HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communications and Automation
Department of Communications and Networking

Efsun Gürlek

Specifying and Verifying Requirements for Transmission of Medical Data in Public Networks

Master's thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Engineering

Espoo, 25 May 2009

Supervisor: Professor Jörg Ott
Instructor: M. Sc. Tero Kinnunen

HELSINKI UNIVERSITY OF TECHNOLOGY | ABSTRACT OF THE MASTER'S THESIS

Author:	Efsun Gürlek		
Title:	Specifying and Verifying Requirements for Transmission of Medical Data in Public Networks		
Date:	25 May 2009	**Language:** English	**Number of Pages:** x+97
Faculty:	Faculty of Electronics, Communications and Automation		
Department:	Department of Communications and Networking		**Code:** S-38
Supervisor:	Professor Jörg Ott		
Instructor:	M. Sc. Tero Kinnunen		

The purpose of this study is to identify and verify the necessary Quality of Service metrics in public networks for an existing remote patient monitoring system called Mobile Viewers. Therefore, any abnormalities can be detected beforehand and the application quality can be seen from the end user's point of view. The name "Mobile Viewers" refer to three different client applications:Web Viewers, Pocket Viewers and Cellular Viewers.

The literature part of this thesis reviews the former research studies dedicated to network performance measurements in 3G, 2.5G and Wireless LAN networks. Based on the review, the most suitable measurement methods, tools, metrics and environments are selected to be utilised during this study.

In the first part of the thesis work, passive live measurement tests are executed within UMTS, GPRS, LAN and Wireless LAN networks in order to find out the delay, jitter and packet loss metrics for the individual Mobile Viewers. As a result, GPRS presents the highest delay, jitter and packet loss values leading to poor application quality.

The second part of the thesis study focuses on identifying the quality requirements for Mobile Viewers. Initially, a network emulator tool is employed to emulate the necessary delay, jitter and packet loss metrics in order to test the application quality under different network conditions. Additional subjective user defined tests are executed to assess the quality for each viewer client. Finally, the limit delay, packet loss and jitter values, where the application quality starts to degrade, are presented.

Additional future work may be carried out by observing the Mobile Viewers' performances with higher technologies for instance, HSDPA. Furthermore, the conclusions derived from the analysis of the measurements and the proposed requirements for Mobile Viewers should be validated by additional experiments with different client devices, measurement tools and longer measurement periods.

Keywords: Quality of Service, passive measurement, network emulation, delay, jitter, packet loss

Preface

This thesis work was done as a part of the Mobile Viewers team at GE Healthcare Finland Corporation. The process has initially been quite difficult due to lack of fine experience with Linux/UNIX operating systems or other networking tools. However, my colleagues gave me the first lift to start the learning experience and continue on my own.

Firstly, I would like to thank GE Healthcare Finland, and especially Esa Kähkönen for providing me this opportunity to work on this thesis project. I also thank Tero Kinnunen, my instructor, for offering feedback, support and guidance during the thesis work. Most of all, I want to state my gratitude to my colleagues for their assistance, especially to Perttu Auramo and Teemu Mäki.

I also wish to thank Professor Jörg Ott, my supervisor, for his contribution on this thesis and would like to express my gratitude to the faculty's English teacher, William Martin for proofreading this thesis.

Finally, great thanks to my sister Gonca Gurlek, to Atila Yukselov and Timo Kärkäs who encouraged me the most throughout the end.

Espoo, May 2008

Efsun Gürlek

Table of Contents

List of Figures

List of Listings

List of Tables

Abbreviations and Definitions

2.5G	Second generation evolution of mobile telecommunications technology.
3G	Third generation mobile telecommunications technology.
Ad-hoc mode	Direct communication between two wireless peers.
B	*Bytes*, eight bits.
CDMA2000 1X EVDO	*Code Division Multiple Access 1 Evolution-Data Only*, a 3G standard operating WCDMA on air interface.
ECG	*Electrocardiogram*, a diagnostic tool that is used to record heart activity.
EDGE	*Enhanced Data rates for GSM Evolution*, considered as a 3G radio technology.
FIN	The FIN packet is sent when sender or receiver wants to terminate the active TCP connection.
FreeBSD	Unix-like free operating system.
Goodput	Effective throughput experienced by user.
GPRS	*General Packet Radio Service*, 2.5G mobile network technology.
GSM	*Global System for Mobile Communications*, cellular phone technology.
HSCSD	*High Speed Circuit Switched Data*, four times higher data rate than GSM.
HSDPA	*High Speed Down Link Packet Access*, evolution to third generation mobile telecommunications technologies. It supports higher data speeds than UMTS for downlink.
HTTP	*Hypertext Transfer Protocol*.
Infrastructure mode	Communication between two peers is through an access point in a wireless LAN.
Jitter	Delay variance, variation in packet transport delay.
KB	*Kilo Bytes*, one thousand bytes.
LAN	*Local Area Network*.
MB	*Mega Bytes*, one million bytes.
MCS	*Mobile Care Server*, a server which collects the patient data and distributes

	to the Mobile Viewer clients.
ms	*Milliseconds*, fraction of time.
MSS	*Maximum Segment Size,* the maximum size of a TCP segment.
OWD	*One Way Delay*, delay between source and destination.
QoS	*Quality of Service,* a measure of quality.
RLC	*Radio Link Control*, link layer protocol for error recovery and flow control in 3G networks.
RLC AM	*RLC Acknowledgement Mode*, option for ensuring error recovery and flow control in link layer.
RTT	*Round Trip Time*, the time when a packet is sent from source and the acknowledgement for that packet is received at the source.
RTT stdev	*Round Trip Time standard deviation*, standard deviation of RTT samples during a connection.
s	*Seconds*, fraction of time.
SACK	*TCP Selective Acknowledgment*, an option in TCP where the receiver informs the sender on the successfully arrived segments; hence, the sender resends the segments not received.
SCTP	*Stream Control Transport Protocol,* reliable and fast transport protocol, which supports multi-homing and multi-streaming.
SIN	The first packet sent during establishment of a TCP connection.
TCP	*Transmission Control Protocol*, a reliable data transport protocol. Transmission is performed in streams of bytes.
UDP	*User Datagram Protocol*, an unreliable data transport protocol. Transmission is performed by datagrams.
UMTS	*Universal Mobile Telecommunications System*, third generation mobile telecommunications technologies, The most common form of UMTS uses WCDMA for air interface. Hence, also can be referred as WCDMA.
VoIP	*Voice over Internet Protocol,* delivery of voice data over IP network.
WLAN	*Wireless Local Area Network.*
WWW	*World Wide Web,* Internet.

1 Introduction

Today, public networks have become the widely adopted universal communication platform to meet the needs of individuals. Over the last decade, they have evolved into a complex infrastructure of numerous networks. These networks all have different transmission characteristics. Moreover, the variety of applications and services that can be provided over these networks have increased together with different quality expectations. Therefore, it became significantly important to understand how these applications or services work, prior to deployment, to ensure successful implementation.

To be able to understand the evolution of Quality of Service in data transmission, it is first necessary to examine the history of cellular data networks.

Public networks have undergone a significant progress especially with the deployment of cellular data networks. Initially, *global system for mobile communicaitons* (GSM) offered circuit switched telephony with limited data services. Later *general packet radio service* (GPRS) became the new packet-oriented data service as an extension to GSM networks. Although other GSM evolutions such as *high speed circuit switched data* (HSCSD) or *enchanced data rates for GSM evolution* (EDGE) have considerably enhanced the data communications, today the third generation cellular systems such as *universal mobile communicaiton systems* (UMTS) or even higher technology, *high speed downlink packet access* (HSDPA), provide much improved support for data services by offering guaranteed quality, which is known as *quality of service(QoS)*.

“QoS can mean different things to different people” [1]. In this study the *application behaviour* is meant as QoS. Different applications show different behaviours; thus, have different data transmission requirements. For instance conversational and streaming traffic are more delay sensitive than interactive and background traffic. In this thesis work, the focus is on the real-time interactive medical data transmission, which is sensitive to *delay*, *jitter* and *packet loss*.

“In addition, a distinction is made between *subjective and objective QoS*” [1]. Objective QoS refers to direcly measurable metrics such as delay, jitter and packet loss whereas the subjective QoS corresponds to how users perceive the quality of the applications [1]. This study focuses on both. For the subjective QoS, application response times will be recorded to assess the quality while, for the objective QoS, network traffic measurements will be employed.

Network traffic measurements are performed to study the network performance or the application itself. One method is to conduct live tests by measuring passively or actively. *Active measurements* are performed by injecting traffic into the network and analysing the injected traffic. *Passive measurements*, on the other hand, monitor the traffic from certain points in the network to understand the traffic behaviour. The thesis focuses on the passive measurements, which will give us a spectrum of delay, jitter and packet loss metrics. In addition, the *network emulator* will be used in this study. With the help of a network emulator, poor network conditions can be emulated so that the limit delay, jitter and packet loss, where the application quality starts to become poor can be estimated.

The main goal of this thesis study is to identify and verify the Quality of Service metrics for an existing remote patient monitoring system, which uses public networks, in order to see the application quality from the end-users' point of view. As discussed above, to reach the goal, passive network measurements, subjective measurements and network emulation will be performed.

The organization of this thesis is as follows. The overall problem statement is identified and introduced in Chapter 2 including the problem definition and the motivation for this work. Chapter 3 presents the basic terms and notions, which are explained in the sense that they are used in this thesis. An intensive background research following a summary of results with comparisons are provided in Chapter 4. Chapter 5 presents the background research for tool selections with regarding passive, active measurements and network emulation. The live passive measurements to be conducted and their environment set ups for each type of network are presented in Chapter 6. Chapter 7 displays the live measurement results and further analyses and evaluates them. The specification and verification of requirements for remote patient monitor client applications via a network emulator tool is discussed in Chapter 8. Finally, Chapter 9 concludes the study by presenting the thesis summary and recommendations for possible future work.

2 Problem Statement

This chapter defines the problem and the problem environment together with stating the motivation for this work.

2.1 Environment

2.1.1 Patient monitoring

The concept of patient care systems began in the mid-1960s. In earlier days, patient caregivers were applying necessary treatments by merely listening to their instincts. Though, as with many other notions, patient care has evolved as well, introducing patient monitoring equipments to hospital environments for assisting in critical decisions and diagnostics. Moreover, patient-monitoring equipment became computer-based systems during the 1990s with database functions and report-generation systems together with decision-making qualifications [95].

Patient care is defined as “Repeated or continuous observations or measurements of the patient, his or her physiological function, and the function of life support equipment, for the purpose of guiding management decisions, including when to make therapeutic interventions, and assessment of those interventions” [96].

Monitoring is the most essential part of patient care systems [96]. The monitor data can be temperature, heart rate, blood pressure, neurological components and so forth. The collected information can be recorded or processed into a report for observation throughout the patient care process. It is ensured by monitoring, that the correct treatment is applied to the patients and signs of complications are noticed in the early stages.

Nowadays there are numerous types of patient monitoring tools. Moreover, the patient monitoring tools can be viewed remotely, which implies that the doctors can view monitor data any time and anywhere. Some of monitoring tools are designed for hospital use only while others can be utilised outside of hospital as well.

2.1.2 Mobile Viewers

GE Healthcare has developed a remote patient monitoring system called *Mobile Viewers*, which consists of *Mobile Care Server* (MCS) and *Mobile Viewer clients*. The principle of this system is to assist the patient care decision-making procedure, independent of time and place.

The Mobile Viewers system architecture is a client-server model where the Mobile Care Server represents the server and the Mobile Viewer clients represent the clients. There are three types of viewer clients: *Web Viewer, Pocket Viewer* and *Cellular Viewer* [92].

The Mobile Care Server collects the data coming from the patient monitors and distributes them to viewer clients. Conversely, viewer clients provide a user interface and display the collected data in different types of formats to users.

The overall architecture of the Mobile Viewers system is illustrated in Figure 1 and the displayed components are explained in detail in Sections 2.1.3 to 2.1.7.

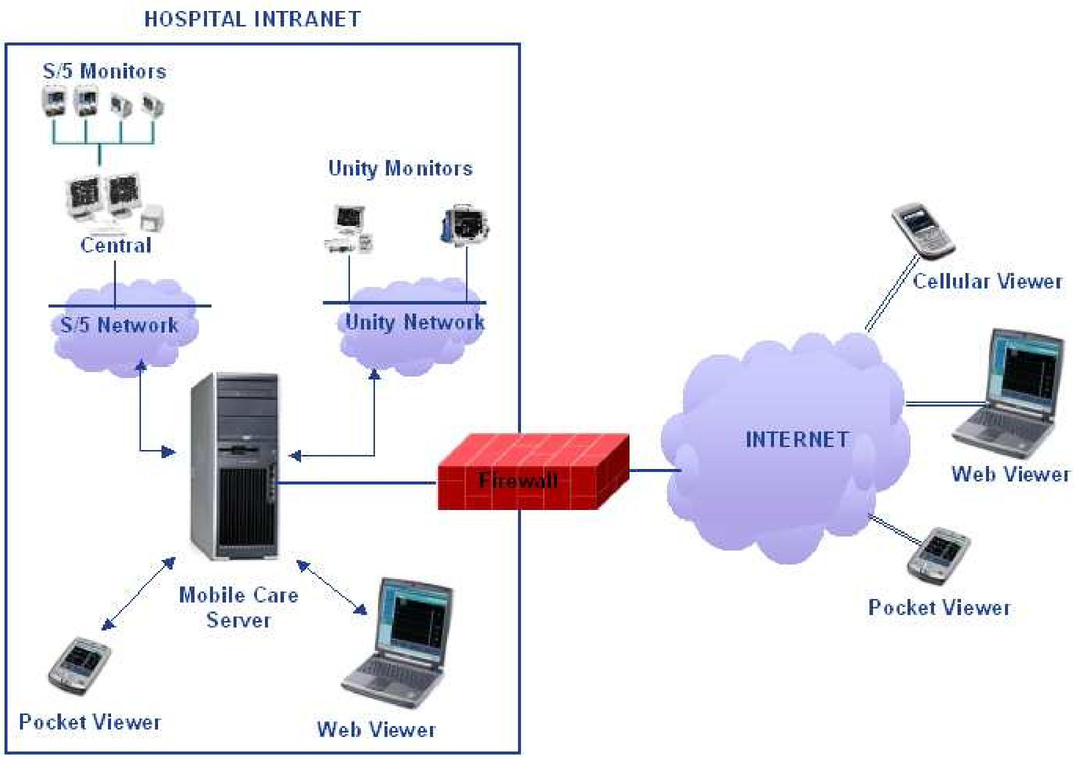

Figure 1 Simplified drawing of Mobile Viewers system architecture

The figure depicted is a simplified version of the real architecture, yet gives a sufficient amount of detail for this thesis work. As seen from the figure, the communication between the MCS and the Mobile Viewer clients is realised by public networks.

2.1.3 Mobile Care Server

The *Mobile Care Server* is the server component of Mobile Viewers architecture, which consists of hardware and Java-based server software operating on a Linux platform interfacing with S/5, Unity and hospital networks.

The principle of MCS is to provide Mobile Viewer clients (Web Viewer, Pocket Viewer, Cellular Viewer) with patient monitor data from GE monitors. In addition, MCS presents web pages for installing the client applications and administering the whole system.

Due to the processing capability of the MCS, 90 concurrent clients can be maintained: 30 Web Viewers, 30 Pocket Viewers, and 30 Cellular Viewers. Moreover, at most 256 patient monitors can be connected to the server.

2.1.4 Web Viewer

The *Web Viewer* is a monitoring application running on a generic PC, which runs on Windows and Mac OSX operating systems that supports the Java Runtime Environment (JRE) and maintains a web browser. It is capable of displaying real time patient monitoring data: waveforms, alarms and numeric data. Moreover, trends in graphical and numerical format and patient demographics can be presented to the user.

The monitored data can be viewed in *Single-View* mode or *Multi-View* mode through the Web Viewers. Single-view mode displays real-time monitoring data for one patient while Multi-View mode can monitor 4 patients, 8 patients or 16 patients simultaneously. This property does not exist either with the Pocket Viewers or the Cellular Viewers. Moreover, trend and waveform data can be printed directly from the Web Viewers.

The Web Viewer can use any type of technology (wired, wireless, mobile, dial-up) to connect to the Mobile Care Server. The communication between the Web Viewer and the MCS is a *transmission control protocol* (TCP) connection with 4040 port number.

In addition, HTTPS protocol and SSL/TLS is used to secure the connection. However, maintaining the security of the connection is beyond the scope of this thesis.

The Web Viewer client is started from a link in the Mobile Viewers homepage http://<server> and the application runs using Java Web Start technology. Figure 2 demonstrates a screenshot of *Web Viewer 5.1* version in real-time waveform single-viewing mode [94].

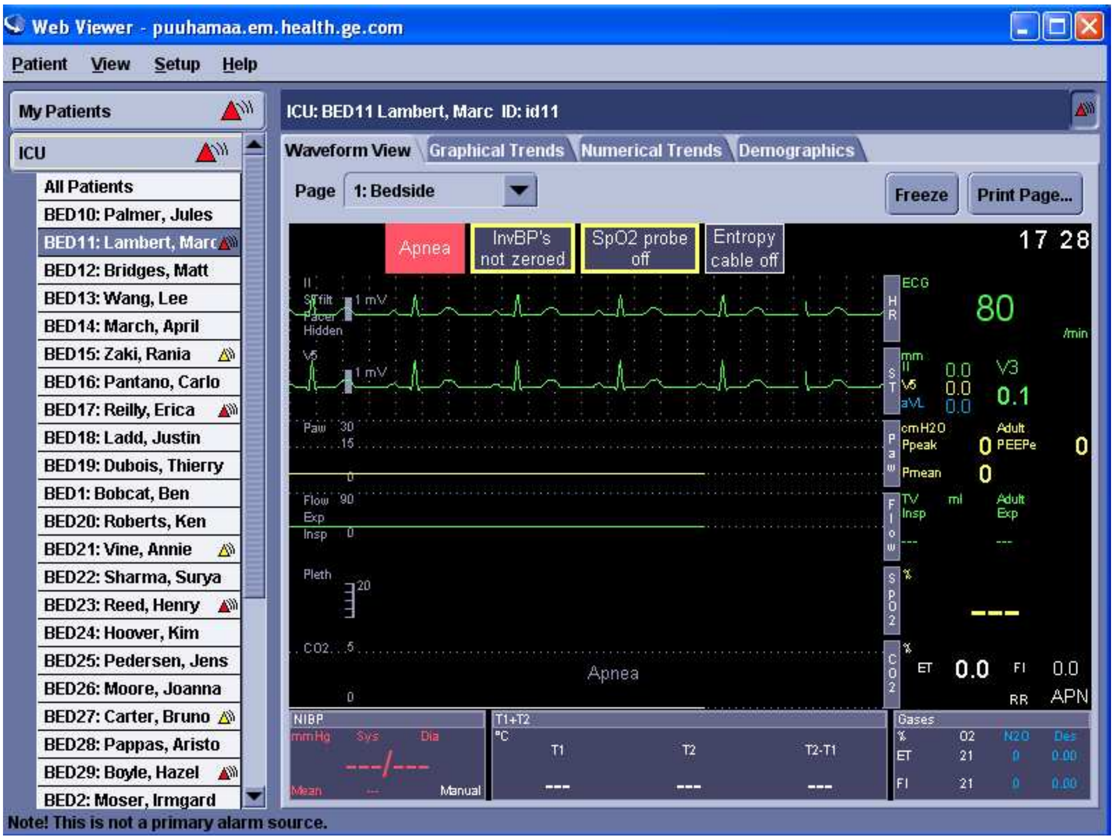

Figure 2 Web Viewer client application

2.1.5 Pocket Viewer

The *Pocket Viewer* is another remote patient monitoring application running on a Windows Mobile / Windows Pocket PC .The Pocket Viewer client application is quite similar to the Web Viewers. The communication between the Mobile Care Server and Pocket Viewers is realized by TCP connection on port 4040 and secured using the HTTPS protocol and SSL/TSL as well. The Pocket Viewer is a PDA-based application and some of the features are not included such as multi-mode viewing and printing capabilities. In addition, the display is smaller and user interface is made simpler [92].

Figure 3 Pocket Viewer client application

Pocket Viewers operate on wireless or cellular technologies depending on the PDA capability in order to connect to the Hospital LAN and the Mobile Care Server. Figure 3 demonstrates a Pocket PC running the Pocket Viewer client application in wave-forms view.

2.1.6 Cellular Viewer

The *Cellular Viewer* is the smallest client application, running on mobile phones supporting *mobile information device profile* (MIDP) 2.0. It is a Java-based client application like the Web Viewer and the Pocket Viewer. The application provides almost similar functionality to Pocket Viewer, but with an even more limited screen size and a simpler user interface [93].

The Cellular Viewer needs only an Internet connection to be able to connect to the MCS. What is needed is just an access point (gateway) to the Internet that the operators provide, and an “Internet service contract” with the operator. Naturally, a phone that supports Internet access is required as well.

The communication between the Cellular Viewer and the Mobile Care Server is a TCP connection on port 80 for non-secure and 443 for secured data. The connection is secured by HTTPS protocol and SSL/TLS as well [92].

Figure 4 Cellular Viewer client application

The Cellular Viewer is able to use any connectivity provided by the mobile phone, such as a GPRS or UMTS, or a WLAN connection if available, in order to connect to the Mobile Care Server remotely. Figure 4, illustrates a mobile phone running the Cellular Viewer client application in waveform viewing mode [94].

2.1.7 GE patient monitors

GE Healthcare manufactures patient monitors as well. In this thesis study two types of GE patient monitors will be connected to the Mobile Care Server: the *S/5* and the *Unity* monitors.

The Unity monitors are connected to the Unity network. The communication between the Mobile Care Servers and Unity monitors is realized by using *the user datagram protocol* (UDP).

The S/5 monitors are connected initially to a central device, which then connects to the S/5 network, as opposed to Unity monitors, which they are connected to the Unity network directly. In addition, unlike Unity monitors the communication between the MCS and S/5 monitors is utilised by TCP connections.

2.2 Motivation and Problem Definition

As stated earlier in Section 2.1, the communication between the Mobile Care Server and Mobile Viewer clients, is realized by public networks. When introducing a new application or service to the public networks, it is essential to analyse the performance of the application before making it accessible for end-users, because the application may not work properly under certain network conditions.

If the Mobile Viewer application quality is considered from the user perspective, the client applications respond within a certain period of time to user actions. The viewed real-time continuous waveforms should flow without any interrupts. There should not occur any disconnections from the MCS while viewing waveforms or downloading trend data.

The end-users should be informed of the resulting poor quality under certain network conditions. Therefore, it is essential to test the application performance by network measurements in order to detect any abnormalities beforehand and see the application quality from the end user's point of view.

2.3 Goals and Scope

The main goal of this thesis is to specify and verify requirements for Mobile Viewer client applications with the intention of informing end-users of the unsatisfying client performances during data transmission in public networks

The main goal can be broken into sub tasks to simplify managing of a large task.

1. Define Quality of Service metrics to be measured
2. Determine the measurement methods to be used
3. Find out the potential measurement tools and select to be utilised
4. Settle on the measurement environments
5. Decide which types of networks are to be measured
6. Determine what type of tests are to be measured for each Mobile Viewer client
7. Conduct measurements
8. Make an analysis of these measurements for each Mobile Viewer client

9. Evaluate the results

10. Emulate different network conditions with a network emulator

11. Conduct user measurement tests under each network condition

12. Determine the performance quality for Mobile Viewer clients

13. Define the criteria for dissatisfactions

14. Settle on requirements and verify for each Mobile Viewer client

15. Evaluate the results

16. Confirm that the main goal has been achieved.

This thesis work is composed of live measurements, network emulations, as well as background research on different networks. However, the scope for this thesis study will be limited to the areas specifically defined below.

The background research on network performance studies will consist of 3G networks, 2.5G networks and WLANs. The studies will be grouped according to the used technology and then will be compared to each other.

Next, the live measurements will only be carried out passively. Active measurements will not be executed. The networks to be measured will be LAN, WLAN, UMTS and GPRS. All three Mobile Viewers clients' network performances will be analysed. In addition, the used measurement tools and devices will be limited to a couple.

Finally, the network emulations will be performed both in wired and wireless environments and the requirements for the user expectations will be evaluated for all viewer clients.

3 Basic Terms and Concepts

In this chapter some basic terms and concepts related to this thesis work are discussed. Section 3.1 presents the necessary network parameters to be measured while Section 3.2 observes the possible measurement methods.

3.1 Vital Network Parameters for Mobile Viewers

After performing some initial background research on network measurements, which are discussed in the next chapter, it was noticed that the researchers' main focus was on delay, jitter and packet loss metrics. These metrics were measured in order to define the application/protocol performances in different networks or to identify the network characteristics. Therefore, the QoS metrics to be measured in this study are selected as *delay*, *jitter*, and *packet loss*. This section explains the delay, jitter and packet loss metrics. These metrics will be further analysed together with the earlier research results in Chapter 6.

3.1.1 Delay

Delay is the time for a packet to travel across the network from one computer to another. Delay is measured in seconds or fractions of seconds. There are two delay metrics used in telecommunications: *one-way delay* (OWD) and *round-trip time* (RTT). Many things can be the cause of delay, such as poor network conditions, congestions, and the processing capability of routers or hosts.

3.1.1.1 One-way delay (OWD)

One-way delay is the time from the sender sends a packet till the receiver receives the corresponding packet. One-way delay calculations usually require expensive sophisticated test gear and programs since both the clocks of sender and receiver must be synchronized accurately for correct measurements. However, it can also be easy if the sender and receiver reside on the same machine so that the same clock would be used for both the sender and the receiver.

The synchronization of clocks can be maintained by a *global positioning system* (GPS) receiver, or a *network time protocol* (NTP) or a *code division multiple access*

(CDMA) based cellular phone. A GPS clock or a CDMA based cellular phone provides microseconds precision, which is much better compared to a *Stratum 1* NTP server that is at best 10 millisecond precision [33][34].

The researchers, who have used one-way delay as the delay metric, have typically developed their own measurement tools as a team, for instance, *Moset* and *QoSMet* [1][2]. However, such tools are usually used for their own research purposes and are not freely available. In this thesis, the sender and the receiver will not be on the same node. For this reason, described above OWD measurement will not be performed in this thesis work due to synchronization problem.

3.1.1.2 Round-trip time (RTT)

Round-trip time is the delay of a packet from the sender to the receiver and back. Unlike one-way delay, measuring RTT is much easier and requires less expensive equipment, since, the sender and the receiver do not need to be synchronized. Fortunately, there are already many open source tools for measuring RTT, such as *tcptrace* [30][31][32].

In this thesis, RTT will be used as the delay metric due to the simplicity of measuring RTT, and the availability of many tools that can measure RTT. Moreover, the majority of previous research considers RTT as the delay metric rather than OWD; hence, this will offer a greater amount of previous research results for comparison with findings in this thesis.

3.1.2 Jitter

Jitter is the variation of delay and is measured as seconds or fraction of seconds like delay. Nowadays at least three different definitions are considered as jitter: *one-way delay/round-trip time variation*, *inter-arrival time variation* and *round-trip time standard deviation*. Some studies find it sufficient to consider categorising only one of these terms as jitter, while others combine them [42].

3.1.2.1 One-way delay/RTT variation

One-way delay variation is the difference of one-way delays between two successive packets whereas RTT variation is the difference of two consecutive round-trip times.

If S_i is the timestamp for packet i and R_i is the time of arrival for packet i, the variance is calculated as the formula below:

$D(i) = (R_i - R_{i-1}) - (S_i – S_{i-1}) = (R_i – S_i) - (R_{i-1} – S_{i-1})$ [35][36].

The majority of active measurement tools facilitate this formula when calculating jitter for one-way delay analysis. However, this metric is not relevant to this thesis work, which is more interested in the spectrum of delay variance rather than instant jitters.

3.1.2.2 Inter-arrival time variation

Inter-arrival time variation is the difference of arrival times between two successive packets. If the inter-arrival time of packet i is R_i, the variance is calculated as the formula below:

$IAT\ (i) = R_{i-1}- R_i$ [38].

Once again, it is desirable to measure the common variability of delay; hence, this metric will not be utilised for this thesis work as well.

3.1.2.3 Round-trip time standard deviation

Standard deviation is a measure of the variability of a data set [37]. If the data set consists of the round-trip times of a number of packets, the standard deviation of this data set will give the variability of RTTs. This is exactly what we are seeking for; hence, this metric is the most relevant measure for our jitter estimation.

The majority of passive measurement tools, which calculate RTT, also estimate the RTT standard deviation. RTT standard deviation is referred to as random jitter where the probability of incoming packets is a normal distribution [39][40][41]. The formula below represents RTT standard deviation. Assume, the round-trip time of packet i is R_i, the average RTT is M, and the number of RTT samples is N.

$$\text{RTT stdev (N)} = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}\left(R_i - M\right)^2\right)}.$$

3.1.3 Packet loss

Packet loss is losing packets along the data path and is measured as a percentage of the lost packets with respect to the total packets sent. This can occur due to fast sender and slow receiver hosts, broken links, packet corruption, bit error rates, or traffic congestions [45][47].

Unfortunately, the Internet is a best effort network and does not guarantee safe arrival of packets. Packet loss can affect applications harmfully whether they use TCP or UDP as the transport protocol. It is crucial for TCP to transmit back the lost packet, where there is no obligation in UDP communication. For instance, during a voice conversation the packet loss can be concealed by re-playing the previous frame from the buffer, although, this degrades the voice quality [46].

As mentioned before, the communication between the Mobile Care Server and the viewer clients are realized by TCP connections. In TCP communication, when a packet is lost, it must be sent back. Therefore, the *retransmission percentage*, which is calculated as a percentage of the retransmitted packets over total packets sent, will be considered in order to find packet loss in this thesis work.

3.2 Measurement Methods

In the previous section, three QoS metrics: delay, jitter and packet loss were selected to be measured in this study. However, it was not clear how to measure these metrics. Several research studies on network measurements, discussed in the next chapter, revealed that there are two types of measurement methods to uncover the delay, jitter and packet loss in different networks: *live testing* (active and passive) and *emulation* (test-bed) [101]. In this section, these terms are explained and clarified.

3.2.1 Live testing (active and passive measurements)

Live testing is analysing the traffic characteristics of a network such as delay, jitter, packet loss, throughput or defining the performance of an application in a network by conducting real network measurements. Live testing requires expensive equipment, time and funding. The measurer needs to use real network components, precise data capturing methods and tools. During testing, it may seem quite monotonic, however;

at the end, the collected data will give researchers the *actual* network/application performance. There are two ways to execute: passively and actively.

3.2.1.1 Passive measurements

Passive measurements are in essence listening to network traffic from one or more points in a given network. In this case, there is no injection of packets into the network which leads to additional network traffic; hence, the measurement is accurate. Passive measurement is best suited to situations where capture points can be selected freely and offers the most accurate results when the whole network is owned, in this case, the measurer is capable of listening to the traffic from any point in the network.

The most popular passive measurement tools are *windump*, *tcpdump* and *wireshark*. These tools are used for capturing the traffic flow and analysing it. There are also some passive measurement tools, which utilise the data captured by windump, tcpdump or wireshark, in order to calculate QoS information. For instance, tcptrace yields round-trip times, retransmission percentage and many other metrics.

In this study, a passive measurement method will be used since the measurement points can be selected freely. The measurements can be conducted both at the server and the client side. In addition, the actual traffic will be captured with no additional packets inserted; hence, the measurements will be more accurate. Chapters 6 and 7 examine the passive measurements and the results in detail.

3.2.1.2 Active measurements

Active measurements, also called probing, do not listen to traffic but send their own probe packets from the sender to the receiver; therefore, active measurements generate additional traffic. The probe packets are usually artificial and they should be carefully selected so that it would not disturb the present network traffic. The measurer observes these packets travelling on the network; for instance, delay can be estimated by calculating the time when the probe packet arrives the destination. Many active measurement tools are present, the most well known are: *traceroute* and *ping*. The modern ones such as *D-ITG, Iperf, OWAMP, Nethawk* not only generate traffic but also have the capability to identify OWD, RTT, jitter and loss.

The active measurement method will not be utilised during this thesis work because of limited time and because passive measurements are adequate for this thesis study. However, the use of active measurements might be considered as a future work.

3.2.2 Network emulator

A *network emulator* emulates the functions of a network so that network traffic characteristics such as delay, jitter, bandwidth and packet loss can be configured. Generally, the emulated type of network is a wide area network, given that it is quite expensive to be able to create a real wide area network environment. Network emulator allows the user to duplicate the behaviour and control of the QoS characteristics of a network. The most popular network emulators are *dummynet, NIST net, NetEm* and *ALTQ*.

3.2.2.1 Methods of emulation

Network emulation can be realized by launching a device/module on the LAN that modifies packet flow (incoming/outgoing) in a way that imitates the behaviour of application traffic in the emulated environment. The module/device includes a number of network parameters in its emulation model: round-trip time across the network (delay), the available bandwidth, packet loss percentage, packet duplication, packet reordering, and network jitter. Desktop PCs, phones, laptops can further be connected to the emulated environment; therefore, users can experience the performance and behaviour of applications in that environment.

In this thesis study, a network emulation method will be used after executing passive measurements, which is discussed in detail in Chapter 8. Additional delay, jitter and packet loss will be given to the network by a network emulator to find out the limit values for delay, jitter and packet loss, where the QoS of the Mobile Viewer clients becomes poor.

4 Background Research

In this chapter, more than eighty articles on Internet traffic measurements have been analysed and the ones that are of most interesting for our thesis study are presented. Certainly, the "interesting" that are of special interests in this thesis are those, which provide information on delay, jitter and packet loss.

These studies are grouped according to the type of technology, which they based their measurements on, such as: 3G, 2.5G and WLAN. Finally each section presents a summary table of research studies, inspired by [97], as a subsection.

4.1 Research on 3G Networks

Multiple research and measurements were carried out to evaluate and characterise the delay, jitter and packer loss characteristics in 3G networks. In this section, eleven of these studies are discussed briefly where the main focus is on delay and jitter. Packet loss behaviour was sometimes discussed in only some of the studies. In addition to UMTS, some studies focused on identifying QoS parameters for the newer mobile technology, which is, HSDPA.

The authors of Study 1 [1] propose a passive approach focusing on one-way delay characteristics measured by a tool that was developed, *QoSMeT*, using a *voice over internet protocol* (VoIP) application in live HSDPA and UMTS networks in 2006. A year later, the same authors publish another study based on HSDPA and WCDMA networks, by focusing also on TCP performances and user perspectives by carrying out goodput measurements [2].

Study 3 [3] is one of the first live measurements performed in an HSDPA network in 2005. The authors identify the average round-trip times by conducting both lab and field measurements whereas another study performed in 2005 in Germany, focuses on TCP performance in two different UMTS networks by conducting active measurements between two handsets [4].

A TCP/IP incoming traffic capture tool for mobile phones was developed to be utilised by the authors of Study 5 [5] for their research on GPRS and UMTS networks undertaken in 2006, in Spain. The study revealed how handovers affect delays. The researchers of Study 6 [6] followed the same approach as the previous study [5] in defining UMTS performance by comparing UMTS to GPRS. Study 7 [7], on the other hand, presents a different approach by experimenting with not only lightly loaded but also fully loaded network conditions by transporting a mixture of data, video and voice traffic over a 3G network.

Another brief study was performed by a TKK student, focusing on an online game application latency in GSM, GPRS and UMTS networks. The author concludes that UMTS is quite good for real-time online action games as opposed to GSM and GPRS [8]. The authors of Study 9 [9] focus on packet delays and packet loss in UMTS networks whereas measurements over a CDMA 2000 1x EVDO 3G were conducted by using TCP bulk data transfer under stable conditions in another study [10].

Lastly, a study very similar to Study 1[1] and Study 2 [2] observed the VOIP performance on WCDMA and HSDPA networks in Finland. They conducted their experiments on user experience by measuring the voice quality with E-model than measuring goodput in Study 2 [2]. The authors prove that the real embedded client measurements show lower performance than laptop-based clients [11].

The following table groups the previous studies on 3G networks according to the year, and the place the research is held in, the measurement method used (such as passive or active), as well as the networks focused on (such as HSDPA or UMTS). The studies are also grouped according to the preferred measurement environments (such as laptop-to-laptop, server-to-mobile, laptop-to-server or mobile-to-mobile) and the number of connected users/clients to the measurement environment. The type of applications used during measurements and even the mobility of measurements (such as mobile or stable conditions) are presented in the table. The studies are further classified into the network load during measurements and the metrics used (such as one-way delay, jitter, packet loss, round-trip time, etc.). Finally, the tools used during measurements are listed together with the packet size sent.

Study	Method	Networks	Environment	Application	Mobility	# User	Load	Live/ Lab	Location	Packet size	Metrics	Year	Tools
1	Passive	UMTS, HSDPA	Laptop 2 laptop	VOIP- UDP	Stable	1	Light	Live	Finland VTT	33-65 Bytes	1-way delay Loss, jitter	2006	QoSMET GPS
2	Passive & Active	UMTS, HSDPA	Laptop 2 laptop	VOIP- UDP HTTP-TCP	Stable	1	Light	Live	Finland VTT	32B-65kB 32B-2MB	1-way delay Jitter RTT	2007	QoSMET, MOSET, GPS, DTIG
3	Passive	HSDPA	Server - laptop	FTP- TCP	Mobile	1	Light	Live	Finland Nokia	32 Bytes	RTT	2006	Unspecified
4	Active	2 different UMTS	Mobile2Mobile Server-Mobile	FTP- TCP HTTP-TCP	Stable & Mobile	1	Light	Live	Germany	3.45 MB	RTT	2005	Tcpdump Iperf
5	Passive	UMTS GPRS	Server-Mobile Mobile2Mobile	FTP-TCP TCP	Stable& Mobile	1	Light	Live	Spain	200KB	RTT	2006	SymPA
6	Active	UMTS GPRS	Server-Mobile	FTP-TCP ICMP HTTP-TCP	Stable	1	Light	Test bed	Austria	32,100, 1450 Bytes	RTT, jitter	2006	Iperf Ping
7	Passive	3 different UMTS	Server-Mobile Laptop-Server	Video,Voice, Data call, TCP Pinging	Stable	5	Heavy &light	Live	China	Video, voice call, ping	RTT, packet loss	2007	NetMonitor, GPS, Etheral, Tcptrace
8	Passive	GPRS UMTS, EDGE	Mobile-Mobile	Online game application	Stable	2	Light	Test bed	Finland Ericsson TKK	80 uplink 100,160 Bytes DL	RTT, packet loss	2005	Protocol Analyzer
9	Active	UMTS	Laptop 2 laptop	UDP Fake TCP	Stable	1	Light	Test bed	Spain	40 bytes	1-way delay Packet loss	2005	Ping,hping2 Tcpdump, Pathchar, pathrate
10	Active	CDMA 2000 1xEVDO	Laptop-Server	TCP	Stable	1	Light	Live	Korea	N/A	RTT, jitter, Packet loss	2006	Iperf, tcpdump, tcptrace
11	Passive	HSDPA WCDMA	Laptop-Laptop Mobile-Mobile	VOIP-UDP	Stable& Mobile	1	Light	Live &lab	Finland &USA	32 Bytes	RTT, jitter, Packet loss	2007	Ping

Table 1 Comparison of 3G network studies on delay, jitter and packet loss

4.1.1 Summary of research on 3G networks

From Table 1, it can be observed that each study covers an individual data point somewhere in the spectrum. It is unreasonable to match their results one-to one in order to identify the exact results for the delay, jitter and packet loss parameters of 3G networks. However, it is possible to point out similar studies and define a spectrum of values. The spectrums of values are discussed in Chapter 7 for the purpose of comparison of this thesis work findings.

Studies 1, 2, 3, 11 are comparable in their used applications, environment and results. Studies 4, 5, 6, 7 are similar in the use of handsets and observations of the delay increase, which the authors suggest, is due to processing, and access delay. Studies 4 and 5 are more closely related, because a mobile environment is used to carry out the tests. Studies 9 and 10, however; are not comparable.

To conclude briefly, the reason behind observing delay spikes and packet losses are due to RLC retransmissions [7][11]. The main issue with 3G delay is not only tied directly to wireless access performance but also mobile device capabilities [4][5]. Therefore, using handsets result in an additional delay of about 250-300 ms compared to laptop performance due to encoding/processing and jitter buffer delay [7]. The studies show that lab measurements are usually too optimistic compared to live measurements [11] and, overall, HSDPA shows better results in delay, jitter and packet loss than in UMTS [1][2]. Moreover, there is a vast difference between lightly and heavily loaded network performances as the difference observed under stable and mobile conditions [7]. To reduce the delay and retransmissions it is wise to use RLC Unacknowledged Mode [11] where the client is mobile and RLC Acknowledged mode, where the client is steady [7].

4.2 Research on 2.5G Networks

This section analyses a few interesting studies performed which concern of the delay, jitter and packet loss measurements of GPRS networks. Only seven of them are discussed briefly. Other than GPRS, a higher performance and more expensive technology, HSCSD, was also focused on by the authors. At the end of this section a summary table is provided which contains the exact column titles as in Section 4.1.1.

The authors of Study 12 [12] observed the TCP performance on GPRS networks by conducting measurements on a test-bed where they had a full control over the radio channel parameters. Study 13 [13] concentrated on performances of HSCSD and GPRS data transmission with the help of a measurement tool, WLT, which was developed by the authors of the study. The study was able to reveal that in mobile environments, a disconnection occurs every 11^{th}–12^{th} minute in HSCSD and a long pause in data transfer occurs in GPRS networks.

The authors of Study 14 [14] focused on TCP bulk transfer performance on GPRS networks by using an event-driven simulator. They demonstrate that the simulation results are too optimistic when compared to live measurements. Study 15 [15], on the other hand, followed a different approach by identifying packet loss in the Internet and in the access side of GPRS networks then comparing their results to those of a wired-line dial-up.

Study 16 [16] focused on identifying TCP behaviour in GPRS networks by generating traffic streams and analysing them. The authors of Study 17 [17] also followed a case similar to Study 15 [15], where the authors evaluated TCP performance by comparing GPRS, ISDN and dial-up modems and denoting that GPRS may not be the right choice for Internet users.

Lastly, the researchers of Study 18 [18] continued their measurements on GPRS one year after publishing Study 13 [13]. They measured the RTT of a GPRS link by using a 32-byte pinging method and they realized that the minimum RTT is improved by 200 ms compared to their earlier measurements. The authors also denote that the test-bed measurement results are quite optimistic compared to live network results.

Study	Method	Network	Environment	Application	Mobility	# User	Load	Live/ Lab	Location	Packet size	Metrics	Year	Tools
12	Active	GPRS	Mobile-Server SACK enabled RLC ACK timestamped	TCP	Stable	1	Light	testbed	Sweden	Unspecified	RTT bulk transfer rexmit	2002	Tcptrace Ethereal, Nethawk
13	Active	GPRS HSCSD	Laptop-Server SACK enabled RLC ACK timestamped	TCP- HTTP bulk-FTP	Stable& Moving	1	Light	live	Finland Sonera	150KB FTP 280-499 B 348-4758 B 539-5070 B	bulk RTT rexmit	2001	WLT
14	Simulat.	GPRS	Laptop-Server RLC ACK	TCP-bulk	Stable	1	Light	simul.	Germany Ericsson	500 kb	bulk RTT		Method by[25]
15	Passive	GPRS Wired-dial up	Packets traced at GGSN Gi interface	TCP flows	Stable	1	Light& Heavy	Live	Hungary Ericsson	N/A	RTT Packet loss	2004	Algorithm s [23], [24]
16	Active	GPRS (2 different)	Mobile-Server SACK, RLC ACK	UDP, TCP File transfer	Stable	1	Light	Live	UK Cambridge Vodaphone	1064B UDP 600KBTCP	One-way delay RTT Packet loss	2002	Ttcp, NTP Tcpdump Tcptrace
17	Passive	GPRS ISDN Dial-up	Laptop-Server RLC ACK	TCP www, email, FTP	Stable	many (6)	Light	Live & Simul.	Netherlands Vodaphone	2.7MB RTT 32 B ping	RTT	2001	WinPcap Perl Script Ping GPRSIM
18	Active	GPRS	Mobile-Server RLC ACK	UDP, TCP- bulk	Stable& Moving	1	Light	Live & testbed.	Finland Sonera	32 B ping	RTT	2002	Ping tcpdump Nethawk

Table 2 Comparison of 2.5G network studies on delay, jitter and packet loss

Once again, a comparison table similar to previous Section 4.1 is presented based upon the exact comparison techniques, however; this time the focus was on 2.5G network studies.

4.2.1 Summary of research on 2.5G networks

As in Section 4.1, each study seats itself as an individual data point somewhere in the spectrum. However, the comparable studies are pointed out with the help of Table 2. Again the spectrum of delay, jitter, and packet loss values will be discussed in Chapter 7.

Study 15 is the only study based on a passive approach, however; it shares the method of comparing GPRS to wired dial up with Study 17. Studies 16 and 18 are similar due to using the same environment and the same applications actively. Study 18 conducts experiments also in a mobile environment and performs measurements on UDP performance over GPRS a year after Study 13. Study 12 experiments with measurements on a test-bed whereas Study 14 uses a simulation environment.

In summary, the studies emphasize that, setting the TCP receiver window size to medium, enabling the SACK option and high MSS with long TCP transactions can altogether improve TCP performance [15]. The simulation results are too optimistic compared to test-bed or live results [18], something which has already been noted for the 3G network studies that the test-bed performance results are perhaps overly optimistic, compared to live measurement results. Also, HSCSD provides better performance than GPRS [13], however; in mobile environment, a disconnection every 11-12th minute in HSCSD and a long pause in data transfer may occur in GPRS networks [13]. Finally, the studies suggest that GPRS may not be the right choice for Internet users [17].

4.3 Research on WLAN Networks

Although, the published research on wireless LANs is as equally outdated as studies performed on GSM [26][27][28], still a few are worth mentioning. The majority of these studies focus on packet loss rather than delay and jitter on WLANs. Six of these studies are presented and a summary table is demonstrated at the end once again.

The authors of Study 19 [19] performed research on identifying QoS parameters over wireless LANs in ad hoc and infrastructure (communication through an access point) mode in 2004 by conducting active measurements whereas the authors of Study 20 [20] present a comprehensive study on TCP and UDP behaviour over WLAN. It was noted that UDP suffers from communication resets due to the absence of any flow control; hence, resulting in lower throughput than TCP.

In Study 21 [21], the author focused on throughput and response time under different network loads for two commercial WLANs, by measuring data at the medium access control sub-layer (data link). He noted that buffering can generally affect WLAN performance.

The authors of Study 22 [22] observed the usability of multicasting for VoIP applications over WLAN whereas Study 23 [23] focused on WLAN performance under different vehicular mobility and peer-distance situations. It was noted that WLAN is suitable for inter-vehicle communications.

Lastly, the authors of [24] constructed a WLAN test-bed for determining the effect of received power, walls, floors and interfering laptops on wireless LANs. The authors pointed out that the RTT and packet loss increases by the number of walls, floors and interfering laptops.

The next page demonstrates a comparison table for WLAN studies similar to Table 1 for 3G networks and Table 2 for 2.5G networks.

Study	Method	Network	Environment	Application	Mobility	# User	Load	Live/ Lab	Location	Packet size	Metrics	Year	Tools
19	Active	Wlan (Ad hoc, infras.)	Laptop-laptop Laptop-palmtop	TCP, UDP	Stable	1	Light, Heavy, Medium	Test-bed	Italy	64B, 128B, 256B, 512B, 1024B, 1500B	RTT, bulk Jitter, packet loss	2005	D-ITG
20	Active	Wlan (ad-hoc)	Laptop-laptop Desktop-laptop	TCP, UDP	Stable	1	Light	Test-bed	USA	100B, 5000B, 1000B, 1500B	Packet loss, Bulk transfer	1999	Ttcp, nstat, tcpdump
21	Active	Wlan (infras)	PC-PC	Link layer, Ethernet frame	Stable	1	Light, Heavy, Medium	Live	USA	72B, 112B, 212B, 512B, 1012B, 1512B, 1526B	Mean response time, inter frame space	1999	Sniffer network analyzer
22	Passive	Wlan (ad-hoc)	Laptop-Laptop PC-PC Laptop-PC	VoIP, UDP	Stable	many	Light	Test-bed	USA, Germany	1472B, 42B	RTT, jitter, packet loss	2004	Un-specified
23	Active	Wlan (ad-hoc)	Laptop-Laptop	UDP	Mobile	1	Light	Live	USA	256B, 1024B	Signal/ Noise Throughput Packet loss	2002	Netperf
24	Active	Wlan	Laptop- PC	Telnet	Stable	1	Light	Test-bed	USA	Unspecified	RTT, packet loss Throughput	1997	CWINS, Harris LAN, WaveLAN Benchmar-king tools

Table 3 Comparison of WLAN network studies on delay, jitter and packet loss

4.3.1 Summary of research on WLAN networks

As mentioned before, there are many studies conducted on WLAN networks, however; they are performed quite some time ago, in the late 90s and early years of this century. These studies' main focus was upon packet loss and throughput. Indeed, there are some similarities between these studies derived from Table 3.

Studies 19 and 22 are the newest studies performed on WLAN, in fact, those are the only ones, which provide any information on delay and jitter. The studies prior to these mainly concentrated on throughput, link quality and loss. Studies 19 and 21 both experiment on lightly, medium and fully loaded network conditions. Study 19 performs measurements on a test-bed similar to Studies 20 and 24 rather than Study 21, which is on a live wireless network. Study 22 is the only research performed with a passive approach, whereas Study 23 is the only study, which utilises a mobile environment.

As a conclusion, knowledge is gained about how wireless performance is affected on lightly, medium, and fully network conditions [19][21]. The existence of peer interference or presence of walls and floors in the wireless environment also degrades the performance [24]. Moreover, the connectivity becomes poor during mobility [23]. It is also suggested that WLAN is suitable for inter-vehicle communication [23] and wireless bridges are capable of extending the wireless coverage area successfully [24]. Finally, the results emphasize that the ad-hoc mode can provide better performance than infrastructure mode in lightly loaded conditions [19].

5 Network Measurement and Emulation Tools

As discussed in the previous chapter, there are several tools for passive measuring, active measuring, and network emulating. In this chapter, the most popular tools are presented based on the previous research contained in Chapter 4. Section 5.1 focuses on passive measurement tools while Section 5.2 presents active measurement tools. Lastly, Section 5.3 discusses some network emulators. At the end of each section, a benchmark of measurement tools is presented for active, passive measurements and network emulation respectively. Based on the benchmark, the selection of the measurement tools to be utilised in this study and in possible future studies is made.

5.1 Passive Measurement Tools

The general property of these tools is to listen to the current traffic on the measurement points in a network. This section will discuss the most popular four passive measurement tools. The passive measurement tools, except *tcptrace*, are neither available for purchase nor *Open Source*, as they are maintained for research purposes only.

QoSMET

QoSMET is a passive measurement tool developed by VTT [55]. The tool is developed as a side product of the Easy Wireless Project ITEA [31], and is capable of monitoring end-to-end QoS performance of a certain application in both end devices over any heterogeneous network where IP is supported. It is based on layer 2 measurements, where received and sent packets are captured from the network interface. As a result, any type of application can be examined. The tool is able to run on the same device as the measured application or on some other device within the network. In addition, the tool can measure one-way delay, jitter, packet loss, connection break duration, throughput, offered load and the volume of data sent and received. Clock synchronization of the end devices is needed for precise measuring by utilising either GPS, NTP or a CDMA based cellular phone if the sender and receiver do not reside in the same node [56][1][2].

NetHawk Protocol Analysers (M5)

NetHawk Protocol Analysers, also developed by VTT, are multi-purpose network-monitoring tools, which allow passive analysis [55]. NetHawk Analyser products are intended for system integration, functional testing, load testing, and network operations on 2G, 2.5G and 3G networks. The difference of NetHawk from other passive tools is that NetHawk provides an environment where multiple users are able to analyse the measurement system. The developers of NetHawk note that the tool is featured to perform multi-interface and multi-technology network monitoring, optimisation, and troubleshooting [57]. The NetHawk Analysers can be included into a fully automated test system and, moreover, are able to utilise multipoint flow measurements with QoSMET. QoSMET handles the endpoint while Nethawk traces within the network path, access and core network interfaces [12][18].

SymPA

SymPA, or the *symbian protocol analyser*, is a passive protocol analyser for mobile phones. The incoming TCP/IP traffic can be captured without interfering with the normal performance of the mobile terminal. The tool provides an interface to process captured information and export it to other environments. SymPA runs on Symbian OS and when it is in capture mode, all IP packets that arrive at the mobile devices from 2.5G and 3G connections are saved in buffers in raw format. The files can be transferred to a computer and transformed from text to *pcap* format by *wireshark* further to be analysed with an analyser tool. [5]

Tcptrace

Tcptrace is a tool for analysing TCP flows, which are captured by a packet capture program such as *tcpdump* [61]. Shawn Ostermann is the developer of tcptrace [58]. Based on previous research results given in Chapter 4, many researchers prefer this tool for analysing TCP communications passively. Because, it is Open Source, fast, up-to-date, easy to use, and is fully documented. Several types of outputs holding the number of TCP connections seen, the round trip times, retransmissions, throughput, TCP segment size and much other information can be generated. In addition, many kinds of graphs for further analysis such as RTT graphs, throughput graphs, TCP sequence graphs, etc. can be created with tcptrace to be later visualised by programs like *xplot* [59] or *jplot* [60].

5.1.1 Selection of a passive measurement tool

It is necessary to find a unique passive measurement tool for listening to the traffic between the Mobile Viewer clients and the Mobile Care Server, something which could work under in all types of operating systems with new mobile phones, pocket PCs, laptops and workstations which is able to generate delay, jitter and packet loss. However, such a complex tool has not yet been developed. As observed in Table 4, *tcptrace* looks the most promising one. Moreover, it is the most considered tool for passive measurement analysis (see Chapter 4).

Traffic Analysers (Passive)	**Operating Systems**			**Testing Device**				**Meter Types**					**Protocols**						**Logging Phase**			**Access**	
	LINUX	***Windows***	***Symbian***	***Laptop / Notebook***	***Mobile Phone***	***Pocket PC***	***PC Worrkstation***	***OWD***	***RTT***	***Jitter***	***Packet loss***	***Throughput***	***UDP***	***TCP***	***ICMP***	***Telnet***	***VOIP***	***DNS***	***Sender***	***Receiver***	***Remote***	***Open Source***	***Research Purposes***
NetHawk	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
Tcptrace	X			X			X		X	X	X	X		X					X	X		X	
QoSMET		X		X	X	X	X	X	X			X	X	X	X	X	X	X	X	X			X
SymPA			X		X									X						X			X

Table 4 Comparison of passive measurement tools

Table 4 classifies four passive measurement tools according to the operation systems which are compatible with (Linux, Windows, Unix like); the type of devices which they can be built on (laptop, PC, mobile phone, pocket PC); the meter types they yield (for instance RTT, OWD, jitter, packet loss, throughput); the protocols to be measured (such as UDP, TCP, ICMP, etc.,); the place where the logging is initiated and stored (sender, receiver, remote device); and finally, the accessibility of the tool.

Tcptrace, the only Open Source tool among others, can provide the required RTT, jitter and loss measurements on TCP applications and can run in Linux operating system providing that the MCS runs on a Linux server. Tcptrace is often utilised with a packet-capturing tool, for instance *tcpdump*. Since the communication between the Mobile Care Server and viewer clients is based on the TCP/IP protocol, tcpdump and tcptrace together is perfectly suited to our measurement environment set up in this thesis.

Tcpdump is a packet capture tool written by Van Jacobson for Linux operating systems, allowing users to capture and view TCP/IP packets transmitted or received over a network. It operates on a packet level and the captured packets can be saved into files. The captured packets can later be analysed by filtering via tcptrace. A section of a tcptrace output of the communication between a Pocket Viewer client and a Mobile Care Server is presented in Figure 5, generated via the following line of code:

```
tcpdump -i eth0 -w tmp.pcap
tcptrace -lr tmp.pcap
```

Listing 1 Code for generating tcptrace output

As depicted in Figure 5, tcptrace traces one TCP flow between the server and the client, where “a” represents the Mobile Care Server and “b” the Pocket Viewer client. The communication is traced in both ways: server to client (a to b) and client to server (b to a). During the capture, the client was viewing the continuous *four waveforms*. The capturing is started after logging in to the Pocket Viewer application; hence, there is no visible *SIN* packet. Moreover the Pocket Viewer application does not end with a *FIN* packet.

Initially, the time at which the first and last packets of the connection seen are reported then follows the lifetime of the connection, and the number of packets seen. Afterwards, the filename currently being processed is listed, following the multiple TCP statistics for the forward (a2b) and the reverse (b2a) directions. Valuable information is documented with tcptrace, which makes it possible to comprehend the communication flow and characteristics, between the Mobile Care Server and the Pocket Viewer client, for instance.

The most practical parameters utilised in order to determine the TCP communication characteristics between the MCS and the viewer client are: *total packets* (the total number of packets seen); *rexmit data pkts* (the count of retransmitted packets); *RTT avg, min*, *max, stdev* (the average, minimum, maximum and standard deviation of all RTT samples); *data xmit time* (the time of the TCP conversation); *avg segm size* (the average segment size seen during the communication); and *RTT samples* (the number of total RTT samples seen) [58].

```
arg remaining, starting with 'tmp5.pcap'
Ostermann's tcptrace -- version 6.6.1 -- Wed Nov 19, 2003

2623 packets seen, 2324 TCP packets traced
elapsed wallclock time: 0:00:50.012795, 52 pkts/sec analyzed
trace file elapsed time: 0:10:04.464834
TCP connection info:
1 TCP connection traced:
TCP connection 1:
       host a:        192.168.1.113:4040
       host b:        192.168.1.104:1044
       complete conn: no      (SYNs: 0)  (FINs: 0)
       first packet:  Wed Jan 21 10:31:54.641034 2009
       last packet:   Wed Jan 21 10:41:58.700190 2009
       elapsed time:  0:10:04.059155
       total packets: 2324
       filename:      tmp5.pcap
   a->b:                            b->a:
     total packets:         1205         total packets:          1119
     ack pkts sent:         1205         ack pkts sent:          1119
     pure acks sent:         592         pure acks sent:          527
     sack pkts sent:           0         sack pkts sent:            0
     dsack pkts sent:          0         dsack pkts sent:           0
     max sack blks/ack:        0         max sack blks/ack:         0
     unique bytes sent:   398185         unique bytes sent:     21380
     actual data pkts:       613         actual data pkts:        592
     actual data bytes:   398185         actual data bytes:     21380
     rexmt data pkts:          0         rexmt data pkts:           0
     rexmt data bytes:         0         rexmt data bytes:          0
     zwnd probe pkts:          0         zwnd probe pkts:           0
     zwnd probe bytes:         0         zwnd probe bytes:          0
     outoforder pkts:          0         outoforder pkts:           0
     pushed data pkts:       613         pushed data pkts:        592
     SYN/FIN pkts sent:      0/0         SYN/FIN pkts sent:       0/0
     urgent data pkts:         0 pkts    urgent data pkts:          0 pkts
     urgent data bytes:        0 bytes   urgent data bytes:         0 bytes
     mss requested:            0 bytes   mss requested:             0 bytes
     max segm size:          987 bytes   max segm size:           101 bytes
     min segm size:           66 bytes   min segm size:            35 bytes
     avg segm size:          649 bytes   avg segm size:            36 bytes
     max win adv:          17152 bytes   max win adv:           33580 bytes
     min win adv:          17152 bytes   min win adv:           31597 bytes
     zero win adv:             0 times   zero win adv:              0 times
     avg win adv:          17152 bytes   avg win adv:           32933 bytes
     initial window:         659 bytes   initial window:            0 bytes
     initial window:           1 pkts    initial window:            0 pkts
     ttl stream length:       NA         ttl stream length:        NA
     missed data:             NA         missed data:              NA
     truncated data:      372439 bytes   truncated data:          590 bytes
     truncated packets:      613 pkts    truncated packets:        10 pkts
     data xmit time:     604.049 secs    data xmit time:      603.334 secs
     idletime max:        1001.5 ms      idletime max:         1110.1 ms
     throughput:             659 Bps     throughput:               35 Bps

     RTT samples:            613         RTT samples:             592
     RTT min:                3.8 ms      RTT min:                 0.0 ms
     RTT max:              206.6 ms      RTT max:                40.0 ms
     RTT avg:               98.9 ms      RTT avg:                 1.2 ms
     RTT stdev:             64.6 ms      RTT stdev:               6.6 ms
```

Figure 5 Section of a tcptrace output between MCS and Pocket Viewer client

The division of *total packets* by *data xmit time* can provide a rough estimation of *inter-departure time* (x packets per second) of TCP packets. The average segment size can give a rough estimation of *TCP segment size,* which the server sends to the client. The *max segm size* and *min segm size* (maximum and minimum segment size seen) can be constructive as well if the packet size between the server and client is desired to be expressed by boundaries. When the inter-departure time and mean segment size of the communication between the server and client is known, it is possible to emulate the traffic actively via an active measurement tool. This eliminates the burden of testing the application passively for a long period of time. This is because maintaining the test equipment and keeping the same environmental conditions can be sometimes difficult.

Additionally *RTT avg, min, max and stdev* can give a spectrum of delay and delay variance of the communication. The round-trip time is calculated after an acknowledgement packet of the sent packet is received successfully. Tcptrace makes a distinction between a normal acknowledgement packet of the sent packet and the retransmitted segment acknowledgement.

Finally tcptrace is able to output the raw RTT samples data format; hence, the user can observe the sequence numbers and the round-trip times in milliseconds of all the packets by the following line of code:

```
tcptrace -Z tmp.pcap
```

Listing 2 Code for generating tcptrace raw RTT samples

5.2 Active Measurement Tools

Active measurements tools share a common feature, which is injecting traffic to the network (probing). This section will discuss the most popular seven active measurement tools. These tools send probe packets and determine the throughput, delay, jitter and loss by analysing how the packets travel in the network. To be able to measure one-way delay, the synchronization of the sender and receiver by an NTP server or GPS clock is definitely required, certainly if the tool supports such a synchronization feature.

MOSET

MOSET is an active mobile service-testing tool developed by VTT [55]. Moset is a client-server tool, where the server is placed in the network and the client resides in a mobile phone. User quality is perceived by measuring HTTP performance from a client. The application provides the user with a variety of tests to select from. The main contribution provided by MOSET, is to find delay in various access networks such as WLAN, GPRS, EDGE, UMTS. Throughout the measurement period, data can be stored on a central server over the network due to the limited memory capability of mobile devices [62].

Netperf

Netperf is a popular active network performance-measuring tool, which includes an up-to-date manual. The tool performs unidirectional bulk data transfer and request/response performance using the TCP, UDP and SCTP transport protocols. Netperf is a client-server model as well. the client creates a connection to the server informing on the measurement test to execute. The tool, unfortunately, does not provide the estimation of the RTT or OWD parameters. In general Netperf is used mainly for throughput estimation [63].

NetProbe

N*etProbe* is a UDP-based multithreaded, active measurement client-server tool. NeProbe measures end-to-end performance parameters (delay and packet loss) of a VoIP connection. This tool is also a client-server oriented application. The client sends a packet to the remote server and then the server instantly sends the same packet back to the client. The UDP packet generated by the client encloses the session ID, sequence number, and time-stamps. The client sends them at a regular interval to the remote server where the packet is time-stamped every time it either leaves or arrives. The session ID, sequence number and the time-stamps of each packet are logged for later analysis [64].

WLT

WLT, or the *wireless link tester***,** is a client-server type active measurement tool developed by the Sonera Corporation. It is designed to measure the throughput, round-trip time and reliability of a wireless link. WLT utilises TCP as its transport protocol. The tool can be installed on a laptop or PC and is capable of measuring bulk or request/response transfers [13].

Iperf

Iperf is a popular server-client based active measurement tool for measuring the bandwidth and the quality of a network link. Iperf was developed in the *National Laboratory for Applied Network Research* (NLANR) project, which ended in 2006 [4]. Jperf can be associated with Iperf to provide a graphical user interface written in Java. Iperf uses the different capacities of TCP and UDP to find out QoS statistics [6]. Bandwidth is identified with TCP tests while one-way delay, round-trip time, jitter and packet loss is via UDP tests [10].

RUDE&CRUDE

RUDE&CRUDE, or the *real-time UDP data emitter and collector for RUDE*, is an active measurement tool for generating UDP traffic developed in Tampere University of Tech. The RUDE part creates the traffic and CRUDE receives and logs on the other side of the connection. This tool is found to be similar with *MGEN*, which is also a UDP traffic generator tool. However, they note at their website that RUDE&CRUDE provides more precise measurements by leaving the synchronization of nodes to GPS or NTP. Hence, they generate one-way delay statistics.

MGEN

MGEN, or the *multi-generator*, is an active measurement tool; which is used for generating UDP traffic. The traffic that is generated can be logged to be analysed later. MGEN log data can be used to calculate the throughput, packet loss, delay, and many other network performance parameters. MGEN runs on Unix-based and Win32 platforms. Later versions of MGEN also include a graphical user interface for users [102].

D-ITG

D- ITG, or the *distributive internet traffic generator*, is an active traffic generator and measurement analysis tool for testing over heterogeneous networks such as Wired, LAN, WLAN, GPRS, Bluetooth, etc. The tool analyses networks via generating network traffic on a packet-by-packet basis. DTIG runs on Linux, Windows and Linux familiar platforms. Both one-way delay and round-trip time meters can be calculated [66]. The tool allows generating multiple flows simultaneously, which means that both the sender and the receiver are multi-threaded applications, each thread managing a single flow. In addition, the sender and receiver can store information on sent and received traffic; it is, therefore, possible to isolate the device capability and network reliance [67]. D-ITG is able to operate on PC desktop, laptop/notebook, pocket PC and smart phones. Moreover, several applications such as (TCP, UDP, ICMP VOIP and HTTP) can be emulated. In addition to application type, the user is capable of specifying packet size, the inter-departure time of packets and various traffic types such as exponential, normal, or uniform [67].

5.2.1 Selection of an active measurement tool

Traffic Generators (Active)	**Operating Systems**			**Protocols**						**Options**						**Operative Mode**			**Logging Phase**			**Meter Types**			
	LINUX	***Windows***	***Unix like***	***UDP***	***TCP***	***ICMP***	***Telnet***	***VoIP***	***DNS***	***TTL***	***TOS***	***Priority***	***Seed***	***Duration***	***Delay***	***Single Flow***	***Remote***	***Multiple Flow***	***Sender***	***Receiver***	***Remote***	***OWD***	***RTT***	***Jitter***	***Packet Loss***
D-ITG	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RUDE/ CRUDE	X		X	X								X		X		X				X		X		X	X
MGEN	X	X	X	X						X	X			X		X		X		X		X		X	X
TG2	X		X	X	X					X	X			X		X		X		X		X		X	X
Iperf	X	X	X	X	X									X		X		X	X	X		X		X	X
NetProbe	X		X	X												X		X	X				X	X	X
TfGen		X		X												X			X						
Traffic	X	X	X	X	X									X		X		X							
MTOOLS	X		X	X							X		X	X	X	X		X	X	X		X	X	X	X
UDP Generator	X		X	X												X				X		X	X	X	X
MOSET	X	X			X																		X		
Netperf	X			X	X									X		X									
Ping	X	X	X	X		X										X			X				X		
Traceroute	X	X	X	X		X				X						X			X				X		
WLT		X			X											X			X	X			X		

Table 5 Comparison of active measurement tools, extended version of [66]

For the purpose of the thesis, it is necessary to find an active measurement tool to emulate the traffic between the Mobile Care Server and Mobile Viewer clients. In the table above, the active measurement tools are evaluated according to the compatible operat-systems (Windows, Linux, Unix like); the emulated protocols (UDP, TCP, VoIP, etc.); measurement options (such as duration of the measurement, *time-to-live* (TTL), traffic prioritisation, delay between sent packets, etc.); operative modes (such as single flow, multiple flow or operating remotely); logging phase, place where the logs are saved and finally, measured meter types. Regarding these criteria, D-ITG is concluded to be the best candidate for executing active measurements between the MCS and clients for a possible investigation in the future.

Following an initial understanding of the traffic between the Mobile Care Server and viewer clients by measuring passively, it is possible to identify the inter-departure time of TCP packets and the average TCP segment size. Thus, a similar flow can be generated by the D-ITG. Moreover, the D-ITG can run on Linux operating system like the Mobile Care Server and one-way delay, jitter and loss metrics can be calculated.

The line of code below, show how to execute the D-ITG for generating a single TCP flow, based on exercises in the D-ITG manual [67]. The traffic flow consists of a constant inter-departure time of 50 packets per seconds and uniformly distributed packet size between 500 and 1000 bytes with logging both in the sender and the receiver:

```
Receiver>./ ITGRecv -l recv_log_file
Sender>  ./ ITGSend -a 10.0.0.3 -rp 9501 -C 50 -u 500 1000 -l
send_log_file
```

Listing 3 Code for generating a TCP flow by D-ITG

The first line initiates the receiver on destination host (IP address: 10.0.0.3 port number: 9501) and creates a log file named r*ecv_log_file* at receiver side. The second line of code initiates the sender to send 50 packets per second until the receiver halts the communication by pressing Ctrl-C.

The user interprets the log file at receiver side by executing the following code:

```
Receiver>. / ITGDec recv_log_file
```

Listing 4 Code for generating the log file at receiver side

The sender can perform the same operation. Figure 6 demonstrates the output generated.

```
----------------------------------------------------------
Flow number: 1
From 10.0.0.4:34771
To 10.0.0.3:9501
----------------------------------------------------------
Total time = 10.001837 s
Total packets = 10000
Minimum delay = 3633.445701 s
Maximum delay = 3633.464808 s
Average delay = 3633.449749 s
Average jitter = 0.000706 s
Delay standard deviation = 0.001364 s
Bytes received = 7498028
Average bitrate = 5997.320692 Kbit/s
Average packet rate = 999.816334 pkt/s
Packets dropped = 0 (0 %)
----------------------------------------------------------
```

```
*************** TOTAL
RESULTS***************
Number of flows = 1
Total time = 10.001837 s
Total packets = 10000
Minimum delay = 3633.445701 s
Maximum delay = 3633.464808 s
Average delay = 3633.449749 s
Average jitter = 0.000706 s
Delay standard deviation = 0.036939 s
Bytes received = 7498028
Average bitrate = 5997.320692 Kbit/s
Average packet rate = 999.816334 pkt/s
Packets dropped = 0 ( 0 %)
Error lines = 0
```

Figure 6 An example output of D-ITG execution retrieved from [67]

As seen in Figure 6, one flow is observed and the elapsed time, total number of packets sent, the minimum, average and maximum delay, average jitter, delay standard deviation, and packet loss percentage are presented including many other metrics.

5.3 Network Emulation Tools

Network emulators, emulate the desired behaviour and characteristics of a network. Thus, the user is able to observe how an application performs under different network conditions. There are a few Open Source network emulators available on the market. In this section, the four most frequently mentioned and used, network emulators are described.

Dummynet

Dummynet is the most familiar network emulator developed by Luigi Rizzo [71]. The emulator is a part of the *FreeBSD* kernel, which can moreover be booted from a floppy image to be used on a regular PC. It emulates queue and bandwidth limitations, as well as product and protocol testing for delays. The aim is to determine the application performance in diverse network conditions via introducing packet losses, packet re-ordering and multi-path effects to the network.

Dummynet consists of two elements: pipes and queues. The pipe emulates a communication link considered as fixed-bandwidth channels whereas the queue represents travelling packets. The pipes and queues can be created dynamically and configured separately via the FreeBSD firewall [72]. With the help of *ipwf* rules, a simple firewall functionality in FreeBSD, it is possible to filter the incoming and outgoing packets, which travel through the pipe. In addition, multi-access, or point-to-point links can be emulated. Likewise, a cascade of pipes can be created to emulate multiple links and paths between the source and destination [73].

The pipes can be configured according to the required bandwidth, propagation delay, random or deterministic packet loss. Jitter, however; cannot be directly configured. In order to generate jitter, delay needs to be altered continuously. The PC running dummynet can operate as a host, router, or bridge [74].

The operation principle is as follows; the incoming packets are inserted into a queue if specified according to *first in first out* (FIFO) or *random error detection* (RED) or other algorithms. Re-ordering of packets can take place at this stage if desired. Packets are departed at a rate as specified bandwidth and delayed for a specified time, losses occur here if specified.

The following line of code creates a pipe that allows TCP traffic from any source to destination configured with a bandwidth of 500 Kbits/s, delay of 12 ms and loss rate of 0.02% [75]:

```
ipfw add pipe 1 ip from any to any
ipfw pipe 1 config bw 500 Kbits/s delay 12ms plr 0.02
ipfw add 1 allow tcp from any to any
```

Listing 5 Code for creating a filter with Dummynet

In the above code, a pipe is created which allows IP packets from any source to any destination. The second line configures this pipe with a limited bandwidth, delay and packet drop percentage. The third line generates a rule to allow only TCP traffic to the pipe.

NIST net

NIST net is another emulation tool yet works in the Linux environment, supports up to Linux 2.4 kernel, and created by Mark Carson of the *north american national institute of standards and technology* (NIST). NIST net is simpler and faster than Dummynet, however its installation is more complex and documentation is not up-to-date [77].

The tool introduces delay, delay variation (standard deviation), packet loss, duplication, re-ordering and bandwidth limitation. The tool can emulate congestion or bursty environments and simulates any protocol that is IP based. NIST net can operate under high data rates as opposed to other emulators. Furthermore, it offers a more precise delay modelling by utilising a separate high-speed clock.

The tool provides two user interfaces: a *command line* and a *graphical user interface*. The graphical user interface is very easy to use, setting the parameters to be affected by NIST net, however; maintains a cumbersome installation [79].

NIST Net offers a PC to be used as a router and consists of two parts: a run-time kernel emulator (NIST net module) and a set of user interfaces. The tool only affects the incoming traffic [77]. The incoming traffic is first intercepted by packet intercept code, and after it is passed to the NIST net module for the operation of packet matching (based on set parameters). Later NIST net passes the packet to Linux IP level code for scheduling delayed packets. Here the fast timer is used to control system clock and generate high-speed clock. NIST net does its own queuing and filtering like Dummynet [78].

The jitter is introduced to the environment by setting the *delsigma* (standard deviation) parameter of NIST net. The value assigned to delsigma is added or subtracted to the previous delay for generating next random delay. It is observed that successive delays or packet losses correlate with each other; hence, the new version of NIST net permits setting a correlation (linear) factor. The correlation factor is a number between –1 and 1, which represents no dependence to complete dependence on the previous delays [80].

NetEm

NetEm is another network emulator tool, which is an extension to NIST Net and developed by Stephen Hemminger in *open source development* (OSD) lab [83]. Most of the functions of NIST net are reused in NetEm[81]. NetEm emulates a wide area network by introducing delay, packet loss, duplication, corruption, re-ordering and rate control. NetEM works with Linux operating systems and has already been a part of the kernel for the version 2.6.7 and later [82].

NetEm's architecture is similar to NIST net where NetEm consists of two main parts as well: *kernel module* and the *command line* unit. The command line is a part of the *iproute2* package. Iproute2 is a collection of utilities for controlling traffic control in Linux and consists of several tools. *Traffic control* (tc) is on of the most important utility in iproute2 [103].

There used to exist a graphical user interface alike NIST net during 2005, however; the GUI [84] has been removed from public access nowadays, due to errors during usage. The command line parameters are given to NetEm using *tc* commands. Like Dummynet and unlike NIST net the kernel timers are limited to PC clock capability, which provides at best 1ms granularity. NIST net is protocol independent and also can be configured as a router or a bridge [81].

ALTQ

ALTQ, or the *alternative queuing*, is a tool developed by Kenjiro Cho for applying queuing disciplines for controlling the output traffic [87]. Like dummynet, ALTQ is a BSD Unix unit and many different queuing disciplines can be applied such as *random error detection* (RED), *weighted fair queuing* (WFQ), *class based queuing* (CBQ), etc., with most of them being for research purposes [88]. ALTQ allows the limitation of bandwidth, however; delay and jitter cannot be emulated. The tool also supports multiple streams and currently is part of the KAME project [89].

5.3.1 Selection of a network emulator

As observed in Table 6, the network emulators are compared according to the operating systems that they are compatible with, clock to be used for affecting the traffic, the affected traffic by the emulator either incoming or outcoming or both, as well as the meter types to set such as packet loss, delay, delay standard deviation, etc. Also, the comparison continues with the protocol types that they can emulate, the ease of the installation, the user interfaces they support (either command line or graphical), the protocols, and, finally, the number of flows to emulate.

After thorough comparison of the four network emulators, NetEm is selected to be utilised in this thesis research. NetEm is able to create delay, jitter, packet loss and is ready to be used in new kernels. In addition, it provides a user interface for filtering both incoming and outcoming traffics and finally, it can run on Linux operating systems [85].

Network Emulators	OS			Clock		Affected Traffic		Meter Types to set								Protocols						Installation			UI		# Flow	
	LINUX	***Windows***	***BSD***	***PC Clock***	***Separate high speed clock***	***Incoming***	***Outgoing***	***Packet duplication***	***Packet re-ordering***	***Mean Delay***	***Delay stdev***	***Correlation***	***Packet loss***	***Bandwidth***	***Packet Corruption***	***UDP***	***TCP***	***ICMP***	***Telnet***	***HTTP***	***Any Protocol***	***Simple***	***Complex***	***Moderate***	***Command line***	***GUI***	***Single***	***Multiple***
Dummy Net			X	X		X	X		X	X			X	X		X	X	X	X	X	X			X	X		X	X
NIST Net	X				X	X		X	X	X	X	X	X	X		X	X	X	X	X	X		X		X	X	X	
NetEm	X			X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	
AltQ			X	X			X							X		X	X	X	X	X				X	X			X

Table 6 Comparisons of network emulators

NetEm introduces a constant delay to the network interface *eth0* by the following first line of code [86], and the second line introduces jitter:

```
tc qdisc add dev eth0 root netem delay 100ms
tc qdisc add dev eth0 root netem delay 100ms 10ms 10%
```

Listing 6 Code for generating delay and jitter by NetEm

The above lines create a random distributed mean delay of 100 ms with a standard deviation (100ms ± 10ms) of 10ms and correlation (dependency on previous delays) factor of 10%.

The re-ordering of packets can also be introduced to the network. In the code below, the NetEM delays every 5th packet 10 ms for causing the re-ordering of packets:

```
tc qdisc add dev eth0 root netem gap 5 delay 10ms
```

Listing 7 Code for re-ordering of packets by NetEm

The packet loss, corruption (bit error) and duplication is generated by the following three lines:

```
tc qdisc add dev eth0 root netem loss 1%
tc qdisc add dev eth0 root netem duplicate 2%
tc qdisc add dev eth0 root netem corrupt .1%
```

Listing 8 Code for packet drop, duplication and corruption by NetEm

5.4 Summary

This chapter selected the most appropriate tools for performing live measurements and network emulation for Mobile Viewers. In order to select, an individual benchmark study is performed for passive measurement tools, active measurement tools and network emulators.

As a passive measurement tool, *tcpdump* is chosen for data capturing and *tcptrace* for calculating delay, jitter and packet loss over the captured data. As an active measurement tool, *D-ITG* is selected, however; this thesis study will not execute active measurements due to the limited scope of the present thesis. The live passive measurements are discussed in more detail in the next chapter. Finally, the network emulator, *NetEm* is preferred to emulate different network conditions to see the quality of the Mobile Viewer clients. The emulation results are discussed in Chapter 8.

6 Live Measurements in Different Networks

In this chapter, passive measurements are performed in different live networks such as LAN, WLAN, 3G and 2.5G to find out delay, jitter and packet loss for the Mobile Viewer client applications. Section 6.1 describes the measurement set up environments while section 6.2 discusses the measurements executed in each environment. The following chapter discusses the results and compares with the previous research results found in Chapter 4.

6.1 Measurement Set Ups

Four different measurement environments: LAN, WLAN, GPRS, UMTS are constructed to be measured by *tcptrace* as selected in Section 4.3.1.

6.1.1 LAN measurements

LAN measurements are performed by creating an isolated network with an *ethernet switch* as illustrated in Figure 7. *Tcptrace* is installed on a Red Hat Linux server, running the Mobile Care Server. The MCS is connected to the Central Network where the S/5 monitors exist by interface *eth0*, to the Unity Network where the Unity monitors reside by *eth2* and to the switch by *eth1*.

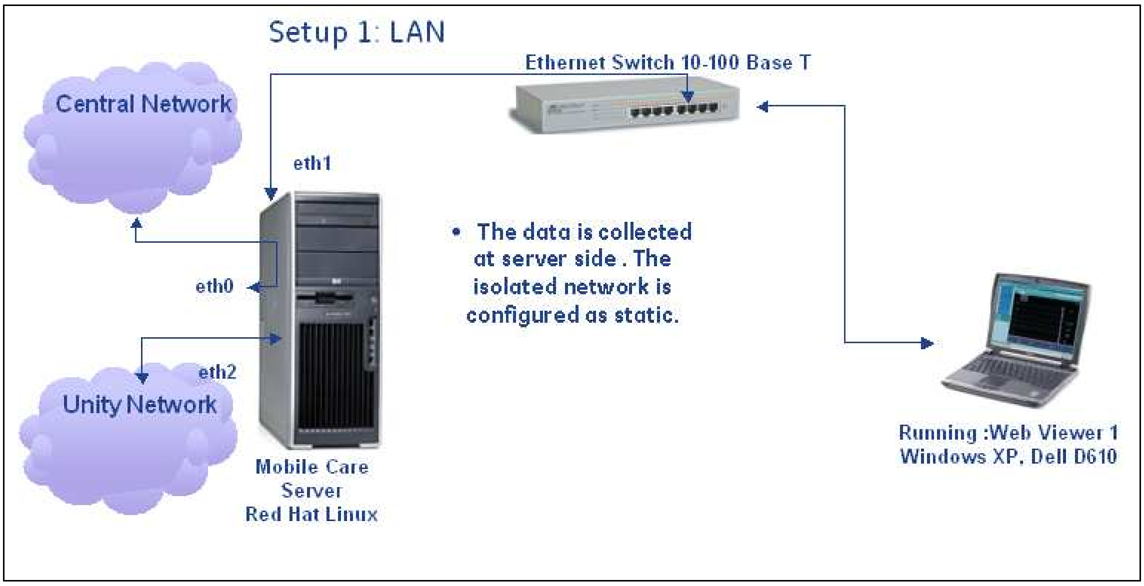

Figure 7 LAN set up for Web Viewer measurements

Approximately five centrals consisting of ten *S/5* monitors, ten *S/5* simulators and six *Unity* monitors are connected to the MCS. In this set up, only Web Viewer performance is analysed, because the Pocket and Cellular Viewer need a wireless connection to connect to the MCS. The distance between the server and Mobile Viewer clients is approximately 3-5 meters.

6.1.2 WLAN measurements

In this set up, all three-client performances are measured. The isolated wireless network is configured dynamically. The measurements were collected at the Linux server where the MCS resides. The Wireless network is generated by the Motorola Access Point. The Web, Pocket and Cellular Viewers are connected to the MCS through the WLAN. The same amount of patient monitors in the LAN set up is connected to the MCS. The distance between the MCS and Mobile Viewer clients is 3 to 5 meters.

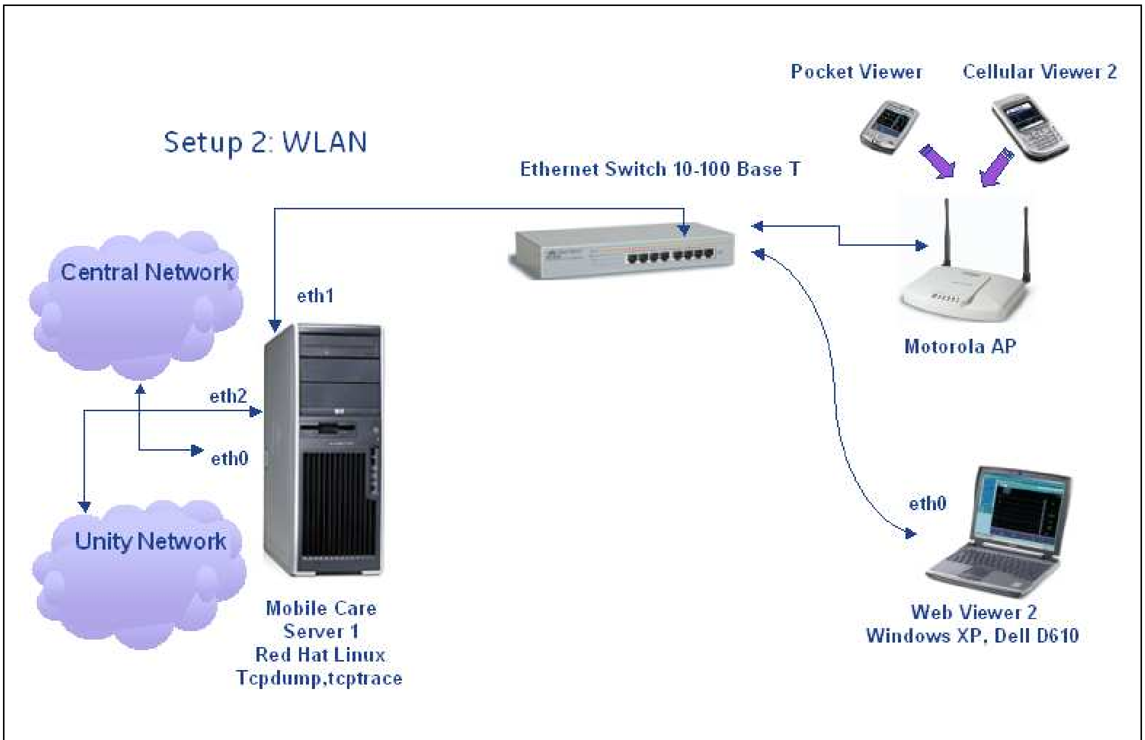

Figure 8 WLAN set up for Web, Pocket and Cellular Viewer measurements

6.1.3 GPRS measurements

GPRS Measurements are performed by executing *ssh* connections from a laptop, equipped with *tcptrace*, to a MCS located at the nearby building of GE Healthcare Corp. This MCS is connected to an *internet service provider* (ISP), and is therefore, accessible through the Internet as in Figure 9. All Mobile Viewer clients are tested in this set up. During measurements data is collected by *tcpdump* at the server to be analysed on the laptop. The MCS is connected to neither the *Unity* nor *S/5* networks. Instead of that, two *central* simulators are configured to be maintaining sixty-four S/5 monitors, and are connected to the Mobile Care Server. Because no Unity monitors were connected to the MCS, the *Unity trend download* tests could not be executed.

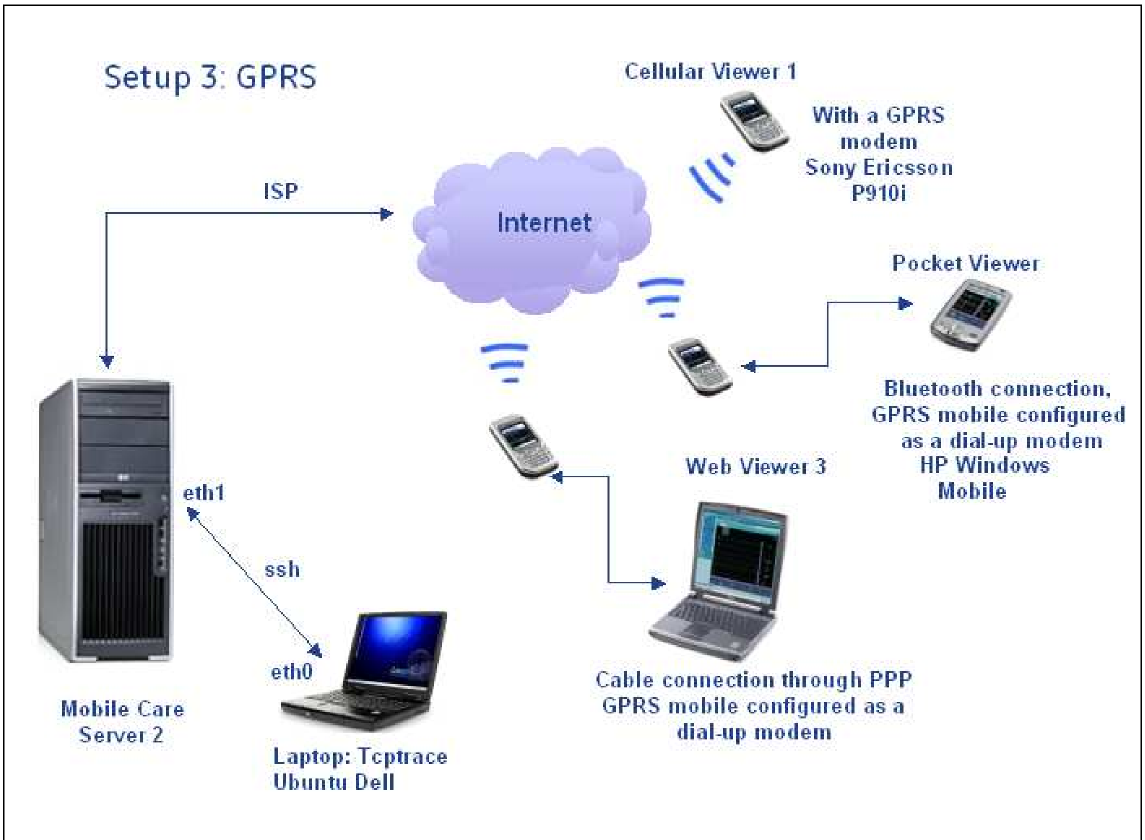

Figure 9 GPRS set up for Web, Pocket and Cellular Viewer measurements

6.1.3.1 Web Viewer

The laptop containing the Web Viewer client is connected to the GPRS phone with a cable though a *point-to-point protocol* (PPP) link. A dial-up connection is created using a GPRS phone modem; thus, the Web Viewer could connect to the MCS successfully. The measurements are collected at the MCS through an *ssh* connection.

6.1.3.2 Pocket Viewer

The Pocket Viewer is paired with the GPRS mobile via a *bluetooth* connection. Afterwards, a *dial-up* connection is created through the GPRS phone modem and the

Pocket Viewer client is connected to the MCS through the GPRS data connection. The measurements are collected on the MCS through an *ssh* connection as well.

6.1.3.3 Cellular Viewer

The GPRS phone (*Sony Ericsson P910i*) is connected to the MCS though GPRS data connection. The measurements are also collected on the MCS through an *ssh* connection.

6.1.4 UMTS measurements

The UMTS Measurements are carried out in a way very similar to the GPRS measurements. However, a 3G phone (*Nokia N81*) was utilised in this set up instead of a GPRS handset. Once more all Mobile Viewer clients are tested.

6.2 Measurements

In this section the performed tests on the Web Viewer, Pocket Viewer and Cellular Viewer are discussed individually.

6.2.1 Measurements on Web Viewer

There were four types of measurement tests performed on the Web Viewer clients. The four tests are listed and described in detail in the following.

1. 600 seconds 16 continuous waveforms of real patient monitors
2. 600 seconds 12 continuous ECG waveforms for a real patient monitor
3. 24 hour trend download of a real S/5 patient monitor
4. 24 hour trend download of a real Unity patient monitor

6.2.1.1 Continuous 16 Multi-View waveforms

In this test, *tcpdump* captures the continuous waveforms of *sixteen* patients' data in periods of 600 seconds. The test is executed ten times. This screenshot demonstrates a Multi-View option of the Web Viewer, consisting of, on ECG waveform and SpO2 digit for sixteen patients. The displayed patient demographic information is *synthetic*.

Figure 10 Continuous 16 Multi-View waveforms

6.2.1.2 Continuous 12-ECG waveforms

In Figure 11, *twelve ECG waveforms* of a patient are illustrated. This screenshot is an option in the Web Viewer where various types of waveform views can be displayed for a patient. The continuous waveform data is captured with 600 seconds of periods and executed ten times.

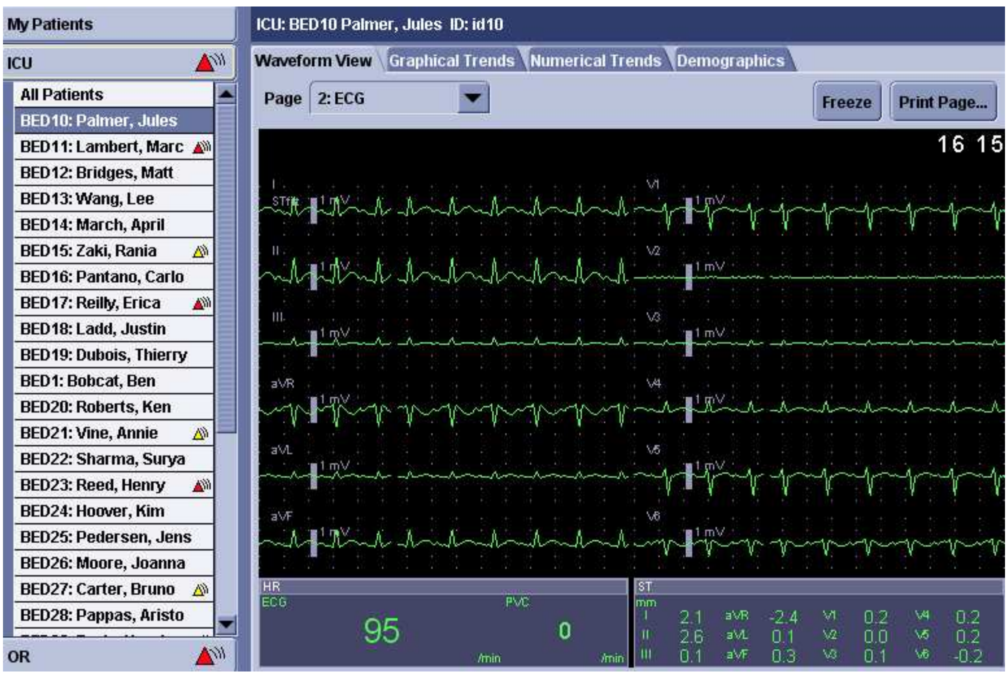

Figure 11 Continuous 12-ECG waveforms

6.2.1.3 Trend download

In the trend download test, nine columns of trend data of a patient are downloaded from the MCS. In Figure 13, *24 hours of trend data* collected at an *S/5* monitor is displayed. The capturing at the MCS is started when the *Numerical Trends* tab is clicked for a patient and stopped when all 24 hours of data appears. This test is executed ten times as well.

The same test is executed also for a *Unity* monitor. Both types of monitors: the S/5 and Unity trend data download needs to be analysed because the Unity trend data is observed to be downloaded much faster than the S/5.

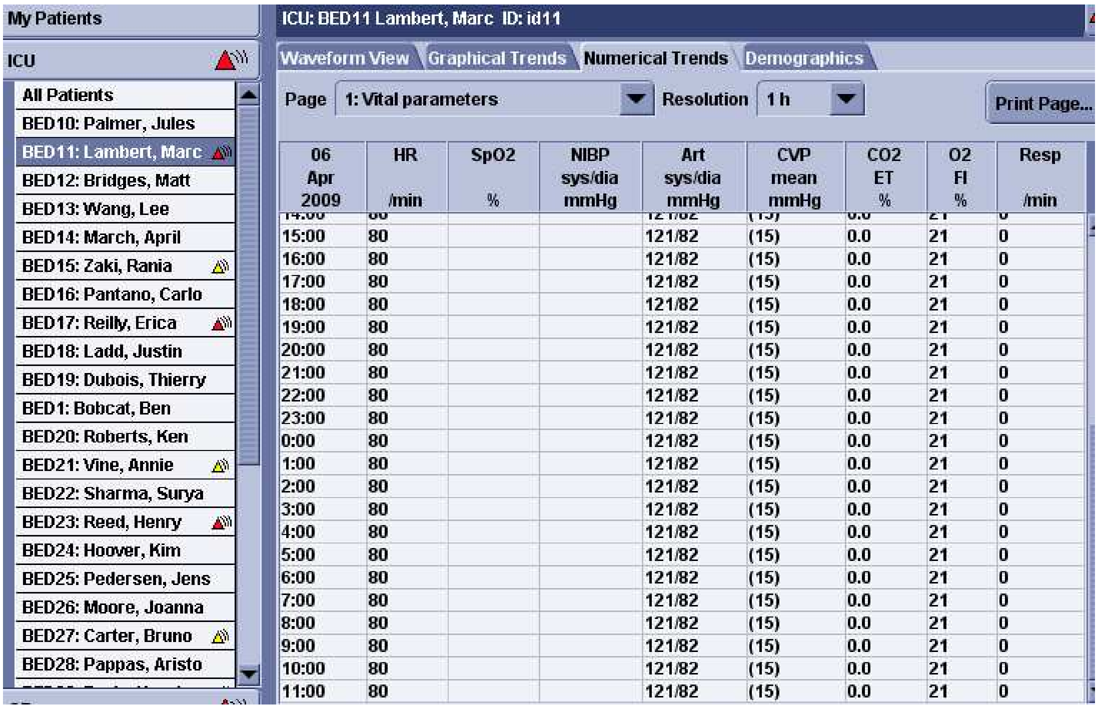

Figure 12 S/5 monitor 24h trend download

6.2.2 Measurements on Pocket Viewer

Three types of measurement tests are performed on the Pocket Viewer clients. The tests are listed below:

1. 600 seconds 4 continuous waveforms of real patient monitors
2. 24 hour trend download of a real S/5 patient monitor
3. 24 hour trend download of a real Unity patient monitor

6.2.2.1 Continuous 4 waveforms and trend download

Figure 13 illustrates the continuous *four waveforms* test and the *24h trend download* test for an S/5 monitor applied on the Pocket Viewer.

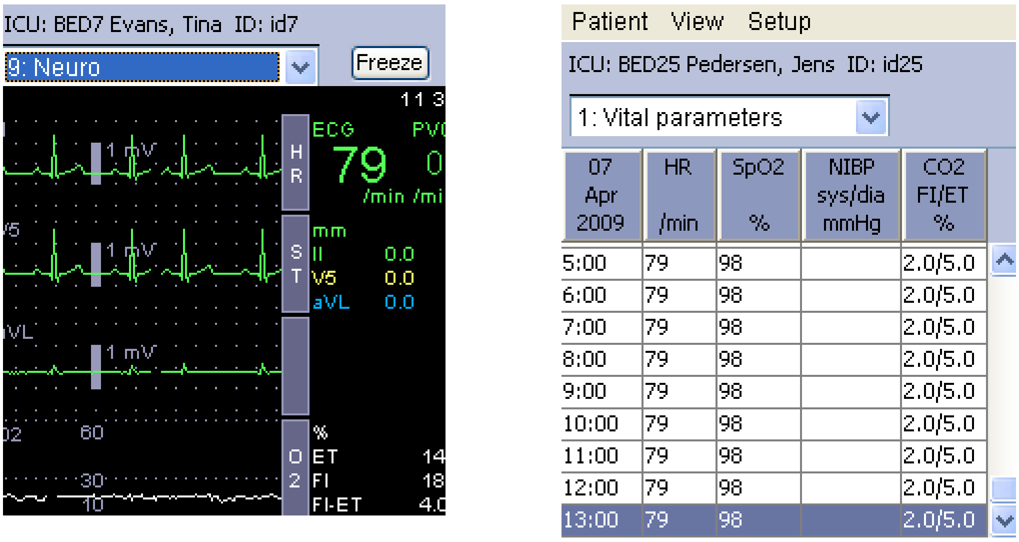

Figure 13 Continuous 4 waveforms and S/5 monitor 24h trend download

a) On the left of Figure 13, the continuous *four waveforms* test is depicted. The Pocket Viewer is able to display maximum of four waveforms; thus, in this test, the waveform view option *9: Neuro*, is configured to view three ECG and one O2 waveforms. The capturing is performed with 600 seconds of periods and executed ten times.

b) On the right of Figure 13, the *trend download* test is demonstrated. The Pocket Viewer can at most display four columns of trend data. In this test 24 hours of trend data for the S/5 and Unity monitors are downloaded individually. The capturing started when the *Numerical Trends* for that patient is selected and stopped when all 24 hours of trend data are filled. This test is executed ten times each.

6.2.3 Measurements on Cellular Viewer

Three types of measurement tests are performed on the Cellular Viewer clients, as listed below:

1. 600 seconds 3 continuous waveforms of real patient monitors
2. 24 hour trend download of a real S/5 patient monitor
3. 24 hour trend download of a real Unity patient monitor

6.2.3.1 Continuous 3 waveforms and trend download

Figure 14 illustrates the continuous *three waveforms* test and the *24h trend download* test for an S/5 monitor applied on the Cellular Viewer.

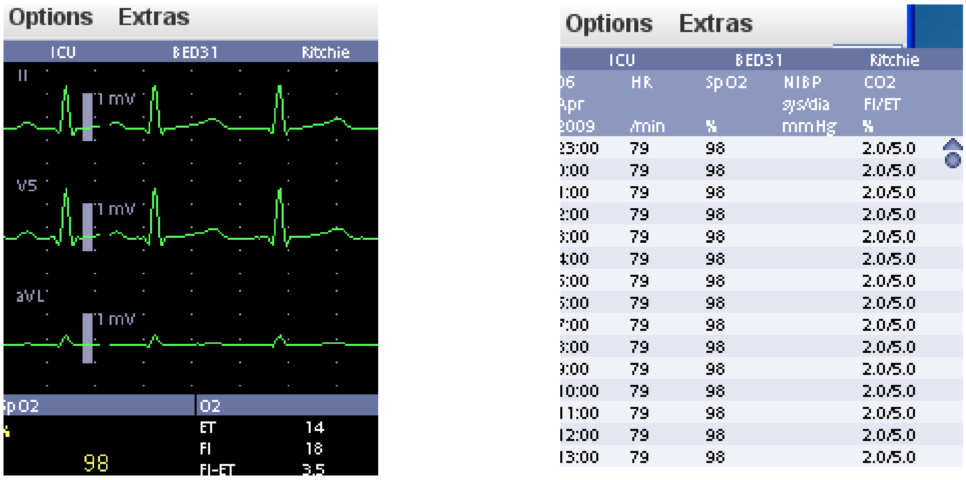

Figure 14 Continuous 3 ECG Waveforms and S/5 monitor 15h trend download

a) On the left of Figure 14, the continuous *three waveforms* test is depicted. The Cellular Viewer is able to display maximum of three waveforms. In this test, the waveform view option *9: Neuro*, is configured to view three ECG waveforms and two digits for SpO2 and O2 values. The capturing is performed with 600s of periods executed ten times.

b) On the right of Figure 14, the *trend download* test is demonstrated. In this test, one-hour resolution is used as in Pocket Viewer and Web Viewer tests. The trend download is captured till 15 hours of data is filled. Because when the slide window is moved to see further, a new TCP connection is initiated. This test is performed individually for S/5 and Unity monitors and executed ten times for each as well.

7 Live Measurement Results and Analysis

In this chapter, the entire the measurements performed on LAN, WLAN, 3G and 2.5G networks are presented. Section 7.1 presents the measurement results and Section 7.2 analyses them in the light of the previous research results discussed in Chapter 4.

The results and analyses are discussed individually for the Web Viewer, Pocket Viewer and Cellular Viewer applications, because the Mobile Viewer clients are diverse applications, which display different traffic characteristics.

In these measurements, server to client communication is taken into account, as the majority of the measurements are collected only at the server side. Moreover, client to server traffic mostly consists of acknowledgement packets, which results in a lot less traffic.

During testing it was observed that, after logging into the Web Viewer, and while viewing waveform pages, one TCP connection is created between the server and client. When another page is viewed, a new TCP connection is established. In addition, during download of all trend data, one TCP connection is created for the Pocket and Web Viewers while Cellular Viewer client establishes a new TCP connection each time the slide bar for viewing trend data a is clicked to display further.

Subsequent to capturing with *tcpdump*, the data is analysed with *tcptrace*. As discussed before in Section 5.1.1, the most important parameters to be analysed with tcptrace are: *RTT min, max, avg* for delay, *RTT stdev* for jitter, and *rexmit* (retransmissions) for packet loss. Since the whole traffic consists of TCP communications, the retransmission percentage makes perfect sense to be determined as the packet loss. In addition, the *inter departure time* (IDT) and the average segment size of TCP segments (*avg segm size*) is presented for future work to be measured actively via the selected active measurement tool in Section 5.2.1, which is, *D-ITG*. Lastly, throughput is also presented in bytes per second for interest.

7.1 Measurement Results

The live measurements results for the Web Viewer, Pocket Viewer and Cellular Viewer applications are presented in Sections 7.1.1 to 7.1.3.

7.1.1 Web Viewer results

In this section, the Web Viewer measurement results in LAN, WLAN, UMTS and GPRS environments are demonstrated.

	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
LAN Web Viewer	16wf	0.1ms	0.9ms	48.2ms	4.2ms	0.00%	410B	17 pps	6853Bps
	12ecg	0.3ms	0.71ms	41.1ms	1.89ms	0.00%	1137B	3 pps	4540Bps
	S/5 Trend	0.3ms	2.47ms	40.0ms	8.03ms	0.00%	286B	5 pps	1287Bps
	Unity Trend	0.3ms	3.83ms	40.1ms	6.44ms	0.00%	323B	22 pps	1999Bps
WLAN Web Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
	16wf	0.1ms	94.71ms	221.2ms	90.04ms	0.00%	740B	9 pps	6951Bps
	12ecg	0.1ms	56.34ms	219.5ms	80.05ms	0.00%	1021B	5 pps	4226Bps
	S/5 Trend	0.3ms	161.37ms	218.9ms	40.63ms	0.00%	265B	4 pps	869Bps
	Unity Trend	108.7ms	172.19ms	218.7ms	23.07ms	0.00%	193B	4 pps	539Bps
UMTS Web Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
	16wf	90.0ms	278.22ms	5223.2ms	179.29ms	0.33%	905B	7 pps	5812Bps
	12ecg	99.2ms	248.42ms	1697.6ms	137.42ms	0.33%	1018B	5 pps	5158Bps
	S/5 Trend	100.4ms	231.42ms	391.2ms	67.86ms	0.10%	570B	5 pps	2485Bps
GPRS Web Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
	16wf	578.3ms	1674.72ms	13764ms	511.2ms	0.44%	1162B	4 pps	4592Bps
	12ecg	600.3ms	1412.34ms	4580.8ms	300.96ms	0.15%	1107B	4 pps	3936Bps
	S/5 Trend	138.7ms	1037.33ms	4005.3ms	231.55ms	5.20%	734B	2 pps	717Bps

Table 7 Web Viewer measurement results in LAN, WLAN, UMTS and GPRS networks

As observed in Table 7, the average delay in LAN are in an order of magnitude of microseconds, which complies with measurements [90][91] whereas, WLAN and UMTS delay are in an order of milliseconds with UMTS conferring to almost the twice values of WLAN. The packet loss occurs with introducing a mobile technology into the environment such as UMTS and GPRS. In GPRS, the packet loss is higher than UMTS and delay values are in an order of magnitude of seconds as expected. The results are discussed in detail in Section 7.2.1.

7.1.2 Pocket Viewer results

The Pocket Viewer measurement results in WLAN, UMTS and GPRS networks are presented in Table 8.

As seen in Table 8, UMTS displays approximately twice the delay values of WLAN and nearly one third of the values achieved in the GPRS network. The packet loss starts to occur with UMTS and increases with GPRS.

	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
WLAN Pocket Viewer	4wf	3.0ms	112.65ms	345.2ms	68.85ms	0.00%	426B	2 pps	562Bps
	S/5 Trend	3.4ms	137.6ms	205.5ms	50.51ms	0.00%	151B	5 pps	556Bps
	Unity Trend	6.2ms	138.8ms	222.8ms	62.25ms	0.00%	139B	16 pps	552Bps
UMTS Pocket Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
	4wf	157.5ms	332.88ms	561.3ms	69.15ms	1.40%	309B	2 pps	474Bps
	S/5 Trend	175.4ms	323.45ms	595.4ms	59.08ms	1.50%	145B	3 pps	292Bps
GPRS Pocket Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
	4wf	512.1ms	959.13ms	1580.5ms	198.43ms	7.00%	452B	2 pps	460Bps
	S/5 Trend	600.4ms	951.45ms	2160.3ms	239.64ms	3.00%	256B	2 pps	270Bps

Table 8 Pocket Viewer measurement results in WLAN, UMTS and GPRS networks

7.1.3 Cellular Viewer results

Table 9 illustrates the measurement results for Cellular Viewers in WLAN, UMTS and GPRS networks.

	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IDT	Xput
WLAN Cellular Viewer	3wf	3.4ms	405.17ms	2012.6ms	337.33ms	0.40%	1166B	1 pps	997Bps
	S/5 Trend	3.4ms	111.91ms	344.4ms	114.67ms	0.00%	1269B	2 pps	2312Bps
	Unity Trend	3.3ms	37.65ms	112.6ms	37.12ms	0.00%	1302B	4 pps	3361Bps
UMTS Cellular Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IAT	Xput
	3wf	229.7ms	407.41ms	878.1ms	102.92ms	0.00%	1177B	1 pps	1031Bps
	S/5 Trend	180.8ms	454.35ms	823.9ms	148.89ms	0.00%	1188B	2 pps	1693Bps
GPRS Cellular Viewer	Server-->Client	RTT min	RTT avg	RTT max	RTT stdev	loss	AvgSeg	IAT	Xput
	3wf	639.6ms	2246.18ms	6227.9ms	1045.84ms	0.70%	1224B	1 pps	1078Bps
	S/5 Trend	767.9ms	1747.79ms	3570.6ms	578.09ms	1.00%	1472B	2 pps	1193Bps

Table 9 Cellular Viewer measurement results in WLAN, UMTS and GPRS networks

As illustrated in Table 9, packet loss occurs in WLAN environments for Cellular Viewers where no loss is experienced in the UMTS network as opposed to the Pocket Viewers and Web Viewers.

7.2 Measurement Analysis

In this section live measurement results for the Web Viewer, Pocket Viewer and Cellular Viewer applications are discussed and compared with the results found in Chapter 4, which presented the background research for the thesis investigation.

7.2.1 Web Viewers

7.2.1.1 Delay Analysis

Figure 15 illustrates the minimum average and maximum round-trip time delay analysis on Web Viewers, regarding the continuous *sixteen waveforms* test. The y-axis represents RTT in milliseconds while the x-axis classifies delay into LAN, WLAN, UMTS and GPRS networks.

According to the figure, the Web Viewer application experiences the highest delay in GPRS networks on average 1.6 seconds up to maximum of 13.7 seconds. There are disconnections from the MCS from time to time even though the client is stable during measurements. However, around 7 seconds of disconnection [5] and a long pause during data transfer [13] are not seen. These symptoms occur during handovers.

The average segment size is measured as 1162 bytes in the GPRS set up for *sixteen waveforms*. From the previous measurements performed on GPRS networks, a laptop to server measuring ICMP/UDP application under stable conditions [6] complies well with the measurements made during this thesis. They show that, the average RTT increases from 500 ms to 2142 s, when the packet size grows from 32 to 1450 bytes.

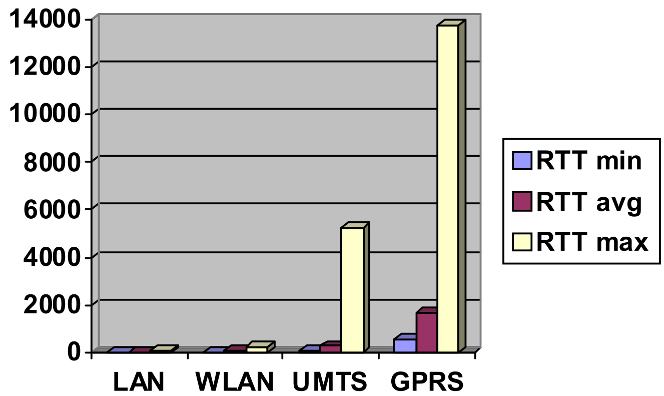

Figure 15 Delay analysis of Web Viewers

The second largest delay, which the Web Viewer application presents, is during UMTS network measurements. The *sixteen waveforms* test results in an average segment size of 905 bytes. The values conform to the measurements results in Study 6 [6] where the average RTT is observed between 150ms and 537 ms, when the packet size grows from 32 B to 1450 B. The WLAN network delay values are much lower compared to previous research results, most likely being due to the distance between server, client and access point being quite small [19][22].

7.2.1.2 Jitter Analysis

Figure 16 analyses the jitter observed while measuring the continuous *sixteen waveforms*, *twelve ECG waveforms* and *S/5 monitor trend download* with Web Viewers. The y-axis represents the RTT standard deviation values in milliseconds while the x-axis groups them into LAN, WLAN, UMTS and GPRS networks.

The round-trip time standard deviation is calculated to find out a spectrum of delay variance by using the formula presented in Section 3.1.2.3. Moreover, *NetEm*, the network emulator to be utilised for emulating different network conditions in Chapter 8, also introduces delay variance via standard deviation.

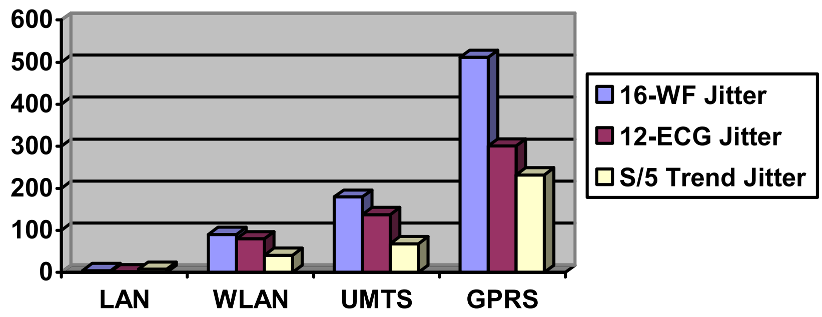

Figure 16 Jitter analysis of Web Viewers

According to the figure, the highest jitter is obtained during the *sixteen waveforms* test in GPRS networks. Naturally, this is due to the highest delay being acquired in GPRS networks during the *sixteen waveforms* tests. In general, except LAN, *16-WF* jitter is the highest, and then follows *12-ECG* jitter and *S/5 Trend* jitter.

7.2.1.3 Packet Loss Analysis

Figure 17 analyses the packet loss observed while measuring the continuous *sixteen waveforms, twelve ECG waveforms* and *S/5 monitor trend download* with Web Viewers. The y-axis represents the retransmission percentage (packet loss) while the x-axis categorizes them into LAN, WLAN, UMTS and GPRS networks.

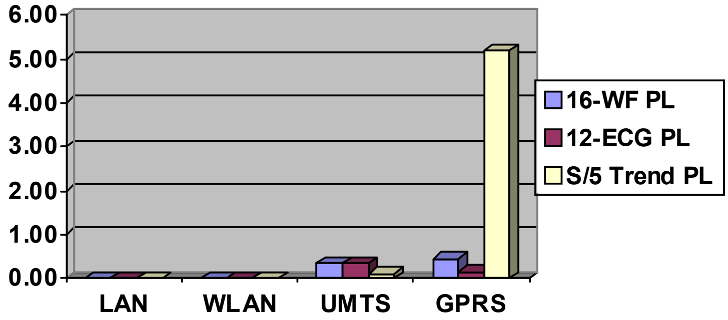

Figure 17 Packet loss analysis of Web Viewers

As observed in Figure 17 and as expected, the GPRS networks introduce the highest packet loss, which complies with the results in Study 12 [12] where loss ranges between 0.11% and 7.58%. The second highest loss is achieved in UMTS networks; moreover, the results comply with studies 1 [1] and 2 [2], where loss is between 0.06% and 0.12%. No packet loss is observed in either LAN or WLAN networks.

In terms of measurement tests, in the GPRS networks, the S/5 monitor trend download introduces the highest packet loss as opposed to the UMTS where the *sixteen waveforms* and the *twelve ECG waveforms* test introduce packet loss higher than the *S/5 trend download*.

7.2.2 Pocket Viewers

7.2.2.1 Delay Analysis

Figure 18 illustrates the minimum average and maximum round-trip time delay analysis on Pocket Viewers, regarding the *four waveforms* test. The y-axis represents RTT in milliseconds while the x-axis classifies delay into WLAN, UMTS and GPRS networks.

As seen in the figure, the highest delay is experienced in the GPRS environments as expected. No server disconnections were seen during measurements. GPRS delay being on average 959 ms, could reach up to 1600 ms with a 452 B average segment size.

Based on this, it can be seen that, GPRS shows better performance with smaller TCP segments.

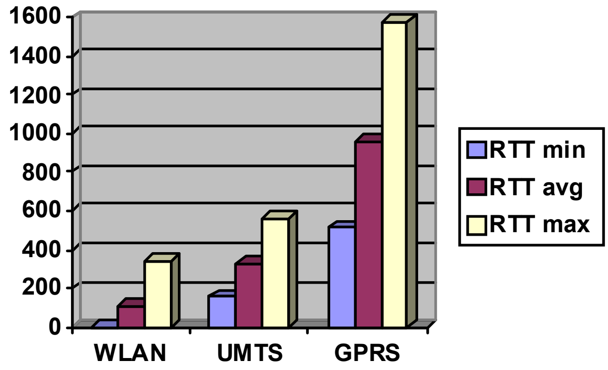

Figure 18 Delay analysis of Pocket Viewers

The UMTS delay ranges around 332 ms with an average segment size of 309 B for the *four waveforms* test, which is comparable with the results found in Study 7 [7] in lightly loaded cells where the average RTT ranges between 50 and 350 ms. In addition, WLAN demonstrates the smallest delay values, which ranges around 112 ms.

7.2.2.2 Jitter Analysis

Figure 19 analyses the jitter observed while measuring the continuous *four waveforms*, *S/5 monitor trend download* and *Unity monitor trend download* with Pocket Viewers. The y-axis represents the RTT standard deviation in milliseconds while the x-axis classifies delay into three types of tests for WLAN and two types of tests for UMTS and GPRS.

As seen in the figure, the Pocket Viewer application produces nearly similar jitter values for WLAN and UMTS networks. GPRS once more provides higher jitter values especially during the *S/5 monitor trend download*. As opposed to GPRS, the continuous *four waveforms* tests display the highest jitter values for WLAN and UMTS networks. Jitter is estimated to be around 70 ms for the *four waveforms* test and 60ms for the *S/5 trend download* test, which is much higher compared to the values found in Study 1 [1] for 33-65B packet sizes.

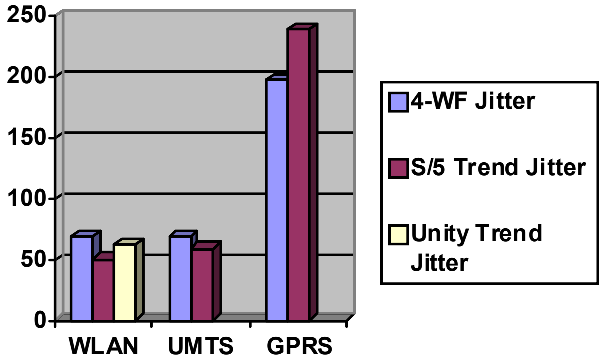

Figure 19 Jitter analysis of Pocket Viewers

7.2.2.3 Packet Loss Analysis

Figure 20 analyses the packet loss observed while measuring the continuous *four waveforms*, *S/5 monitor trend download* and *Unity monitor trend download* with Pocket Viewers. The y-axis represents the retransmission percentage while the x-axis classifies the packet loss into WLAN, UMTS and GPRS networks

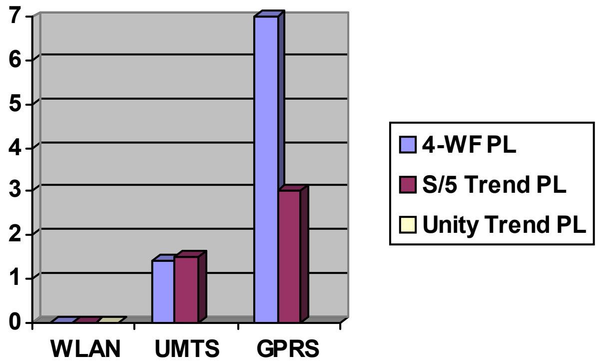

Figure 20 Packet loss analysis of Pocket Viewers

As seen in the figure, the highest packet loss is achieved in the GPRS networks ranging from 3% to 7% while UMTS shows better results ranging around 1.5% and there is no loss in WLAN. The packet loss results in UMTS, show higher values than in Study 8 [8] with 0.9% loss. However, Study 8 [8] executes tests with a smaller packet size than the Pocket Viewer application. Hence, the loss percentage being higher is not a surprise.

7.2.3 Cellular Viewers

7.2.3.1 Delay Analysis

Figure 21 illustrates the minimum average and maximum round-trip time delay analysis on Cellular Viewers, concerning the *three waveforms* test only. The y-axis represents RTT in milliseconds while the x-axis classifies delay into WLAN, UMTS and GPRS networks.

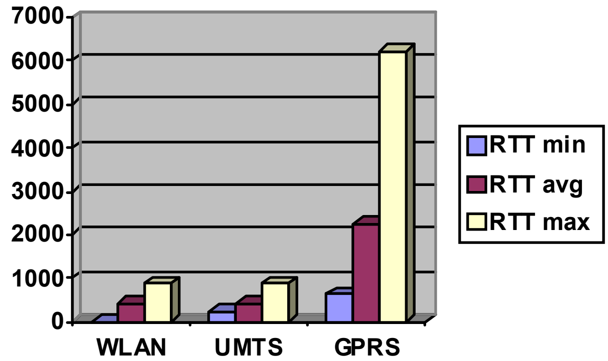

Figure 21 Delay analysis of Cellular Viewers.

The figure shows that the highest delay is achieved in the GPRS networks, being on average 2.3 seconds up to maximum of 6.2 seconds with an average segment size of 1224 bytes. Server disconnections are not detected. The average delay complies well with the results of Study 6 [6], where the authors observe a 2.1 seconds of average delay with 1450-byte sized packets.

The average delay in WLAN is quite similar to UMTS, being approximately 400 ms, which complies with the results found in Study 19 [19] with 128-byte probe packets in a medium loaded wireless LAN.

7.2.3.2 Jitter Analysis

Figure 22 analyses the jitter observed while measuring the continuous *three waveforms, S/5 monitor trend download* and *Unity monitor trend download* with Cellular Viewers. The y-axis represents the RTT standard deviation in milliseconds while the x-axis classifies delay in to three types of tests for WLAN and two types of tests for UMTS and GPRS

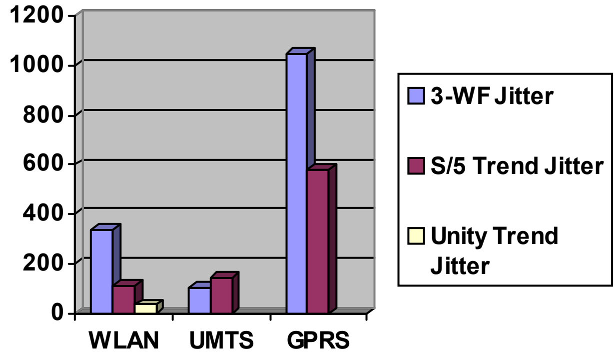

Figure 22 Jitter analysis of Cellular Viewers.

According to the figure, the highest jitter is obtained with the GPRS set up during *three waveforms test* on Cellular Viewers. Wireless LAN displays higher jitter values for *three waveforms* test than in UMTS. Compared to the *S/5 monitor trend download*, the *Unity monitor trend download* test results in much lower jitter, which is approximately 37 ms.

7.2.3.3 Packet Loss Analysis

Figure 23 analyses the packet loss observed while measuring the continuous *three waveforms*, *S/5 monitor trend download* and *Unity monitor trend download* with Cellular Viewers. The y-axis represents the retransmission percentage while the x-axis categorizes the packet loss into WLAN, UMTS and GPRS networks

Surprisingly UMTS experiences no packet loss while 0.4% loss occurs in WLAN during *three waveforms* test. GPRS only results in loss between 0.7% and 1%.

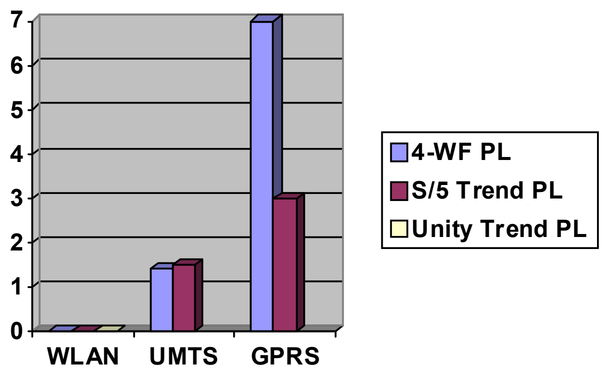

Figure 23 Packet loss analysis of Cellular Viewers

7.3 Summary and Result Evaluation

This chapter examined different networks, and reported the performances of the Mobile Viewer client applications for each network by carrying out passive live measurements. In addition, these measurements were compared with the previous research performed in Chapter 4. Based on these observations, it can be expected that LAN, WLAN and UMTS networks can meet the certain expectations of end-users. These expectations include things such as fast response to user selections, waveform flowing without any pause or interrupts, and no disconnections from the server. However, GPRS may bring difficulties when using the Web Viewers [15].

According to the measurement results and detailed analysis of the Web Viewer, Pocket Viewer and Cellular Viewer applications, generally, the GPRS network measurements, results in the highest delay (in an order of seconds), jitter and packet loss, and then follows UMTS as observed in previous research [5][6][8].

Server disconnections do appear in GPRS when the average RTT is estimated as 1.6 seconds with the Web Viewers as opposed to the UMTS and WLAN networks where server disconnections do not occur. However, after disconnections, connection between the Mobile Care Server and the client is re-established in at most two minutes.

The measurements in the WLAN showed much better results compared to previous research studies, which might be due to the measurement devices being in quite close proximity to each other.

Out of all the tests executed, continuous waveform viewing tests display, in most cases, the highest delay, jitter and packet loss values as opposed to trend downloads.

Packet loss usually occurs when the measurement environment consists of a mobile technology such as UMTS and GPRS as observed in previous studies [15][17].

Additionally, it can be emphasized once again that according to the findings in this thesis and the previous research, the delay in Local Area Networks are in an order of magnitude of microseconds [90][91] whereas UMTS [1][2][5][6] [7][8][9][10] and Wireless LANs [19][22] are in amount of milliseconds while GPRS, seconds [12][13][17].

Finally, the performed live passive measurement results can be a contribution to the other studies related to WLAN, LAN, GPRS and UMTS network performance measurements for medical data transmission focusing mainly on delay, jitter and packet loss. Also, the reliability of the results can be improved by utilising other client devices during measurements, and increasing the measurement periods.

8 Requirements for Mobile Viewer Clients

As mentioned earlier in Chapter 2, the main purpose of this thesis is to specify and verify the requirements for the Mobile Viewers in order to analyse the application performance before it is introduced to public networks. Before testing, it is not possible to guarantee that the applications will work well under all network conditions.

The live network measurements and analysis performed in Chapters 6 and 7 showed how Mobile Viewer clients work in different types of networks and the spectrum of values for delay, jitter and packet loss that are generated. However, the measurement set ups were too optimistic overall, which means that the Mobile Viewer clients were in close proximity to the server and/or access point, and/or clients were stable during measurements. Therefore, we need to present some network conditions that can help us to determine the maximum delay, jitter and packet loss values where the application performance quality starts to degrade.

As a result, the decision was made to place a network emulator between the Mobile Care Server and the Mobile Viewer clients in order to take control of the QoS components of the network. With the help of the network emulator, it is possible to introduce the needed amount of delay, packet loss and jitter into the network, allowing the tolerable limit values for delay, jitter and packet loss to be identified.

8.1 Emulated Wired Network Environment

In this section, an environment is built where the communication between the Mobile Care Server and the Web Viewer client is relayed through a network emulator. The utilised network emulator is *NetEm*, which is selected in Section 5.3.1.

8.1.1 NetEm wired network set up

As illustrated in Figure 24, a server running the Fedora Operating System, which contains the ready-to-use *NetEm* application, is placed between the MCS and Web Viewer client. NetEm application is already enabled in Kernel 2.6 and later. The server running the NetEm is configured as a router between two subnets which forwards the incoming traffic from the MCS to the Web Viewer client and vice versa. In addition, the server running the MCS and Laptop is equipped with *tcptrace*.

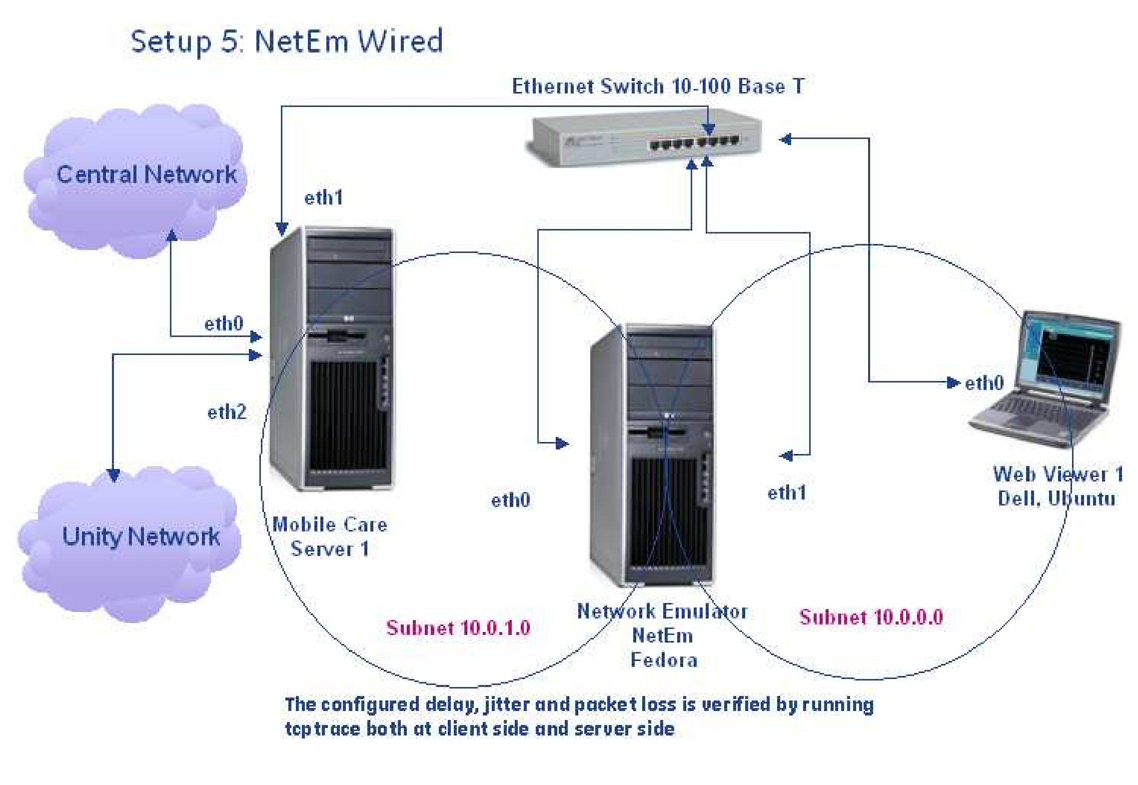

Figure 24 NetEm wired set up

The Mobile Care Server is connected to the same amount of patient monitors as in the live measurement tests presented in Chapter 6, and only the Web Viewer is tested in this set up. More information on network configuration is presented in Appendix A.1.

8.1.2 NetEm rules

Following a successful set up of the environment, the *NetEm* application is tested. A NetEm rule introducing delay is set on *eth1* interface of the Fedora server and delay is verified by pinging from the MCS to the Web Viewer client and vice versa.

8.1.2.1 Delay

Initially, only delay is generated by the first command line below. The delay is altered between 0 and 6 seconds via the second line and finally deleted via the last line of code:

```
Netem rule: tc qdisc add dev eth1 root netem delay ___ ms
Netem rule: tc qdisc change dev eth1 root netem delay ___ ms
Netem rule : tc qdisc del dev eth1 root netem delay ___ ms
```

Listing 9 NetEm codes for delay

8.1.2.2 Jitter

Soon after, jitter is applied between 25 and 750 ms together with some delay. The RTT standard deviation is applied and modified by the following three lines of NetEm rules. The second space for ms stands for the requested deviation value for delay:

```
Netem rule: tc qdisc add dev eth1 root netem delay ___ ms___ms
Netem rule: tc qdisc change dev eth1 root netem delay ___ ms___ms
Netem rule: tc qdisc del dev eth1 root netem delay ___ ms___ms
```

Listing 10 NetEm codes for jitter

8.1.2.3 Packet loss

The packet loss, *retransmission percentage*, is generated by the following NetEm rules. The retransmission percentage ranged between 0% and 50%:

```
Netem rule: tc qdisc add dev eth1 root netem loss ___%
Netem rule: tc qdisc change dev eth1 root netem loss ___%
Netem rule: tc qdisc del dev eth1 root netem loss ___%
```

Listing 11 NetEm codes for packet loss

8.1.3 User measurements on Web Viewer

During each delay, jitter and packet loss test, it was difficult to come up with a solid explanation for the Web Viewer client application behaviour. Therefore, inspired by VoIP application quality assessments made using the *mean opinion score* (MOS) test [98], it was decided to grade the quality of the Web Viewer client in each environment by user measurements via utilising a stopwatch.

The following user measurements were performed in terms of seconds:

1. **Start up**: The stopwatch is started when the Web Viewer application shortcut is clicked and stopped when the Login page appears. The stopwatch is again started after the login information is filled and logged in, and stopped when the Web Viewer application appears. These two measurements are separated by the plus sign.

2. **16 waveforms appearance**: The stopwatch is started when “All Patients” is clicked and stopped when all *sixteen waveforms* appear.

3. **Maximum pause interval:** While viewing continuous *sixteen waveforms*, the waveforms pause for couple of seconds when there is a high delay, jitter or packet loss in the network. The longest pause is measured using a stopwatch.

4. **S/5 24h monitor download**: The stopwatch is started when the numerical trends of an *S/5* monitor is clicked and stopped when all data are filled.

5. **Unity 24h monitor download**: Step 4 is carried out for a *Unity* monitor.

6. **12-ECG waveforms appearance**: The stopwatch is started when "ECG" waveform view is clicked and stopped when *twelve ECG waveforms* appear.

7. **Maximum pause interval (12-ECG)**: Step 3 is executed this time for the *twelve ECG waveforms*.

8. **Server disconnections:** This measurement denotes whether the client is disconnected from the Mobile Care Server or not.

9. **Unity waveforms appearance:** The stopwatch is started when a *Unity* type patient monitor is selected and stopped when the waveforms for that monitor appear.

10. **S/5 waveforms appearance:** Step 9 is carried out for an *S/5* monitor.

Note: Steps 2, 4, 5, 6, 9 and 10 are carried out three times for each delay, jitter and packet loss analysis in order to present an average value.

8.1.3.1 Quality assessment

A *MOS* like subjective measurement is performed to grade the quality of Web Viewer application by user perspective. However, boundaries had to be drawn somewhere to assess the quality. In this case, the *sixteen waveforms* appearance time is taken into account. The quality is expressed based on how fast the *sixteen waveforms* appear.

1. **Excellent**: 16 waveforms appearance time is between 2-3 seconds.

2. **Good:** 16 waveforms appearance time is between 4-6 seconds.

3. **Moderate:** 16 waveforms appearance time is between 7-10 seconds.

4. **Poor:** Server disconnections happen. Even a single disconnection from the Mobile Care Server is considered as a poor quality.

8.1.4 Results for Web Viewers

As we observe Table 10, in the leftmost columns, the configured delay, jitter and packet loss by the network emulator is presented. Next follows the results of 10 user measurement tests and then appears the assessed quality which is excellent, good, medium and poor.

In addition, the configured delay, jitter and packet loss is verified by capturing data by *tcpdump* at the server side while viewing the continuous *sixteen waveforms* between 300 and 600 seconds. Later, the captured data is analysed with *tcptrace* and the results are placed to the rightmost three columns of Table 10, as the actual delay, jitter and packet loss in the network. This is performed for each row in Table 10 where each row displays a different network condition created via *NetEm*; thus, the configured delay, jitter and packet loss are verified.

It can be noted that the actual delay, jitter and packet loss at the rightmost columns can differ from the configured values. Because the Mobile Viewer applications already generate some delay, jitter or packet loss when there is no addition of these to the network. Hence, the actual delay is observed to be higher, whereas jitter is either more or less. However, the perceived packet loss seems to be in parallel with the configured value until 20%.

8.1.4.1 Limit delay, jitter and packet loss values for Web Viewers

The limit values are chosen to be the data points when the quality becomes poor. The grey rows in Figure 10 separate the delay, jitter and packet loss configurations by *NetEm*.

If the table is examined closely, the first poor quality, which is marked with red, denotes the *limit delay* when the application quality becomes poor. The actual delay value, which is adjacent to the assessed quality (in the rightmost three columns) is decided to be the limit delay. The limit value for *delay* is observed as *4 seconds***.** This is when the server disconnections start to appear, interruptions in continuous drawing of waveforms begin and the packet loss starts.

The second poor quality in the middle of the table, again indicated in red denotes the *limit jitter*. Again the actual jitter value, the one to the right of the assessed quality is considered as the limit jitter. The limit value for *jitter* is detected to be approximately *450 milliseconds* together with a delay of 1.2 seconds and packet loss of 7 percent.

The third poor quality at nearly the end of the table, marked with red, is the limit packet loss. Once again, the actual packet loss right next to the assessed quality is determined as the limit packet loss. The limit value for *packet loss* is perceived as *20 percent*. The application performance until 20% loss was still convenient. However, at higher than 20 percent of packet loss, server disconnections start to happen.

8.1.4.2 . Comparison with live measurement results

While executing user measurement tests with the network emulator, we wanted to see if there are any values in Table 10 similar to the average delay, jitter and packet loss values found for the *sixteen waveforms* measurement tests in Chapter 6. Thus, it is possible to observe whether the Web Viewer application displays the same performance in the live measurement tests and the equivalent network condition created by the network emulator.

The row highlighted with cyan, presents the average Web Viewer delay, jitter and packet loss observed in LANs for the *sixteen waveforms* test as obtained in Section 7.1.1. The application quality is excellent and the user measurement results are comparable. For instance, no server disconnections occur and the *sixteen waveforms* appearance time is similar.

The pink row represents the approximate delay, jitter and packet loss generated by the Web Viewer application in 3G networks, which is still an excellent quality. The client application displayed similar user measurement results with the presented values.

Lastly, the yellow row stands for the delay, jitter and packet loss values observed in GPRS networks by the Web Viewer application. As detected in live tests, the quality is poor with regarding server disconnections.

CONFIGURED VALUES(NetEm)			USER MEASUREMENTS on WEB VIEWER (seconds)											ACTUAL VALUES (tcptrace)		
Delay ms	Jitter ms	PacketLoss%	Start up (s)	16wf app.	max pause int.	S/5 24h trend dw	Unity 24h trend dw	12ecg app.	max pause int.	Server disc.	Unitywf app.	S/5wf app.	Quality	Delay ms	Jitter ms	PacketLoss%
0	0	0	6+22	2	0	10	2	3	0	no	1	1	Excellent	0.9	3.9	0
300	0	0	7+24	2	0	11	3	3	0	no	1	1	Excellent	307	11.5	0
500	0	0	12+27	2	0	11	3	3	0	no	1	1	Excellent	525	33.2	0
1000	0	0	23+30	4	0	12	3	5	0	no	2	2	Good	1037	54.5	0
1500	0	0	31+36	6	0	13	5	6	0	no	3	3	Good	1561	64.3	0
2000	0	0	35+42	7	0	15	6	7	0	no	3	3	Moderate	2072	63.3	0
3000	0	0	55+51	9	0	16	8	7	0	no	7	7	Moderate	3081	65.9	0
4000	**0**	**0**	**70+60**	**11**	**9**	**20**	**9**	**10**	**4**	**yes**	**8**	**8**	**Poor**	**4027**	**57.4**	**2**
5000	0	0	92+70	20	16	22	10	12	6	yes	11	11	Poor	5025	66.5	3
6000	0	0	103+81	N/A	N/A	22	10	17	8	yes	12	12	Poor	6021	52.2	3
300	25	0	8+24	2	0	11	3	3	0	no	1	1	Excellent	308	38.3	0
500	25	0	12+27	2	0	11	3	4	0	no	1	1	Excellent	514	42	0
300	50	0	8+24	2	0	12	3	4	0	no	2	2	Excellent	313	75	0
500	50	0	12+27	2	0	12	3	4	0	no	2	2	Excellent	526	83	0
500	100	0	12+28	2	0	12	4	4	0	no	2	2	Excellent	554	127	0
1000	100	0	24+30	4	0	13	5	5	0	no	2	2	Good	1036	135	0.1
250	150	0	12+29	3	0	12	4	4	0	no	2	2	Excellent	296	172	0
500	150	0	12+29	3	0	12	4	4	0	no	2	2	Excellent	569	211	0.4
1000	150	0	24+30	4	0	13	5	5	0	no	2	2	Good	1045	187	0
1000	250	0	24+31	5	0	14	5	5	0	no	3	3	Good	1156	254	0.3
1500	250	0	31+36	6	0	15	5	6	0	no	3	3	Good	1570	251	0.4
1500	500	0	37+37	9	3	15	5	7	0	no	4	4	Moderate	1718	351	0.5
1000	500	0	24+33	5	2	13	5	5	2	no	3	3	Good	1264	338	2
1000	525	0	25+34	6	4	14	6	6	3	no	4	4	Moderate	1235	390	3
1000	**550**	**0**	**26+34**	**11**	**6**	**17**	**10**	**8**	**6**	**yes**	**4**	**4**	**Poor**	**1244**	**454**	**7**
1250	550	0	29+45	12	6	17	11	8	6	yes	5	5	Poor	1556	498	5
1500	675	0	40+80	15	7	18	12	9	7	yes	5	5	Poor	1898	423	2
1000	750	0	30+36	25	7	20	14	10	6	yes	7	7	Poor	1408	550	6
1500	750	0	94+105	N/A	N/A	N/A	N/A	N/A	N/A	yes	N/A	N/A	Poor	1961	561	4
0	0	0.1	6+22	2	0	11	2	3	0	no	1	1	Excellent	1.1	4.9	0.1
0	0	0.3	6+22	2	0	11	2	4	0	no	2	2	Excellent	0.8	3.2	0.3
0	0	0.5	6+22	2	0	11	2	4	0	no	2	2	Excellent	0.8	3.7	0.5
0	0	1	6+22	2	0	11	2	4	0	no	2	2	Excellent	1	4.3	1
0	0	2	6+22	2	0	12	2	4	0	no	2	2	Excellent	1.1	4.5	2
0	0	3	6+22	2	0	12	2	4	0	no	2	2	Excellent	0.9	3.2	3
0	0	5	6+22	2	0	12	2	4	0	no	2	2	Excellent	0.8	3	5
0	0	10	6+22	2	0	12	2	4	0	no	2	2	Excellent	0.8	3	10
0	0	15	6+22	2	0	12	2	4	0	no	2	2	Excellent	0.7	2.5	15
0	**0**	**20**	**6+22**	**2**	**2**	**12**	**2**	**4**	**0**	**yes**	**2**	**2**	**Poor**	**0.9**	**3.7**	**20**
0	0	25	8+28	3	4	13	3	4	0	yes	2	2	Poor	0.9	3.4	24
0	0	30	10+33	3	6	13	3	4	2	yes	2	2	Poor	1	4.1	28
0	0	40	10+72	5	6	N/A	4	5	5	yes	3	3	Poor	1.3	5.4	36
0	0	50	26+145	N/A	N/A	N/A	N/A	N/A	N/A	yes	N/A	N/A	Poor	1.1	1.9	49

Table 10 User measurements for Web Viewers

8.2 Emulated Wireless Network Environment

This time a wireless environment is emulated between the Mobile Care Server the viewer clients so that the Pocket Viewer and the Cellular Viewer application performances can be tested. The communication is relayed through the network emulator, *NetEm*, as in Section 8.1, which covered the wired environment.

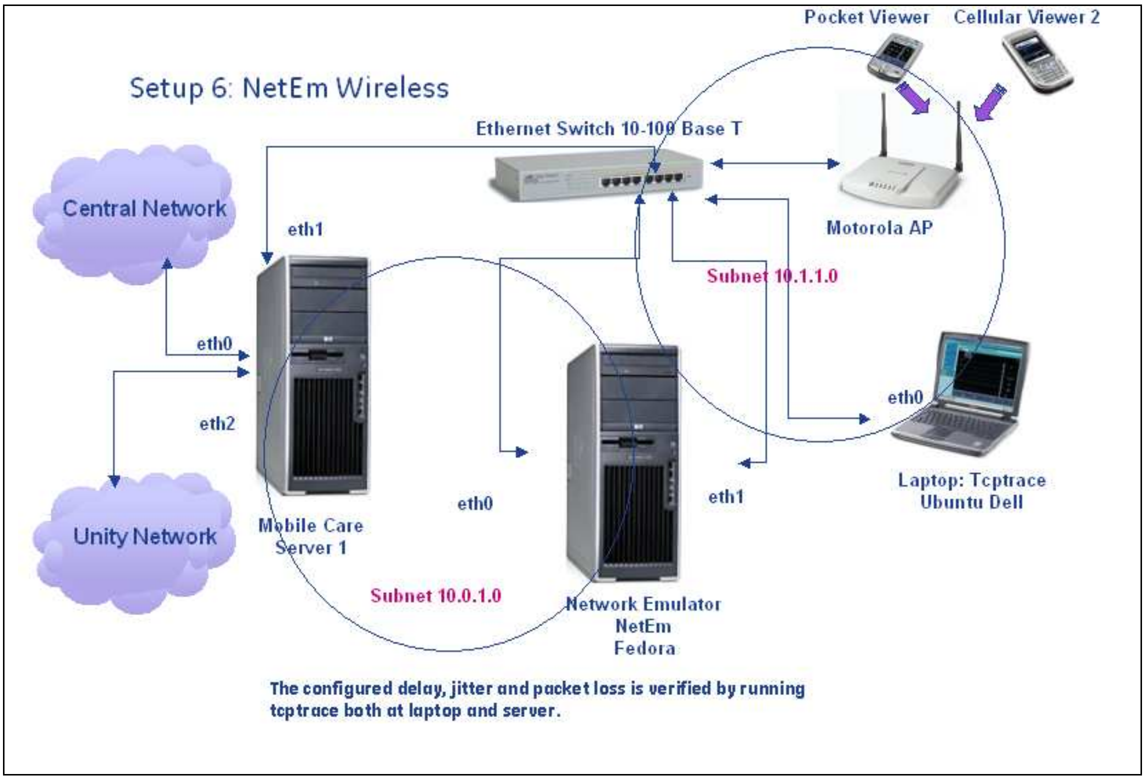

Figure 25 NetEm wireless set up

8.2.1 NetEm wireless network set up

As demonstrated in Figure 25, the same server running the NetEm application is placed between the MCS and the Motorola Access Point. The server acts as a *router* forwarding traffic from Subnet 10.0.1.0 to Subnet 10.1.1.0 and vice versa. The access point assigns IP addresses to the viewer clients in a specified range. The necessary static routes are configured in the routing table of the Access Point. Additionally, the server running the MCS and the laptop is equipped with *tcptrace* to verify the configured delay, jitter and packet loss by NetEm in each subnet. The set up was confirmed by configuring the NetEm to apply delay on *eth1* then pinging from the MCS to laptop, laptop to MCS, MCS to Pocket viewer and MCS to Cellular Viewer, so that the applied delay is verified and the set up works successfully.

8.2.2 User measurements on Pocket Viewer

User measurements on the Pocket Viewer are carried out in a way quite similar to the Web Viewers. However, the *twelve ECG waveforms* appearance and the related *maximum pause interval* tests are missing due to the function being unavailable in the Pocket Viewers. In addition, the *four waveforms* appearance test is conducted instead of *sixteen waveforms*, since the Pocket Viewer can at most view four waveforms. The *NetEm* rules for delay, jitter and packet loss are identical.

8.2.2.1 Quality assessment

The same technique is used as in Section 8.1.3.1. Unlike the Web Viewers, the continuous *four waveforms* appearance test result is taken into account and the time interval for the quality assessment is changed, because the Pocket Viewers are more tolerant to delay than the Web Viewers.

1. **Excellent**: 4 waveforms appearance time is between 1-4 seconds.
2. **Good:** 4 waveforms appearance time is between 5-7 seconds.
3. **Moderate:** 4 waveforms appearance time is between 8-12 seconds.
4. **Poor:** Server disconnections happen. Even a single disconnection from the Mobile Care Server is considered as a poor quality.

Note: When the application performance became really poor, a few user measurement tests could not be carried out due to the application response slowness or server disconnections; hence, those failed steps were marked as *not applicable* (N/A).

8.2.3 Results for Pocket Viewers

Table 11 presents the user measurement results for the Pocket Viewers. From left to right, the configured delay, jitter and packet loss values by *NetEm* are displayed, right after, the utilised eight user measurement results are placed. Then the assessed application quality and the actual delay, jitter and packet loss measured by *tcprace* are presented.

Still, the configured delay, jitter and packet loss values by NetEm may not be exactly alike with the resulted delay, jitter and packet loss. It is seen that delay constantly increases, whereas jitter is either raised or reduced and the resulted packet loss appears to be half of the configured loss, which might be caused by the set up configurations.

8.2.3.1 Limit delay, jitter and packet loss values for Pocket Viewers

The limit QoS parameter values which end-users can endure are as follows. Again the rightmost measured delay, jitter and packet loss values where the quality becomes poor are taken into account (red marked text).

Delay limit is detected to be *7 seconds.* More than 7 seconds server disconnections start, discontinuity in continuous waveforms begin and packet loss starts.

Jitter limit is identified as approximately *2.3 seconds* together with a delay of 3.9 seconds and a packet loss of 0.4 percent.

Packet loss limit is *14 percent.* It seems that the application still responds with a small delay, however; server disconnections occur. Despite the configured loss being 25 percent, analysis with *tcptrace* shows a packet loss of 14 percent.

8.2.3.2 Comparison with live measurement results

The approximate delay, jitter and packet loss generated by the Pocket Viewer application in live 3G networks is once again represented in the pink row in Table 11. The quality is observed to be excellent both in the live 3G networks and the corresponding row in the configured wireless network environment with *NetEm*.

The green highlighted row demonstrates the observed average Pocket Viewer application delay, jitter and packet loss in WLANs as presented in Section 7.1.2. The application displays an excellent performance with no packet loss in both conditions.

Finally, the delay, jitter and packet loss values observed in GPRS networks by the Pocket Viewer application for the continuous *four waveforms* test is highlighted with yellow, which is an excellent performance as well. There are no server disconnections when delay is around 1 second and jitter 225 milliseconds in the configured network and the similar delay and jitter in the live GPRS network.

CONFIGURED VALUES(NetEm)			USER MEASUREMENTS on POCKET VIEWER (seconds)									ACTUAL VALUES (tcptrace)		
Delay ms	Jitter ms	PacketLoss%	Start up (s)	4wf app.	max pause int.	S/5 24h trend	Unity 24h trend	Server disc.	Unitywf app.	S/5wf app.	Quality	Delay ms	Jitter ms	PacketLoss%
0	0	0	12+46	1	0	12	2	no	1	1	Excellent	106	74	0
200	0	0	12+48	1	0	12	2	no	1	1	Excellent	303	77	0
300	0	0	12+50	1	0	12	2	no	1	1	Excellent	401	75	0
500	0	0	12+52	2	0	13	3	no	2	2	Excellent	617	68	0
1000	0	0	12+56	3	0	15	4	no	3	3	Excellent	1036	44	0.001
1500	0	0	12+65	5	0	16	5	no	5	5	Good	1561	64	0.004
2000	0	0	12+70	6	0	17	7	no	5	5	Good	2050	57	0
3000	0	0	12+84	7	0	18	7	no	6	6	Good	3062	60	0.001
4000	0	0	12+100	9	1	19	10	no	7	7	Moderate	4074	118	0.001
5000	0	0	12+105	10	2	24	12	no	9	9	Moderate	5095	212	0.001
6000	0	0	12+122	12	6	27	15	no	11	10	Moderate	6096	164	0.002
7000	**0**	**0**	**12+136**	**30**	**7**	**28**	**16**	**yes**	**22**	**23**	**Poor**	**7078**	**263**	**0.15**
8000	0	0	12+150	60	21	N/A	N/A	yes	40	39	**Poor**	8045	62	6
10000	0	0	12+224	N/A	132	N/A	N/A	yes	28	25	**Poor**	10039	56	0.13
300	25	0	12+49	2	0	12	2	no	2	2	Excellent	400	81	0
500	25	0	12+53	3	0	12	3	no	3	3	Excellent	615	79	0
500	50	0	12+54	3	0	12	3	no	3	3	Excellent	600	90	0.001
500	125	0	12+54	3	0	12	3	no	3	3	Excellent	604	165	0.001
1000	200	0	12+60	4	0	13	4	no	4	4	Excellent	1107	225	0.001
1000	500	0	12+63	5	0	14	5	no	5	5	Good	1140	539	0
1500	500	0	12+65	7	0	17	6	no	5	5	Good	1560	558	0
1500	750	0	12+69	7	0	20	7	no	5	5	Good	1695	796	0
2000	750	0	12+67	7	0	17	6	no	7	6	Good	2071	794	0
2000	1000	0	12+73	7	0	20	7	no	7	6	Good	2109	1073	0
2000	2000	0	12+73	10	3	18	6	no	9	8	Moderate	2364	1333	0
3000	1500	0	12+84	10	4	19	7	no	10	10	Moderate	2953	1464	0
3000	2000	0	12+86	11	4	22	12	no	10	10	Moderate	3493	1732	0.009
1000	2500	0	12+79	8	4	18	9	no	8	8	Moderate	1812	1635	0
3000	2500	0	12+83	12	5	17	10	no	9	10	Moderate	3369	2014	0.03
4000	2500	0	12+103	12	4	24	10	no	10	10	Moderate	4205	1999	0.04
3000	**3000**	**0**	**12+108**	**15**	**6**	**26**	**12**	**yes**	**12**	**12**	**Poor**	**3809**	**2338**	**0.04**
3000	5000	0	N/A	N/A	N/A	N/A	N/A	yes	N/A	N/A	**Poor**	5186	3618	0.05
0	0	1	12+46	2	0	12	2	no	2	2	Excellent	123	75	0.6
0	0	2	12+46	2	0	12	2	no	2	2	Excellent	125	74	1
0	0	3	12+46	2	0	12	2	no	2	2	Excellent	125	70	1.6
0	0	5	12+46	2	0	12	2	no	2	2	Excellent	125	72	2
0	0	10	12+54	2	0	12	2	no	2	2	Excellent	129	77	5
0	0	20	12+54	4	0	12	2	no	2	2	Excellent	117	71	10
0	**0**	**25**	**12+59**	**4**	**2**	**13**	**3**	**yes**	**3**	**3**	**Poor**	**128**	**75**	**14**
0	0	30	12+64	10	4	14	5	yes	7	7	**Poor**	124	74	16
0	0	40	12+72	12	14	N/A	N/A	yes	10	10	**Poor**	127	73	21
0	0	50	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**Poor**	N/A	N/A	N/A

Table 11 User measurements for Pocket Viewers

8.2.4 User measurements on Cellular Viewer

User Measurements on the Cellular Viewer are conducted in the same way as both the Pocket Viewer and the Web Viewer. Similar to the Pocket Viewer tests, eight user measurements are carried out with the Cellular Viewers and utilising the same *NetEm* rules for the configuration of delay, jitter and packet loss.

8.2.4.1 Quality assessment

Unlike the Pocket Viewers, the continuous *three waveforms* appearance test result is taken into account during quality assessment including modification to time intervals. This is because the Cellular Viewers response time is higher than the Pocket Viewers or the Web Viewers.

1. **Excellent**: 3 waveforms appearance time is between 8-12 seconds.
2. **Good:** 3 waveforms appearance time is between 13-18 seconds.
3. **Moderate:** 3 waveforms appearance time is between 19-25 seconds.
4. **Poor:** Server disconnections happen. Even a single disconnection from the Mobile Care Server is considered as a poor quality.

8.2.5 Results for Cellular Viewers

Table 12 tabulates the user measurement results for the Cellular Viewers. Once more, the table displays from left to right, the configured delay, jitter and loss parameters, eight user measurements, assessed delay and the resulted delay, jitter and loss values.

Still, it is visibly clear that the configured delay, jitter and packet loss values are not exactly alike with the resulted delay, jitter and packet loss. The actual delay is higher compared to the configured delay whereas the jitter is higher or less. Alternatively, the actual packet loss is comparable to the configured loss.

8.2.5.1 Limit delay, jitter and packet loss values for Cellular Viewers

The limit QoS parameter values which end-users can tolerate are as follows. Once more the red marked text at the rightmost columns where the quality becomes poor is taken into account.

Delay limit is observed as *5.2 seconds*. Higher this limit, server disconnections and packet loss start in addition to discontinuous waveforms.

Jitter limit is identified as around *1.7 seconds* together with a delay of 3.8 seconds and a packet loss of 10 percent.

Packet loss limit occurs to be *21 percent*. The configured loss of 25 percent results in a similar loss result, which is 21 percent.

8.2.5.2 Comparison with live measurement results

The average Cellular Viewer application delay, jitter and packet loss values in WLANs found in Section 7.1.3 are observed to be similar with the values in the row highlighted in green. The application displays an excellent performance with no packet loss.

Live tests on the 3G network resulted in around 407 ms delay with 102 ms jitter as in Section 7.1.3. The user measurement results in Table 12 could not be compared with the live test results because the jitter is in the region of 350 ms, which is about 3 times more than the live tests when delay is roughly 400 ms.

Lastly, the row highlighted in yellow represents the delay, jitter and packet loss values observed in GPRS networks by the Cellular Viewer application. The performance is considered to be good. Disconnections do not happen with the Cellular Viewers when the delay is in the region of 2 seconds and jitter is 1 second.

CONFIGURED VALUES(NetEm)			USER MEASUREMENTS on CELLULAR VIEWER (seconds)									ACTUAL VALUES (tcptrace)		
Delay ms	Jitter ms	PacketLoss%	Start up (s)	3wf app.	max pause	S/5 24h trend	Unity 24h trend	Server disc.	Unitywf app.	S/5wf app.	Quality	Delay ms	Jitter ms	PacketLoss%
0	0	0	4+8	8	0	13	5	no	8	8	Excellent	437	349	0
300	0	0	4+10	10	0	13	5	no	10	10	Excellent	616	312	0.1
500	0	0	4+11	11	0	14	6	no	11	11	Excellent	925	323	0.2
1000	0	0	4+12	12	0	15	7	no	12	12	Excellent	1477	401	0.6
1500	0	0	4+13	13	0	16	8	no	13	13	Good	1983	322	1.6
2000	0	0	4+16	14	0	19	9	no	14	14	Good	2346	317	0.7
3000	0	0	4+22	19	0	20	12	no	19	19	Moderate	3320	298	0.3
4000	0	0	4+28	25	2	22	15	no	25	25	Moderate	4327	287	0.3
5000	**0**	**0**	**4+32**	**27**	**4**	**75**	**21**	**yes**	**27**	**27**	**Poor**	**5260**	**295**	**0.7**
6000	0	0	4+38	31	6	N/A	28	yes	31	31	Poor	6412	500	1.7
7000	0	0	4+42	45	11	N/A	N/A	yes	45	45	Poor	8034	216	4.7
300	25	0	4+10	10	0	13	5	no	10	10	Excellent	776	358	0.4
500	25	0	4+11	11	0	14	6	no	11	11	Excellent	1051	378	0.3
500	500	0	4+12	12	2	15	7	no	12	12	Excellent	909	507	0.7
1000	750	0	4+13	13	2	16	8	no	13	13	Good	1416	772	2.3
1500	750	0	4+16	14	0	18	9	no	14	14	Good	2130	850	0.2
1500	1000	0	4+17	14	0	18	9	no	14	14	Good	2258	1177	0.2
2000	750	0	4+18	15	0	19	9	no	15	15	Good	2543	893	0.3
2000	1500	0	4+19	18	0	20	11	no	18	18	Good	2650	1441	4.8
2000	2000	0	4+20	19	3	20	12	no	19	19	Moderate	3267	1577	6
3000	**2000**	**0**	**4+22**	**26**	**4**	**22**	**15**	**yes**	**25**	**25**	**Poor**	**3878**	**1787**	**10**
3000	3000	0	4+38	27	6	28	19	yes	27	27	Poor	4679	2167	11
0	0	0.1	4+8	8	0	13	5	no	8	8	Excellent	146	204	0.3
0	0	0.3	4+8	8	0	13	5	no	8	8	Excellent	123	169	0.4
0	0	0.5	4+8	8	0	13	5	no	8	8	Excellent	117	98	0.8
0	0	1	4+8	8	0	13	5	no	8	8	Excellent	317	336	1.1
0	0	2	4+8	9	0	13	5	no	9	9	Excellent	172	228	3.5
0	0	3	4+8	9	0	13	5	no	9	9	Excellent	289	323	4.2
0	0	5	4+8	9	0	13	5	no	9	9	Excellent	338	332	5.6
0	0	10	4+9	9	0	13	5	no	9	9	Excellent	464	359	10
0	0	20	4+10	10	0	13	6	no	10	10	Excellent	306	337	18
0	**0**	**25**	**4+11**	**11**	**3**	**14**	**8**	**yes**	**11**	**11**	**Poor**	**237**	**283**	**21**
0	0	30	4+18	16	15	N/A	57	yes	16	16	Poor	193	132	29
0	0	40	4+20	22	17	N/A	N/A	yes	22	22	Poor	165	153	35
0	0	50	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Poor	155	138	48

Table 12 User measurements for Cellular Viewers

8.3 Summary and Result Evaluation

This chapter identifies the limit values for delay, jitter and packet loss parameters for the Mobile Viewer clients (Web Viewer, Pocket Viewer and Cellular Viewer). As a result, these values can be utilised in order to warn end-users in circumstances of possible dissatisfactions with certain network conditions.

As mentioned earlier, end-users have certain expectations, for instance, the client applications should response quickly to user actions. Also, the viewed real-time continuous waveforms should flow without any interrupts or pause. There should not occur any disconnections from the Mobile Care Server while viewing waveforms or downloading trend data.

When determining the requirements for the Mobile Viewer clients, these expectations are taken into account. Therefore, some network conditions are emulated with a network emulator by inserting additional delay, jitter and packet loss into the network. In addition, some user measurement tests are performed in each network condition to find out whether the applications responded within a certain amount of time, the interrupts during trend download or discontinuity in waveform drawings appeared (or not) and if there were any disconnections from the server.

Finally, the live network results found in Chapter 6 are compared to the user measurements to see if the emulated network environments actually present an equivalent application performance.

We can see from the results that the Web Viewers do not meet the user expectations when the delay between the Web Viewer and the MCS is more than 4 seconds, jitter is more than 450 milliseconds or the packet loss is higher than 20 percent. It was observed that, with these limit values the Web Viewers' *sixteen waveforms* view does not flow continuously. The waveforms stop drawing for about 6-9 seconds. In addition, disconnections from the server occur and the *24h trend download* becomes delayed by up to 10 seconds.

Also, when the Pocket Viewers were observed, we see the same waveform behaviour; the *four waveforms* stop drawing for about 6-7 seconds. In addition, the *24h trend download* can delay by up to 16 seconds and disconnections from the MCS occur when the delay between the Pocket Viewer and the MCS is about 7 seconds, jitter is 2.3 seconds or packet loss is 14 percent.

The Cellular Viewer observations show that when the delay between the Cellular Viewer and the Mobile Care Server is about 5.2 seconds, jitter is 1.7 seconds or packet loss is 21 percent, the end-user can be frustrated. The *three waveforms* appearance time can reach up to 27 seconds, waveforms can stop drawing for up to 4 seconds, and the *15h trend download* time can increase up to 75 seconds.

Additionally, the reliability of the emulated network is estimated. For instance, the Web Viewer application results in an average delay of 1.6 seconds and 511 ms jitter when viewing the *sixteen waveforms* in GPRS network. In this condition, server disconnections occur, waveforms stop drawing and the *24h trend download* becomes delayed. When observing Table 10, the row containing an actual delay of 1.5 seconds and the jitter of 498 seconds seem to be the closest values to the GPRS result. And if we look at the user measurement results in the row, there are server disconnections, interrupts in *waveform* drawing and increased response time during the *24h trend download*. Hence, we conclude that the emulated network displays similar application performances when the delay, jitter and packet loss is similar to the live results.

The requirements defined in this chapter appear to be reliable, however; the results are specific to the Mobile Viewers. Therefore, they are not transferable to other applications. In addition, the results cannot be evaluated in the light of the works cited in this thesis. This is because the author was unable to detect any previous work similar to this one, where the requirements for the Mobile Viewer clients are specified by utilising a network emulator, user measurement tests and verified by live passive measurements.

Lastly, the defined requirements can be validated in future by carrying out the same measurement techniques with the same emulator a few more times and also by utilising different emulators and different client devices.

9 Conclusions and Remarks

Quality of Service is becoming increasingly important with the development of new access technologies and sophisticated equipment. Public networks have become such a complex infrastructure that it is impossible to promise any guaranteed quality for applications and services. Therefore, an endless intensive research effort has been dedicated to this field. This thesis work attempted to give insight into what “Quality of Service” stands for and why it is such an important notion, in particular, for data transmission in public networks.

The main goal of this thesis was to specify and verify the requirements for Mobile Viewer client applications developed by General Electric Healthcare, to identify the key Quality of Service parameters during data transmission in public networks with the intention of warning the end-users of certain dissatisfactions.

In order to reach this goal, first, an intensive literature study on network performance measurements, metrics, methods and tools was performed. Based on the background study results, the best-suited performance measurement method, QoS metrics, measurement environment, measurement tools and network emulator were selected.

First of all, live measurement analysis of different networks such as LAN, WLAN, GPRS and UMTS were conducted for each Mobile Viewer client application (Web Viewer, Pocket Viewer and Cellular Viewer) during the chosen tests. The results were compared according to the network technology for each client application. The results showed that, in general, GPRS network measurements cover the *highest* delay, jitter and packet loss compared to UMTS, and WLAN network measurements. In addition, server disconnections were observed in the GPRS networks with Web Viewer clients. As a result, delay in Local Area Networks is observed to be in an order of magnitude of microseconds whereas UMTS and Wireless Local Area Networks of milliseconds and GPRS of seconds.

At last, with the help of a network emulator, different network conditions were emulated, consisting of the necessary delay, jitter and packet loss in order to detect the critical data points where each client application starts to bring dissatisfaction to customers. In order to find out the data points, a number of user measurement tests were conducted, performance quality for each network environment is determined and the criteria for dissatisfaction are defined. These points were identified and verified simultaneously with passive measurements. Furthermore, the live network analysis results are compared to the user measurements to see if the emulated network environments actually confer to similar performance results when there are similar delay, jitter and packet loss in the network.

The results showed that the Web Viewers do not meet the user expectations when *delay* between the Web Viewer and the MCS is more than 4 seconds, *jitter* is more than 450 milliseconds or the *packet loss* is higher than 20 percent. Also, the Pocket Viewers show poor performance when the *delay* between the Pocket Viewer and the MCS is about 7 seconds, *jitter* 2.3 seconds or *packet loss* 14 percent. Finally, the Cellular Viewer observations show that when the *delay* between the Cellular Viewer and the Mobile Care Server is about 5.2 seconds, *jitter* is 1.7 seconds or *packet loss* is 21 percent the end-user can be dissatisfied. The resulting poor performance was defined as the irritating delay in waveforms appearance time or trend download, disconnecting from server and the interruption in the continuous waveform flowing.

Even though the goal and objectives of this thesis were met, further measurements regarding some other types of access networks such as EDGE, HSCSD, HSDPA, needs to be conducted in order to carry out a comprehensive analysis. Furthermore, the measurement periods, type of client devices, and number of tests needs to be increased to provide more precise results. In addition to passive live network measurements and emulated environment measurements, it would be interesting to conduct active measurements tests via an active measurement tool to find out one-way delay and one-way delay variations. Finally, it is believed that this presented study will provide a solid foundation for future work.

References

[1] J.Prokkola, M. Jurvansuu, M. Immonen, "*Measuring WCDMA and HSDPA Delay Characteristics with QoSMET*", in Proceedings of IEEE International Conference on Communications (ICC 2007), June 2007, pp 492-498.

[2] M. Jurvansuu, J. Prokkola, M. Hanski, P. Perälä, "*HSDPA Performance in Live Networks*", in Proceedings of IEEE International Conference on Communications (ICC 2007), June 2007, pp 467-471.

[3] H. Holma and J. Reunanen, "*3GPP Release 5 HSDPA Measurements*", in proc. PIMRC'06, Helsinki, Finland, September 11-14, 2006.

[4] M. Kohlwes, J. Riihijarvi, P. Mahonen, "*Measurements of TCP performance over UMTS networks in near-ideal conditions*", Vehicular Technology Conference, June 2005, Vol.4, pp 2235-2239.

[5] A. Diaz, P. Merino, J. Rivas, "*Performance monitoring and analysis of wireless communication protocols for mobile devices*", International Conference on Ubiquitous Computing, 2006.

[6] P. Reichl, M. Umlauft, "*Project WISQY: A measurement based End-to-End Application-Level performance comparison of 2.5G and 3G Networks*", Wireless Telecommunications Symposium, April 2005, pp 9-14.

[7] W. Tan, F. Lam, W. Lau, "*En Empirical study on 3G Network Capacity and performance*", in Proceedings of IEEE International Conference on Computer Communications, May 2007, pp. 1514-1522.

[8] A. Hämäläinen, M. Mäkinen, R. Kantola, "*Measuring online game application in GPRS and UMTS*", master's thesis, URL= *www.netlab.hut.fi/opetus/s38310/04-05/Kalvot_04-05/H%E4m%E4l%E4inen_070605 .ppt*

[9] J.Cano-Garcia, E. Gonzalez-Parada, E. Casilari, "*Experimental Analysis and Characterization of Packet Delay in UMTS Networks*", Lecture notes in Computer Science, Sept 2006. Vol. 4003, pp.397- 406.

[10] Y. Lee, "*Measured TCP Performance in CDMA 1xEV-DO Network*", ACM International Conference Proceeding Series, 2006, Vol. 220, No. 2.

[11] A. Arjona, C. Westphal, A. Yla-Jaaski, M. Kristensson, "*Towards High Quality VoIP in 3G Networks An Empirical Study*", June 2008, AICT '08. Fourth Advanced International Conference on Telecommunications, pp. 143-150.

[12] A. Wennstrom, A. Brunstrom, J. Rendon, J. H. Gustafsson, "*A GPRS Testbed for TCP Measurements*", in Proceedings of IEEE, International Conference on Mobile and Wireless Communicaitons Network, Dec. 2002, pp.320-324.

[13] J. Korhonen, 0.Aalto, A.Gurtov, H. Laamanen, "*Measured Performance of GSM HSCSD and GPRS*", in Proceedings of IEEE International Conference on Communications (ICC'OI), June 2001, pp. 1330-1334.

[14] M. Meyer, "*TCP Performance over GPRS*", In proceedings of IEEE Wireless Communications and Networking Conference, Sep. 1999.

[15] P. Benko, G. Malicsko, A. Veres, "*A Large-Scale, Passive Analysis of End-to-End TCP Performance over GPRS*", in Proceedings of IEEE, Conference on Computers and Communications Societies, Vol. 3, March 2004, pp. 1882-1892.

[16] R. Chakravorty, I. Pratt, "*Practical Experience with TCP over GPRS*", in Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM 2002), Nov. 2002.

[17] P. Stuckmann, N. Ehlers, B. Wouters, "*GPRS Traffic Performance Measurements*" in Proceedings of the IEEE Vehicular Technology Conference (VTC 2002 fall), Sep.2002, pp. 1289-1293.

[18] A. Gurtov, M. Passoja, O. Aalto, M. Raitola, "*Multi-Layer Protocol Tracing in a GPRS Network*", in Proceedings of the IEEE Vehicular Technology Conference (Fall VTC2002), Sep. 2002.

[19] P. Lorenz and P. Dini, "*Measuring Quality of Service Parameters over Heterogeneous IP Networks*", ICN 2005, LNCS 3421, 2005, pp. 718-727.

[20] G. Xylomenos, G. C. Polyzos, "*TCP and UDP Performance Over a Wireless LAN*", INFOCOM '99. Vol. 2, March 1999 pp. 439 –446.

[21] B. Bing, "*Measured Performance of the IEEE 802.11 Wireless LAN", in Proceedings of IEEE*, Conference on Local Computer Networks (LCL99), Oct. 1999, pp. 34-42.

[22] M. Kappes, "*An Experimental Performance Analysis of MAC Multicast in 802.11b Networks for VoIP Traffic*", in Proceedings of ScienceDirect, Computer Communications Vol. 29, No. 8, May 2006, pp. 938-948.

[23] J.P. Singh, N. Bambos, B. Srinivasan, D. Clawin, "*Wireless LAN Performance Under Varied Stress Conditions in Vehicular Traffic Scenarios*", in Proceedings of IEEE VTC2002, Vol. 2, Sept. 2002, pp. 737-747.

[24] A. Messier, J. Robinson, K. Pahlavan, "*Performance Monitoring of a Wireless Campus Area Network*", in Proceedings of IEEE, Local Computer Networks (LCN97), Nov. 1997, pp. 232-238.

[25] R. Chakravorty, I. Pratt, "*Performance Issues with General Packet Radio Service*", Journal of Communications and Networks (JCN), Vol. 4, 2002, pp. 266-281.

[26] T. Alanko, M. Kojo, H. Laamanen, M. Liljeberg, M. Moilanen, K. Raatikainen, "*Measured Performance of Data Transmission Over Cellular Telephone Networks*", in Proceedings of ACM SIGCOMM99 Computer Communication Review, Vol. 24, No. 5, Oct. 1994, pp. 24-44.

[27] R. Ludwig, B. Rathonyi, A. Konrad, K. Oden, A. Joseph, "*Multi-Layer Tracing over a Reliable Wireless Link*", in Proceedings of ACM SIGMETRICS99 International Conference on Measurement and Modelling of Computer Systems, June 1999, pp. 144-154.

[28] G. Xylomenos, G. Polyzos, "*TCP Performance Issues over Wireless Links*", in Proceedings of IEEE Communications Magazine, Vol. 39, No. 4, Apr. 2001, pp. 52-58.

[29] W. Wei, B. Wang, C. Zhang, J. Kurose, D. Towsley, "*Classification of Access Network Types: Ethernet, Wireless LAN, ADSL, Cable Modem or Dialup?*" in Proceedings of IEEE Conference on Computer and Communications Societies, Vol. 2, No. 13, March 2005, pp. 1060-1071.

[30] R. Feng, J. Song, "*Some QoS Issues in 3G Wireless Networks*", in Proceedings of IEEE Conference on Computers, Communications, Control and Power Engineering (TENCON '02), Oct 2002, Vol. 2.pp 724-727.

[31] Easy Wireless Project, *International Technology Education Association* (ITEA). URL: http://ew.thales.no/index.html

[32] A3-WISQY: Wireless InterSystem QoS.
URL: http://www.ftw.at/ftw/research/projects/ProjekteFolder/A3

[33] W. Lewandowski, J. Azoubib, W. J. Klepczynski, "*GPS: Primary Tool for Time Transfer*", In Proceedings of IEEE, VOL. 87, NO. 1, January 1999.pp 163-172.

[34] F. Roosbeek, P. Defraigne, A. Somerhausen, "*Setting up an NTP Server at the Royal Observatory of Belgium*", 36th Annual Precise Time and Time Interval (PTTI) Meeting, 2004, pp 237-246.

[35] Vern Paxson et al, "*Framework for IP Performance Metrics*", RFC2330, May 1998.

[36] Mail Archieve of Wireshark Users, URL:
http://www.mailarchive.com/wiresharkusers@wireshark.org/msg01653.html.

[37] J. But, U. Keller, D. Kennedy, G. Armitage, "*Passive TCP Stream of RTT and Jitter Parameters*" In Proceedings of IEEE, Local Computer Networks, Vol. 17, No. 17, pp. 433-441, Nov 2005.

[38] R. Prasad, C. Dovrolis, M. Murray, K. Claffy, "*Bandwidth estimation: metrics, measurement techniques, and tools*" Network, IEEE, Vol.17, No. 6, pp. 27-35, 2000.

[39] Wolfram Math World- Normal Distribution,
URL: http://mathworld.wolfram.com/NormalDistribution.html.

[40] Stanley Seat, "*An Eye-Opening Look at Jitter*", EDN, Nov 2002, URL: http://www.edn.com/article/CA250810.html.

[41] D. J. Paul, "*Characterizing Digital Audio Transformers with Induced Jitter Histograms*", AES E-Library, No. 5448, Nov. 2001.

[42] A. Botta, A. Pescape, G. Ventre, "*Quality of Service Statistics over Heterogeneous Networks: Analysis and Applications*", In proceedings of Elseiver, European Journal of Operational Research, Vol.191, No. 3, Dec. 2008, pp. 1075-1088.

[43] C. Bettstetter, H. J. Vogel, and J. Eberspecher, "*GSM Phase 2i; General Packet Radio Service GPRS: Architecture. Protocols and Air Interface*", IEEE Communications Surveys, Vol. 2, No. 3, Third Quarter 1999.

[44] S. Hoff, M. Meyer, J. Sachs, "*Analysis of the General Packet Radio Service (GPRS) of GSM as Access to the Internet*", ICUPC'98, and Oct. 1998.

[45] P. Benko. A Veres. "*A Passive Method for Estimating End-to-End TCP Packet Loss*", in Proceedings of IEEE Global Conference 2002.

[46] T. J. Kostas, M. S. Borella, I. Sidhu, G. M. Schuster, J. Grabiec, J. Mahler, "*Real-Time Voice Over Packet-Switched Networks*", in Proceedings of IEEE Network, Vol. 12, No. 1, Feb 1998, pp. 18-27.

[47] M. S. Borella, D. Swider, S. Uludag, G. B. Brewster, "*Internet Packet Loss: Measurement and Implications for End-to-End QoS*", in Proceedings of IEEE ICCP98 Workshop, Aug. 1998, pp. 3-12.

[48] M. Crovella, B. Krishnamurthy, "*Internet Measurement: Infrastructure, Traffic and Applications*", Published by Wiley, May 2006, pp. Sections 3.5 and 6.3.

[49] P. Calyam, D. Krymskiy, M. Sridharan, P. Schopis, "*Active and Passive Measurements on Campus, Regional and National Network Backbone Paths*", in Proceedings of IEEE, International Conference on Computer Communications and Networks, Oct. 2005, pp. 537-542.

[50] IEEE, Institute *of Electrical and Electronics Engineers*, URL: http://ieee.org/portal/site

[51] ACM, *Association for Computing Machinery*, URL: http://www.acm.org/

[52] SpringerLink, URL: http://www.springerlink.com/home/main.mpx

[53] ScienceDirect, URL: http://www.sciencedirect.com/

[54] Lecture notes on *Internet Traffic Measurements and Measurement Analysis*, S-38.3183, TKK, and URL: http://www.netlab.tkk.fi/opetus/s383183/k06/schedule.shtml

[55] VTT, *Technical Research Centre of Finland*, URL: http://www.vtt.fi/

[56] QoSMET, *A Passive Tool for Measuring Quality of Service of Networking Applications,* URL: http://www.vtt.fi/liitetiedostot/innovaatioita/**qosmet**.pdf

[57] NetHawk, *Testing Tools for Converging Networks*, URL: https://www.nethawk.fi/products/nethawk_analyser/nethawk_m5/

[58] Tcptrace, *A Passive Analysis Open Source Application*, URL: http://www.tcptrace.org/

[59] Xplot, *Tool for Monitoring Tcptrace Output Graphs in Linux OS*, URL: http://www.xplot.org/

[60] Jplot, *A Java Version of Xplot*, URL: http://masaka.cs.ohiou.edu/software/jPlot/

[61] Tcpdump, *A Packet-Capturing Tool*, URL: http://www.tcpdump.org/

[62] Moset, *Mobile Service Performance Tools*, URL: http://www.vtt.fi/liitetiedostot/innovaatioita/MOSET_technolog.pdf

[63] Netperf, *Active Measurement Tool*, URL: http://www.netperf.org/netperf/

[64] NetProbe, *Active Measurement Tool*, URL: http://www.netprobe.net.

[65] RUDE&CRUDE, *Real-time UDP Data Emitter and Collector for RUDE*, http://rude.sourceforge.net/

[66] D-ITG, *Distributed Internet Traffic Generator Presentation*, and URL: http://www.grid.unina.it/software/ITG/codice/ditg-pres.pdf

[67] S. Avallone, A. Botta, A. Dainotti, W. Donato, A. Pescape, " *Distributed Internet Traffic Generator V.2.7.0-Beta Manual*", Dec. 2008, retrieved from URL: http://www.grid.unina.it/software/ITG/documentation.php

[68] NS-2, *Network Simulator*, URL: http://www.isi.edu/nsnam/ns/

[69] REAL, *Network Simulator*, URL: http://minnie.tuhs.org/REAL/

[70] Netsim, *Network Simulator*, URL: http://ontwerpen1.khlim.be/projects/netsim/index.html

[71] Dummynet, *Network Emulator*, URL: http://info.iet.unipi.it/~luigi/ip_dummynet/

[72] L. Rizzo, "*Dummynet: A Simple Approach to the Evaluation of Network Protocols*", in Proceedings of ACM SIGCOMM97, Computer Communication Review, Vol. 27, No. 1, Jan. 1997, pp. 31-41.

[73] G. Armitage, L. Stewart, "*Some Thoughts on Emulating Jitter for User Experience Trials*", in Proceedings of ACM SIGCOMM04 Workshop on Network and System Support for Games, Sep. 2004, pp. 157-160.

[74] P. Huang, Lecture Notes on Network Simulation and Testing, "*Testing on Real Systems- Dummynet*", retrieved from URL: http://cc.ee.ntu.edu.tw/~phuang

[75] A Basic Dummynet Tutorial, URL: http://cs.baylor.edu/~donahoo/tools/dummy/tutorial.htm

[76] M. Carson, D. Santay, "*NIST Net- A Linux-based Network Emulation Tool*", in Proceedings of ACM SIGCOMM Computer Communications Review, Vol. 33, No. 3, July 2003, pp. 111-126.

[77] NIST Net Home Page, URL: http://snad.ncsl.nist.gov/nistnet/

[78] K. Mahdee, H. Tarek, "*LAN Simulator*", Term project retrieved from URL: http://comnet.technion.ac.il/projects/spring03/cn02s03/index_files/current_final.doc

[79] NIST Net Installation Instructions, URL: http://w3.antd.nist.gov/tools/nistnet/install.html

[80] NIST Net Installation and Configuration, URL: http://www.cisco.com/en/US/products/ps5981/products_release_and_installation_notes09186a00803abf68.html#wp23144

[81] A. Keller, " *Network Emulator, Manual tc Packet Filtering and NetEm*", July 2006, ETH Zurich, retrieved from URL: http:// tcn.hypert.net/tcmanual.pdf

[82] S. Hemminger, "*Network Emulation with NetEm*", April 2005, OSDL, retrieved from URL: http://devresources.linux-foundation.org/shemminger/**netem**/LCA2005_paper.pdf

[83] NetEm Network Emulator homepage, URL: http://www.linuxfoundation.org/en/Net:Netem.

[84] NetEm GUI, software retrieved from URL: http://www.mail-archive.com/nistnet@antd.nist.gov/msg00346.html

[85] Lecture Notes on Internet Programming & Protocols, retrieved from URL: http://www.cs.utk.edu/~dunigan/ipp05/class15.pdf.gz

[86] NetEm, "*How to Use NetEm*" , retrieved from URL: http://nfsv4.bullopensource.org/doc/HowToUseNetEm.pdf

[87] K. Cho, "*Fitting Theory into Reality in the ALTQ Case*", Sony Computer Science Laboratories, Inc., retrieved from URL: http://www.sonycsl.co.jp/~kjc/papers/fittingtheory.pdf

[88] K. Cho, "*ALTQ Tips*", Sony Computer Science Laboratories, Inc., retrieved from URL: http://www.sonycsl.co.jp/~kjc/software/Tips.txt

[89] ALTQ, HomePage, URL. http://pf4freebsd.love2party.net/altq.html

[90] T. Skeie, S. Johannessesn, Q. Holmedie, "*The Road to an End-to-End Deterministic Ethernet*" in Proceedings of IEEE International Workshop on Factory Communication Systems, Aug. 2002, pp. 3-9.

[91] B. L. Barnett, M. K. Molloy, "*A Measurement Study of Two Local Area Network Data Link Level Protocols*", in Proceedings of IEEE Conference on Local Computer Networks, Oct. 1988, pp. 364- 374.

[92] J. Paananen, *GE Mobile Viewers System Architecture*, June 2005.

[93] J. Paananen, *GE Mobile Cellular Viewer Design Description*, June 2005.

[94] GE Healthcare, Mobile Viewers, Technical Reference Manual, 1st ed., 2006.

[95] A. A. Al-Ghanem, A.A. Sedki, F. M. Matboli,M.S.Al-Ghamdi, "Patient Monitors", Technical Report retrieved from URL: http://www.ksu.edu.sa/sites/Colleges/Arabic%20Colleges/AppliedMedicalSciences/.../Documents/**Patient**%20Monitoring%20Report.pdf

[96] R. M. Gardner, M. M Shabot, "Patient Monitoring Systems", book chapter published by SpringerLink, Dec. 2006, pp. 585-625.

[97] L. Cottrell, "Comparison of some Internet Active End-to-End Performance Measurement Projects", retrieved from URL: http://www.slac.stanford.edu/comp/net/wan-mon/iepm-cf.html, July 1999.

[98] N. Unuth, "Mean Opinion Score (MOS) - A Measure Of Voice Quality", 2009, retrieved from URL: http://voip.about.com/od/voipbasics/a/MOS.htm

[99] A. Botta, D. Emma, A. Pescape, G. Ventre, "*Systematic Performance Modelling and Characterization of Heterogeneous IP Networks*", in Proceedings of IEEE ICPADS'05, Vol. 2, No. 22, July 2005, pp. 120-124.

[100] S.I. Maniatis, E.G. Nikolouzou, I.S. Venieris, "*QoS Issues in the Converged 3G Wireless and Wired Networks*", in Proceedings of IEEE Communications Magazine, Vol. 40, No. 8, Aug. 2002, pp. 44-53.

[101] M. Malkowski, D. ClaSen, "*Performance of Video Telephony Services in UMTS using Live Measurements and Network Emulation*", in Proceedings of SpringerLink in Journal of Wireless Personal Communications, Vol. 46, No. 1, July 2008, pp. 19-32.

[102] MGEN, Multi-Generator-active measurement tool, URL:http://cs.itd.nrl.navy.mil/work/mgen/index.php

[103] Iproute2, collection of utilities for traffic control in Linux, URL: http://www.linuxfoundation.org/en/Net:Iproute2

[104] Setting up a Linux Router, URL: http://www.comptechdoc.org/os/linux/usersguide/linux_ugrouting.html

[105] Adding a static route, URL: http://ubuntuforums.org/showthread.php?t=279371

[106] Traceroute Manual, URL: http://www.exit109.com/~jeremy/news/providers/traceroute.html#reverse

Appendix A

A.1 Wired Emulated Network Configurations

Mobile Care Server 1 Configurations:

Ifcfg-eth1
IP ADDRESS 10.0.1.2
GATEWAY 255.255.255.0
DEFAULT 10.0.1.1

static routes
route add –net 10.0.0.0 netmask 255.0.0.0 gw 10.0.1.1 dev eth1
echo 1 > /proc/sys/net/ipv4/ip-forward

Fedora Server (Router) Configurations:

Ifcfg-eth0
IP ADDRESS 10.0.1.1
GATEWAY 255.255.255.0

Ifcfg-eth1
IP ADDRESS 10.0.0.1
GATEWAY 255.0.0.0

static routes
route add –net 10.0.0.0 netmask 255.0.0.0 gw 10.0.0.1 dev eth1
route add –net 10.0.1.0 netmask 255.255.255.0 gw 10.0.1.1 dev eth0
echo 1 > /proc/sys/net/ipv4/ip-forward

iptables
- A FORWARD –i eth0 –o eth1 –j ACCEPT
- A FORWARD –i eth1 –o eth0 –j ACCEPT

Web Viewer 1 (Ubuntu Laptop)Configurations:

Ifcfg-eth0
IP ADDRESS 10.0.0.2
GATEWAY 255.0.0.0
DEFAULT 10.0.0.1

static routes
route add –net 10.0.1.0 netmask 255.255.255.0 gw 10.0.0.1 dev eth0
echo 1 > /proc/sys/net/ipv4/ip-forward

A.2 Wireless Emulated Network Configurations

Mobile Care Server 1 Configurations:

Ifcfg-eth1
IP ADDRESS 10.0.1.2
GATEWAY 255.255.255.0
DEFAULT 10.0.1.1

static routes
route add –net 10.1.1.0 netmask 255.255.255.0 gw 10.0.1.1 dev eth1
echo 1 > /proc/sys/net/ipv4/ip-forward

Fedora Server (Router) Configurations:

Ifcfg-eth0
IP ADDRESS 10.0.1.1
GATEWAY 255.255.255.0

Ifcfg-eth1
IP ADDRESS 10.1.1.10
GATEWAY 255.255.255.0

static routes
route add –net 10.1.1.0 netmask 255.255.255.0 gw 10.1.1.10 dev eth1
route add –net 10.0.1.0 netmask 255.255.255.0 gw 10.0.1.1 dev eth0
echo 1 > /proc/sys/net/ipv4/ip-forward

iptables
- A FORWARD –i eth0 –o eth1 –j ACCEPT
- A FORWARD –i eth1 –o eth0 –j ACCEPT

Ubuntu Laptop (tcptrace) Configurations:

Ifcfg-eth0
IP ADDRESS 10.1.1.11
GATEWAY 255.255.255.0
DEFAULT 10.1.1.10

static routes
route add –net 10.0.1.0 netmask 255.255.255.0 gw 10.1.1.10 dev eth0
echo 1 > /proc/sys/net/ipv4/ip-forward

Access Point Configurations

LAN1: This interface is a DHCP server.

Address Assignment Range: 10.1.1.52 to 10.1.1.56

IP Address: 10.1.1.1
Network Mask: 255.255.255.0
Default Gateway: 10.1.1.10
Primary DNS: 10.1.1.1
Secondary DNS Server: 0.0.0.0
WINS Server: 0.0.0.0

User Defined Routes
Destination: 10.0.1.0
Subnet Mask: 255.255.255.0
Gateway: 10.1.1.10
Interface: LAN1
Metric: 1

Appendix B

B.1 Measurement Devices and Their Properties

The tools and device properties utilised in measurement environments are listed below.

Web Viewer Client 1	Laptop Dell D610, Windows XP Professional 2002, Service Pack 2, Intel(R) CPU
Web Viewer Client 2	Same device as Web Viewer Client 1
Web Viewer Client 3	Laptop Dell D620, Windows XP 2002, Service Pack 2, Intel(R) CPU T2300 @ 1.66GHz, 0.99 GB RAM
Measurements Collected Device	Laptop, Acer TravelMate 5310, Intel Celeron ,Ubuntu Release 7.10 Operating System Kernel Linux 2.6.22-14-generic GNOME 2.10.1 Memory: 1002.1MB Processor: Intel(R) Celeron(R) M CPU: 520 160 GHz
Cellular Viewer Client 1	Sony Ericsson P910i, 2G Network GSM 900 / 1800 / 1900 - P910i, Data: Class 8 (4+1 slots), 32 - 40 kbps, Operating System: Symbian OS v7.0, UIQ v2.1 UI, 32-bit Philips Nexperia PNX4000 156 MHz processor
Cellular Viewer Client 2	Nokia N81, 3G Network: UMTS 2100, Operating System: Symbian OS 9.2, Series 60 v3.1 UI, ARM 11 369 MHz processor, WLAN: Wi-Fi 802.11b/g with UPnP
Mobile Care Server 1	Intel HP xw4300 Workstation Red Hat Linux Operating System mcserver 4.2, Mobilecare Version 5.1
Mobile Care Server 2	ISP URL: mobilcare.em.health.ge.com, Red Hat Linux Operating System Release 4S-4.1.i386, Kernel: 2.6.9-34.EL.i686, Mobilecare Version 5.1 Intel HP xw___ Workstation
Emulator Server	Intel HP xw4200 Workstation Fedora 10 Operating System, Kernel Linux 2.6.27.12-170.2.5.fc10.i686 Memory: 3.5GB, Processor: Intel(R) Pentium(R) D CPU 3.20 GHz
Access Point	Motorola AP-5131, Network standards: Wi-Fi: 802.11a/b/g, WPA2, WMM, Data rates supported: 1, 2, 5.5, 6,9, 11, 12, 18, 24, 36, 48, 54 Mbps, Transmitter power: 22 dBm Maximum, Wireless medium: DSSS, OFDM
Ethernet Switch	Allied Telesyn AT-FS708, 10 Base-T/100 Base-TX 8 Port Fast Ethernet Switch
Central Simulator	Central Simulator 1.ORC1, built 2006-08-03 GE
Stop Watch	ORIGO 31-3769, Professional Stopwatch
Pocket Viewer	HP ipaQ 114, Microsoft® Windows Mobile® 6 Classic Operating System, Marvell PXA310 processor 624 MHz, Memory:64 MB SDRAM, 802.11b/g with WPA2 security; Bluetooth 2.0 with EDR

Appendix C

C.1 3G Data Spectrum

C.1.1 UMTS

Laptop --> Server VOIP/UDP 33-65 B Stable [1] [2]								
Avg Delay		Max Delay		Min Delay		Jitter	Packet loss	
UL	DL	UL	DL	UL	DL	Max	Min	Max
57ms	100ms	330ms	430ms	65ms	80ms	30ms	0.06%	0.12%
Laptop --> Laptop Sending UDP and Fake TCP packets up to 40B IAT:20ms Stable [9]								
DL-Delay		UL- Delay		Delay Standard Deviation DL/UL				Packetloss
TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	DL
101.3ms	108.3ms	105.9	101.1ms	81.4ms	83.1ms	79.9ms	78.2ms	0.5%
Mobile Phone--> Mobile Phone FTP/TCP 200KB Mobile [5]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Max	Min	Min	Max	Min	Max	Min	Max
768ms	2421ms	468ms	N/A	N/A	N/A	N/A	N/A	7000ms
Laptop--> Server ICMP/UDP Stable [6]								
Avg Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
32B	100B	1450B	Min	Max	Min	Max	Min	Max
150ms	160ms	537ms	N/A	N/A	N/A	N/A	N/A	N/A
Mobile Phone--> Server Iperf/TCP Stable [7]								
Heavily Loaded Cell			Lightly Loaded		Packet loss		Handover/Conn.break	
Round Trip Time Bands			Avg RTT		Min	Max	Min	Max
1000ms	3000ms	6000ms	50ms	350ms	0.1%	6%	N/A	N/A
Laptop--> Server FPS Game 80B UL, 100B-160B DL Stable [8]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Min	Max	Min	Max	UL	DL	Min	Max
150ms	N/A	N/A	N/A	N/A	0.02%	0.09%	N/A	N/A
Laptop --> Server VOIP/UDP 33-65 B Stable [10]								
Avg RTT		Max RTT		Min RTT		Jitter	Avg Packet loss	
UL	DL	UL	DL	UL	DL	Avg	UL	DL
730.9ms	414.4ms	2097.1ms	1278.7ms	170.3ms	109.2ms	N/A	4.7%	0.2%

C.1.2 HSDPA

Laptop --> Server VOIP/UDP 33-65 B Stable [1]						
One Way Delay DL			Jitter Bands		Packet loss	
Avg	Max	Min	Min	Max	Min	Max
50ms	60ms	45ms	0ms	20ms	N/A	N/A
Laptop --> Server HTTP GET/ TCP 32B- 2MB Stable [2]						
Round Trip Time			Jitter		Packet loss	
Avg	Max	Min	Avg		Min	Max
80ms	90ms	65ms	13ms		N/A	N/A
Mobile Phone--> Server VOIP/ UDP 32B Mobile [3]						
Round Trip Time			Jitter Avg		Packet loss	
Avg	Max	Min	Min	Max	Min	Max
331ms	N/A	N/A	19ms	22ms	0.4%	1.9%

C.2 2.5G Data Spectrum

C.2.1 GPRS

Mobile Phone--> Mobile Phone FTP/TCP 200KB Mobile [5]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Max	Min	Min	Max	Min	Max	Min	Max
2146ms	3343ms	1281ms	N/A	N/A	N/A	N/A	N/A	7000ms
Laptop--> Server ICMP/UDP Stable [6]								
Avg Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
32B	100B	1450B	Min	Max	Min	Max	Min	Max
500ms	735ms	2142ms	N/A	N/A	N/A	N/A	N/A	N/A
Laptop--> Server HTTP/TCP Request/Reply 280B-5070B Stable and Mobile [13]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Max	Min	Min	Max	Min	Max	Long pause during data transfer	
5.5s	254s	1.5s	N/A	N/A	N/A	N/A		
Laptop--> Server TCP Bulk Transfer, Stable [12]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Max	Min	Min	Max	Min	Max	Min	Max
N/A	47.5s	980ms	N/A	N/A	0.11%	7.58%	N/A	N/A
Laptop--> Server FPS Game 80B UL, 100B-160B DL Stable [8]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Min	Max	Min	Max	UL	DL	Min	Max
896,6ms	N/A	N/A	N/A	N/A	0.82%	0.04%	N/A	N/A
Laptop--> Server WWW/HTTP Retrieval of a popular web page Stable [17]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Min	Max	Min	Max	UL	DL	Min	Max
100s	10s	180s	N/A	N/A	N/A	N/A	N/A	N/A
Laptop--> Server FTP download 2.7MB Stable [17]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Min	Max	Min	Max	UL	DL	Min	Max
26mins	13min	31min	N/A	N/A	N/A	N/A	N/A	N/A
Laptop--> Server 1064B UDP Datagram Transfer, Stable [16]								
Uplink delay			Downlink Delay			Jitter	Handover/Conn.break	
Avg	Min	Max	Avg	Min	Max	Loss	Min	Max
0.7s	0.3s	1.8s	1s	0.2s	3.4s	N/A	N/A	N/A

C.2.2 HSCSD

Laptop--> Server HTTP/TCP Request/Reply 280B-5070B Stable and Mobile [13]								
Avg Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Max	Min	Min	Max	Min	Max	Every 11th-12th minute disconnection	
2.7s	26.8s	0.8s	N/A	N/A	N/A	N/A		

C.2.3 EDGE

Laptop--> Server FPS Game 80B UL, 100B-160B DL Stable [8]								
Round Trip Time			Jitter Avg		Packet loss		Handover/Conn.break	
Avg	Min	Max	Min	Max	UL	DL	Min	Max
696,7ms	N/A	N/A	N/A	N/A	0.18%	0.12%	N/A	N/A

C.3 WLAN Data Spectrum

C.3.1 Infrastructure mode

Laptop--> Server UDP Probe packet 128B, Medium Load, Stable [19]					
Round Trip Time		Jitter		Packet loss	
Min	Max	Min	Max	Min	Max
302ms	741ms	0.08ms	3.74ms	0.20%	66.60%
Laptop--> Server UDP Probe packet 256B, Medium Load, Stable [19]					
Round Trip Time		Jitter		Packet loss	
Min	Max	Min	Max	Min	Max
313ms	908ms	0.14ms	3.64ms	0.20%	72.40%
Laptop--> Server UDP Probe packet 512B, Medium Load, Stable [19]					
Round Trip Time		Jitter		Packet loss	
Min	Max	Min	Max	Min	Max
380ms	1248ms	0.33ms	7.52ms	0.00%	74.50%

C.3.2 Ad-hoc mode

Laptop--> Laptop UDP Voice pkt 42B, Light load,Ad-hoc,Stable [22]					
Round Trip Time		Jitter		Packet loss	
Avg	Stdev	Avg	Stdev	Avg	Stdev
2.96ms	1.55ms	0.18ms	0.06ms	0.01%	0.00%