0.741605	0.742957
8	**Delay q1**	0.829682	0.001499	0.000929	0.828753	0.830611
9	**Delay q1**	0.913042	0.002279	0.001412	0.911629	0.914454
10	**Delay q1**	0.991455	0.002748	0.001703	0.989752	0.993158
1	**Delay q3**	0.115502	0.000144	0.000089	0.115413	0.115592
2	**Delay q3**	0.235435	0.000283	0.000175	0.235259	0.235610
3	**Delay q3**	0.3536651	0.000413611	0.000256	0.353409	0.353921
4	**Delay q3**	0.4593412	0.001867069	0.001157	0.458184	0.460498
5	**Delay q3**	0.5418491	0.007774024	0.004818	0.537031	0.546667
6	**Delay q3**	0.6370766	0.000679172	0.000421	0.636656	0.637498
7	**Delay q3**	0.7205419	0.000676476	0.000419	0.720123	0.720961
8	**Delay q3**	0.7977926	0.000753914	0.000467	0.797325	0.798260
9	**Delay q3**	0.8701875	0.000894186	0.000554	0.869633	0.870742
10	**Delay q3**	0.9379900	0.000839898	0.000521	0.937469	0.938511

Appendix Table 12 : q1 5UEs, q3 increasing, QoS OFF

		QoS ON				
q2 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q1**	1.422000	0.007483	0.004638	1.417362	1.426638
5	**Throughput_r q1**	1.419000	0.008307	0.005148	1.413852	1.424148
10	**Throughput_r q1**	1.418000	0.008718	0.005403	1.412597	1.423403
15	**Throughput_r q1**	1.422000	0.010770	0.006675	1.415325	1.428675
20	**Throughput_r q1**	1.419000	0.010440	0.006471	1.412529	1.425471
25	**Throughput_r q1**	1.416000	0.011136	0.006902	1.409098	1.422902
30	**Throughput_r q1**	1.417000	0.009000	0.005578	1.411422	1.422578
35	**Throughput_r q1**	1.422000	0.006000	0.003719	1.418281	1.425719
40	**Throughput_r q1**	1.418000	0.008718	0.005403	1.412597	1.423403
45	**Throughput_r q1**	1.425000	0.006708	0.004158	1.420842	1.429158
100	**Throughput_r q1**	1.420000	0.011832	0.007334	1.412666	1.427334
1	**Throughput_r q2**	0.891000	0.135753	0.084139	0.806861	0.975139
5	**Throughput_r q2**	0.884000	0.089017	0.055172	0.828828	0.939172
10	**Throughput_r q2**	0.914000	0.148068	0.091772	0.822228	1.005772
15	**Throughput_r q2**	0.799000	0.084077	0.052111	0.746889	0.851111
20	**Throughput_r q2**	0.741000	0.114145	0.070746	0.670254	0.811746
25	**Throughput_r q2**	0.638000	0.081707	0.050641	0.587359	0.688641
30	**Throughput_r q2**	0.498000	0.047074	0.029176	0.468824	0.527176
35	**Throughput_r q2**	0.419000	0.059405	0.036819	0.382181	0.455819
40	**Throughput_r q2**	0.397000	0.059000	0.036568	0.360432	0.433568
45	**Throughput_r q2**	0.366000	0.038262	0.023715	0.342285	0.389715
100	**Throughput_r q2**	0.160000	0.021909	0.013579	0.146421	0.173579
1	**Delay q1**	0.007206	0.000014	0.000008	0.007197	0.007214
5	**Delay q1**	0.007463	0.000027	0.000017	0.007446	0.007479
10	**Delay q1**	0.007768	0.000023	0.000014	0.007754	0.007783
15	**Delay q1**	0.008038	0.000031	0.000019	0.008019	0.008057
20	**Delay q1**	0.008207	0.000028	0.000017	0.008190	0.008224
25	**Delay q1**	0.008265	0.000017	0.000010	0.008255	0.008275
30	**Delay q1**	0.008279	0.000011	0.000007	0.008272	0.008286
35	**Delay q1**	0.008275	0.000014	0.000009	0.008266	0.008283
40	**Delay q1**	0.008277	0.000013	0.000008	0.008269	0.008285
45	**Delay q1**	0.008274	0.000012	0.000007	0.008267	0.008282
100	**Delay q1**	0.0082764	1.70746E-05	0.000011	0.008266	0.008287
1	**Delay q2**	0.0041216	1.10568E-05	0.000007	0.004115	0.004128
5	**Delay q2**	0.0041233	7.3483E-06	0.000005	0.004119	0.004128
10	**Delay q2**	0.0041265	1.29051E-05	0.000008	0.004118	0.004134
15	**Delay q2**	0.0041191	8.0946E-06	0.000005	0.004114	0.004124
20	**Delay q2**	0.0041133	1.54031E-05	0.000010	0.004104	0.004123
25	**Delay q2**	0.0041001	1.24133E-05	0.000008	0.004092	0.004108
30	**Delay q2**	0.0040735	1.11276E-05	0.000007	0.004067	0.004080
35	**Delay q2**	0.0040521	2.03915E-05	0.000013	0.004040	0.004065
40	**Delay q2**	0.0040453	2.12054E-05	0.000013	0.004032	0.004058
45	**Delay q2**	0.0040299	1.94009E-05	0.000012	0.004018	0.004042
100	**Delay q2**	0.0038433	3.85918E-05	0.000024	0.003819	0.003867

Appendix Table 13: q1 5 UEs, q2 increasing, QoS ON

q2 UE		QoS OFF average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q1**	1.422000	0.007483	0.004638	1.417362	1.426638
5	**Throughput_r q1**	1.418000	0.008718	0.005403	1.412597	1.423403
10	**Throughput_r q1**	1.417000	0.009000	0.005578	1.411422	1.422578
15	**Throughput_r q1**	1.424000	0.008000	0.004958	1.419042	1.428958
20	**Throughput_r q1**	1.415000	0.011180	0.006930	1.408070	1.421930
25	**Throughput_r q1**	1.412000	0.014000	0.008677	1.403323	1.420677
30	**Throughput_r q1**	1.406000	0.008000	0.004958	1.401042	1.410958
35	**Throughput_r q1**	1.409000	0.005385	0.003338	1.405662	1.412338
40	**Throughput_r q1**	1.404000	0.009165	0.005681	1.398319	1.409681
45	**Throughput_r q1**	1.403000	0.007810	0.004841	1.398159	1.407841
100	**Throughput_r q1**	1.294000	0.023324	0.014456	1.279544	1.308456
1	**Throughput_r q2**	0.901000	0.147949	0.091698	0.809302	0.992698
5	**Throughput_r q2**	0.893000	0.098392	0.060983	0.832017	0.953983
10	**Throughput_r q2**	0.926000	0.155962	0.096664	0.829336	1.022664
15	**Throughput_r q2**	0.812000	0.067941	0.042110	0.769890	0.854110
20	**Throughput_r q2**	0.715000	0.113864	0.070572	0.644428	0.785572
25	**Throughput_r q2**	0.632000	0.089978	0.055768	0.576232	0.687768
30	**Throughput_r q2**	0.545000	0.075928	0.047060	0.497940	0.592060
35	**Throughput_r q2**	0.445000	0.064692	0.040096	0.404904	0.485096
40	**Throughput_r q2**	0.424000	0.040299	0.024977	0.399023	0.448977
45	**Throughput_r q2**	0.372000	0.048744	0.030211	0.341789	0.402211
100	**Throughput_r q2**	0.150000	0.018974	0.011760	0.138240	0.161760
1	**Delay q1**	0.007704	0.000126	0.000078	0.007626	0.007782
5	**Delay q1**	0.010635	0.000378	0.000234	0.010401	0.010869
10	**Delay q1**	0.018505	0.001148	0.000711	0.017794	0.019216
15	**Delay q1**	0.035530	0.002374	0.001472	0.034058	0.037001
20	**Delay q1**	0.073091	0.007059	0.004375	0.068716	0.077466
25	**Delay q1**	0.131977	0.008101	0.005021	0.126956	0.136998
30	**Delay q1**	0.201358	0.012355	0.007658	0.193700	0.209015
35	**Delay q1**	0.276838	0.006098	0.003780	0.273059	0.280618
40	**Delay q1**	0.355444	0.013758	0.008527	0.346917	0.363971
45	**Delay q1**	0.450133	0.028156	0.017451	0.432681	0.467584
100	**Delay q1**	1.9303840	0.150255027	0.093127	1.837257	2.023511
1	**Delay q2**	0.0041222	1.16021E-05	0.000007	0.004115	0.004129
5	**Delay q2**	0.0041238	7.95623E-06	0.000005	0.004119	0.004129
10	**Delay q2**	0.0041270	1.33066E-05	0.000008	0.004119	0.004135
15	**Delay q2**	0.0041195	6.947E-06	0.000004	0.004115	0.004124
20	**Delay q2**	0.0041083	1.40316E-05	0.000009	0.004100	0.004117
25	**Delay q2**	0.0040979	1.31736E-05	0.000008	0.004090	0.004106
30	**Delay q2**	0.0040810	1.59024E-05	0.000010	0.004071	0.004091
35	**Delay q2**	0.0040612	2.16027E-05	0.000013	0.004048	0.004075
40	**Delay q2**	0.0040546	1.24013E-05	0.000008	0.004047	0.004062
45	**Delay q2**	0.0040292	2.4765E-05	0.000015	0.004014	0.004045
100	**Delay q2**	0.0038318	3.15205E-05	0.000020	0.003812	0.003851

Appendix Table 14: q1 5UEs, q2 increasing, QoS OFF

		QoS Enabled and Disabled				
UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_rq2	0.859000	0.127941	0.079297	0.779703	0.938297
10	Throughput_rq2	0.959000	0.121610	0.075373	0.883627	1.034373
20	Throughput_rq2	0.899000	0.149763	0.092822	0.806178	0.991822
30	Throughput_rq2	0.757000	0.084267	0.052229	0.704771	0.809229
40	Throughput_rq2	0.517000	0.061164	0.037909	0.479091	0.554909
50	Throughput_rq2	0.471000	0.097821	0.060629	0.410371	0.531629
60	Throughput_rq2	0.424000	0.056071	0.034753	0.389247	0.458753
70	Throughput_rq2	0.357000	0.048795	0.030243	0.326757	0.387243
80	Throughput_rq2	0.304000	0.081633	0.050596	0.253404	0.354596
90	Throughput_rq2	0.273000	0.025710	0.015935	0.257065	0.288935
1	Delay q2	0.004118	0.000013	0.000008	0.004110	0.004127
10	Delay q2	0.004131	0.000008	0.000005	0.004126	0.004136
20	Delay q2	0.004129	0.000015	0.000009	0.004120	0.004138
30	Delay q2	0.004118	0.000009	0.000006	0.004113	0.004124
40	Delay q2	0.004079	0.000016	0.000010	0.004070	0.004089
50	Delay q2	0.004069	0.000026	0.000016	0.004052	0.004085
60	Delay q2	0.004064	0.000022	0.000014	0.004050	0.004078
70	Delay q2	0.004033	0.000024	0.000015	0.004018	0.004048
80	Delay q2	0.0039956	4.08433E-05	0.000025	0.003970	0.004021
90	Delay q2	0.0039828	2.38251E-05	0.000015	0.003968	0.003998

Appendix Table 15: q2 only

		QoS ON				
q0 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q0**	0.471000	0.003000	0.001859	0.469141	0.472859
5	**Throughput_r q0**	0.474000	0.004899	0.003036	0.470964	0.477036
9	**Throughput_r q0**	0.473000	0.004583	0.002840	0.470160	0.475840
10	**Throughput_r q0**	0.474000	0.004899	0.003036	0.470964	0.477036
11	**Throughput_r q0**	0.471000	0.003000	0.001859	0.469141	0.472859
12	**Throughput_r q0**	0.473000	0.004583	0.002840	0.470160	0.475840
13	**Throughput_r q0**	0.471000	0.003000	0.001859	0.469141	0.472859
14	**Throughput_r q0**	0.476000	0.004899	0.003036	0.472964	0.479036
15	**Throughput_r q0**	0.471000	0.003000	0.001859	0.469141	0.472859
1	**Throughput_r q2**	0.847000	0.118832	0.073651	0.773349	0.920651
5	**Throughput_r q2**	0.874000	0.156601	0.097061	0.776939	0.971061
9	**Throughput_r q2**	0.889000	0.081664	0.050615	0.838385	0.939615
10	**Throughput_r q2**	0.795000	0.103755	0.064307	0.730693	0.859307
11	**Throughput_r q2**	0.785000	0.097596	0.060490	0.724510	0.845490
12	**Throughput_r q2**	0.772000	0.097242	0.060270	0.711730	0.832270
13	**Throughput_r q2**	0.656000	0.129089	0.080009	0.575991	0.736009
14	**Throughput_r q2**	0.469000	0.062040	0.038452	0.430548	0.507452
15	**Throughput_r q2**	0.192000	0.025219	0.015631	0.176369	0.207631
1	**Delay q0**	0.007241	0.000013	0.000008	0.007233	0.007249
5	**Delay q0**	0.007468	0.000024	0.000015	0.007453	0.007483
9	**Delay q0**	0.007863	0.000029	0.000018	0.007845	0.007881
10	**Delay q0**	0.008021	0.000032	0.000020	0.008001	0.008041
11	**Delay q0**	0.008237	0.000041	0.000025	0.008212	0.008262
12	**Delay q0**	0.008501	0.000043	0.000027	0.008474	0.008527
13	**Delay q0**	0.008844	0.000042	0.000026	0.008818	0.008870
14	**Delay q0**	0.009280	0.000053	0.000033	0.009247	0.009313
15	**Delay q0**	0.010094	0.000036	0.000022	0.010071	0.010116
1	**Delay q2**	0.004144	0.000008	0.000005	0.004139	0.004149
5	**Delay q2**	0.004290	0.000022	0.000014	0.004276	0.004304
9	**Delay q2**	0.004569	0.000025	0.000015	0.004538	0.004568
10	**Delay q2**	0.004671	0.000028	0.000017	0.004635	0.004669
11	**Delay q2**	0.004823	0.000039	0.000024	0.004759	0.004807
12	**Delay q2**	0.0050085	3.4768E-05	0.000022	0.004926	0.004969
13	**Delay q2**	0.0052595	4.30687E-05	0.000027	0.005117	0.005170
14	**Delay q2**	0.0055461	6.9521E-05	0.000043	0.005378	0.005464
15	**Delay q2**	0.0066331	0.000168725	0.000105	0.005841	0.006050

Appendix Table 16 : q2 5 UEs, q0 increasing, QoS ON

		QoS OFF				
q0 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q0**	0.471000	0.003000	0.001859	0.472859	0.472859
5	**Throughput_r q0**	0.474000	0.004899	0.003036	0.477036	0.477036
9	**Throughput_r q0**	0.473000	0.004583	0.002840	0.475840	0.475840
10	**Throughput_r q0**	0.472000	0.004000	0.002479	0.474479	0.474479
11	**Throughput_r q0**	0.473000	0.004583	0.002840	0.475840	0.475840
12	**Throughput_r q0**	0.473000	0.004583	0.002840	0.475840	0.475840
13	**Throughput_r q0**	0.471000	0.003000	0.001859	0.472859	0.472859
14	**Throughput_r q0**	0.470000	0.000000	0.000000	0.470000	0.470000
15	**Throughput_r q0**	0.458000	0.004000	0.002479	0.460479	0.460479
1	**Throughput_r q2**	0.847000	0.118832	0.073651	0.920651	0.920651
5	**Throughput_r q2**	0.893000	0.159000	0.098547	0.991547	0.991547
9	**Throughput_r q2**	0.859000	0.110132	0.068259	0.927259	0.927259
10	**Throughput_r q2**	0.867000	0.123454	0.076516	0.943516	0.943516
11	**Throughput_r q2**	0.845000	0.123875	0.076777	0.921777	0.921777
12	**Throughput_r q2**	0.854000	0.099418	0.061619	0.915619	0.915619
13	**Throughput_r q2**	0.704000	0.110200	0.068301	0.772301	0.772301
14	**Throughput_r q2**	0.534000	0.061677	0.038227	0.572227	0.572227
15	**Throughput_r q2**	0.325000	0.050050	0.031021	0.356021	0.356021
1	**Delay q0**	0.008984	0.000184	0.000114	0.009097	0.009097
5	**Delay q0**	0.011046	0.000393	0.000244	0.011290	0.011290
9	**Delay q0**	0.016039	0.000840	0.000521	0.016560	0.016560
10	**Delay q0**	0.020127	0.001641	0.001017	0.021144	0.021144
11	**Delay q0**	0.026171	0.001822	0.001130	0.027301	0.027301
12	**Delay q0**	0.041991	0.007116	0.004410	0.046402	0.046402
13	**Delay q0**	0.076397	0.015649	0.009699	0.086096	0.086096
14	**Delay q0**	0.205321	0.026348	0.016330	0.221651	0.221651
15	**Delay q0**	0.602664	0.077986	0.048335	0.651000	0.651000
1	**Delay q2**	0.004144	0.000009	0.000006	0.004150	0.004150
5	**Delay q2**	0.004290	0.000014	0.000009	0.004298	0.004298
9	**Delay q2**	0.004553	0.000019	0.000012	0.004581	0.004581
10	**Delay q2**	0.004652	0.000031	0.000020	0.004690	0.004690
11	**Delay q2**	0.004783	0.000041	0.000026	0.004848	0.004848
12	**Delay q2**	0.0049477	2.9685E-05	0.000018	0.005027	0.005027
13	**Delay q2**	0.0051438	6.60267E-05	0.000041	0.005300	0.005300
14	**Delay q2**	0.0054209	6.38521E-05	0.000040	0.005586	0.005586
15	**Delay q2**	0.0059456	0.000279049	0.000173	0.006806	0.006806

Appendix Table 17 : q2 5UEs, q0 increasing, QoS OFF

		QoS ON				
q1 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q1**	1.426000	0.006633	0.004111	1.421889	1.430111
5	**Throughput_r q1**	1.419000	0.008307	0.005148	1.413852	1.424148
10	**Throughput_r q1**	1.417000	0.011874	0.007360	1.409640	1.424360
11	**Throughput_r q1**	1.425000	0.008062	0.004997	1.420003	1.429997
12	**Throughput_r q1**	1.418000	0.013266	0.008223	1.409777	1.426223
13	**Throughput_r q1**	1.417000	0.007810	0.004841	1.412159	1.421841
14	**Throughput_r q1**	1.423000	0.009000	0.005578	1.417422	1.428578
15	**Throughput_r q1**	1.418000	0.007483	0.004638	1.413362	1.422638
1	**Throughput_r q2**	0.956000	0.139442	0.086425	0.869575	1.042425
5	**Throughput_r q2**	0.884000	0.089017	0.055172	0.828828	0.939172
10	**Throughput_r q2**	0.790000	0.144637	0.089646	0.700354	0.879646
11	**Throughput_r q2**	0.798000	0.127421	0.078975	0.719025	0.876975
12	**Throughput_r q2**	0.771000	0.084906	0.052624	0.718376	0.823624
13	**Throughput_r q2**	0.625000	0.078390	0.048586	0.576414	0.673586
14	**Throughput_r q2**	0.449000	0.060902	0.037746	0.411254	0.486746
15	**Throughput_r q2**	0.196000	0.045869	0.028430	0.167570	0.224430
1	**Delay q1**	0.007258	0.000009	0.000006	0.007252	0.007263
5	**Delay q1**	0.007463	0.000027	0.000017	0.007446	0.007479
10	**Delay q1**	0.008020	0.000041	0.000025	0.007995	0.008045
11	**Delay q1**	0.008246	0.000043	0.000027	0.008219	0.008273
12	**Delay q1**	0.008516	0.000045	0.000028	0.008488	0.008544
13	**Delay q1**	0.008810	0.000036	0.000022	0.008787	0.008832
14	**Delay q1**	0.009274	0.000030	0.000019	0.009255	0.009292
15	**Delay q1**	0.010094	0.000106	0.000066	0.010029	0.010160
1	**Delay q2**	0.004128	0.000011	0.000007	0.004121	0.004135
5	**Delay q2**	0.004124	0.000007	0.000005	0.004119	0.004128
10	**Delay q2**	0.004117	0.000018	0.000011	0.004101	0.004122
11	**Delay q2**	0.004120	0.000013	0.000008	0.004106	0.004122
12	**Delay q2**	0.004121	0.000010	0.000006	0.004106	0.004118
13	**Delay q2**	0.004109	0.000014	0.000008	0.004084	0.004101
14	**Delay q2**	0.004068	0.000017	0.000011	0.004044	0.004065
15	**Delay q2**	0.004019	0.000059	0.000037	0.003861	0.003934

Appendix Table 18 : q2 5UEs, q1 increasing, QoS ON

		QoS OFF				
q1 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_r q1	1.426000	0.006633	0.004111	1.421889	1.430111
5	Throughput_r q1	1.418000	0.008718	0.005403	1.412597	1.423403
10	Throughput_r q1	1.420000	0.014142	0.008765	1.411235	1.428765
11	Throughput_r q1	1.425000	0.008062	0.004997	1.420003	1.429997
12	Throughput_r q1	1.415000	0.011180	0.006930	1.408070	1.421930
13	Throughput_r q1	1.413000	0.007810	0.004841	1.408159	1.417841
14	Throughput_r q1	1.407000	0.007810	0.004841	1.402159	1.411841
15	Throughput_r q1	1.376000	0.015620	0.009682	1.366318	1.385682
1	Throughput_r q2	0.956000	0.139442	0.086425	0.869575	1.042425
5	Throughput_r q2	0.893000	0.098392	0.060983	0.832017	0.953983
10	Throughput_r q2	0.847000	0.143739	0.089089	0.757911	0.936089
11	Throughput_r q2	0.865000	0.146850	0.091017	0.773983	0.956017
12	Throughput_r q2	0.873000	0.122560	0.075962	0.797038	0.948962
13	Throughput_r q2	0.750000	0.054955	0.034061	0.715939	0.784061
14	Throughput_r q2	0.517000	0.080380	0.049819	0.467181	0.566819
15	Throughput_r q2	0.341000	0.054672	0.033885	0.307115	0.374885
1	Delay q1	0.009197	0.000234	0.000145	0.009052	0.009342
5	Delay q1	0.010635	0.000378	0.000234	0.010401	0.010869
10	Delay q1	0.018964	0.001448	0.000898	0.018067	0.019862
11	Delay q1	0.026327	0.002894	0.001794	0.024533	0.028120
12	Delay q1	0.039832	0.004995	0.003096	0.036736	0.042928
13	Delay q1	0.073491	0.010482	0.006497	0.066994	0.079988
14	Delay q1	0.216792	0.025871	0.016035	0.200757	0.232827
15	Delay q1	0.616694	0.061115	0.037879	0.578815	0.654573
1	Delay q2	0.004128	0.000011	0.000007	0.004121	0.004135
5	Delay q2	0.004123	0.000008	0.000005	0.004119	0.004129
10	Delay q2	0.004112	0.000017	0.000010	0.004107	0.004128
11	Delay q2	0.004114	0.000014	0.000009	0.004111	0.004128
12	Delay q2	0.004112	0.000011	0.000007	0.004114	0.004128
13	Delay q2	0.004093	0.000007	0.000004	0.004105	0.004113
14	Delay q2	0.004055	0.000021	0.000013	0.004055	0.004081
15	Delay q2	0.003897	0.000029	0.000018	0.004001	0.004037

Appendix Table 19: q2 5UEs, q1 increasing, QoS OFF

		QoS ON			
q3 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q2**	0.911000	0.124133	0.076937	0.834063	0.987937
2	**Throughput_r q2**	0.985000	0.146918	0.091059	0.893941	1.076059
3	**Throughput_r q2**	0.918000	0.164912	0.102212	0.815788	1.020212
4	**Throughput_r q2**	0.976000	0.193298	0.119805	0.856195	1.095805
5	**Throughput_r q2**	0.927000	0.093172	0.057747	0.869253	0.984747
6	**Throughput_r q2**	0.938000	0.153610	0.095207	0.842793	1.033207
7	**Throughput_r q2**	0.885000	0.212474	0.131690	0.753310	1.016690
8	**Throughput_r q2**	0.911000	0.087115	0.053993	0.857007	0.964993
9	**Throughput_r q2**	0.935000	0.143335	0.088838	0.846162	1.023838
10	**Throughput_r q2**	0.959000	0.139818	0.086658	0.872342	1.045658
1	**Throughput_r q3**	17.685000	0.320289	0.198513	17.486487	17.883513
2	**Throughput_r q3**	8.758000	0.111427	0.069062	8.688938	8.827062
3	**Throughput_r q3**	5.932000	0.081093	0.050261	5.881739	5.982261
4	**Throughput_r q3**	4.416000	0.121589	0.075361	4.340639	4.491361
5	**Throughput_r q3**	3.618000	0.157911	0.097873	3.520127	3.715873
6	**Throughput_r q3**	3.073000	0.089448	0.055440	3.017560	3.128440
7	**Throughput_r q3**	2.602000	0.095163	0.058982	2.543018	2.660982
8	**Throughput_r q3**	2.316000	0.099820	0.061868	2.254132	2.377868
9	**Throughput_r q3**	2.085000	0.064226	0.039807	2.045193	2.124807
10	**Throughput_r q3**	1.900000	0.084617	0.052445	1.847555	1.952445
1	**Delay q2**	0.004125	0.000012	0.000007	0.004118	0.004133
2	**Delay q2**	0.004131	0.000011	0.000007	0.004124	0.004138
3	**Delay q2**	0.004124	0.000017	0.000010	0.004114	0.004135
4	**Delay q2**	0.004129	0.000013	0.000008	0.004121	0.004137
5	**Delay q2**	0.004127	0.000008	0.000005	0.004122	0.004132
6	**Delay q2**	0.004127	0.000012	0.000007	0.004119	0.004134
7	**Delay q2**	0.004119	0.000024	0.000015	0.004104	0.004134
8	**Delay q2**	0.004127	0.000007	0.000004	0.004122	0.004131
9	**Delay q2**	0.004127	0.000010	0.000006	0.004120	0.004133
10	**Delay q2**	0.004129	0.000011	0.000007	0.004122	0.004135
1	**Delay q3**	0.097889	0.001873	0.001161	0.096728	0.099050
2	**Delay q3**	0.203560	0.002649	0.001642	0.201918	0.205201
3	**Delay q3**	0.3021982	0.004369779	0.002708	0.299490	0.304907
4	**Delay q3**	0.3976134	0.007422935	0.004601	0.393013	0.402214
5	**Delay q3**	0.4802466	0.01528191	0.009472	0.470775	0.489718
6	**Delay q3**	0.5592001	0.014412261	0.008933	0.550267	0.568133
7	**Delay q3**	0.6372158	0.013825968	0.008569	0.628647	0.645785
8	**Delay q3**	0.7199800	0.013481999	0.008356	0.711624	0.728336
9	**Delay q3**	0.7829744	0.009167498	0.005682	0.777292	0.788656
10	**Delay q3**	0.8486948	0.012133127	0.007520	0.841175	0.856215

Appendix Table 20: q2 5UEs, q3 increasing, QoS ON

		QoS OFF				
q3 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q2**	0.644000	0.073648	0.045647	0.598353	0.689647
2	**Throughput_r q2**	0.489000	0.093856	0.058172	0.430828	0.547172
3	**Throughput_r q2**	0.418000	0.041665	0.025824	0.392176	0.443824
4	**Throughput_r q2**	0.369000	0.033601	0.020825	0.348175	0.389825
5	**Throughput_r q2**	0.306000	0.037470	0.023224	0.282776	0.329224
6	**Throughput_r q2**	0.298000	0.038158	0.023650	0.274350	0.321650
7	**Throughput_r q2**	0.281000	0.029816	0.018480	0.262520	0.299480
8	**Throughput_r q2**	0.274000	0.046519	0.028832	0.245168	0.302832
9	**Throughput_r q2**	0.234000	0.035833	0.022209	0.211791	0.256209
10	**Throughput_r q2**	0.242000	0.056356	0.034929	0.207071	0.276929
1	**Throughput_r q3**	18.989000	0.158142	0.098016	18.890984	19.087016
2	**Throughput_r q3**	9.889000	0.058898	0.036505	9.852495	9.925505
3	**Throughput_r q3**	6.715000	0.050050	0.031021	6.683979	6.746021
4	**Throughput_r q3**	5.170000	0.025298	0.015680	5.154320	5.185680
5	**Throughput_r q3**	4.229000	0.041097	0.025472	4.203528	4.254472
6	**Throughput_r q3**	3.685000	0.034424	0.021336	3.663664	3.706336
7	**Throughput_r q3**	2.983000	0.020518	0.012717	2.970283	2.995717
8	**Throughput_r q3**	2.673000	0.021932	0.013593	2.659407	2.686593
9	**Throughput_r q3**	2.420000	0.014832	0.009193	2.410807	2.429193
10	**Throughput_r q3**	2.223000	0.004583	0.002840	2.220160	2.225840
1	**Delay q2**	0.004095	0.000013	0.000008	0.004086	0.004103
2	**Delay q2**	0.004061	0.000024	0.000015	0.004046	0.004076
3	**Delay q2**	0.004043	0.000017	0.000010	0.004033	0.004053
4	**Delay q2**	0.004021	0.000016	0.000010	0.004011	0.004031
5	**Delay q2**	0.003988	0.000018	0.000011	0.003977	0.003999
6	**Delay q2**	0.003983	0.000024	0.000015	0.003969	0.003998
7	**Delay q2**	0.003972	0.000023	0.000014	0.003958	0.003986
8	**Delay q2**	0.003963	0.000037	0.000023	0.003940	0.003986
9	**Delay q2**	0.003931	0.000039	0.000024	0.003906	0.003955
10	**Delay q2**	0.003930	0.000050	0.000031	0.003899	0.003961
1	**Delay q3**	0.090712	0.000796	0.000494	0.090218	0.091206
2	**Delay q3**	0.179484	0.001149	0.000712	0.178772	0.180196
3	**Delay q3**	0.2665191	0.001939166	0.001202	0.265317	0.267721
4	**Delay q3**	0.3467387	0.001154522	0.000716	0.346023	0.347454
5	**Delay q3**	0.4224410	0.001389428	0.000861	0.421580	0.423302
6	**Delay q3**	0.4847042	0.002404075	0.001490	0.483214	0.486194
7	**Delay q3**	0.5593268	0.002686054	0.001665	0.557662	0.560992
8	**Delay q3**	0.6191223	0.002625175	0.001627	0.617495	0.620749
9	**Delay q3**	0.6784909	0.001906169	0.001181	0.677309	0.679672
10	**Delay q3**	0.7365034	0.002694779	0.001670	0.734833	0.738174

Appendix Table 21: q2 5UEs, q3 increasing, QoS OFF

		QoS Enabled and Disabled				
UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q3**	22.180000	0.000000	0.000000	22.180000	22.180000
3	**Throughput_r q3**	7.400000	0.000000	0.000000	7.400000	7.400000
5	**Throughput_r q3**	4.557000	0.050408	0.031243	4.525757	4.588243
7	**Throughput_r q3**	3.173000	0.021000	0.013016	3.159984	3.186016
9	**Throughput_r q3**	2.582000	0.021354	0.013235	2.568765	2.595235
11	**Throughput_r q3**	2.170000	0.000000	0.000000	2.170000	2.170000
13	**Throughput_r q3**	1.740000	0.000000	0.000000	1.740000	1.740000
15	**Throughput_r q3**	1.520000	0.000000	0.000000	1.520000	1.520000
17	**Throughput_r q3**	1.350000	0.000000	0.000000	1.350000	1.350000
19	**Throughput_r q3**	1.222000	0.009798	0.006073	1.215927	1.228073
1	**Delay q3**	0.076796	0.000000	0.000000	0.076796	0.076796
3	**Delay q3**	0.240889	0.000000	0.000000	0.240889	0.240889
5	**Delay q3**	0.393152	0.000907	0.000562	0.392590	0.393714
7	**Delay q3**	0.527902	0.001745	0.001082	0.526820	0.528984
9	**Delay q3**	0.647911	0.000668	0.000414	0.647497	0.648325
11	**Delay q3**	0.759037	0.000211	0.000131	0.758906	0.759168
13	**Delay q3**	0.884257	0.000000	0.000000	0.884257	0.884257
15	**Delay q3**	0.985794	0.000068	0.000042	0.985752	0.985836
17	**Delay q3**	1.0778390	5.95735E-05	0.000037	1.077802	1.077876
19	**Delay q3**	1.1619780	0.002221953	0.001377	1.160601	1.163355

Appendix Table 22 : q3 only

		QoS ON				
q0 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q0**	0.474000	0.004899	0.003036	0.470964	0.477036
5	**Throughput_r q0**	0.472000	0.004000	0.002479	0.469521	0.474479
9	**Throughput_r q0**	0.473000	0.004583	0.002840	0.470160	0.475840
10	**Throughput_r q0**	0.472000	0.004000	0.002479	0.469521	0.474479
11	**Throughput_r q0**	0.473000	0.004583	0.002840	0.470160	0.475840
12	**Throughput_r q0**	0.472000	0.004000	0.002479	0.469521	0.474479
13	**Throughput_r q0**	0.472000	0.004000	0.002479	0.469521	0.474479
14	**Throughput_r q0**	0.474000	0.004899	0.003036	0.470964	0.477036
15	**Throughput_r q0**	0.473000	0.004583	0.002840	0.470160	0.475840
1	**Throughput_r q3**	20.757000	0.009000	0.005578	20.751422	20.762578
5	**Throughput_r q3**	15.088000	0.025219	0.015631	15.072369	15.103631
9	**Throughput_r q3**	9.422000	0.020881	0.012942	9.409058	9.434942
10	**Throughput_r q3**	7.992000	0.027857	0.017265	7.974735	8.009265
11	**Throughput_r q3**	6.585000	0.026173	0.016222	6.568778	6.601222
12	**Throughput_r q3**	5.154000	0.032924	0.020406	5.133594	5.174406
13	**Throughput_r q3**	3.742000	0.022271	0.013803	3.728197	3.755803
14	**Throughput_r q3**	2.315000	0.021095	0.013075	2.301925	2.328075
15	**Throughput_r q3**	0.896000	0.024576	0.015232	0.880768	0.911232
1	**Delay q0**	0.009083	0.000018	0.000011	0.009072	0.009094
5	**Delay q0**	0.009211	0.000036	0.000022	0.009188	0.009233
9	**Delay q0**	0.009447	0.000032	0.000020	0.009428	0.009467
10	**Delay q0**	0.009524	0.000029	0.000018	0.009505	0.009542
11	**Delay q0**	0.009657	0.000031	0.000019	0.009638	0.009676
12	**Delay q0**	0.009793	0.000024	0.000015	0.009778	0.009808
13	**Delay q0**	0.009975	0.000038	0.000024	0.009951	0.009999
14	**Delay q0**	0.010284	0.000029	0.000018	0.010265	0.010302
15	**Delay q0**	0.010992	0.000046	0.000028	0.010964	0.011020
1	**Delay q3**	0.082436	0.000036	0.000023	0.082413	0.082458
5	**Delay q3**	0.115463	0.000197	0.000122	0.115341	0.115585
9	**Delay q3**	0.188012	0.000415	0.000257	0.187755	0.188269
10	**Delay q3**	0.222386	0.000737	0.000457	0.221929	0.222843
11	**Delay q3**	0.270777	0.001115	0.000691	0.270085	0.271468
12	**Delay q3**	0.3466271	0.002153333	0.001335	0.345292	0.347962
13	**Delay q3**	0.4775734	0.003114909	0.001931	0.475643	0.479504
14	**Delay q3**	0.7674512	0.007218219	0.004474	0.762977	0.771925
15	**Delay q3**	1.8974570	0.050781523	0.031474	1.865983	1.928931

Appendix Table 23 : q3 1UE, q0 increasing QoS ON

		QoS OFF				
q0 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q0**	0.471000	0.003000	0.001859	0.469141	0.472859
5	**Throughput_r q0**	0.472000	0.004000	0.002479	0.469521	0.474479
9	**Throughput_r q0**	0.469000	0.003000	0.001859	0.467141	0.470859
10	**Throughput_r q0**	0.469000	0.003000	0.001859	0.467141	0.470859
11	**Throughput_r q0**	0.469000	0.003000	0.001859	0.467141	0.470859
12	**Throughput_r q0**	0.470000	0.000000	0.000000	0.470000	0.470000
13	**Throughput_r q0**	0.466000	0.004899	0.003036	0.462964	0.469036
14	**Throughput_r q0**	0.460000	0.000000	0.000000	0.460000	0.460000
15	**Throughput_r q0**	0.450000	0.000000	0.000000	0.450000	0.450000
1	**Throughput_r q3**	20.762000	0.006000	0.003719	20.758281	20.765719
5	**Throughput_r q3**	15.107000	0.026096	0.016174	15.090826	15.123174
9	**Throughput_r q3**	9.490000	0.020000	0.012396	9.477604	9.502396
10	**Throughput_r q3**	8.103000	0.031639	0.019609	8.083391	8.122609
11	**Throughput_r q3**	6.717000	0.017349	0.010753	6.706247	6.727753
12	**Throughput_r q3**	5.357000	0.034366	0.021300	5.335700	5.378300
13	**Throughput_r q3**	4.040000	0.024083	0.014927	4.025073	4.054927
14	**Throughput_r q3**	2.835000	0.025788	0.015983	2.819017	2.850983
15	**Throughput_r q3**	1.963000	0.014866	0.009214	1.953786	1.972214
1	**Delay q0**	0.080152	0.000046	0.000029	0.080123	0.080180
5	**Delay q0**	0.113211	0.000247	0.000153	0.113057	0.113364
9	**Delay q0**	0.185543	0.000501	0.000310	0.185233	0.185854
10	**Delay q0**	0.219174	0.000925	0.000573	0.218601	0.219747
11	**Delay q0**	0.267094	0.000858	0.000532	0.266562	0.267626
12	**Delay q0**	0.339554	0.002413	0.001496	0.338058	0.341049
13	**Delay q0**	0.460745	0.003045	0.001888	0.458857	0.462632
14	**Delay q0**	0.688918	0.007061	0.004377	0.684542	0.693295
15	**Delay q0**	1.098431	0.010615	0.006579	1.091852	1.105010
1	**Delay q3**	0.082420	0.000022	0.000013	0.082407	0.082433
5	**Delay q3**	0.115345	0.000210	0.000130	0.115214	0.115475
9	**Delay q3**	0.186724	0.000418	0.000259	0.186465	0.186983
10	**Delay q3**	0.219571	0.000843	0.000523	0.219048	0.220093
11	**Delay q3**	0.265866	0.000737	0.000457	0.265409	0.266322
12	**Delay q3**	0.3343803	0.002132184	0.001322	0.333059	0.335702
13	**Delay q3**	0.4442679	0.002708721	0.001679	0.442589	0.445947
14	**Delay q3**	0.6320600	0.007061319	0.004377	0.627683	0.636437
15	**Delay q3**	0.9026128	0.006338668	0.003929	0.898684	0.906541

Appendix Table 24 : q3 1UE, q0 increasing, QoS OFF

		QoS ON				
q1 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_r q1	1.419000	0.008307	0.005148	1.413852	1.424148
5	Throughput_r q1	1.419000	0.007000	0.004339	1.414661	1.423339
9	Throughput_r q1	1.424000	0.008000	0.004958	1.419042	1.428958
10	Throughput_r q1	1.417000	0.009000	0.005578	1.411422	1.422578
11	Throughput_r q1	1.416000	0.009165	0.005681	1.410319	1.421681
12	Throughput_r q1	1.417000	0.006403	0.003969	1.413031	1.420969
13	Throughput_r q1	1.421000	0.008307	0.005148	1.415852	1.426148
14	Throughput_r q1	1.415000	0.005000	0.003099	1.411901	1.418099
15	Throughput_r q1	1.420000	0.007746	0.004801	1.415199	1.424801
1	Throughput_r q3	20.761000	0.008307	0.005148	20.755852	20.766148
5	Throughput_r q3	15.094000	0.018547	0.011495	15.082505	15.105495
9	Throughput_r q3	9.419000	0.034191	0.021191	9.397809	9.440191
10	Throughput_r q3	8.005000	0.042249	0.026186	7.978814	8.031186
11	Throughput_r q3	6.588000	0.041425	0.025675	6.562325	6.613675
12	Throughput_r q3	5.163000	0.002955	0.001831	5.161169	5.164831
13	Throughput_r q3	3.736000	0.045431	0.028158	3.707842	3.764158
14	Throughput_r q3	2.316000	0.046087	0.028564	2.287436	2.344564
15	Throughput_r q3	0.897000	0.052163	0.032330	0.864670	0.929330
1	Delay q1	0.009074	0.000022	0.000014	0.009060	0.009087
5	Delay q1	0.009219	0.000019	0.000012	0.009207	0.009230
9	Delay q1	0.009451	0.000024	0.000015	0.009436	0.009466
10	Delay q1	0.009529	0.000023	0.000014	0.009514	0.009543
11	Delay q1	0.009650	0.000021	0.000013	0.009637	0.009663
12	Delay q1	0.009777	0.000021	0.000013	0.009764	0.009790
13	Delay q1	0.009985	0.000021	0.000013	0.009972	0.009998
14	Delay q1	0.010272	0.000035	0.000022	0.010250	0.010294
15	Delay q1	0.011066	0.000077	0.000048	0.011019	0.011114
1	Delay q3	0.082444	0.000035	0.000022	0.082422	0.082466
5	Delay q3	0.115590	0.000154	0.000096	0.115494	0.115686
9	Delay q3	0.188498	0.000732	0.000454	0.188045	0.188952
10	Delay q3	0.222623	0.001180	0.000731	0.221891	0.223354
11	Delay q3	0.271398	0.001762	0.001092	0.270306	0.272490
12	Delay q3	0.3470021	0.002954582	0.001831	0.345171	0.348833
13	Delay q3	0.4791866	0.005768947	0.003576	0.475611	0.482762
14	Delay q3	0.7685904	0.015065807	0.009338	0.759253	0.777928
15	Delay q3	1.9079420	0.095875649	0.059423	1.848519	1.967365

Appendix Table 25: q3 1UE, q1 increasing QoS ON

q1 UE		QoS OFF average	standard deviation	Confidence Interval	average-conf	average+conf
1	Throughput_r q1	1.421000	0.007000	0.004339	1.416661	1.425339
5	Throughput_r q1	1.414000	0.009165	0.005681	1.408319	1.419681
9	Throughput_r q1	1.410000	0.007746	0.004801	1.405199	1.414801
10	Throughput_r q1	1.412000	0.009798	0.006073	1.405927	1.418073
11	Throughput_r q1	1.411000	0.011358	0.007040	1.403960	1.418040
12	Throughput_r q1	1.403000	0.006403	0.003969	1.399031	1.406969
13	Throughput_r q1	1.400000	0.007746	0.004801	1.395199	1.404801
14	Throughput_r q1	1.384000	0.009165	0.005681	1.378319	1.389681
15	Throughput_r q1	1.347000	0.009000	0.005578	1.341422	1.352578
1	Throughput_r q3	20.760000	0.004472	0.002772	20.757228	20.762772
5	Throughput_r q3	15.107000	0.018466	0.011445	15.095555	15.118445
9	Throughput_r q3	9.492000	0.030265	0.018758	9.473242	9.510758
10	Throughput_r q3	8.102000	0.036551	0.022654	8.079346	8.124654
11	Throughput_r q3	6.725000	0.042249	0.026186	6.698814	6.751186
12	Throughput_r q3	5.363000	0.044508	0.027586	5.335414	5.390586
13	Throughput_r q3	4.042000	0.044227	0.027411	4.014589	4.069411
14	Throughput_r q3	2.840000	0.038987	0.024164	2.815836	2.864164
15	Throughput_r q3	1.964000	0.024576	0.015232	1.948768	1.979232
1	Delay q1	0.080187	0.000038	0.000023	0.080163	0.080210
5	Delay q1	0.113394	0.000145	0.000090	0.113305	0.113484
9	Delay q1	0.186032	0.000692	0.000429	0.185602	0.186461
10	Delay q1	0.219774	0.000692	0.000429	0.219344	0.220203
11	Delay q1	0.267590	0.001808	0.001121	0.266470	0.268711
12	Delay q1	0.340153	0.003069	0.001902	0.338251	0.342055
13	Delay q1	0.461425	0.005794	0.003591	0.457834	0.465016
14	Delay q1	0.688788	0.011974	0.007421	0.681367	0.696209
15	Delay q1	1.099317	0.020245	0.012548	1.086769	1.111865
1	Delay q3	0.082451	0.000029	0.000018	0.082434	0.082469
5	Delay q3	0.115502	0.000144	0.000089	0.115413	0.115592
9	Delay q3	0.187185	0.000655	0.000406	0.186780	0.187591
10	Delay q3	0.220174	0.001028	0.000637	0.219537	0.220811
11	Delay q3	0.266266	0.001748	0.001083	0.265182	0.267349
12	Delay q3	0.3349293	0.002860561	0.001773	0.333156	0.336702
13	Delay q3	0.4450838	0.004832674	0.002995	0.442089	0.448079
14	Delay q3	0.6320311	0.008408795	0.005212	0.626819	0.637243
15	Delay q3	0.9050840	0.010521545	0.006521	0.898563	0.911605

Appendix Table 26: q3 1UE, q1 increasing, QoS OFF

		QoS ON				
q2 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q2**	1.001000	0.196288	0.121658	0.879342	1.122658
5	**Throughput_r q2**	0.911000	0.124133	0.076937	0.834063	0.987937
9	**Throughput_r q2**	0.919000	0.143140	0.088717	0.830283	1.007717
10	**Throughput_r q2**	0.941000	0.112379	0.069652	0.871348	1.010652
11	**Throughput_r q2**	0.910000	0.192094	0.119059	0.790941	1.029059
12	**Throughput_r q2**	0.914000	0.115603	0.071650	0.842350	0.985650
13	**Throughput_r q2**	0.863000	0.142341	0.088222	0.774778	0.951222
14	**Throughput_r q2**	0.871000	0.071477	0.044301	0.826699	0.915301
15	**Throughput_r q2**	0.844000	0.119264	0.073919	0.770081	0.917919
1	**Throughput_r q3**	21.178000	0.195745	0.121322	21.056678	21.299322
5	**Throughput_r q3**	17.685000	0.320289	0.198513	17.486487	17.883513
9	**Throughput_r q3**	14.146000	0.498321	0.308857	13.837143	14.454857
10	**Throughput_r q3**	12.986000	0.365026	0.226241	12.759759	13.212241
11	**Throughput_r q3**	12.407000	0.308385	0.191135	12.215865	12.598135
12	**Throughput_r q3**	11.738000	0.357514	0.221585	11.516415	11.959585
13	**Throughput_r q3**	10.730000	0.802857	0.497607	10.232393	11.227607
14	**Throughput_r q3**	9.665000	0.388336	0.240689	9.424311	9.905689
15	**Throughput_r q3**	9.225000	0.575104	0.356447	8.868553	9.581447
1	**Delay q2**	0.004129	0.000015	0.000009	0.004120	0.004138
5	**Delay q2**	0.004125	0.000012	0.000007	0.004118	0.004133
9	**Delay q2**	0.004126	0.000012	0.000007	0.004119	0.004134
10	**Delay q2**	0.004129	0.000010	0.000006	0.004123	0.004136
11	**Delay q2**	0.004124	0.000017	0.000011	0.004114	0.004135
12	**Delay q2**	0.004127	0.000010	0.000006	0.004121	0.004133
13	**Delay q2**	0.004122	0.000016	0.000010	0.004112	0.004131
14	**Delay q2**	0.004125	0.000007	0.000004	0.004121	0.004130
15	**Delay q2**	0.004121	0.000011	0.000007	0.004114	0.004129
1	**Delay q3**	0.080723	0.000805	0.000499	0.080224	0.081222
5	**Delay q3**	0.097889	0.001873	0.001161	0.096728	0.099050
9	**Delay q3**	0.123798	0.004572	0.002834	0.120965	0.126632
10	**Delay q3**	0.135053	0.004109	0.002547	0.132506	0.137600
11	**Delay q3**	0.141512	0.003761	0.002331	0.139181	0.143843
12	**Delay q3**	0.1499157	0.004500719	0.002790	0.147126	0.152705
13	**Delay q3**	0.1650363	0.012807694	0.007938	0.157098	0.172974
14	**Delay q3**	0.1821917	0.007401519	0.004587	0.177604	0.186779
15	**Delay q3**	0.1910522	0.012179085	0.007549	0.183504	0.198601

Appendix Table 27: q3 1UE, q2 increasing, QoS ON

		QoS OFF				
q2 UE		average	standard deviation	Confidence Interval	average-conf	average+conf
1	**Throughput_r q2**	0.640000	0.081854	0.050732	0.589268	0.690732
5	**Throughput_r q2**	0.644000	0.073648	0.045647	0.598353	0.689647
9	**Throughput_r q2**	0.616000	0.059867	0.037105	0.578895	0.653105
10	**Throughput_r q2**	0.650000	0.058992	0.036563	0.613437	0.686563
11	**Throughput_r q2**	0.646000	0.084404	0.052313	0.593687	0.698313
12	**Throughput_r q2**	0.643000	0.163832	0.101542	0.541458	0.744542
13	**Throughput_r q2**	0.673000	0.090227	0.055923	0.617077	0.728923
14	**Throughput_r q2**	0.605000	0.088572	0.054896	0.550104	0.659896
15	**Throughput_r q2**	0.657000	0.092201	0.057146	0.599854	0.714146
1	**Throughput_r q3**	21.539000	0.080926	0.050157	21.488843	21.589157
5	**Throughput_r q3**	18.989000	0.158142	0.098016	18.890984	19.087016
9	**Throughput_r q3**	16.567000	0.211237	0.130924	16.436076	16.697924
10	**Throughput_r q3**	15.956000	0.199860	0.123872	15.832128	16.079872
11	**Throughput_r q3**	15.399000	0.202210	0.125329	15.273671	15.524329
12	**Throughput_r q3**	14.843000	0.281995	0.174779	14.668221	15.017779
13	**Throughput_r q3**	14.238000	0.337366	0.209098	14.028902	14.447098
14	**Throughput_r q3**	13.743000	0.311000	0.192756	13.550244	13.935756
15	**Throughput_r q3**	13.231000	0.285253	0.176798	13.054202	13.407798
1	**Delay q2**	0.004093	0.000012	0.000008	0.004085	0.004100
5	**Delay q2**	0.004095	0.000013	0.000008	0.004086	0.004103
9	**Delay q2**	0.004092	0.000010	0.000006	0.004085	0.004098
10	**Delay q2**	0.004098	0.000009	0.000006	0.004092	0.004104
11	**Delay q2**	0.004097	0.000012	0.000008	0.004089	0.004104
12	**Delay q2**	0.004092	0.000026	0.000016	0.004076	0.004108
13	**Delay q2**	0.004101	0.000013	0.000008	0.004093	0.004109
14	**Delay q2**	0.004090	0.000014	0.000009	0.004081	0.004099
15	**Delay q2**	0.004099	0.000013	0.000008	0.004091	0.004107
1	**Delay q3**	0.079258	0.000320	0.000199	0.079060	0.079457
5	**Delay q3**	0.090712	0.000796	0.000494	0.090218	0.091206
9	**Delay q3**	0.104850	0.001445	0.000896	0.103954	0.105746
10	**Delay q3**	0.109088	0.001461	0.000905	0.108183	0.109994
11	**Delay q3**	0.113281	0.001556	0.000964	0.112317	0.114245
12	**Delay q3**	0.1177541	0.002354587	0.001459	0.116295	0.119213
13	**Delay q3**	0.1230369	0.003054883	0.001893	0.121143	0.124930
14	**Delay q3**	0.1276550	0.003083191	0.001911	0.125744	0.129566
15	**Delay q3**	0.1328038	0.003039239	0.001884	0.130920	0.134688

Appendix Table 28: q3 1UE, q2 increasing, QoS OFF