A Network Traffic Analysis Tool for the Prediction of Perceived VoIP Call Quality

by

Gert Stephanus Herman Maritz

Thesis presented in partial fulfilment of the requirements for the degree of
Master of Science in Engineering
at the Faculty of Engineering, Stellenbosch University

Supervisor: Dr R. Wolhuter
Department of Electrical and Electronic Engineering

December 2011

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2011

Abstract

A Network Traffic Analysis Tool for the Prediction of Perceived VoIP Call Quality

G.S.H. Maritz

Department of Electrical & Electronic Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Thesis: MScEng (E&E)

December 2011

The perceived quality of Voice over Internet Protocol (IP) (VoIP) communication relies on the network which is used to transport voice packets between the end points. Variable network characteristics such as bandwidth, delay and loss are critical for real-time voice traffic and are not always guaranteed by networks. It is important for network service providers to determine the Quality of Service (QoS) it provides to its customers. The solution proposed here is to predict the perceived quality of a VoIP call, in real-time by using network statistics.

The main objective of this thesis is to develop a network analysis tool, which gathers meaningful statistics from network traffic. These statistics will then be used for predicting the perceived quality of a VoIP call. This study includes the investigation and deployment of two main components. Firstly, to determine call quality, it is necessary to extract the voice streams from captured network traffic. The extracted sound files can then be analysed by various VoIP quality models to determine the perceived quality of a VoIP call.

The second component is the analysis of network characteristics. Loss, delay and jitter are all known to influence perceived call quality. These characteristics are, therefore, determined from the captured network traffic and compared with the call quality. Using the statistics obtained by the repeated comparison of the call quality and network characteristics, a network specific algorithm is generated. This Non-Intrusive Quality Prediction Algorithm (NIQPA) uses basic characteristics such as time of day, delay, loss and jitter to predict the quality of a real-time

VoIP call quickly in a non-intrusive way. The realised algorithm for each network will differ, because every network is different.

Prediction results can then be used to adapt either the network (more bandwidth, packet prioritising) or the voice stream (error correction, change VoIP codecs) to assure QoS.

Uittreksel

’n Netwerk Verkeersanaliseerder vir die Voorspelling van VoIP Oproepkwaliteit

(“A Network Traffic Analysis Tool for the Prediction of Perceived VoIP Call Quality”)

G.S.H. Maritz

Departement Elektriese en Elektroniese Ingenieurswese
Universiteit van Stellenbosch
Privaatsak X1, 7602 Matieland, Suid Afrika

Tesis: MScIng (Elektronies)

Desember 2011

Die kwaliteit van spraak oor die internet (VoIP) kommunikasie is afhanklik van die netwerk wat gebruik word om spraakpakkies te vervoer tussen die eindpunte. Netwerk eienskappe soos bandwydte, vertraging en verlies is krities vir intydse spraakverkeer en kan nie altyd gewaarborg word deur netwerkverskaffers nie. Dit is belangrik vir die netwerk diensverskaffers om die vereiste gehalte van diens (QoS) te verskaf aan hul kliënte. Die oplossing wat hier voorgestel word is om die kwaliteit van ’n VoIP oproep intyds te voorspel, deur middel van die netwerkstatistieke.

Die belangrikste doel van hierdie projek is om ’n netwerk analise-instrument te ontwikkel. Die instrument versamel betekenisvolle statistiek deur van netwerkverkeer gebruik te maak. Hierdie statistiek sal dan gebruik word om te voorspel wat die gehalte van ’n VoIP oproep sal wees vir sekere netwerk toestande. Hierdie studie berus op die ondersoek en implementering van twee belangrike komponente.

In die eerste plek, moet oproep kwaliteit bepaal word. Spraakstrome word uit die netwerkverkeer onttrek. Die onttrekte klanklêers kan dan geanaliseer word deur verskeie spraak kwaliteitmodelle om die kwaliteitdegradasie van ’n spesifieke VoIP oproep vas te stel.

Die tweede komponent is die analise van netwerkeienskappe. Pakkieverlies, pakkievertraging en bibbereffek is bekend vir hul invloed op VoIP kwaliteit en is

waargeneem. Hierdie netwerk eienskappe word dus bepaal uit die netwerkverkeer en daarna vergelyk met die gemete gesprekskwaliteit.

Statistiek word verkry deur die herhaalde vergelyking van gesprekkwaliteit en netwerk eienskappe. Uit die statistiek kan 'n algoritme (vir die spesifieke network) gegenereer word om spraakkwaliteit te voorspel. Hierdie Nie-Indringende Kwaliteit Voorspellings-algoritme (NIKVA), gebruik basiese kenmerke, soos die tyd van die dag, pakkie vertraging, pakkie verlies en bibbereffek om die kwaliteit van 'n huidige VoIP oproep te voorspel. Hierdie metode is vinnig, in 'n nie-indringende manier. Die gerealiseerde algoritme vir die verskillende netwerke sal verskil, want elke netwerk is anders.

Die voorspelling van spraakgehalte kan dan gebruik word om òf die netwerk aan te pas (meer bandwydte, pakkie prioriteit) òf die spraakstroom aan te pas (foutkorreksie, verander VoIP kodering) om die goeie kwaliteit van 'n VoIP oproep te verseker.

Acknowledgements

The author would like to thank the following people for their contribution towards this project.

- My supervisor, Dr. Riaan Wolhuter, for his time, knowledge, suggestions, help, friendliness and interesting conversation.
- Christiaan Brand for helping with the real-world captured network traffic and for telling me about OpenBTS and getting it to Stellenbosch University.
- Devon Louw for the use of his static IP and for wasting his Saturday.
- The guys in my lab at the electronic engineering department: Dirk Warnich and Tim Verschaeve
- My biggest fans: Pa, Ma, Jacques, Stanley van den Heever and Peter Crafford.
- For inspiration: The OpenBTS guys, Twitter followers, Pink Floyd, Google and The Narrow.

Contents

Abbreviations

3SQM Single Sided Speech Quality Measurement

AMR Adaptive Multi-Rate

API Application Programming Interface

ATA Analogue Telephone Adapter

CSRC Contributing Source

DiffServ Differentiated Services

EFR Enhanced Full Rate

FEC Forward Error Correction

FR Full Rate

GPL General Public License

GUI Graphical User Interface

HTTP Hypertext Transport Protocol

IETF Internet Engineering Task Force

ICMP Internet Control Message Protocol

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IntServ Integrated Services

ITU International Telecommunication Union

ITU-T International Telecommunication Union - Telecommunications Standardisation Sector

MDC Multiple Description Coding

MOS Mean Opinion Score

NIQPA Non-Intrusive Quality Prediction Algorithm

NTP Network Time Protocol

PBX Private Branch Exchange

PCAP Packet Capture

PCM Pulse Code Modulation

PESQ Perceptual Evaluation of Speech Quality

PLC Packet Loss Concealment

PMCC Pearson product-Moment Correlation Coefficient

PSTN Public Switched Telephone Network

PT Payload Type

QoS Quality of Service

QoE Quality of Experience

RFC Requests for Comments

RSVP Resource Reservation Protocol

RTCP Real Time Control Protocol

RTP Real Time Protocol

SIP Session Initiation Protocol

SMTP Simple Mail Transport Protocol

SSRC Synchronization Source

TCP Transmission Control Protocol

UA User Agent

UDP User Datagram Protocol

URI Uniform Resource Identifier

VAD Voice Activity Detection

VoIP Voice over Internet Protocol

List of Figures

List of Tables

Chapter 1

Introduction

This chapter serves as the introduction and summary for the thesis. The chapter commences with Section 1.1 where the importance of the work is cited. Section 1.2 provides basic background research, which is needed to understand the remainder of this chapter. In Section 1.3 the objectives of this project are highlighted. In order achieve these objectives, certain contributions, as conveyed in Section 1.4, were made. Section 1.5 then gives a synopsis of the project as a whole.

1.1 Motivation

Voice over Internet Protocol (VoIP), as the name suggests, is the transmission of voice conversation over Internet Protocol (IP) based networks or, more simply put, a telephone service over the Internet. These networks were not designed to transfer packets real time. Network characteristics, such as packet loss, delay and jitter, deteriorates real-time transfers and therefore VoIP call quality [4] [5] [6] [7] [8].

A need exists to measure the speech quality of a VoIP call [9]. This measurement should be done without invading the privacy of the people taking part in the call and without using more bandwidth than is necessary. The results of such measurements can then be used to adapt the IP based networks, and is important for Quality of Service (QoS) service agreements.

There is, therefore, the need to develop a VoIP call quality analysis tool. There are several objective speech quality estimation models. However, all these existing models are either bandwidth consuming or time consuming.

The aim of this work is to analyse captured network traffic characteristics during VoIP calls. The existing objective VoIP speech quality estimation models can then be utilised to set up statistical models for that network. This will enable the fast prediction of VoIP call quality on that specific network using its own network characteristics.

If a live VoIP call is to be analysed using current objective methods, the speech needs to be extracted from the network traffic. Privacy implications exist here.

Furthermore, the method presented here allows for the quality prediction of multiple VoIP calls. The algorithms used by the objective methods also needs processing power, which may be a problem when analysing the speech quality multiple calls.

It is important to note that this project does not evaluate VoIP, it evaluates the developed prediction tool.

In the thesis, *Speech Quality Prediction for Voice over Internet Protocol Networks* by Lingfen Sun [10], the main goal was "to develop novel and efficient models for non-intrusive speech quality prediction". In this thesis, we will build on the work done in [10]. We design and develop our own real world VoIP data capturing and analysing techniques, which are then used to verify and extend the work done by [10].

1.2 Background

VoIP is the transmission of voice conversation over IP based networks. VoIP technology is replacing existing telephone networks or Public Switched Telephone Networks (PSTNs).

In these PSTNs voice traffic moves on a circuit-switched network, where a fixed path is established and kept throughout the duration of the call. Network resources are allocated to the call and thus time-based pricing is incorporated. These networks are extremely reliable but very expensive. Data-traffic moves on packet-switched networks. VoIP traffic, together with files, media and messages, is broken up into packets and sent onto the network. Each of these packets have header information containing the destination address and descriptors. These packets move individually on the network and different paths may be taken by different packets. Packets do not necessarily arrive in the same order in which they were sent, and may even be lost. This method is very efficient, as network resources are shared and thus bandwidth is saved, but not very reliable.

In VoIP architecture a voice call has two ends, the source and a destination. At the source, as seen in Figure 1.1 a Codec converts the analogue voice generated by the speaker to the digital form which is transmitted. This digital data is then packetised into packets of equal size for transmission over a network. These packets also contain information about the source, the destination and a timestamp for reconstructing the data in the correct order. At the destination, speech frames are buffered and put into the right order, depacketised and converted back to the ana-

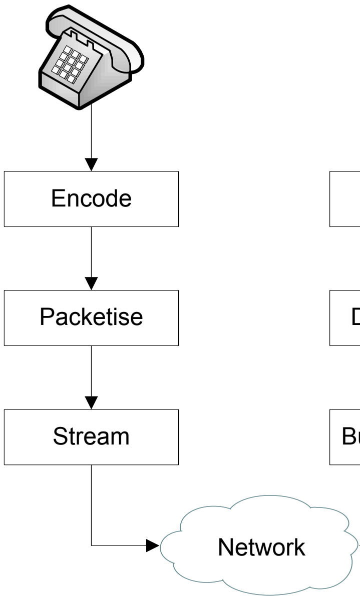


Figure 1.1: Basic VoIP architecture

logue voice signal.

VoIP popularity has exploded because of this improved utilisation of bandwidth and an effective monetary saving.

A packet-switched network is, however, not very efficient for real-time voice transfer, if the network is not correctly managed.

1.2.1 Quality of Service and Quality of Experience in VoIP

Quality of Service (QoS) refers to the network's capability to deliver a guaranteed level of service to the user. Voice packets are transmitted over IP networks in the same manner as data packets. These data and voice packets travel over the same network. In a well planned network, QoS is used to distinguish and route traffic based on its priority. There are three major network impairments that compromise voice quality:

- Packet loss - Too much traffic in the network causes the network to drop packets. Packet loss occurs frequently on data networks. Most protocols provide reliable delivery by requesting retransmission of lost packets (TCP). Voice packets are usually sent using real-time protocols, thus dropped voice packets are discarded and not retransmitted. This significantly influences the quality of a VoIP call.

- Delay - Voice cannot tolerate much delay. Latency is the average time it takes for a packet to travel between its source and its destination. Callers usually notice round trip voice delays of 250 milliseconds or more. If there is too much traffic on a network, voice packets may be delayed behind other data packets for too long, and the quality of the call will be compromised.

- Jitter - In order for the voice to be of good quality, voice packets need to arrive at regular intervals. Jitter is caused by variation in the rate of packet delivery. Jitter is caused by too much data on a line, as well as by bursts of network traffic.

VoIP can deliver the required quality, and can even outperform dedicated voice networks, if QoS mechanisms are properly implemented to control these factors. It is thus very important to understand the factors concerned to enable efficiently implemented VoIP networks.

Quality of Experience (QoE) is the user's perception of the performance of a device, a service, or an application. User perception of quality is a fundamental determinant of acceptability and performance for any service platform [11].

Carriers and service providers are concerned with defining network technologies and QoS mechanisms to deliver adequate QoE to their client. This should be done while minimizing operating cost and maximising link utilisation. It is these issues that are addressed by this project. This is done by developing a method to predict QoE by using QoS statistics.

1.2.2 Measuring QoS

There are two approaches to the measurement of perceived speech quality in telecommunication networks: Subjective and Objective.

Subjective testing is the most reliable approach to assess speech quality because it measures the way humans perceive speech. Voice QoS is measured with widely accepted Mean Opinion Scores (MOSs), with a range of between 1 and 5.0. A MOS score of 1 means that all users would be dissatisfied. When a MOS is lower than 2.5 it is generally accepted as having an unacceptable level of call quality.

Objective methods refer to those algorithms which are carried out by machines without the involvement of human listeners. Objective methods can be classified into two categories, intrusive or non-intrusive, based on whether a reference speech signal is employed during the test or not.

The two main objective QoS measurement models are Single Sided Speech Quality Measurement (3SQM) and Perceptual Evaluation of Speech Quality (PESQ)

Despite the work done in [10], there exists no accepted standard to predict speech quality in VoIP networks using network characteristics [12], only speech quality assesment standards for example PESQ [13] and 3SQM [14].

1.3 Objectives of this Study

This thesis serves as an exploration into the state of VoIP call quality measurement technology. The measurement models, found in the research, will then be used to find the relationships between network characteristics and impairments and speech quality. These relationships will be used for the prediction of speech quality.

The main objectives of this study are:

- To investigate how IP networks handle real-time voice communication.
- To investigate VoIP at packet level
- To investigate the different models and methods used for measuring perceived speech quality.
- To design and develop a tool which:
  - Intercepts network traffic.
  - Extracts the voice streams.
  - Converts the voice streams to speech signals.
  - Analyse the resulting speech signal by using objective perceived speech quality measurement models.
  - The results obtained from these measurements must then be compared to the network impairments, to enable us to quantify their impact on perceived speech quality.
  - The development of a non-intrusive speech quality predictor. This predictor tool will be able predict VoIP call quality by obtaining a specific network's characteristics.
- To show that for different IP networks a different relationship between network impairments and speech quality exist. This is widely known, but this fact is used to show that a generic model to predict speech quality using network statistics on any IP network can not be used. Therefore, a different prediction model will be generated exist for each network.

These objectives were successfully met and demonstrated.

1.4 Contributions

The contributions resulting from the above mentioned objectives, are as follows:

- A verification and extension of the work done in [10]. In terms of extension, a realtime packet capture and analysis tool was developed.
  - A prediction tool was designed to enable us to predict perceived speech quality using only network impairments. This tool uses a network specific derived algorithm, the Non-Intrusive Quality Prediction Algorithm (NIQPA).
  - The NIQPA uses quickly obtainable network characteristics such as time of day, delay, loss and jitter to predict the quality of a real-time VoIP call quickly and in a non-intrusive way.

1.5 Thesis Overview

Chapter 2 Chapter 2 provides information on the research carried out. The basic principles regarding VoIP systems and characteristics, capabilities and limitations of the common protocols used in VoIP, are discussed. The perceived quality of speech measurement models will also be discussed. The relevance of the different topics will be made clear. The various issues and characteristics concerning VoIP and QoS will be discussed. The information presented governs the design of the VoIP analysis tool.

In Section 2.1 an overview of VoIP features are given. The inner workings of VoIP as a technology are discussed and examined. This includes protocol and infrastructure. This will be useful for the design and implementation of the network traffic analysis tool, as well as for extracting speech from the captured network traffic.

Voice coding technologies and the codecs used in the project (G 7.11 and GSM 06.10) are discussed in Section 2.2. The impact codecs have on packet size is also investigated, as packet size is relevant to this topic.

In Section 2.4 the concept of perceived QoS and factors affecting speech quality are presented. These factors, packet loss, delay and jitter are discussed in depth. The subjective and objective speech quality measurement methods are presented. Subjective voice quality measurement, such as MOS, is discussed and its limitations are presented. More importantly, the objective voice quality measurements including both intrusive and non-intrusive voice quality measurements are discussed.

Section 2.5 gives an overview of different VoIP implementations. This is important, as this overview highlights the value of this project.

In Section 2.6, an indroduction to regression analysis is given. Regression will be used to determine the relationship between the network characteristics and speech quality.

All the information discussed in this chapter is required for a realistic approach toward designing the VoIP analysis and prediction tool.

Chapter 3 Chapter 3 presents the theoretical design of the tests done on the QoS measurement models as well as the prediction tool created to predict the quality from network characteristics. This chapter utilises the information already discussed in Chapter 2, in the theoretical design phase of this project. After the project concept is put forward in Section 3.1, the basic concept for analysing captured network traffic is discussed (in Section 3.2).

A network analysis tool is designed, as described in Section 3.3. This tool utilises existing network monitoring tools and speech quality measurment models to generate network and quality statistics. These statistics can then be used to derive algorithms to model the relationship between the extracted network characteristics and speech quality. The derived algorithms can then be used to predict speech quality, as explained in Section 3.4.

Chapter 4 Chapter 4 implements and refines the designs and presents detailed breakdowns of the components highlighted in Chapter 3. The issues that were encountered, as well as the solutions for these issues, will be discussed.

The third party software components used in this project are discussed in Section 4.1. These applications are then utilised to achieve the objectives of the previous chapter.

The implementation of the analysis of network traffic is discussed in Section 4.2. The first step is to isolate the different VoIP streams. The network parameters, such as packet loss, inter-packet delay and jitter can then be calculated. The speech content of each of these streams is also extracted. These extracted speech files are then analysed using the QoS measurement models discussed in Section 2.4.

Next, in Section 4.3, this analysis of network traffic is implemented into a semi-real-time network analysis tool. This tool captures network traffic while analysing previously captured data as described in Section 4.2. This tool generates statistics in tabular form.

These gathered statistics are then used to derive algorithms (Section 4.4), which model the relationship between perceived speech quality and network parameters. The prediction tool uses these algorithms to predict perceived speech quality using only easily obtainable network parameters.

Chapter 5 Chapter 5 evaluates the project against the objectives presented in this present chapter. The different tools implemented are evaluated. Simulations and tests, which comprise lab generated and real world tests, are discussed and the results thoroughly dissected. This chapter is divided into two main sections; Section 5.1 and Section 5.2.

In Section 5.1, the impact of the networks' impairments, which were discussed in Section 2.4, on speech quality are studied and results are shown. The effect that network impairments have on speech quality is shown using simulation results. The difference between the speech quality measurement models (PESQ and Single Sided Speech Quality Measurement (3SQM)) can already been seen in these results. After the simulation results are discussed the network analysis tool test set-up is discussed. Four test servers have been used for the evaluation of this tool; The lab, local, distant and commercial servers. The results from the different servers are compared and discussed. It is shown that the 3SQM measurement model does not measure the speech degradation as expected. This is also found in the research done [8], therefore 3SQM is not used further for this project.

The network impairments (packet loss, inter-packet delay and jitter) of the different servers are discussed. Plots show the effect that these impairments have on speech quality. The correlation between these impairments is also discussed, and it is assumed that the correlation between inter-packet delay and packet loss is an indicator of network congestion. As congestion causes delay, and after a certain delay packets are dropped.

Another important parameter is time. Network usage differs at different times of the day. The variation of speech quality and the network impairments are shown.

The results are also shown for tests done under different test conditions (different sample size and codecs).

In Section 5.2, the designed and implemented speech quality prediction tool is discussed. It is shown that straight line least squares fitted models give more prediction error than second order polynomial models. It was shown that the relationship between jitter and speech quality does not provide a statistically meaningful result. This is because jitter is highly depended on the route taken by the packets. These routes vary. Time stamp information is used.

Packet loss, inter-packet delay and time are shown to be parameters that give the lowest prediction error. Therefore a combination of these three parameters is used to derive a final algorithm, the NIQPA.

The NIQPA, is then implemented in an application. This application quickly and efficiently predicts speech quality using only network traffic.

Chapter 6 Chapter 6 concludes with a summary of the project. Suggestions are made on how the concept, which is presented in this thesis, can be extended in future work.

Chapter 2

Background and Research

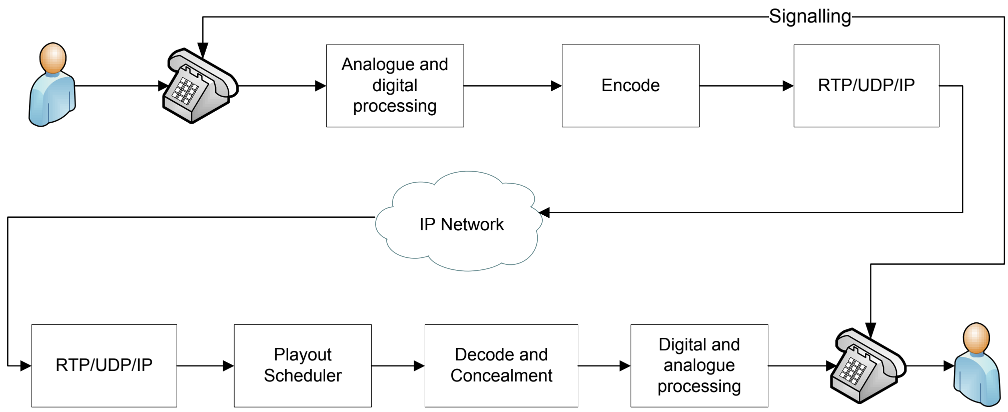


Figure 2.1: VoIP call components

The Internet is a very popular means of communication. It was set-up as a network and thus provides many services by which the user can communicate, including email, instant messaging and social networking.

Internet telephony is a combination of hardware and software that enables people to use the Internet as the transmission medium for telephone calls, which is simply called Voice over Internet Protocol (VoIP). As shown in Figure 2.1, there are various components which are part of a basic VoIP call.

Voice is captured by a microphone, sampled and digitised and encoded to a format which is chosen by the application. Typically a voice frame is of 20 ms duration and contains 160 voice samples, where each sample is 8 bits of information sampled at 8000 Hz. Forward Error Correction (FEC) or Multiple Description Coding (MDC) can be added to create redundant samples of the existing samples. These redundant samples are then transmitted with a time shift from the original samples, to reduce the probability of losing both the original and redundant samples [4].

The encoded voice frames are then packetised. This means gathering samples to a transmission block. Addressing information, which usually includes Internet Protocol (IP), User Datagram Protocol (UDP) and Real Time Protocol (RTP) headers, is added to the block. The packets are then sent via a network interface.

Once they have been received by the receiver, the packet headers are removed and any FEC or MDC are applied, if used. Packets need to be decoded in continuous blocks, therefore packets are buffered. The packets may arrive in a non-continuous fashion; the timing information of the RTP header is used to place frames in the correct sequence.

The application can also take measures to improve quality by using a technique called Packet Loss Concealment (PLC) where lost speech frames are compensated for by creating approximations of the lost packets from those received. In some cases PLC and speech encoding are combined in one algorithm. After the decoding the audio samples are transferred to the receiver's audio output device.

The following sections will describe the VoIP components in more detail.

2.1 Protocols used in VoIP

The protocol structure of VoIP is shown in Figure 2.2 [4].

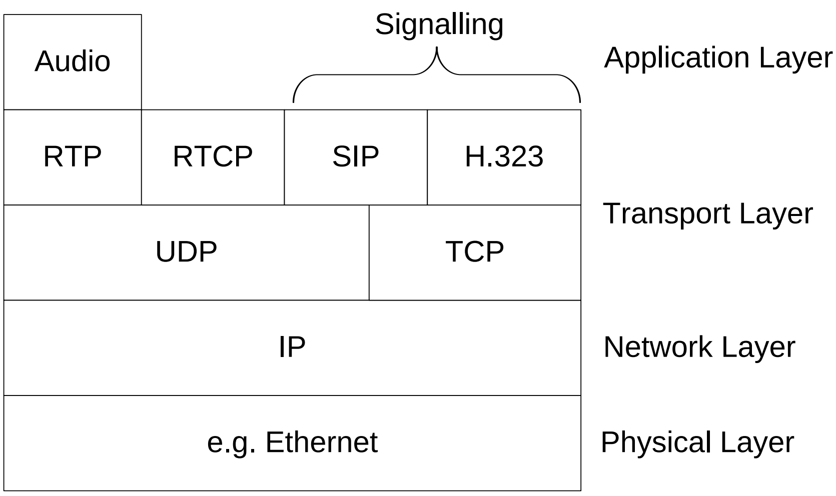


Figure 2.2: VoIP protocol structure

As the name suggests, VoIP uses IP in the *Network Layer*. IP is designed to deliver packets reliably between two nodes. In this project the physical layers are not considered.

For this research two *Transport Layer Protocols* need to be discussed, Transmission Control Protocol (TCP) and UDP. TCP is a higher level protocol, which is connection oriented and reliable. This means that in practice TCP handles acknowledgements and the retransmission of lost data. These retries and acknowledgements take valuable time and TCP is, therefore, not the best protocol for transferring voice in real time [15].

An alternative higher level protocol is UDP. UDP is connectionless and unreliable. It does not support acknowledgements or re-transmissions of lost packets. Each UDP packet consists of a small header and user data and is called a UDP datagram [15].

For VoIP, real-time data transmission or timeliness is more important than reliability. It is, therefore, understandable why UDP is utilised for VoIP communication.

In VoIP, protocols are used to perform two essential tasks: *Signalling* and *Data transfer*.

Signalling refers to the initiation of a call session, whilst data transfer is concerned with getting the voice data from the sender to the receiver.

2.1.1 Protocols used for Voice Transfer

It is important for this project to understand how voice data is transferred over the network. This will enable us to be able to extract the relevant data from the voice data streams. With the streaming of real time media, mechanisms need to be in place to ensure correct reconstruction of the media and detection of network delays and impairments.

2.1.1.1 Real Time Protocol (RTP)

RTP is the main transport protocol used for the VoIP media streams. The RTP protocol, as the name states, was developed to carry data that has real-time properties. RTP has many capabilities, which include support for unicast and multicast network services. This includes the ability to synchronise multiple streams arriving at a single receiver, if this is supported by the underlying network [3].

RTP is generally used with UDP, but it may be used with other suitable network and transport protocols.

The main role of RTP, with regard to voice stream, is to ensure intelligible reconstruction of the streamed data.

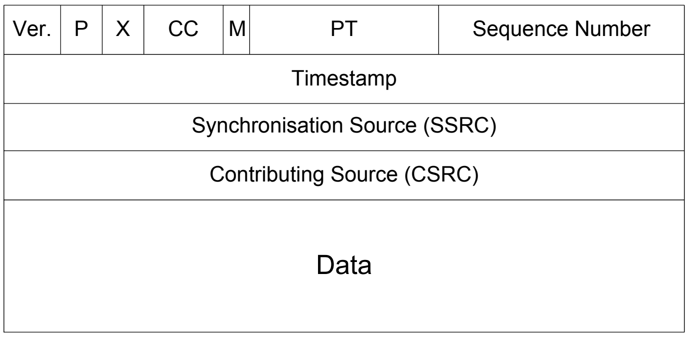


Figure 2.3: RTP header structure [1]

Table 2.1: RTP Payload Types [1]

Payload type	Name
0	PCM-Ulaw
2	G.721
3	GSM
31	H.261 (Video)

The RTP header structure is shown in Figure 2.3. The important fields for this study as shown in Figure 2.3 are [1]:

- Payload Type (PT) - This specifies the payload type and is filled by an application at the source and used at the destination. Some of the payload types specified by the IETF are shown in Table 2.1. VoIP CODECs will be discussed later in this chapter.

- Packet sequence number - This field is used to store the current packet number in the stream. It is used by the receiver to restore packet sequence when packets arrive out of order. It is also used to determine when packet loss has occurred.

- Timestamp - Based upon the sampling instant of the first octet in the RTP data. This allows jitter to be detected.

- Synchronization Source (SSRC) - Identifies the source of packet streams. This value should be unique and is chosen randomly, with the intent that no two pairs of synchronization sources within the same RTP session will have the same SSRC.

- Contributing Source (CSRC) - Identifies a specific media source from several others that have been mixed together. This is useful for multiparty conferences.

This header information describes how the codec bit streams are packetised and tells the receiver how to reconstruct the data [2].

RTP does not have an fixed UDP port, instead it is generally assigned to use a even port. While the next port is used for Real Time Control Protocol (RTCP).

RTP is an application layer protocol and does not provide any QoS guarantees. It does, however, allow for transmission impairments to be detected. RTCP is used to send back the impairment statistics.

2.1.1.2 Real Time Control Protocol (RTCP)

The Real Time Control Protocol (RTCP) is a control protocol to RTP. It provides control services for a data stream that uses RTP. The main feature of RTP is to provide feedback on the quality of the transmission link. Information that is conveyed back to the sender includes the following [2]:

- The Network Time Protocol (NTP) timestamps, which can be used to determine round trip delay
- Packet counts
- Lost packets reported as a fraction of the total and as a cumulative number
- Highest sequence number received
- Inter-arrival jitter

To calculate round trip delay, the sender transmits a report containing the time the report was sent. When the receiver transmits the report back to the sender it subtracts the time it took to process the report. Thus, the receiver can calculate round trip delay excluding the time spent at the end points. This can be done in both directions.

RTP and RTCP are the media transport protocols. Call set up and tear down is done by a signalling protocol.

2.1.2 Signalling Protocols

It is necessary to understand the call setup process, which will be used to set up VoIP calls. Signalling establishes a virtual circuit over the network for the media stream over which the packeted voice will flow. Signalling is independent of the media flow. It also determines the type of media to be used in the call. There are currently two popular signalling protocols in VoIP: H.323 and SIP [3]. These two protocols are shown in Figure 2.2.

H.323 was developed by the International Telecommunication Union - Telecommunications Standardisation Sector (ITU-T) in 1997 for packet switched multimedia communication. H.323 was designed before the emergence of VoIP. As it was not specifically designed for VoIP, it is planned that the installed base of H.323 will be replaced by Session Initiation Protocol (SIP). Almost all VoIP service providers use SIP-based networks, therefore only SIP will be discussed for this project [16]. Neither H.323 nor SIP was designed to interface with the PSTN [15].

2.1.2.1 Session Initiation Protocol (SIP)

SIP was created by the Internet Engineering Task Force (IETF) in 1999 [17]. SIP is an application-layer control protocol that can initiate, modify and terminate multimedia sessions (conferences) such as internet telephony calls. Its primary function is session setup, but it also allows for other functions and uses, such as notifications for presence and messaging. It is a text-encoded protocol based on elements of the Hypertext Transport Protocol (HTTP), which is used for web browsing, and also Simple Mail Transport Protocol (SMTP), which is used for email [2]. This makes it easy to debug because the messages are easy to construct, if you are a developer, and easy to see, if you are a network manager. Contrasted with H.323, SIP is an exceedingly simple protocol. Nevertheless, it has enough powerful features to model the behaviour of a very complex traditional telephone Private Branch Exchange (PBX).

SIP can run over Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) and it can use either TCP or UDP. The most common implementations, though, use IPv4 and UDP. This minimizes overhead, thereby speeding performance [2].

A SIP client generates a SIP request. A SIP server responds to the request by generating a response. The expanding set of request types or methods, are shown in Table 2.2 [18]. The rest of the methods are extensions of SIP and are defined in separate Requests for Comments (RFC) or internet drafts. New methods are continually proposed to add functionality to the protocol [2]. After the SIP methods have been received, the other party replies with a response call, as seen in Table 2.3. An example of the use of SIP methods and response codes can be seen in Figure 2.4.

In both IP telephony and traditional telephony calls there are two distinct phases. This can be seen in Figure 2.4. The first phase is call set-up and it includes the details to initiate a voice call. After the call has been set up, the data transfer phase of

Table 2.2: SIP Methods [2]

INVITE	Session setup
ACK	Acknowledgement of final response to INVITE
BYE	Session termination
CANCEL	Pending session cancellation
REGISTER	Registration of a user's URI
OPTIONS	Query of options and capabilities
INFO	id-call signalling transport
PRACK	Provisional response acknowledgement
UPDATE	Update session information
REFER	Transfer user to a URI
SUBSCRIBE	Request notification of an event
NOTIFY	Transport of subscribed event notification
MESSAGE	Transport of an instant message body
PUBLISH	Upload presence state to a server

Table 2.3: SIP response codes [3]

100 Trying	413 Request, entity too large
180 Ringing	414 Request, URI too large
181 Call is being	415 Unsupported media type forwarded
182 Queued	420 Bad extension
183 Session progress	480 Temporarily not available
200 OK	481 Call leg/transaction does not exist
202 Accepted	482 Loop detected
300 Multiple choices	483 Too many hops
301 Moved permanently	484 Address incomplete
302 Moved temporarily	485 Ambiguous
305 Use proxy	486 Busy here
380 Alternative service	487 Request cancelled
400 Bad request	488 Not acceptable here
401 Unauthorized	500 Internal severe error
402 Payment required	501 Not implemented
403 Forbidden	502 Bad gateway
404 Not found	503 Service unavailable
405 Method not allowed	504 Gateway timeout
406 Not acceptable	505 SIP version not supported
407 Proxy authentication	600 Busy everywhere
408 Request timeout	603 Decline
409 Conflict	604 Does not exist anywhere
410 Gone	606 Not acceptable

the call begins, which uses transfer protocols to transfer the voice packets.

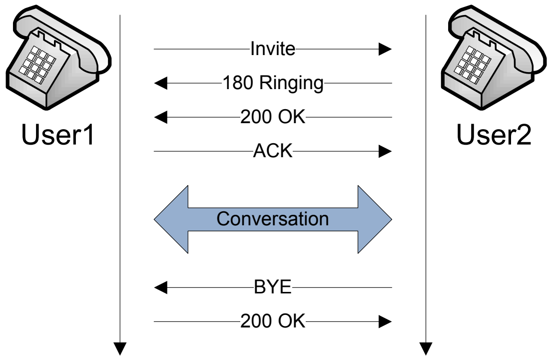


Figure 2.4: SIP Call Setup Process

SIP was designed to set up real-time sessions for any media between groups of participants. Therefore, SIP has bigger potential than simply IP telephony, as it can also be used for video and audio streaming, instant messaging and online multicast conferences. Some of the major functions within SIP, from a VoIP perspective, can be seen in Table 2.4 [19].

SIP devices or User Agents (UAs) can talk directly to each other. However, generally an intermediary system or PBX is used which acts as a SIP proxy. SIP proxies only participate in the SIP request and responses, and when the call has been set up, the voice packets are sent directly between the parties, without going through the proxy.

SIP proxies handle some of the tasks of the UAs. For example, if a UA makes a call to an extension, "1234", it is difficult for the phone to determine who "1234" is. It could be a UA or several UAs to be called at once. The SIP proxy knows what or who "1234" is.

SIP uses Uniform Resource Identifiers (URIs) to identify UAs, for example:
sip:user1@there.com.

SIP is only used for call control; features such as voice mail and auto-attendant are not part of the SIP protocol. These features are provided by end points which are UAs themselves (that send and receive calls). These end points are called "SIP servers" or "SIP PBXs".

The following is an example of a SIP request from User 1 [19]:

1. INVITE sip:user2@there.com SIP/2.0
2. Via: SIP/2.0/UDP 4.3.2.1:5060 ;branch=z9hG4bK765d
3. Max-Forwards: 70

Table 2.4: Functions of SIP

Function	Description]
User location and registration	End points (users/phones) notify SIP proxies of their location and determine which end points will participate in a call
User availability	SIP is used by end points to determine whether they will answer a call.
User capabilities	SIP is used by end points to negotiate media capabilities. For example, a mutually supported voice codec.
Session setup	SIP tells the end point that the phone should be ringing. SIP is used to determine the session attributes which are agreed upon by all users in a call.
Session management	SIP is used to transfer and terminate calls, and change call parameters in mid-session (such as adding another user to an existing call)

4. To: User2 <sip:user2@there.com>
5. From: User1 <sip:user1@here.com>;tag=1928301774
6. Call-ID: a84b4c76e66710@123.123.123.123
7. CSeq: 314159 INVITE
8. Contact: <sip:user1@123.123.123.123>
9. Content-Type: application/sdp
10. Content-Length: 142

The steps in this example explained:

1. INVITE method, Request URI, current version of SIP
2. Version of SIP and transfer protocol , IP address or hostname of originator of request, branch parameter
3. Stipulates the maximum number of hops to the destination
4. Display name and URI to which request was directed
5. Display name and URI of originator of request. Tag parameter is used for identification.
6. Unique identifier for this session

7. Count of successive requests in a call
8. SIP URIs that provide information to the other party to enable it to contact User 1
9. Content-Type describes the message body, which is not shown here
10. Byte count of the message body

2.2 VoIP Speech Codecs

Codecs are used to convert an analogue voice signal to a digitally encoded version. Codecs vary in sound quality, the bandwidth required, the computational requirements, etc. In VoIP, codecs are used to convert the analogue voice generated by the speaker to the digital form which is transferred over the network. This digital form is usually compressed to save network bandwidth.

Each component of a VoIP system, such as the server, program, phone or gateway, typically supports several different codecs. These components negotiate which codec they will use. Some codecs support silence suppression, where silence is not digitised or transmitted. Comfort noise may be generated to fill these voids. Several codecs exist. The most popular codecs used for VoIP are listed in Table 2.5.

Each codec has a different bit rate which samples the analogue waveform at specific intervals. These samples are accumulated for a fixed period and create a frame of data. A packetisation time (frame duration) differs from codec to codec, as seen in Table 2.5 [20] [21]. Higher bandwidth allows for higher bit rate, which enables lower compression and thus better voice quality.

Table 2.5: Codec Characteristics

Codec	Bit Rate [kbps]	Sampling rate [kHz]	Frame size [ms]
G.711	64	8	20
GSM 06.10	13	8	20
G.729 (ACELP)	8	8	10
iLBC	8	13.3	30
G.726 (ADPCM)	16,24,32,40	8	10
Speex (Wide Band)	8,16,32	4-44,2	34

The sizes of packets in VoIP transmission is based upon the voice packetisation which is encoded differently by different voice codecs (coder and decoder). Packet size may vary between 10 and 320 bytes of digital voice, depending on the codec.

The bandwidth required for VoIP is determined by [22]:

- Codec and sample period

- Overhead
- Transmission medium (Each medium adds its own headers, framing and checksums)
- Sound processing (such as silence suppression) which is usually integrated with the codec.

The voice payload size can be calculated using Equation 2.2.1. In this example, for the G.711 u-law codec (with a bit rate of 64 kbps) and 20 milliseconds packetisation time:

$$\begin{aligned}
\text{Voice payload size [bits]} &= (\text{Codec bit rate})(\text{Packetisation time}) \\
&= (64kbps)(20ms) \\
&= 1280bits \\
\text{Voice payload size [bytes]} &= \frac{\text{Voice payload size [bits]}}{8} \\
&= \frac{1280}{8} \\
&= 160bytes
\end{aligned} \tag{2.2.1}$$

The IP overhead is the overhead occurring at layer 3 and above. For VoIP this usually means IP (20 Bytes), UDP (8 Bytes), and RTP (12 Bytes). This is a total of 40 Bytes of IP overhead. An Ethernet header adds a further 18 Bytes. Thus for a transfer over Ethernet the total overhead is 58 bytes Total VoIP packet size in this case can therefore be calculated as [22]:

$$\begin{aligned}
\text{VoIP packet size} &= \text{Voice payload size} + \text{Total overhead} \\
&= 160 + 58 \\
&= 218bytes
\end{aligned} \tag{2.2.2}$$

Putting more audio frames in each IP packet would be more efficient, as this will decrease the transmission overhead. It does, however, mean that if a packet is lost, more speech frames are lost. It also increases the latency of long audio frames and this affects the perceived audio quality of the call.

Some codecs require payment of royalties for their use in a product or program. In this project only G.711 and GSM 06.10 will be studied. The quality of the different codecs is beyond the scope of this project, but is recommended for future study.

2.2.1 G.711

G.711 is an ITU-T standard for audio compression [23].

G.711, a specific implementation of Pulse Code Modulation (PCM), is a very commonly used waveform codec.

There are two slightly different versions [24]:

- μ-law - Used primarily in North America and Japan. Provides slightly more dynamic range than A-law. This is a 64 kbps PCM (Pulse Code Modulation) codec and a companding variant of the ITU-T G.711 standard. All such codecs impose a minimal load on the CPU.
- A-law - Used in most other countries outside North America. Requires even less CPU processing power than μ-law.

2.2.2 GSM 06.10

GSM 06.10 or Full Rate (FR) was the first digital speech coding standard used in the GSM mobile phone system.

The bit rate of the codec is 13 kbit/s, or 1.625 bits/audio sample, it is often padded out to 33 bytes/50 ms or 13.2 kbit/s. , but at the time of development it was a good compromise between computational complexity and quality, requiring only in the order of a million additions and multiplications per second [25].

The quality of the coded speech is quite poor by modern standards. Therefore FR is being replaced by Enhanced Full Rate (EFR) and Adaptive Multi-Rate (AMR) standards, which provide much higher quality at a lower bit rate. GSM was developed to compromise between computational complexity and speech quality. It is still widely used in networks around the world, as it has no licensing requirements and offers excellent CPU-related performance [24].

2.3 Sound Processing

Sound processing attempts to counter the elements that have a negative influence on the interactive voice quality. This is usually integrated with the codec used. It includes [26]:

- Noise cancellation - Noise generated from the network or acoustic environment can be filtered. This includes echo generated.

- Voice Activity Detection (VAD) - Stops the transmit packets if there is silence, this enables bandwidth efficient transmission.
- Comfort noise generation - Generates background noise for voice communications during periods of silence that occur during the course of conversation.

These processes are beyond the scope of this project, but are recommended for future study.

2.4 Quality of Service

In the first chapter a brief introduction to Quality of Service (QoS) was given. In this section it will be discussed in more detail. QoS refers to a network's capability to deliver a guaranteed level of service to a user [3]. There are three major network impairments that influence QoS of a network [4] [5] [6] [7] [8]:

Delay Latency or delay is important in a voice call. If delay exceeds a certain value, a conversation can degrade into half-duplex communication. This will be uncomfortable for the users.

In VoIP, delay can be broken up into two parts: Processing and Network delay. Processing delay is that delay which is the result of processing in end systems, for example encoding.

Network delay has two parts; a fixed part, which depends on the performance of the network nodes in the transmission path, and includes transmission delay and propagation delay. The second part of network delay is the variable part. This is the time spent in queues, and depends on the network load. This variable delay can be reduced by implementing better packet scheduling mechanisms (forwarding and prioritising).

Packet delay can also be reduced by sending smaller packets, as was discussed in the previous section.

The ITU-T recommendation G.114 is generally accepted throughout the telephone industry [2]. The following are the values for one-way delay [27]:

- Less than 150 ms for acceptable conversation quality.
- No more than 400 ms for tolerable conversational quality.
- Delay over 400 ms is deemed unacceptable.

Jitter Delay variation, or jitter, obstructs the reconstruction of voice frames to their original sequence because frames do not arrive in a regular periodic pattern.

Jitter is defined as the difference in end-to-end delay of two consecutive packets. Jitter is managed by storing packets to allow the slower arriving packets to be placed in the correct order to be played in the original sequence. This is called de-jitter or playout buffers.

These buffers add further to the end-to-end delay, and are usually only effective on delay variations less than 100 ms. Jitter must therefore be minimised [28].

Packet Loss Packet loss occurs when a packet of data traveling over an IP network fails to reach it's destination.

The techniques used to manage packet loss include:

- Forward Error Correction (FEC) - Corrects the lost frames. Requires additional processing.
- Packet Loss Concealment (PLC) - Conceals lost frames by playing the last received frame. Effective at low loss rate.

For example, the PLC algorithm in G.711 codecs can recover reasonably well from a packet loss of up to 5%. But with very low bitrate codecs, even with their built-in PLC or PLR algorithms, packet loss rates of less than 1% are normally perceptible, and are often annoying [29].

Advanced packet scheduling mechanisms (forwarding and prioritising) also reduce packet loss due to full network queues.

2.4.1 Measurement of QoS

Service providers are challenged with monitoring networks to enable them to find the causes of voice quality problems over networks [9]. QoS measurement usually depends on network traffic measurements. However, measuring speech quality is more related to the QoS experienced by the user, and it is therefore important to service providers.

As mentioned in previous chapter, there are two approaches to the measurement of perceived speech quality in telecommunication networks: Subjective and Objective.

2.4.2 Measuring Speech Quality

Subjective speech quality is a subjective characteristic of speech quality. It refers to how speech is perceived by a listener, as it is the listener which rates the quality of a particular speech sample. The Mean Opinion Score (MOS) test method (defined in the International Telecommunication Union (ITU) specification, ITU-T Rec. P.800 [30]) provides a numerical indication of perceived quality, by expressing a single number, ranging between 1 and 5, where 1 is the worst and 5 the best perceived speech quality. Running subjective speech quality tests are quite expensive in terms of time and human resources, therefore, objective tests are necessary.

Objective tests run on machines and are developed to model subjective tests. There are two types of objective tests [5]:

- Intrusive - A reference signal is send through a network and the degraded signal (on the other side of the network) is compared to the reference signal to determine the signal degradation.

- Non-intrusive - No reference signal is sent into the network .

2.4.2.1 E-model

The E-model is an objective, non-intrusive approach to measure speech quality in networks [31]. It assesses the combined effects of a wide range of end-to-end transmission parameters (loss, delay, jitter, speech coding and echo parameters) that affect the conversational speech quality of narrowband telephone networks. Despite the fact that E-model is intended as an off-line transmission planning tool, due to its simple form it has found applications into on-line assessments as well [31].

The primary output of the E-model is a transmission rating factor, R, which can be transformed into other quality measures, such as MOS, as shown in Table 2.6. Equation 2.4.1 is used to calculate transmission factor, R [31].

$$R = R_o - I_s - I_d - I_{e-eff} + A \tag{2.4.1}$$

Where R = rating value

R_o = signal to noise ratio (noise sources)

I_s = voice impairments (side-tones and quantisation distortion)

I_d = delay and echo impairments

I_{e-eff} = packet loss impairment (including codec distortion)

A = advantage factor (compensates for poor quality where diminished quality can be tolerated)

The rating factor ranges from 0 to 100 can be transformed into a five-point MOS by using Equation 2.4.2 [31].

$$MOS = \begin{cases} 1 & R < 0 \\ 1 + 0.035R + R(R-60)(100-R)7.10^{-6} & 0 < R < 100 \\ 4.5 & R > 100 \end{cases} \tag{2.4.2}$$

Table 2.6: E-model and MOS Scores

User satisfaction	R-value	MOS score
Very satisfied	90	4.3
Satisfied	80	4.0
Some users dissatisfied	70	3.6
Many users dissatisfied	60	3.1
Nearly all users dissatisfied	50	2.6

The E-model is popular for estimating VoIP quality, but it has shortcomings. The burst effects of packet loss on speech are not well addressed in the E-model. However, as it is based on a complex set of fixed, empirical formulae and is applicable to a restricted number of codecs and network conditions (because subjective tests are required to derive model parameters), this hinders its use in new and emerging applications. Therefore it will not be implemented in this project. It can, however, be used together with PESQ [7].

2.4.2.2 Perceptual Evaluation of Speech Quality

Perceptual Evaluation of Speech Quality (PESQ) is based on a recommendation (P.862) by the Telecommunication Standardisation Sector of ITU. It is an objective method to assess end-to-end speech quality of narrow band speech codecs [13].

The PESQ principle works by sending a reference signal through a coding system (intrusive) and then comparing the degraded signal with the original. It outputs a prediction (in MOS score) of the perceived quality that would be given to the degraded signal by subjects in a subjective listening test.

Although PESQ is a popular tool and has been widely adopted in the industry, it does not always predict perceived speech quality accurately. PESQ is not capable of handling certain quality aspects, such as side-tones and delay. PESQ is, therefore, not intended to replace subjective listening tests as the benchmark for speech codec quality tests. It does, however, have an advantage over the subjective listener test because it provides a fast evaluation of speech.

Figure 2.5 shows the basic structure of the PESQ algorithm [13].

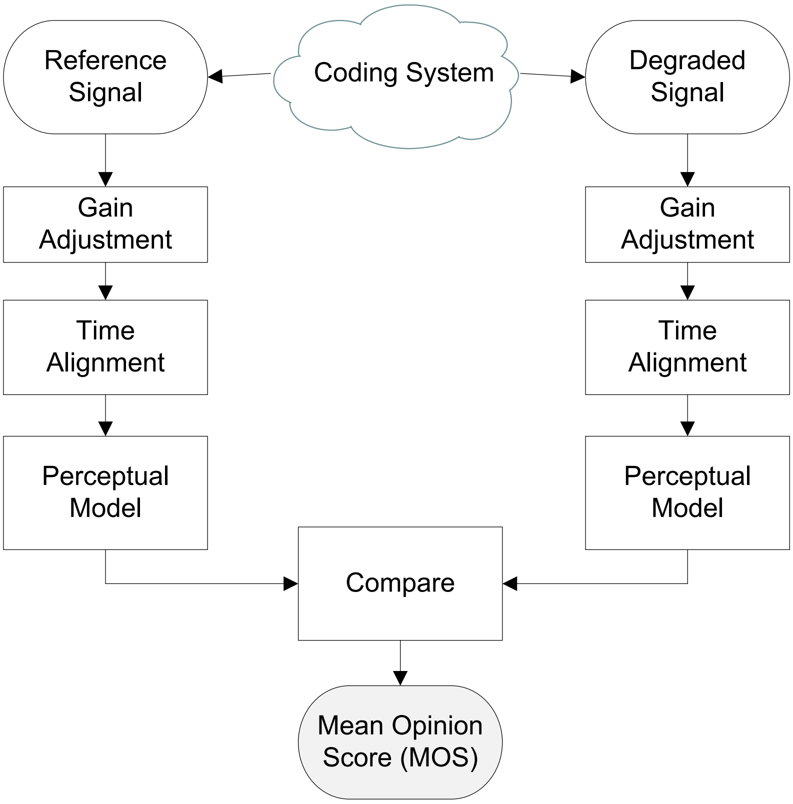


Figure 2.5: PESQ functional units

Gain Adjustment The gain of the coding system is not known. Therefore, the level of the original and degraded signals should be aligned. This is done by adjusting the frequency weighted energy of the two signals to a reference value. This step compensates for overall spectral balance difference which is usually caused by the coding scheme and should not affect the score.

Time Alignment Voice coding systems, as with packets sent over a network, may introduce an unknown and varying delay in the degraded speech signal. The PESQ algorithm performs time alignment between the original and degraded speech signals. This is done by firstly determining the average delay of the entire signal and then performing alignment on the sections marked as silent. These sections are called *utterances*.

These utterances are further split, and the smaller subsections are aligned with the original signal using a dynamic time warping algorithm. Subsections of utterances are aligned to maximise correlation of short term energy of the original and degraded signals.

Perceptual Model PESQ estimates the degradation of speech using a model of the human auditory system. The algorithm is performed by processing speech frames. Frames overlap each other by 16ms and are 32ms in length. The algorithm calculates the difference between the original and the degraded speech samples by comparing the power spectra of the corresponding frames of the two signals. This comparison of power spectra of the frames is based on perceptual considerations, including [13]:

- The power spectrum is binned to Bark bands (psychoacoustical scale, ranges from 1 to 24 Barks, corresponding to the first 24 critical bands of hearing). This is done to reflect that the human ear has better frequency resolution at lower frequencies.
- Non-linearities in the perception of sounds are converted to *loudness* values. The converted spectrum is referred to as the *loudness* spectrum. This is achieved by using Zwicker's loudness model, which is a standard for predicting the loudness of a sound.
- The difference between the loudness spectra of the two signals is referred to as *disturbance density*.
- A second (asymmetric) disturbance density is intended to reflect spectral information which is present in the degraded signal but not in the original. This error signal is composed of the introduced spectral components in the degraded signal [13].

MOS Estimate The MOS estimate as predicted by PESQ algorithm is a linear combination of average values of the disturbances. PESQ provides raw scores in the range -0.5 to 4.5. It is desired to provide a MOS-LQO (MOS-Listening Quality Objective) score from PESQ to allow a linear comparison with MOS. A PESQ Raw score can be mapped to MOS by using Equation 2.4.3 [32].

$$LQO = 0.999 + \frac{4}{1 + e^{-1.4945.PESQ+4.6607}} \tag{2.4.3}$$

PESQ's validity has been shown by its ratings being sufficiently correlated to subjective ratings. The average correlation over all 22 known ITU benchmark experiments is 0.935. These experiments cover 7 languages (English, French, German, Dutch, Swedish, Italian and Japanese) [13]. Intrusive methods, such as PESQ, have a number of disadvantages, such as consuming network capacity, when used for testing live networks [8].

2.4.2.3 Single Sided Speech Quality Measurement

ITU's Single Sided Speech Quality Measurement (3SQM) was developed for non-intrusive voice quality testing. It is based on recommendation ITU P.563 [14].

Recommendation P.862 (PESQ), as discussed in the previous subsection, requires a reference signal. The 3SQM model, however, according to ITU P.563 [14], is applicable to speech quality predictions without a separate reference signal. Therefore, it can be utilised for non-intrusive speech quality estimation, live network monitoring and assessment by unknown speech sources. This is of huge advantage to operators as this enables them to monitor the quality of live calls, without the need to test the network with artificial calls .

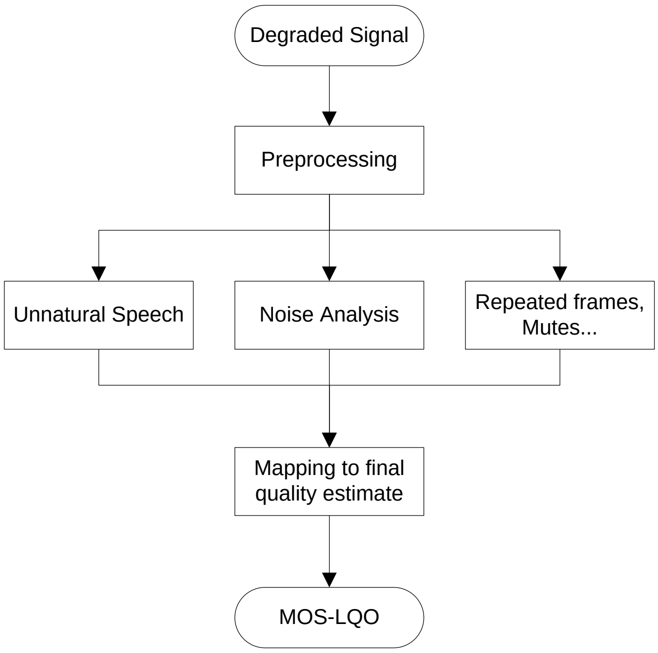


Figure 2.6: Block diagram of the 3SQM model

As shown in Figure 2.6, the 3SQM model works as follows [33]:

- Preprocessing of the input signal - this includes: IRS receive filtering, speech level adjustment and VAD, which separates voice and non-voice.
- Determination the speech and distortion parameters of the input signal - these parameters include [14]:
  - Speech level

  - Noise
  - Delay
  - Repeated Frames
  - Unnatural speech (beeps and clicks)

- These parameters are then combined to form the final speech quality estimation in MOS-LQO scale.

3SQM has lower correlation with subjective results and requires much more processing power and time than PESQ [8]. This is not ideal.

2.4.3 Methods to Improve QoS

The different management policies are adopted to ensure QoS. There are several approaches to provide QoS, two of the more popular methods are [15]:

- **Integrated Services (IntServ)** - Includes the specifications to reserve network resources in support of a specific application. IntServ uses Resource Reservation Protocol (RSVP) to allow the application or user to request and allocate bandwidth to support a connection. IntServ does not scale well, as each networking device (routers and switches) must maintain and manage the information across multiple paths.

- **Differentiated Services (DiffServ)** - DiffServ uses a different mechanism to handle flow across the network from that employed by IntServ. DiffServ uses bits in the header to recognise the need for QoS. DiffServ is more scalable than IntServ

2.5 VoIP Implementations

There are three main types of VoIP service in use today [34]:

Analogue Telephone Adapter (ATA) - A device used to connect one or more standard analogue telephones to a digital and/or non-standard telephone system such as a Voice over IP based network.

IP Phones - These phones look like normal phones except they have Ethernet connector ports instead of phone ports.

Softphone - Computers, mobile phones and other internet enabled devices running VoIP software. and speakers.

As mentioned in Chapter 1, a major challenge exists to measure the quality of voice calls over such IP networks [9]. This information is needed for Quality of

Service (QoS) monitoring and control purposes, which are essential to ensure that commercial and technical requirements, such as service level agreements, are met.

Several VoIP systems have been developed. Skype is one of the most popular of these [35].

2.5.1 Skype

Skype is a peer-to-peer (p2p) VoIP client developed by KaZaa that allows users make voice calls over the internet. It encrypts calls and the user directory is entirely decentralised and distributed among the nodes of the network.

Skype's main features include:

- Codecs - Wideband iLBC and iSAC [36]
- Contact list - Contact information stored by Skype
- Encryption - AES (Advanced Encryption Standard) is used to encrypt information [37]
- Host Cache - List of super node IP addresses and corresponding ports that a Skype client builds.

Skype's protocol is proprietary, unlike other open standards such as SIP.

Skype's ability to transverse NATs and firewalls as well as the ability to call landlines and mobile phones makes it one of the most successful implementations of VoIP. At peak times, there are up to 20 million people online. Skype is responsible for 8% of global international calling minutes, with its users making 3.1 billion minutes of calls to landlines and mobiles in the third quarter of 2009 [38].

2.5.2 VoIP Security

As VoIP becomes more popular, the security of these systems becomes more of a concern.

Unencrypted VoIP packets are sent over the internet, which means anyone that is between the source and destination can intercept these packets. This allows them to listen in to phone calls.

To help protect the privacy of VoIP users, these conversations can be encrypted. The following are some of the encryption methods used by VoIP systems [39]:

- **Built-in Encryption** - Many VoIP clients have responded to customer concerns over security by building encryption into their existing software. Skype, for example, has built-in encryption capability in their proprietary software.

- **Transport Layer Security and IP Security** - Transport Layer Security (TLS) and IP Security (IPSec) are some of the most common ways to encrypt VoIP calls. TLS encrypts information, like a VoIP call, that is traveling between two applications while IPSec encrypts data for two devices and all the applications running on them.

- **Secure Real-Time Transfer Protocol** - Secure Real-Time Transfer Protocol (SRTP) encrypts a call using a unique key. This ideal for protecting VoIP traffic because it has a minimal effect on the quality of the calls it encrypts.

- **Virtual Private Network** - Many companies already have VPN set up for securely transmitting data, and adding VoIP can be relatively simple. This process is similar to adding a PC or server to a traditional network. Calls on the VPN would then be secure, allowing users from remote offices, or even from their laptops, to communicate with other offices on the VPN network.

The United States of America government and military organizations are using Voice over Secure IP (VoSIP), Secure Voice over IP (SVoIP), and Secure Voice over Secure IP (SVoSIP) to protect confidential, and/or classified VoIP communications [40]. With the increased usage of VoIP, the security thereof becomes a necessity.

2.6 Regression Analysis

In statistics regression analysis is a tool used to investigate the relationships between variables.

Types of regression models include [41]:

- Linear regression model
- Simple linear regression
- Logistic regression
- Non-linear regression

Linear least squares regression is by far the most widely used modelling method [41]. Linear regression is also fast to compute and therefore useful for doing repetitive experiments for exploring different parameter settings. Therefore, for the purpose of the developing the concept, in this thesis, we will focus on the least squares method of regression.

2.6.1 Least Squares Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.

Linear regression models are extremely powerful, and have the power to empirically tease out very complicated relationships between variables. The most common form of linear regression is least squares fitting. Least squares fitting of lines and polynomials are both forms of linear regression [42].

The Least squares method is a mathematical procedure for finding the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations (residuals) from each data point to the line. When a point lies on the fitted line exactly, then its vertical deviation is zero [42]. A discussion of the mathematics behind the least squares method is presented in Appendix A.

2.7 Summary

In this chapter the various issues and characteristics concerning VoIP and QoS have been discussed. The information presented governs the design of the VoIP analysis tool.

In Section 2.1 an overview of VoIP features was given. The inner workings of VoIP as a technology were discussed and examined. This includes VoIP protocol and infrastructure. This will be useful for the design and implementation of the network traffic analysis tool, as well as for extracting speech from the UDP packets.

The codecs (G.711 and FR) are used in this project and are therefore discussed in Section 2.2. The impact that codecs have on packet size was also investigated.

In Section 2.4 the network impairments which impact speech quality were presented. These impairments, which are delay, packet loss and jitter, were discussed extensively. The subjective and objective methods speech quality measurement were presented. Subjective voice quality measurement, such as MOS, was discussed and its limitations were presented. More importantly, the objective voice quality measurement including both intrusive (PESQ) and non-intrusive (E-model and 3SQM) voice quality measurement was discussed.

In Section 2.5, an overview of different VoIP implementations was given. This is important, as it shows that the work being done of significance and has potential.

Finally, in Section 2.6, an indroduction to regression analysis was given. Regression will be used to determine the relationship between the network characteristics and speech quality.

All the information discussed in this chapter is required to develop a realistic approach toward designing the VoIP analysis and prediction tool. The information discussed here, is implemented in the theoretical design phase of this project, in Chapter 3. Therefore, the relevance of the information presented in this chapter will thus become clear in the subsequent chapters.

Chapter 3

Theoretical Design

The objective of this project is to develop an application to measure QoS in VoIP calls using captured network traffic. In Chapter 2 various issues related to VoIP and QoS measurement were discussed. The relationship between perceived speech quality and network characteristics based on these measurements can then be used to derive an model for the prediction of speech quality. This will enable the prediction of perceived speech quality, using only easily obtainable network traffic characteristics.

In this chapter we use the knowledge obtained in the previous chapter to design the network analysis and prediction tools. Block diagrams will be used to explain what needs to be done to complete the thesis objectives successfully.

3.1 Project Concept

There is no standardised scheme for monitoring VoIP quality. Ideally, the quality of calls should be measured by the users: Humans should evaluate the perceived Quality of Service (QoS). Unfortunately for statistically meaningful results, these subjective measurements are time consuming and costly.

VoIP applications usually determine networking QoS, by measuring observable network or transport variables, such as packet loss, inter-packet arrival times, and jitter. These metrics for networking QoS, however, do not reflect the perceived service quality accurately.

In this project we correlate perceived QoS with networking QoS to enable us to derive a prediction tool for quickly and efficiently determining perceived QoS in real-time. This is achieved by comparing the quickly obtainable network variables with the quality given by the results of perceptual quality measurement. When enough statistics have been gathered, quality prediction models can be formulated.

Time, day, location and instantaneous network conditions largely dictate the quality of Voice over IP calls. Therefore it is important to determine different models for different network situations.

The network traffic contains packets which are transmitted to the receiver (destination) from the sender (source). The first step in the design of this project is capturing the network traffic, which can then be analysed.

3.2 Captured Network Traffic

There are five steps in determining the relationship between the network characteristics and perceived speech quality from captured network traffic.

1. Find VoIP streams
2. Extract network characteristics
3. Extract speech
4. Analyse speech
5. Process data

These steps, which are also shown in Figure 3.1, will now be discussed.

3.2.1 Find VoIP Streams

Before the packets can be analysed it is necessary to determine which packets are used to carry VoIP data. There should be at least two streams in the captured data, one for each direction of a call. Once a VoIP stream's packets have been isolated, that specific stream can be analysed.

3.2.2 Packet Quality Characteristics Analysis

The transmission delay, jitter and loss of packets are networking variables that influence speech quality [43]. These variables (or characteristics) depend on the current network condition and the routing path. These networking characteristics need to be extracted for each of the relevant VoIP streams. This will be done by analysing the packet information of the captured packets.

The information that needs to be extracted from the packets includes:

- Inter-packet arrival time
- Number of packets

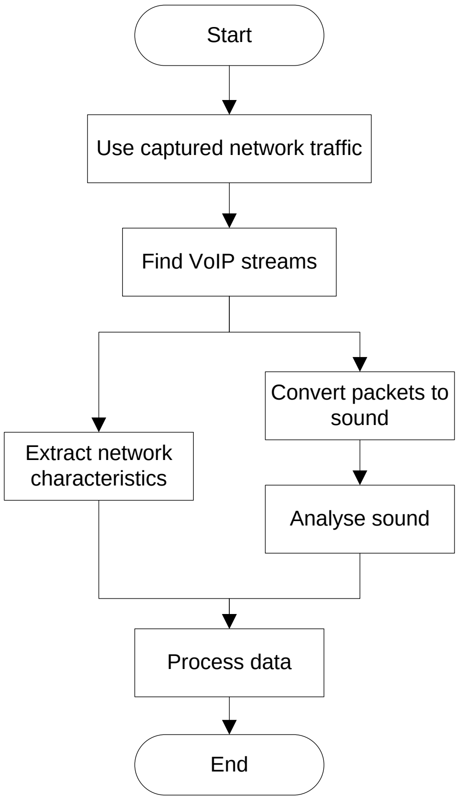


Figure 3.1: Block diagram of the captured network traffic analysis tool

- Packet loss
- Jitter

3.2.3 Packets to Sound Conversion

To enable voice quality estimation, the packets containing VoIP streams need to be converted to sound files. The actual speech data need to be extracted from the captured packets and converted into WAV format. It is also necessary to detect which CODEC is used to be able convert packets to sound for different CODECs. The WAV files must be in the correct format to be evaluated by the speech quality measurement models (Perceptual Evaluation of Speech Quality (PESQ) and Single Sided Speech Quality Measurement (3SQM)) discussed in Section 2.4.

3.2.4 Speech Analysis

PESQ and 3SQM will be used to analyse the extracted speech files. It is important to determine which of the intercepted streams are the reference and which the degraded signals. This is necessary for the PESQ to be effective, as was described in Section 2.4. An echo call is a call which repeats and sends back the same signal which is received. If a call is not a echo call, the PESQ model will be useless.

It is thought that the difference and correlation between the reference and degraded signal's 3SQM MOS score will also be an indicator of the degradation of speech quality. Therefore, signals will also be evaluated by the 3SQM speech quality measurement model.

3.2.5 Process Data

The network characteristics and the results of the speech analysis will then be logged in a table, called the *Data Table*. This table can then be used to determine the relationship between the network characteristics and speech quality. However, for long periods of network probing, averages of the variables in this table will be used.

3.3 The Network Analysis Tool

The process described in the previous section can be put to more effective use. In this section this captured network traffic analysis process will be used in the design of the semi-real-time network analysis tool. A block diagram of the design of this tool can be seen in Figure 3.2.

As seen in Figure 3.2, the first step of the application is to make a call.

This tool will make a call and the call will be echoed (enabling the use of the PESQ model) back from the destination, as shown in Figure 3.3. For the data to be statistically meaningful, a standardised call has to be made every time. As discussed, this call needs to be made to a server with echo call functionality.

The concept of this tool is to capture a set number of network traffic samples. The length of these samples is also to be set. While the tool is capturing network traffic, the previously captured traffic sample is analysed. Therefore the tool is only one sample length away from being real-time.

After the set number of samples has been analysed, the resulting table of the captured samples analysis will be processed. The data from this table (*Data Table*) will be used to obtain averages. These averages and the packet information is then added to a table, the *Statistics Table*. The Statistics Table allows us to predict the VoIP call quality from network variables. The tool is then terminated.

The tool will again be executed after a set length of time. It is necessary to stop the tool and restart it, as different calls may be routed differently. Thus, the more calls and samples available, the better the prediction.

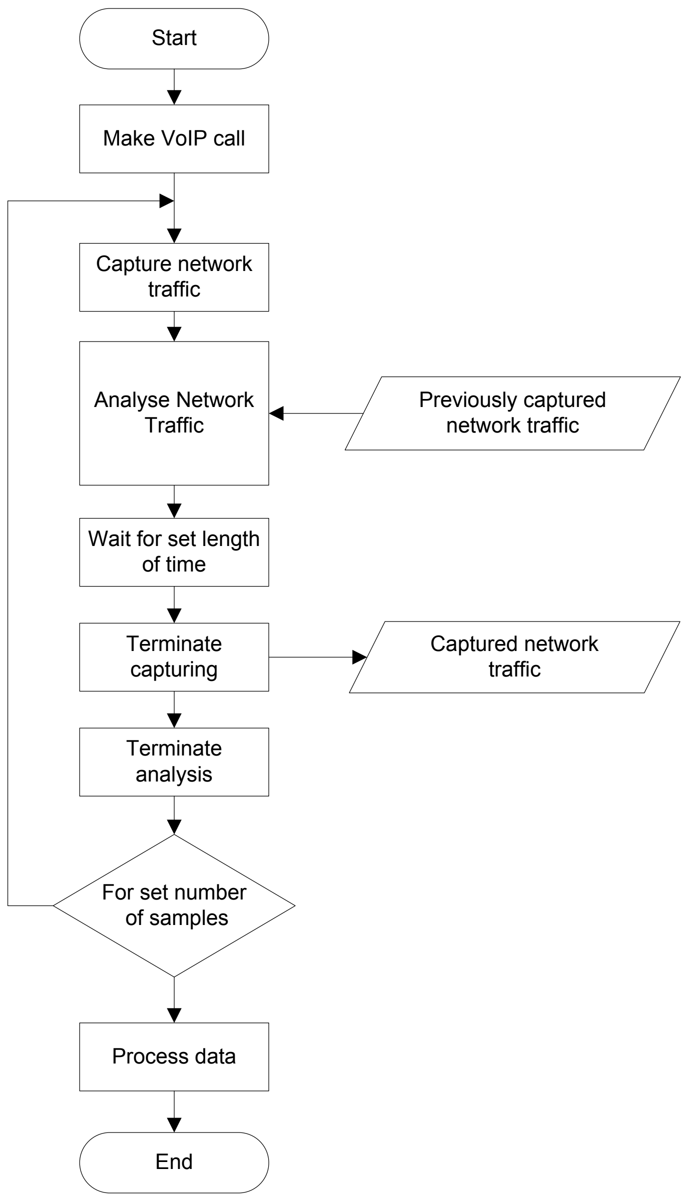


Figure 3.2: Block diagram of the semi-real-time network analysis tool

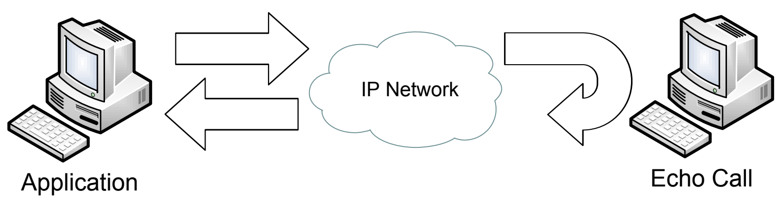


Figure 3.3: The basic concept of the network analysis tool

It is important to note that the degraded VoIP stream is degraded twice, as it effectively travels over the network twice, to the echo server and back.

3.4 The Perceived Speech Quality Prediction Tool

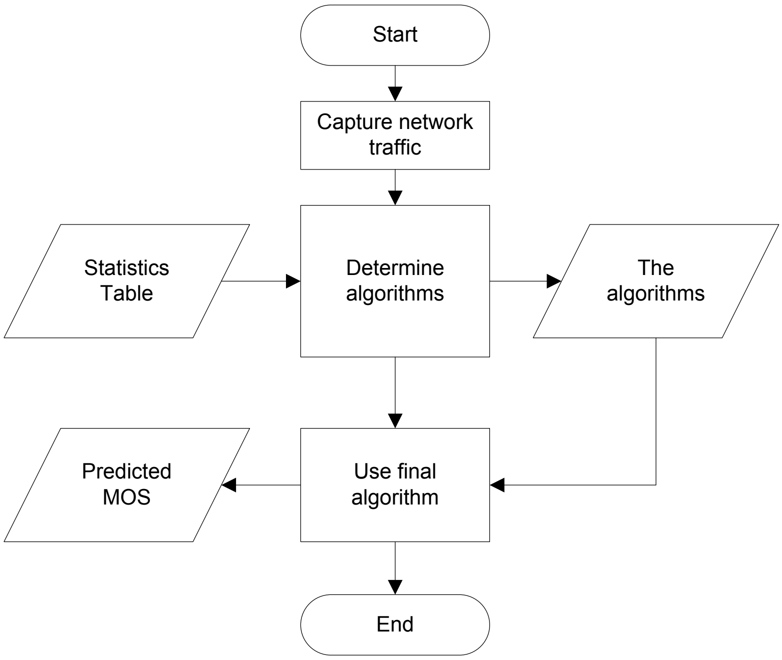


Figure 3.4: Block diagram of the perceived speech quality prediction tool

The design of the prediction tool is shown in Figure 3.4. This tool is designed to capture a few seconds of network traffic. Only the received stream needs to be isolated. The network characteristics are then extracted from the packet stream.

The statistics table is used to determine the algorithms. These algorithms model the relationship between the different network characteristics and perceived speech quality.

Correlation and regression are the most commonly used techniques for investigating the relationship between two quantitative variables.

Correlation The most important part of this project is describing the relationship between network character variables and speech quality. One of the most fundamental concepts in research is the concept of correlation. If two variables are correlated, this means that you can use information about one variable to predict the values of the other variable.

The correlation measures the strength of the linear relationship [44].

Regression The goal of a correlation analysis is to see whether two measurement variables co vary, and to quantify the strength of the linear relationship between the variables. Regression expresses the relationship in the form of an equation. In statistics, regression is used [45]:

- To derive a formula to predict a value for a variable given values of other variables
- To test if and how a variable is related to other variables.

The analysis consists of choosing and fitting an appropriate model, done by the method of least squares, with a view to exploiting the relationship between the variables to help estimate the expected response for a given value of the independent variable. The least-squares method makes the sum of the squares of the vertical distances of the data points from the line as small as possible.

Using regression analysis, a final algorithm can then be derived and speech quality (in MOS) can be predicted.

3.5 Summary

In this chapter the theoretical design methods of using network traffic to determine speech quality was discussed by using block diagrams. After the concept of the project was put forward in Section 3.1, the basic concept for analysing captured network traffic was discussed in Section 3.2. To use this process of analysing network traffic more effectively, a network analysis tool was designed as described in Section 3.3. This tool will generate statistics, which can be used to generate algorithms to model the relationship between the extracted network characteristics and speech quality. The derived algorithms can then be used to predict speech quality, as explained in Section 3.4.

The next chapter implements and refines the designs and presents detailed breakdowns of the components highlighted in this chapter.

Chapter 4

Implementation

In this chapter, the implementation of the Voice over Internet Protocol (VoIP) quality estimation tool design stages is discussed. Problems encountered during implementation, as well as the appropriate solutions, are discussed.

4.1 Applications Used

The various applications used to achieve different parts of this implementation are discussed in this section.

4.1.1 VoIP Call

For statistically meaningful results the same call needs to be made at the beginning of each test.

PJSUA is a command line Session Initiation Protocol (SIP), User Agent (UA) or softphone written using PJSIP, an open source SIP stack. PJSUA is used to simulate a legitimate VoIP client [46].

For the requirements of this project, PJSUA is used to initiate a call to an echo test number at the beginning of the network analysis tool process. See Figure 3.2.

This is done by the following command:

```
sudo pjsua --config-file=config.conf --play-file out.wav sip:4444@sip2sip.info
```

Where **out.wav** is a sound file containing speech, **sip:4444@sip2sip.info** is an echo test extension and config.conf contains:

```
--null-audio
--local-port=5067
--id sip:2233397413@sip2sip.info
--realm=sip2sip.info
--registrar sip:sip2sip.info
--username 2233397413
--password 123456
```

```
--contact=sip:2233397413@sip2sip.info
--proxy=sip:proxy.sipthor.net
--add-codec=gsm
--dis-codec=pcma
--dis-codec=pcmu
--auto-play
--duration=5000
```

The options specified in the above example are as follows:

- Specifying SIP account settings:
  - **–id=URL** - Set URL of the account. For example "sip:1234@dsplab.sun.ac.za"
  - **–registrar=URL** - Set the URL of the registrar server. Needs to be specified or account will not register. An example of URL: "sip:dsplab.sun.ac.za"
  - **–proxy=URL** - Optionally set the URL of proxies to build initial route set for all requests using this account.Example proxy URL "sip:proxy.dsplab.sun.ac.za;lr"
  - **–reg-timeout=SECONDS** - Set optional time-out for SIP account registration.
  - **–realm=string** - Set authentication realm. The realm is used to match this credential against challenges issued by downstream servers.
  - **–username=string** - Set authentication user ID.
  - **–password=string** - Set authentication password (clear text).

  Transport options:
  - **–local-port=PORT** - Set local port for SIP transport. Default is 5060 for UDP and TCP, and 5061 for TLS.
  - **–ip-addr=IP** - Use the specified address as SIP and RTP addresses.

  Media options:
  - **–add-codec=NAME** - Set codec NAME to have higher priority to use. Codec names include: pcma, pcmu, speex/8000, speex/16000, speex/32000, ilbc, gsm, l16/44100/2.
  - **–dis-codec=NAME** - Disable codecs with matching NAME.
  - **–null-audio** - Disable the sound device.
  - **–dis-codec=NAME** - Disable codecs with matching NAME.
  - **–play-file=WAVFILE** - Add file port to play WAV file to the conference bridge.
  - **–auto-play** - Stream the WAV file to incoming calls.

- **–auto-loop** - Automatically loop-back call to itself.

Media transport options:

- **–rtp-port=N** - Set the start RTP port. Default is 4000.

User agent options:

- **–duration=SEC** - Set maximum call duration to SEC seconds.

PJSUA can read these commands from a configuration file if **–config-file=FILE** is specified.

4.1.2 The VoIP Server

As seen in Figure 3.3, there is a need for a echo call. This is necessary to determine the degradation of the reference signal.

An Asterisk server is used for the echo call. Asterisk is software that turns an ordinary computer into a voice communications server. Asterisk is the world's most powerful and popular telephony development tool-kit [47]. It is used by small and large businesses, call centres, carriers and governments worldwide. Asterisk is open source and is available free to all under the terms of the General Public License (GPL) [48].

The Asterisk server is very simple to set up and use. Asterisk is controlled by editing a set of configuration files. The configuration file for Asterisk SIP channels, for both inbound and outbound calls is *sip.conf*.

Two channels (users) can be created by editing the *sip.conf* configuration file. This

```
[softPhone1]
callerid=2001
canreinvite=no
type=friend
context=test
host=dynamic

[softPhone2]
callerid=2002
canreinvite=no
type=friend
context=test
host=dynamic
```

In this example two users are created, **softPhone1** and **softPhone2**. The callerid is the number which is dialled for this specific channel.

The configuration file *extensions.conf*, contains the dialplan and controls the operational flow of Asterisk. For this project the dialplan is as follow.

```
[test]
exten => 1111,1,Answer()
exten => 1111,2,Echo()
exten => 1111,3,Hangup()
exten => 2001,1,Macro(dialSIP,softPhone1)
exten => 2002,1,Macro(dialSIP,softPhone2)

[macro-dialSIP]
exten => s,1,Dial(SIP/\${ARG1},20)
exten => s,2,Goto(s-\${DIALSTATUS},1)
exten => s-CANCEL,1,Hangup
exten => s-NOANSWER,1,Voicemail(u\${MACRO_EXTEN})
exten => s-BUSY,1,Voicemail(u\${MACRO_EXTEN})
exten => s-CONGESTION,1,Congestion(30)
exten => s-CHANUNAVAIL,1,Voicemail(u\${MACRO_EXTEN})
exten => s-CANCEL,1,Hangup
exten => s,n,Hangup
```

The Echo() application is used to simply echo whatever the caller says. Therefore when "1111" is dialled, in this example, your voice will be echoed. This is useful when testing a VoIP server. It is used in this application to test the degradation of speech for a certain network.

extensions.conf is further set up to give instructions on what to do when "2001" or "2002" is dialled. For example, when "2001" or "2002" is dialled the dialSIP macro is used. This macro is used to lead the call in the same way for different numbers. This includes how long the phone should ring before it goes to voice mail, and what to do if the user is busy, not available or if the server is congested.

One of the few problems found when setting up the Asterisk server, was that the IP to be used when a PC connects through a local network to the internet needs to be specified. This can be done by setting *host* to the correct IP. The same can be done for the echo call by adding the following to the *extensions.conf* file.

```
[general]
localnet=192.168.0.0/255.255.0.0 ; the local network
externip=x.x.x.x                ; the public address to be used
```

4.1.3 Capture Network Traffic

Capturing of network traffic is done by the one of the most popular of networking debugging tools, TCPDump. It is used to intercept and display packets which are transmitted and received through the network connection. Raw packets are usually saved in a ".pcap" files. In the field of computer network administration, pcap (**p**acket **cap**ture) consists of an Application Programming Interface (API) for capturing network traffic. The syntax to use TCPDump is as follows [49]:

```
tcpdump [options] [filter expression]
```

Commonly used tcpdump options include:

- **-n** - Saves time by not converting host addresses to names. Avoid DNS lookups.
- **-w <filename>** - Write the raw packets to the specified file instead of printing them out.
- **-r <filename>** - Read packets from the specified file. The file should ideally be created with the -w option.
- **-q** - Quiet output. Prints less information per output line
- **-s <number of bytes>** - tcpdump usually does not analyse and store the entire packet. This option allows the number of bytes specified of each packet's payload to be stored and analysed.
- **-A** - Print each packet in ASCII.

We are often not interested in all packets flowing through the network. Filters are used to restrict analysis to packets of interest. The following are commonly used filters:

- Specifying the hosts we are interested in
  - "dst host <name/IP>" - Destination name or IP
  - "src host <name/IP>" - Source name or IP
  - "host <name/IP>" - Either source or destination name or IP
- Specifying the ports we are interested in
  - "dst port <number>" - Destination port
  - "src port <number>" - Source port
  - "port <number>" - Either source or destination port
- Specifying different types of packets to capture
  - "icmp" - ICMP packets
  - "udp" - UDP packets
  - "tcp" - TCP packets
- Filters can be combined using:
  - "&&" - and
  - "||" - or

– "!" - not

For example all the tcp packets which are not from or to host dsplab.sun.ac.za:

```
tcpdump "tcp and ! host dsplab.sun.ac.za"
```

tcpdump is thus perfect to use for capturing VoIP packets.

The capturing of the UDP packets is done by using the following command:

```
sudo tcpdump -w filename.pcap -n udp -s 300
```

It is important to specify option -s <number> with a number larger than 214, which is the number of bytes in an RTP payload. The tcpdump default packet capture size is 50 [49].

4.1.4 Using Captured data

After the network traffic has been captured using tcpdump it needs to be decoded using a packet analyser. One of the most popular packet analysers is Wireshark. Wireshark is an open source packet analyser used for network troubleshooting, analysis and software and communications protocol development. It decodes over 750 protocols.

Wireshark is very similar to tcpdump, but it has a graphical front-end, and many more information sorting and filtering options. It allows the user to see all traffic being passed over the network.

Captured network data can be browsed via a Graphical User Interface (GUI), or via the terminal (command line) version of the utility, tshark [50].

In this project, tshark is used to decode the captured network traffic.

Tshark is used as follows:

```
tshark [options]
```

Tshark's options include:

- **-r <filename>** - Set the filename to read from
- **-T pdml|ps|psml|text** - Output format
- **-d <layer type>==<selector>,<decode as protocol>** - Decode as. For example "tcp.port==8888,http"

4.1.5 Decoding Sound

It is necessary to use a sound editor to convert different CODEC']s to WAV.

Sound **E**xchange, abbreviated SoX, is a free digital audio editor with a command line interface.

SoX features include [51]:

- Cross-platform (Windows, Linux, Mac OS X)
- Reading and writing AU, WAV, AIFF, MP3 (via an external LAME MP3 encoder), Ogg Vorbis, FLAC and other audio file formats
- Recording and playing audio (on many systems).
- Processing via echo, phaser, compressor, delay, filtering, etc
- Editing via concatenate, trim, pad, repeat, reverse, volume, fade, splice, normalise
- Speed (pitch and tempo), pitch (without tempo), tempo (without pitch), and sample-rate adjustment
- Noise removal
- Silent passage removal
- Statistical analysis; spectrogram analysis

4.2 Analysing the Captured Network Traffic

As discussed in the previous chapter and shown in Figure 3.1, we need to analyse the captured network traffic to extract the network characteristics and speech. The different applications discussed in the first section of this chapter will be used to achieve this.

4.2.1 Find VoIP streams

Before the packets can be analysed it is necessary to determine what packets are used to carry VoIP data.

4.2.1.1 Ports

The first step when converting captured network data, is finding the ports used for VoIP calls. RTP does not have an assigned standard UDP port, instead it is generally configured to use a even port (as UDP). Each stream has a different port.

The port is found by running the following tshark command:

```
tshark -r captured_Data.pcap -T fields -e udp.port 2>/dev/null | sort | uniq >
    logs/ports.txt
```

This command analyses the .pcap and writes all the UDP ports to a file, (ports.txt).

These ports are then tested to determine if they are used for VoIP call or not. This is done by determining whether it has a Sequence Number (seq) and CODEC. If the CODEC is unknown, the stream will not be analysed.

Another command which can be used is:

```
tshark -r captured_Data.pcap -n -T fields -e udp.port -tdd 2>/dev/null |sort|
    uniq -c | sort -k1,1rn -k2 | sed 's/ *[^ ]* *//;q'
```

This command returns the most recurring UDP port.

For each call two ports are used, one each for outgoing and incoming streams.

4.2.1.2 Detecting Streams

After the UDP ports used for VoIP have been found, VoIP streams can be detected. This is done by searching for Synchronization Source (SSRC)). A SSRC identifies the synchronisation source. For each VoIP call there are at least two streams and therefore at least two SSRCs.

The SSRC is unique and is chosen randomly, with the intent that no two pairs of synchronisation sources within the same RTP session will have the same SSRC.

To find all the SSRCs at port 8000 the following command is used:

```
tshark -r captured_Data.pcap "udp.port ==8000" -d "udp.port ==8000,rtp" -T
    fields -e "rtp.ssrc"
```

The SSRCs are then added to a list. These SSRCs (VoIP streams) are then analysed to determine network characteristics and perceived speech quality.

4.2.2 Packet Quality Characteristics Analysis

Certain statistics can be retrieved by analysing the information from captured packets of the different streams.

A VoIP stream can be analysed by using the following command:

```
tshark -r captured_Data.pcap -d "udp.port==4000,rtp" "rtp.ssrc==0x765d4611" -
    tdd
```

The packet information of the 0x765d4611 SSRC at port 4000 is displayed. An example of the packet information shown is:

```
  2   0.006883 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
      =0x765D4611, Seq=29529, Time=202720
  4   0.019214 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
      =0x765D4611, Seq=29530, Time=202880
  6   0.026332 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
      =0x765D4611, Seq=29531, Time=203040
  8   0.013121 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
      =0x765D4611, Seq=29532, Time=203200
```

```
10   0.021997 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
    =0x765D4611, Seq=29533, Time=203360
12   0.033577 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
    =0x765D4611, Seq=29534, Time=203520
14   0.016594 146.232.220.183 -> 146.232.222.59 RTP PT=ITU-T G.711 PCMU, SSRC
    =0x765D4611, Seq=29535, Time=203680
```

This information shows:

- Packet number
- Inter-packet delay (Time delta)
- Source IP
- Destination IP
- Payload Type (PT)
- Synchronisation Source (SSRC)
- Sequence number (Seq)
- Packet time stamp (Time)

General statistics, such as inter-packet delay, jitter and packet loss can be calculated from this information.

4.2.2.1 Average Delay

The average inter-packet delay of the packets in the captured network traffic can be calculated by accumulating the inter-packet delays of the packets and dividing it by the total amount of packets.

4.2.2.2 Loss

The number of packets sent by the source can be calculated by subtracting the first sequence number from the last. The first sequence number is chosen randomly, and is between 1 and 65536. The sequence is reset to 1 if it goes over 65536.

Loss can be determined by comparing the total number of packets received and the total number of packets sent.

4.2.2.3 Jitter

This implementation calculates jitter according to IETF RFC3550 [1].

Jitter can be calculated by using Equation 4.2.1

$$J_i = J_{i-1} + \frac{|D_{i-1,i}| - J_{i-1}}{16} \tag{4.2.1}$$

Where:
J_i - Jitter of packet nr i
J_{i-1} - Jitter of packet nr $i-1$
$D_{i-1,i}$ is the difference and is calculated by using Equation 4.2.2

$$D_{i-1,i} = (R_i - R_{i-1}) - (S_i - S_{i-1}) \tag{4.2.2}$$

Where:
R_i - Time of arrival of packet i
S_i - RTP timestamp of packet i
This information is available, therefore jitter can be calculated.

4.2.3 Converting captured network traffic to speech

The VoIP SSRCs are then used used to convert the specific streams to audio. The raw hexidecimal payload of each captured packet in the .pcap file is written to a .raw file using the following command:

```
tshark -r captured_Data.pcap -d "udp.port==8000,rtp" "rtp.ssrc==0x21351483" -T
    pdml 2>/dev/null | grep payload | cut -f10 -d'"' >> logs/0x21351483.raw
```

The hexidecimal raw file is then converted to a raw binary file.

4.2.3.1 Converting Different CODECs

As described in Table 2.1, the payload type field of the RTP header specifies the CODEC. This can be used to decode the VoIP stream.

Three CODECs are supported in the implementation:

- G.711 PCMU
- G.711 PCMA
- GSM

The raw binary file is then converted to WAV using SoX.

For converting the G.711 PCMU CODEC to WAV the following command is used:

```
sox -c 1 -r 8000 -U -t raw logs/0x21351483.raw -e signed-integer out.wav
```

This command outputs a mono uncompressed 16-bit PCM audio file with a sample rate of 8000 Hz.

The same can be done for the PCMA CODEC:

```
sox -c 1 -r 8000 -A -t raw logs/0x21351483.raw -e signed-integer out.wav
```

And the GSM 06.10 CODEC:

```
sox -t gsm logs/0x21351483.raw -e signed-integer out.wav
```

4.2.4 Analysing Speech

The audio files generated are in the correct format to be evaluated by the PESQ and 3SQM algorithms.

4.2.4.1 Single Sided Speech Quality Measurement

The ITU reference of the Single Sided Speech Quality Measurement (3SQM) implementation is used. This implementation can be found on ITU's website [14].

The usage of this implmentation is:

```
./p563 <SpeechFile>
```

Where:

- **<SpeechFile>** - The wav file to be analysed

An example of 3SQM being used in this implementation is:

```
./p563 logs/capture2.pcap.wav
```

Where the resulting output is:

```
Filename  MOS
logs/capture2.pcap.wav  3.542878
```

4.2.4.2 Perceptual Evaluation of Speech Quality

As with 3SQM, the PESQ algorithm is not implemented in this project; instead the ITU reference implementation is used. This can be found on ITU's website [13].

The

```
./pesq [options] ref deg
```

where:

- **[options]** - Specify the sampling rate of the input files, which can either set to "+8000" (8 kHz) or +16000 (16 kHz)

- **ref** - reference signal filename
- **deg** - degraded signal filename

An example of pesq being used in this implementation is:

```
./pesq +8000 capture1.wav capture1.wav
```

where the resulting output is:

```
 Level normalization...
 IRS filtering...
 Variable delay compensation...
 Acoustic model processing...

Prediction : PESQ_MOS = 4.383
```

4.3 The Network Analysis Tool

As discussed in Section 3.3 and seen in Figure 3.2, an application is needed to control the components described in the previous section. This implementation needs to be as close to real-time as possible. This can be accomplished by capturing network traffic while analysing the previous sample of captured data.

The simultaneous execution of two processes is done by using the **subprocess** [52] and **processing** [53] packages in Python. For example, if 10 network traffic samples of 10 seconds each are to be captured and analysed, the following code is to be executed:

```
Count = 1
#the number of samples that should be analysed
for (Count <= 10):
  #capture network traffic
  capt = Process(target=QualityClass().capture, args=(str(Count),))
  capt.start()

  #if there is previous captured network traffic
  if (Count > 1):
    #analyse previously captured network traffic
    evalu = Process(target=QualityClass().evaluate, args=(str(Count),))
    evalu.start()

  #wait for 10 seconds
  time.sleep(10.0)

  #terminate the capturing of network traffic
  capt.terminate()
  #if there is previous captured network traffic
  if (Count > 1):
    #terminate the analysing of network traffic
    evalu.terminate()
  Count = Count + 1
```

where the **QualityClass()** is the following:

```
class QualityClass():
  def capture(self, Num):
    #determine local IP address
    IP=os.popen("ifconfig | grep Bcast").read()
    IP=IP[IP.find("addr")+5:IP.find("Bcast")-2]
    #capture the udp packets on 300
    subprocess("sudo tcpdump -w logs/capture"+Num+".pcap -n \"udp and host "+
        IP+"\" -s 300 2>/dev/null", shell=True)
    #capture 10 seconds
    time.sleep(10.0)

  def evaluate(self, Num):
    #evaluate the previous captured packets
    evalNum = int(Num) - 1
    #analyse the captured network traffic to determine the speech quality and
        network characteristics
    subprocess.call("python pcap2wav logs/capture"+str(evalNum)+".pcap", shell
        =True)
```

Please note the in code comments for an explanation of the code.

The tool is executed every few minutes to obtain statistically meaningful results. The results of the analysis are logged in a table.

4.3.1 Generated Tables

Two tables are generated by the network analysis tool, the Data table and the Statistics table. The Data table is used to log the results of the analysis of the captured network traffic samples (.pcap). After the required number of samples have been analysed, the data from Data table are processed and saved in the Statistics table. The Statistics table allows us to predict the VoIP call quality from network variables.

4.3.1.1 Data table

The *Data table*, as shown in Table 4.1, is used to log the results from the evaluation of the captured network traffic samples by the network analysis tool.

The information logged:

- Sample - The sample number of the captured data. It also shows whether the information is the reference or degraded signal.
- P563 - The 3SQM MOS score of a specific audio stream
- PESQ - The PESQ MOS score of the reference and the degraded signals
- SSRC - The SSRC of the stream

Table 4.1: Data table - Log of the results of the evaluation of 10 captured samples

Sample	P563	PESQ	SSRC	Source	Destination	Packets	Loss	Dmean	Jitter
1ref	4.894599	0.000	0x66a044ca	146.232.220.183	196.15.178.165	479	0	0.0200076	0.0002937
1deg	4.935620	3.958	0x48101910	196.15.178.165	146.232.220.183	477	2	0.0201258	0.0061023
2ref	4.881613	0.000	0x66a044ca	146.232.220.183	196.15.178.165	450	0	0.0210221	0.0007074
2deg	4.812512	4.499	0x48101910	196.15.178.165	146.232.220.183	454	0	0.0208702	0.0058097
3ref	4.430233	0.000	0x66a044ca	146.232.220.183	196.15.178.165	462	0	0.0204776	0.0005684
3deg	4.367947	3.455	0x48101910	196.15.178.165	146.232.220.183	462	0	0.0204499	0.0067022
4ref	2.674510	0.000	0x66a044ca	146.232.220.183	196.15.178.165	466	0	0.0199987	0.0005938
4deg	2.653378	3.841	0x48101910	196.15.178.165	146.232.220.183	459	0	0.0202555	0.0063171
5ref	3.231357	0.000	0x66a044ca	146.232.220.183	196.15.178.165	458	0	0.0207163	0.0005653
5deg	3.357212	3.746	0x48101910	196.15.178.165	146.232.220.183	459	0	0.0206842	0.0064456
6ref	4.560469	0.000	0x66a044ca	146.232.220.183	196.15.178.165	458	0	0.0207037	0.0005355
6deg	4.503937	3.707	0x48101910	196.15.178.165	146.232.220.183	455	1	0.0208217	0.0060572
7ref	5.000000	0.000	0x66a044ca	146.232.220.183	196.15.178.165	474	0	0.0200014	0.0005898
7deg	4.539965	4.348	0x48101910	196.15.178.165	146.232.220.183	474	0	0.0199896	0.0047047
8ref	2.793146	0.000	0x66a044ca	146.232.220.183	196.15.178.165	472	0	0.0200870	0.0005359
8deg	2.865420	3.536	0x48101910	196.15.178.165	146.232.220.183	472	0	0.0200586	0.0059247
9ref	4.979681	0.000	0x66a044ca	146.232.220.183	196.15.178.165	471	0	0.0201590	0.0007573
9deg	4.863674	4.164	0x48101910	196.15.178.165	146.232.220.183	471	0	0.0201635	0.0058330
10ref	4.233141	0.000	0x66a044ca	146.232.220.183	196.15.178.165	448	0	0.0211841	0.0006210
10deg	4.326812	3.422	0x48101910	196.15.178.165	146.232.220.183	443	2	0.0214126	0.0052229

- Source - The source IP of the stream
- Destination - The destination IP of the stream
- Packets - Number of packets in the stream.
- Loss - Number of packets lost
- Delay - The average delay of the packets in the stream
- Jitter - The average jitter of the packets in the stream

4.3.1.2 Statistics table

The data collected in in the Data table is then processed and saved in the Statistics table. This is necessary because the network analysis tool needs to be stopped and re-executed several times for meaningful results to be obtained.

The Statistics table contains averages of the different Data tables generated. Each Data table (or execution of the network analysis tool) will have two entries on it. One each for the incoming and outgoing streams. Table 4.2 shows an example of the Statistics table.

The information logged:

- Source - The source IP of the stream
- Samp - The number of samples
- Correl - Correlation between the two 3SQM
- avg_3sqm - Average 3SQM MOS score
- avg_pesq - Average PESQ MOS score
- pkts - The total number of packets in the stream
- loss - Number of packets lost
- %loss - Percentage lost packets
- avg_delay - The average delay of the packets in the stream
- avg_jitter - The average jitter of the packets in the stream
- Time - The date and time the entry was added to the Stats table

The statistics table can then be used to derive models for the prediction of perceived speech quality using captured network traffic. Thus, it is this table which will be used by the prediction tool.

Table 4.2: Statistics table - The averages of 15 different Data tables

Source	Smpls	Correl	avg_3sqm	avg_pesq	pkts	loss	percent	avg_delay	avg_jitter	Time
146.232.220.183	10	0.53133	4.0805440	0.0000	4629	0	0.0000	0.020508	0.0005331	2010-05-16 19:44:44 SAST
196.15.178.165	10	0.53133	4.0670941	3.5817	4606	38	0.8250	0.020626	0.0036995	2010-05-16 19:44:44 SAST
146.232.220.183	10	0.87138	4.2744615	0.0000	4639	0	0.0000	0.020430	0.0005241	2010-05-16 19:49:20 SAST
196.15.178.165	10	0.87138	4.0035904	3.8806	4625	4	0.0865	0.020486	0.0030196	2010-05-16 19:49:20 SAST
146.232.220.183	10	0.99676	4.1301808	0.0000	4640	0	0.0000	0.020421	0.0005311	2010-05-16 19:53:56 SAST
196.15.178.165	10	0.99676	4.1402714	3.8746	4647	4	0.0861	0.020388	0.0039085	2010-05-16 19:53:56 SAST
146.232.220.183	10	0.85774	4.2816845	0.0000	4642	0	0.0000	0.020423	0.0005644	2010-05-16 19:58:32 SAST
196.15.178.165	10	0.85774	3.9632444	3.8760	4630	5	0.1080	0.020473	0.0027701	2010-05-16 19:58:32 SAST
146.232.220.183	10	0.81533	4.2869604	0.0000	4638	0	0.0000	0.020433	0.0005710	2010-05-16 20:03:07 SAST
196.15.178.165	10	0.81533	4.1019010	3.8481	4620	2	0.0433	0.020502	0.0030161	2010-05-16 20:03:07 SAST
146.232.220.183	10	0.93384	4.1204616	0.0000	4635	0	0.0000	0.020432	0.0005433	2010-05-16 20:07:43 SAST
196.15.178.165	10	0.93384	3.9845813	3.8676	4627	7	0.1513	0.020469	0.0030899	2010-05-16 20:07:43 SAST
146.232.220.183	10	0.91909	4.1754014	0.0000	4629	0	0.0000	0.020460	0.0005394	2010-05-16 20:12:18 SAST
196.15.178.165	10	0.91909	3.9460606	3.9109	4617	7	0.1516	0.020502	0.0038959	2010-05-16 20:12:18 SAST
146.232.220.183	10	0.93646	4.1050681	0.0000	4636	0	0.0000	0.020426	0.0005256	2010-05-16 20:16:54 SAST
196.15.178.165	10	0.93646	3.9830101	3.9224	4634	4	0.0863	0.020441	0.0042545	2010-05-16 20:16:54 SAST
146.232.220.183	10	0.72341	4.0287875	0.0000	4640	0	0.0000	0.020426	0.0005592	2010-05-16 20:21:30 SAST
196.15.178.165	10	0.72341	3.7559563	3.7561	4656	4	0.0859	0.020327	0.0057743	2010-05-16 20:21:30 SAST
146.232.220.183	10	0.86292	4.0641875	0.0000	4643	0	0.0000	0.020422	0.0005344	2010-05-16 20:26:05 SAST
196.15.178.165	10	0.86292	3.9663799	3.8447	4635	2	0.0431	0.020438	0.0056485	2010-05-16 20:26:05 SAST
146.232.220.183	10	0.59571	3.9930989	0.0000	4637	0	0.0000	0.020399	0.0005473	2010-05-16 20:30:41 SAST
196.15.178.165	10	0.59571	4.2087403	3.7613	4605	27	0.5863	0.020547	0.0064634	2010-05-16 20:30:41 SAST
146.232.220.183	10	0.87616	4.1342869	0.0000	4639	0	0.0000	0.020436	0.0007624	2010-05-16 20:35:16 SAST
196.15.178.165	10	0.87616	3.8981020	3.7173	4625	8	0.1730	0.020500	0.0069245	2010-05-16 20:35:16 SAST
146.232.220.183	10	0.70172	4.0085422	0.0000	4919	0	0.0000	0.020402	0.0007796	2010-05-16 20:39:52 SAST
196.15.178.165	10	0.70172	4.1796290	3.8948	5084	2	0.0393	0.020455	0.0069544	2010-05-16 20:39:52 SAST
146.232.220.183	10	0.33334	4.2882657	0.0000	4615	0	0.0000	0.020533	0.0007333	2010-05-16 20:44:28 SAST
196.15.178.165	10	0.33334	4.0082030	3.6979	4545	19	0.4180	0.020860	0.0088466	2010-05-16 20:44:28 SAST
146.232.220.183	10	0.71172	4.1095360	0.0000	4633	0	0.0000	0.020431	0.0009314	2010-05-16 20:49:04 SAST
196.15.178.165	10	0.71172	3.8453125	3.6797	4613	27	0.5853	0.020511	0.0081361	2010-05-16 20:49:04 SAST

4.4 Prediction of perceived speech quality

As was discussed in Section 3.4 there are two commonly used techniques for investigating the relationship between two quantitative variables, correlation and regression.

4.4.1 Correlation

Correlation is used twice in this project.

- The relationship between the 3SQM MOS scores of the reference signal and the degraded signal is investigated.
- The relationship between packet loss and delay is investigated.

In this project the Pearson product-Moment Correlation Coefficient (PMCC) (typically denoted by r) will be used as a measure of how two variables are correlated. This is widely used in the sciences as a measure of the strength of linear dependence between two variables [54].

In Python this can be done by using the **lpearsonr()** function [55]:

```
correlation = stats.lpearsonr(x,y)[0]
```

where x and y are lists of equal length.

4.4.2 Regression

Regression is used to derive least squares polynomial fit equations which model the relationship between speech quality and each of the different network characteristics.

Two different tools are used to get the least squares fit equations.

Firstly R (programming language and software environment for statistical computing) was used to fit the data to a straight line. This is done by using the *lsfit()* function, from the *stats* package.

In this project it is used as follows in Python [56]:

```
    ls_fit = r.lsfit(delay,pesq)
    m = ls_fit['coefficients']['X']
    c = ls_fit['coefficients']['Intercept']
```

where : $y = mx + c$

However polynomials, are needed to model the relationships. The *polyfit* function, which is part of the *numpy* package for Python [57], is used to achieve this.

For this project it is used as follows:

```
w = polyfit(x, y, deg)
```

Where the parameters are:

- x - An array of independent sample variables
- y - An array of dependent sample variables
- deg - An integer which specifies the degree of the fitting polynomial
- w - An Array of size deg + 1 containing polynomial coefficients, highest power first. For example $w[0]x^2 + w[1]x + w[2]$.

Second order polynomials are the highest order polynomials which are used in this project. The reason for this will be made clear in the next chapter.

The standard deviation is used to to determine the efficiency of the prediction model. This means if variance of the errors is small the prediction model is efficient.

When fitting (with the least squares method) data that contains random variations, there are two important assumptions that are usually made about the error [58]:

- The error exists only in the response (dependent/model) data , and not in the predictor(independent/measured) data.
- The errors are random and follow a normal (Gaussian) distribution with zero mean and constant variance.

This means we can use the 68-95-99.7 rule [59]:

- About 68% of the values lie within one standard deviation of the mean.
- About 95% of the values lie within two standard deviations of the mean.
- Nearly all (99.7%) of the values lie within three standard deviations of the mean.

In this project the standard deviation is determined by:

```
def stddev(Errors):
  numErrors = numarray.array(Errors)
  n, = numError.shape
  sum = numarray.sum(numErrors)
  sum_of_squares = numarray.sum(numErrors * numErrors)
  return sqrt(sum_of_squares / n - (sum / n) ** 2
```

This function works by:

1. Converting the array to a numarray
2. Getting the length of the array

3. Adding the numbers in the array

4. Computing the sum of squares of the squares of the errors.

5. Returning the standard deviation.

The standard deviation in the last step is calculated as shown in Equation 4.4.1 [60].

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{4.4.1}$$

4.4.3 Using Time

At different times of the day, the usage of the internet and networks are different. Therefore the call quality will be different on different days and at different times. It is well known that network traffic exhibits non-stationary behaviour [61]. This will be seen in Subsection 5.1.4. It is, however, assumed that a repeatable network usage pattern exists. For example, it is assumed that more network bandwidth will be utilised at 16:00 than at 4:00. This assumption will confirmed in Subsection 5.1.4.

This project looks only at the effect the time of day has on network impairments and speech quality. Unfortunately, a lot of data capturing would be needed for this project to include the effect of different days, weeks and months.

Predicting MOS score using time of day is achieved by taking the average over time or time bands. These timebands, which can be set, will then model the MOS score for a 24 hour period.

4.4.4 The Prediction Tool

This tool is designed to capture a few seconds of network traffic. The network characteristics are extracted from the incoming packet stream. Regression can then be used to determine the algorithms which model the relationships between the different network characteristics and perceived speech quality.

For example, for the packet loss second order polynomial prediction model.

```
w = polyfit(LossArray,pesqArray,2)
lossa = w[0]
lossb = w[1]
lossc = w[2]

lossturn = -(lossb)/(2*lossa)

if (LossPercent > lossturn):
```

```
  LossPercent = lossturn

Predicted_loss_MOS_Score = lossa*(LossPercent)**2 + lossb*(LossPercent) +
    lossc
```

where the variables are:

- LossArray - Array of the percentage loss measurements from the statistics table.
- pesqArray - Array of the PESQ MOS measurements from the statistics table
- lossa - Float which is the quadratic coefficient of the quadratic Equation 4.4.2
- lossb - Float which is the linear coefficient in the quadratic Equation 4.4.2
- lossc - Float which is the constant term of the quadratic Equation 4.4.2
- lossturn - The turning point of quadratic Equation 4.4.2
- LossPercent - The measured packet loss, from which the MOS score must be predicted
- Predicted_loss_MOS_Score - The predicted MOS score.

$$f(x) = ax^2 + bx + c \tag{4.4.2}$$

The same is done for each of the other characteristics and for time.

The second order polynomial has a turning point and, as it is not expected that speech quality will improve when the network impairments increase, the model is stopped at this point. As seen with the *lossturn* variable, if a network impairment is larger at the turning point, it is modelled as if it were at the turning point.

In the next chapter it will be shown which of the network characteristics are statistically meaningful for the prediction of speech quality.

4.5 Summary

The implementation of the network analysis tool and the speech quality prediction tool has been discussed in this chapter.

In Section 4.1 the major third party software components were discussed. These applications are used to achieve what was designed in the previous chapter.

The implementation of the analysis of network traffic is discussed in Section 4.2. The first step is to isolate the different VoIP streams. The network parameters, such as packet loss, inter-packet delay and jitter can then be calculated. The speech component of each of these streams is also extracted. These extracted speech files are then analysed using the QoS measurement models discussed in Section 2.4.

Next, in Section 4.3, this analysis of network traffic is implemented into a semi-real-time network analysis tool. This tool captures network traffic while analysing previously captured data, as described in Section 4.2. It also generates statistics in tabular form.

These statistics are then used to derive algorithms (Section 4.4), which model the relationships between perceived speech quality and network parameters. The prediction tool uses these algorithms to predict perceived speech quality using only quickly obtainable network parameters.

The different tools implemented, are evaluated in the following chapter.

Chapter 5

Evaluation and Tests

In this chapter the designs and implementations derived in the previous chapters are evaluated against the thesis objectives, which were presented in Chapter 1.

There are two main sections in this chapter; Section 5.1 and Section 5.2. In Section 5.1, the impact of a network's impairments (discussed in Section 2.4), on speech quality are studied and results are shown. In Section 5.2, the quality prediction tool designed and implemented, is discussed.

Three different environments are used to evaluate the derived implementations: Simulation, laboratory and real world environments.

5.1 Evaluation of Network Analysis Tool

In this section the Network Analysis Tool is evaluated. The impact that network impairments have on perceived speech quality is shown in the results.

5.1.1 Simulations

The impact that packet loss and delay have on the quality of Voice over Internet Protocol (VoIP) calls is simulated. These simulated VoIP streams are then analysed by the speech evaluation implementation used in the network analysis tool. The simulations of packet loss and delay are done using *wav2rtp* [62]. Wav2rtp is a tool which generates Real Time Protocol (RTP) data packets from sound files. It converts a wav file to **pcap!** (**pcap!**) format. Once in **pcap!** format, wav2rtp manipulates the **pcap!** structure to simulate loss and delay. The pcap file can then be analysed by the designed network analysis tool. The resulting wav file can then be extracted from the generated **pcap!** file.

Wav2rtp models the User Agent (UA) and channel and does not simulate a real network. It is only used to show the effect that packet loss and delay have on perceived speech quality using the objective speech quality measurement models. All simulations were done for the G.711 codec.

5.1.1.1 The Impact of Loss on Speech Quality

The impact of independent random loss on the overall perceptual speech quality is investigated in the following simulation. A speech file is used by wav2rtp to generate RTP packets and then random loss is simulated by uniformly excluding random packets. This does not simulate a real channel, but gives us an idea of the impact of packet loss on speech quality. The resulting Packet Capture (PCAP) files are then analysed using the implemented network traffic analysis tool. The percentage of random loss can be set in wav2rtp. The impact of packet loss can be seen by looking at the speech waveforms in Figures 5.1a to 5.1d.

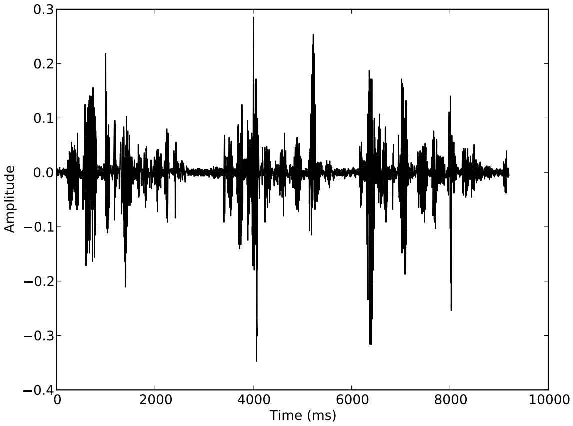
In Figure 5.1a the waveform of the reference speech signal is shown. The reference signal is a 9 second speech clip with the words "Will regard such claims with sympathy. The naming of the post war MTBs". Figures 5.1b,5.1c and 5.1d show the speech waveform with 0%, 25% and 50% random packet loss respectively.

These degraded waveforms can then be evaluated with the Perceptual Evaluation of Speech Quality (PESQ) and Single Sided Speech Quality Measurement (3SQM) models. The simulation results of the impact that random packet loss has on speech quality can therefore be seen. The results for PESQ and 3SQM are shown in Figures 5.2a and 5.2b respectively. These results are obtained by using wav2rtp to generate random independent loss (at different rates) to the a single input sound file. An output pcap file is generated for each of these different loss rates. PESQ and 3SQM can then be used to determine the perceived speech quality. In this simulation the speech file was degraded 100 times in steps of 2.5% between 0% and 70%. Every degraded sample was then analysed with the PESQ and 3SQM models. The standard deviation of the 100 samples for each of the steps is represented in each datapoint's errorbar.

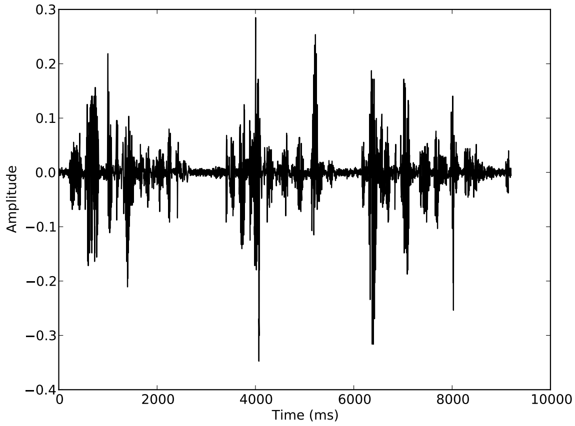
In Figure 5.2 the relationship between packet loss and speech quality seems to fit to a quadratic (second order) function. Second order polynomials will, therefore, be used to model this relationship. PESQ (Figure 5.2a) also seems to measure the quality of speech more effectively than 3SQM. Figure 5.2a shows a lower mean standard deviation (0.2243) than 3SQM case in Figure 5.2a (0.3780). This will further be discussed in Subsection 5.1.3.

5.1.1.2 The Impact of Delay on Speech Quality

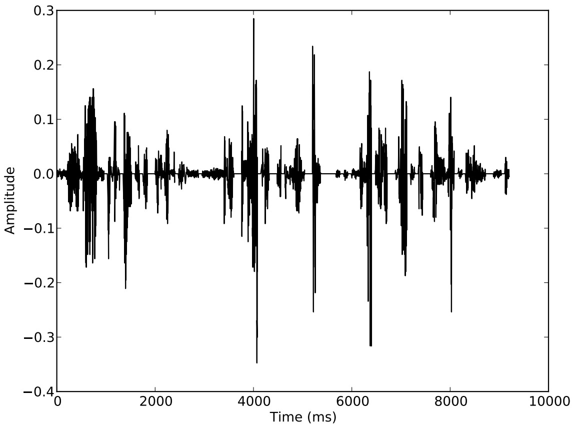
In this simulation, the impact of delay on the overall perceptual speech quality is investigated. The same speech file that was used in the simulation of the impact



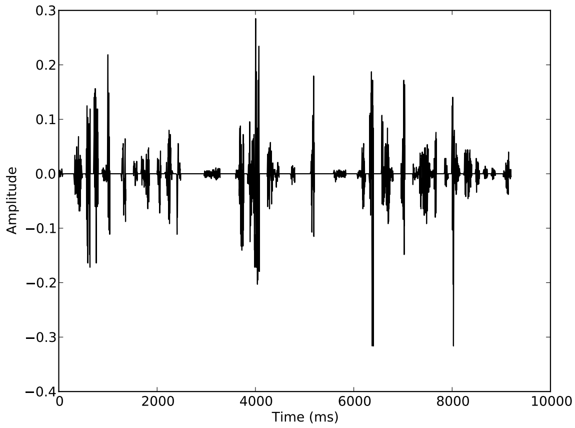
(a) Reference Signal



(b) Degraded Signal - 0% loss



(c) Degraded Signal -25% loss

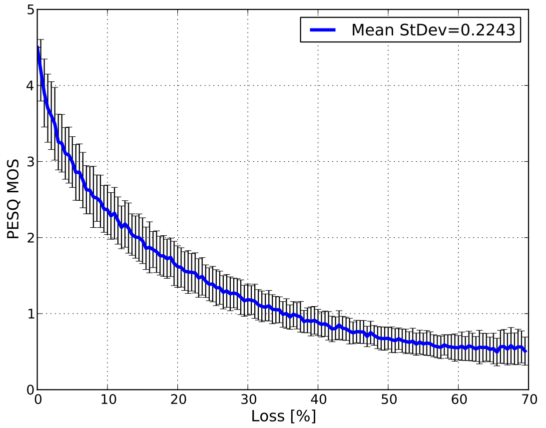


(d) Degraded Signal -50% loss

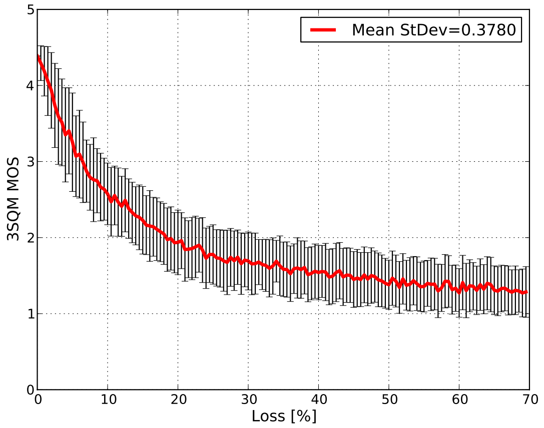
Figure 5.1: Speech signal waveforms showing the effect of simulated random packet loss

of loss on speech quality is used in the simulation. The minimum and maximum delay can be set in wav2rtp. We investigate the impact of delay on speech quality by increasing the delay from 0 to 4000 milliseconds in 100 milliseconds step. Each data point (or step) represents the average MOS of 100 generated speech samples. The error bar for each data point represents the standard deviation of that 100 MOS measurements. When the minimum and maximum delay are set to the same value, as seen in Figure 5.3, there is no degradation of speech quality. This is because when delay is constant, speech takes longer to arrive at the destination but packets are not dropped and, therefore, do arrive and in the correct order.

However, when the delay is set varied, the packets do not arrive in time and

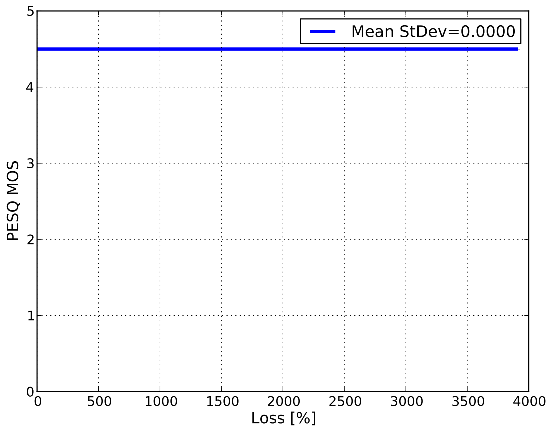


(a) PESQ

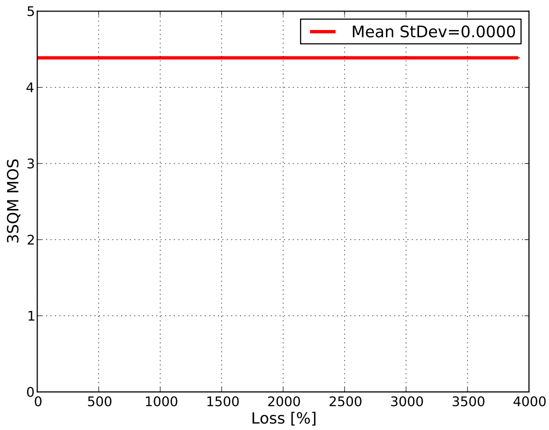


(b) 3SQM

Figure 5.2: Simulation results showing the impact of packet loss on speech quality using two objective speech quality models



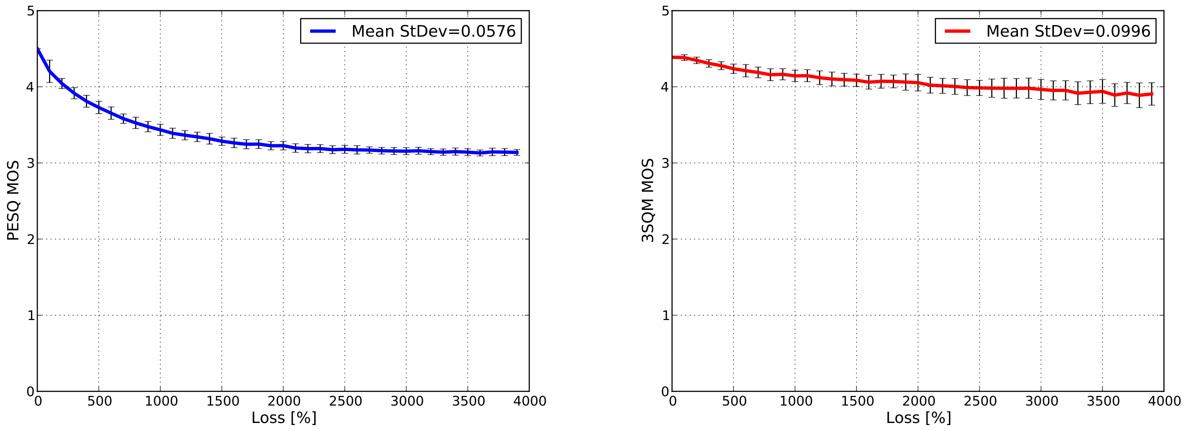
(a) PESQ



(b) 3SQM

Figure 5.3: Simulation results showing the impact of delay on speech quality using two objective speech quality models

are dropped. Therefore the speech quality is degraded, as seen in Figure 5.4. Thus, it is shown that it is variation in packet delay which affects the measured quality, not the delay itself. Therefore, for this project, inter-packet delay will be used to measure delay. It is also of value to note that the variation of delay does not have the same effect on speech quality that packet loss has. As with packet loss, PESQ (Figure 5.4a) seems to model the relationship more effectively than 3SQM (Figure 5.4b) as the mean standard deviation is higher for the 3SQM case. This will be further discussed in Subsection 5.1.3.



(a) PESQ

(b) 3SQM

Figure 5.4: Simulation results showing the impact of variable delay on speech quality using two objective speech quality models

These simulation results can be compared to the results from the laboratory and to real world tests.

5.1.2 The Test Set-up

This was discussed in general in the previous chapter, but will now be recapitulated for specific results.

After an echo test call has been initiated, the network traffic is captured and analysed. The size of the data sample which is captured can be set. For these tests the sample size is 10 seconds of captured data and number of samples captured is 10 samples (unless otherwise stated). Therefore, a 100 second phone call is captured in these test cases. Data from each of these samples is added to the *Data table*. An example of the Data table is shown in Table 4.1. Each Data table has two entries, one for the reference stream and one for the degraded stream

In Figures 5.5 - 5.7 some of the data from the Data table is shown in graphical form.

It can already be seen from Figures 5.5 and 5.6 that PESQ and 3SQM are not correlated.

Figure 5.6 shows that the two streams (reference and degraded) are well correlated. It had been thought that the difference between the reference and degraded

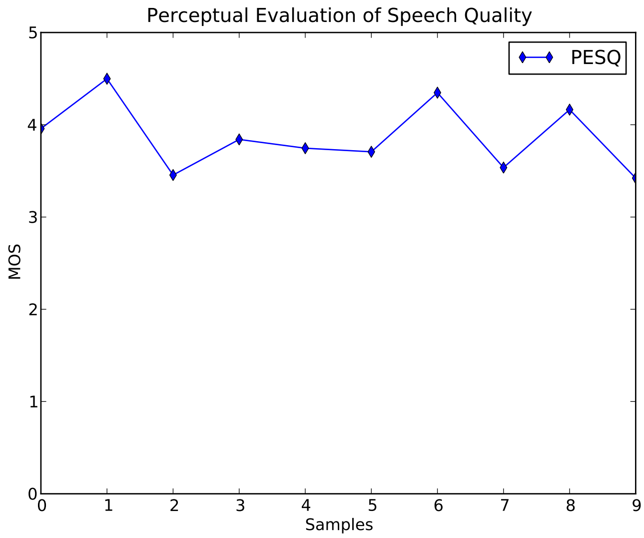


Figure 5.5: From Table 4.1 - The 10 samples captured analysed using PESQ

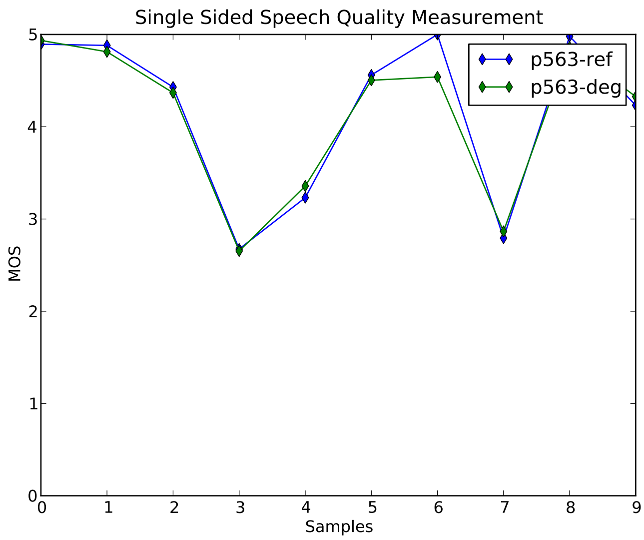


Figure 5.6: From Table 4.1 - The 10 samples captured analysed using 3SQM. The reference signal (p563-ref) and the degraded signal (p563-ref) are analysed

signals' 3SQM score might reflect the degradation in quality. This is shown in Figure 5.7. This is, however not, an accurate assumption, as will be found later in this chapter.

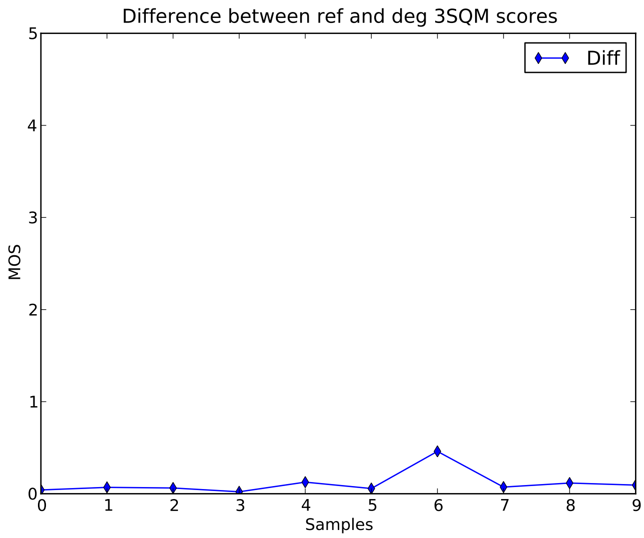


Figure 5.7: From Table 4.1 - The difference between the 3SQM scores of the reference signals and degraded signals of the 10 samples

After the set number of samples has been captured, the data collected in the Data table is then processed and saved in the Statistics table. The *Statistics table* contains averages of the different Data tables generated. It is in the statistics table where the core of this work lies. An example of a small statistics table is shown in Table 4.2.

5.1.2.1 Different Test Servers

One of the objectives, which is put forward in Chapter 1, is to prove that for every IP Network there is a different relationship between the network impairments and the resulting speech quality. The theory behind this is to prove that VoIP calls are influenced differently by different networks, e.g. by low quality network equipment and bad routing principles. Therefore, different IP networks are used for testing. These networks cannot be changed or be influenced. The network can be seen as a black box which is probed by using the network tool developed.

Four different servers are used for testing.

- Lab server (146.232.222.59) - A server situated in the laboratory. No packet loss and very little delay expected

- Local server (196.15.178.165) - A server situated within 30 km (estimated distance is 23 km) of the call. Packet loss and delay should be low.
- A distant server (89.179.245.37) - A test server situated in Moscow, Russia. The estimated distance to the server is 10142 km. As it is a large physical distance, there is huge delay and loss should be much higher
- A commercial server (voiptalk.org) - A commercial VoIP server situated in London, England. The estimated distance to this server is 9689 km. This server is used for commercial calls, therefore it may handle many calls, but it may also become overloaded.

To test these servers a network diagnosis tool called MTR (My Traceroute) [63] is used. MTR sends a sequence Internet Control Message Protocol (ICMP) ECHO request to each one of the hops (routers) to determine the quality of the link to each machine. As it does this, it prints running statistics about each machine. A sudden increase in packet loss or response time is often an indication of a bad (or simply overloaded) link. This usually means that the link is congested. MTR provides valuable statistics regarding the durability of that connection in the seven columns that follow [63]:

- Loss% - The percentage of packet loss at the hop.
- Snt - The number of packets sent.
- Last - The latency of the last packet sent, in milliseconds.
- Avg - The average latency of all packets.
- Best - The best (shortest) round trip time for a packet to this host
- Wrst - The worst (longest) round trip time for a packet to this host.
- StDev - The standard deviation of the latencies to this host. The higher the standard deviation, the greater the difference is between measurements of latency. Standard deviation allows you to assess whether the mean (average) provided represents the true centre of the data set, or has been skewed by some sort of phenomenon or measurement error.

The resulting tables, generated when using MTR to diagnose the four VoIP servers which were used for testing in this project, are shown in Tables 5.1 - 5.4. The data presented cannot be used as a measure for packet loss and delay of VoIP packets, as other protocols (and not UDP) are used for these purposes. It does, however, show how the test server conditions differ.

Table 5.1: The network diagnosis results for the lab VoIP test server

HOST:	dsplab.sun.ac.za	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	dawie.dsp.sun.ac.za	0.0%	50	2.1	1.3	0.2	2.1	4.7

Table 5.2: The network diagnosis results for the local VoIP test server

HOST:	dsplab.sun.ac.za	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	web.esl.sun.ac.za	0.0%	50	1.7	1.6	0.3	19.9	2.9
2.	196.212.135.129	0.0%	50	1.2	3.0	1.0	22.1	4.1
3.	196.212.178.241	0.0%	50	33.7	37.1	32.8	60.0	5.6
4.	cdsl1-rba-vl144.isdsl.net	0.0%	50	36.8	38.5	33.2	98.5	11.4
5.	core5a-rba-gi0-0-0.ip.isnet.	0.0%	50	37.7	36.0	33.2	52.0	3.2
6.	196.26.0.10	0.0%	50	37.5	41.2	33.4	117.1	15.3
7.	rrba-ip-spe-2-wan.telkom-ipn	0.0%	50	34.5	36.7	33.6	49.1	3.2
8.	196.43.25.137	0.0%	50	35.2	37.3	33.3	60.0	5.1
9.	196.25.13.253	0.0%	50	56.3	61.4	54.1	119.4	12.6
10.	155.239.255.249	0.0%	50	60.8	62.7	56.9	113.1	9.6
11.	louwd4.engineering.telkom-ip	2.0%	50	90.2	93.0	87.8	120.4	6.0

Table 5.3: The network diagnosis results for the distant VoIP test server

HOST:	dsplab.sun.ac.za	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	web.esl.sun.ac.za	0.0%	50	0.5	1.9	0.3	19.9	4.9
2.	196.212.135.129	0.0%	50	4.5	1.7	0.9	10.8	2.0
3.	196.212.178.241	0.0%	50	34.8	35.4	33.3	45.7	2.4
4.	cdsl1-rba-vl144.isdsl.net	0.0%	50	33.5	36.2	33.3	55.8	4.2
5.	core2b-rba-te2-0-1.ip.isnet.	0.0%	50	34.8	38.7	32.9	156.7	17.4
6.	mi-za-rba-p6-gi3-0-2-104.ip.	0.0%	50	235.7	235.5	233.1	240.9	1.7
7.	mi-uk-dock-p2-po2-2.ip.isnet	0.0%	50	234.0	236.2	233.4	263.7	4.2
8.	core2a-dock-gi1-0-19-104.ip.	0.0%	50	209.8	207.3	204.7	215.4	2.1
9.	168.209.246.65	0.0%	50	236.1	237.7	234.9	262.0	4.7
10.	cat01.Frankfurt.gldn.net	2.0%	50	580.3	567.7	547.1	604.2	12.2
11.	cat01.Stockholm.gldn.net	8.0%	50	859.5	870.2	852.1	922.9	14.9
12.	bankrost-gw.Moscow.gldn.net	6.0%	50	624.4	620.1	437.4	707.1	36.3
13.	ko-bb-po1.msk.corbina.net	2.0%	50	629.0	622.9	582.5	818.5	30.8
14.	korova-bb.msk.corbina.net	6.0%	50	886.7	880.9	861.2	921.3	10.0
15.	stpetr-bb.msk.corbina.net	6.0%	50	899.8	893.3	865.2	1053.	34.3
16.	mitin-bb-teng3-1.msk.corbina	10.0%	50	873.1	856.5	615.6	941.0	46.7
17.	ipse.static.corbina.ru	6.1%	49	873.3	876.3	857.1	931.7	12.5

Table 5.4: The network diagnosis results for the commercial VoIP test server

HOST:	dsplab.sun.ac.za	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	web.esl.sun.ac.za	0.0%	50	0.6	2.7	0.3	19.9	4.9
2.	196.212.135.129	0.0%	50	1.3	2.8	0.9	20.1	4.9
3.	196.212.178.241	0.0%	50	34.3	36.2	33.1	71.2	5.8
4.	cdsl1-rba-vl144.isdsl.net	0.0%	50	36.1	37.2	33.8	63.0	5.3
5.	core1b-rba-te2-0-1.ip.isnet.	0.0%	50	50.3	41.2	32.6	167.8	21.6
6.	mi-za-rba-p5-gi0-1-105.ip.is	0.0%	50	253.3	241.9	236.7	287.5	8.7
7.	mi-uk-dock-p3-po3-0-2.ip.isn	0.0%	50	238.5	242.1	237.7	279.5	7.1
8.	core1a-dock-gi1-0-0-105.ip.i	0.0%	50	208.4	211.6	207.3	247.8	7.2
9.	168.209.246.65	0.0%	50	238.5	240.3	236.5	277.4	5.7
10.	linx-thn.hotlinks.co.uk	4.0%	50	595.7	552.4	227.3	595.7	56.2
11.	217.14.130.46	2.0%	50	802.7	818.2	797.5	834.7	7.6
12.	77.240.48.94	4.0%	50	808.4	806.4	780.6	830.0	9.5

It can be seen in Table 5.3 that the connection to the test server in Moscow has the worst packet loss and delay and, therefore, the worst connection.

These servers will now be used to investigate the relationship between the various impairments and speech quality. It would have been ideal to test the different servers simultaneously. However, evaluating VoIP performance is beyond the scope of this project. The results of each tool are discussed.

5.1.3 The Impact Network Impairments have on Speech Quality

The impact of Loss, Delay and Jitter on speech quality will now be investigated.

5.1.3.1 Using 3SQM

The results of the simulations in Subsection 5.1.1 show that PESQ are more accurate than 3SQM at assessing speech quality when network characteristics are changed. This was also found in other studies [64] [65]. The same can be seen when using 3SQM to measure speech quality in real world VoIP calls (Section 5.1.3).

It is well known that the quality (e.g. MOS score) deteriorates as packet loss increases [66] [4]. This is not the case when using 3SQM, as can be seen in Figure 5.2b, where a steady decrease in MOS score was expected as network characteristics deteriorate [66] [4]. In Figure 5.8 3SQM was used to determine the perceived speech quality achieved in a VoIP call to the distant test server. The speech quality degradation due to packet loss is much slower than would be expected [66] [4].

It had been thought that the difference between the reference and degraded signals' 3SQM score might reflect the degradation in quality. This was discussed in Subsection 5.1.2. It can be seen in Figure 5.9a that the correlation of 3SQM MOS

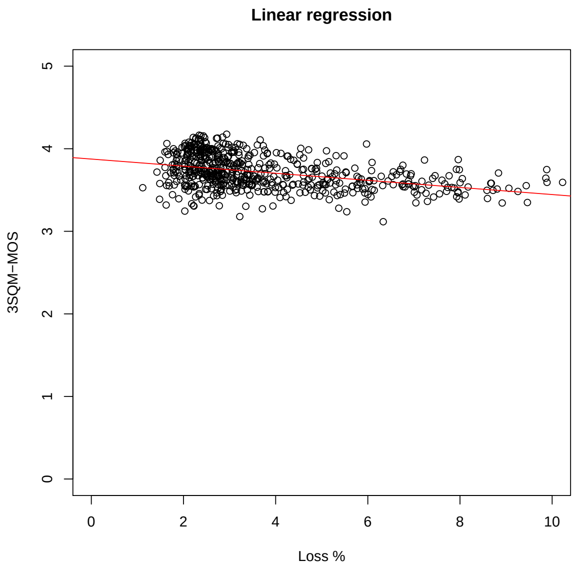
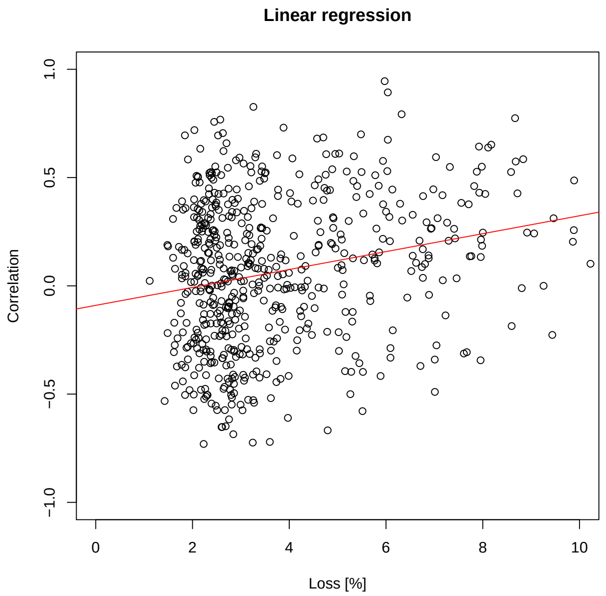
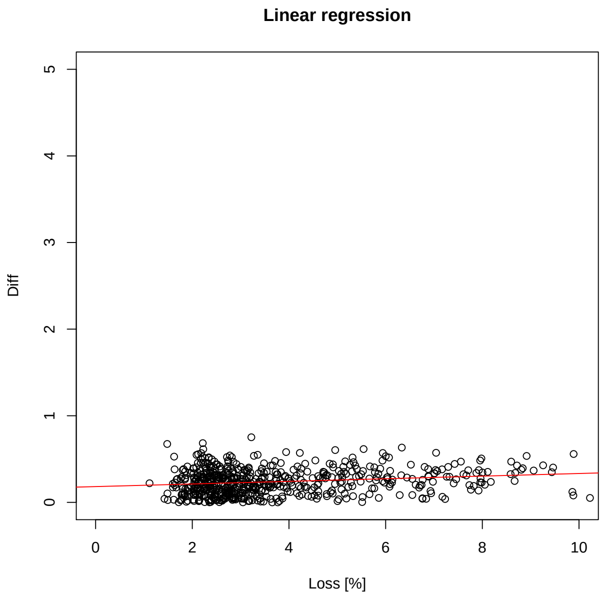


Figure 5.8: The relationship between packet loss and 3SQM MOS



(a) Correlation



(b) Difference

Figure 5.9: The correlation and difference between the 3SQM scores for the reference and degraded signal packet loss.

does decrease as loss increases, but the results are too saturated to be statistically meaningful. In Figure 5.9b the difference between the 3SQM scores of the reference signal and the degraded signal is shown. The difference does increase as packet loss increases. This increase in difference is much slower than expected and therefore, the difference in 3SQM scores can not be used to accurately measure speech quality.

On the basis of these results, the simulation results (where 3SQM had a bigger standard deviation between measurments) and the results from other studies (where PESQ and 3SQM results were correlated with subjective measurements [64] [65]), 3SQM will not further be used to analyse VoIP calls in this project.

5.1.3.2 The Impact of Loss on Speech Quality

The impact that packet loss has on speech quality will be investigated. All four of the test servers mentioned in the previous section will be used.

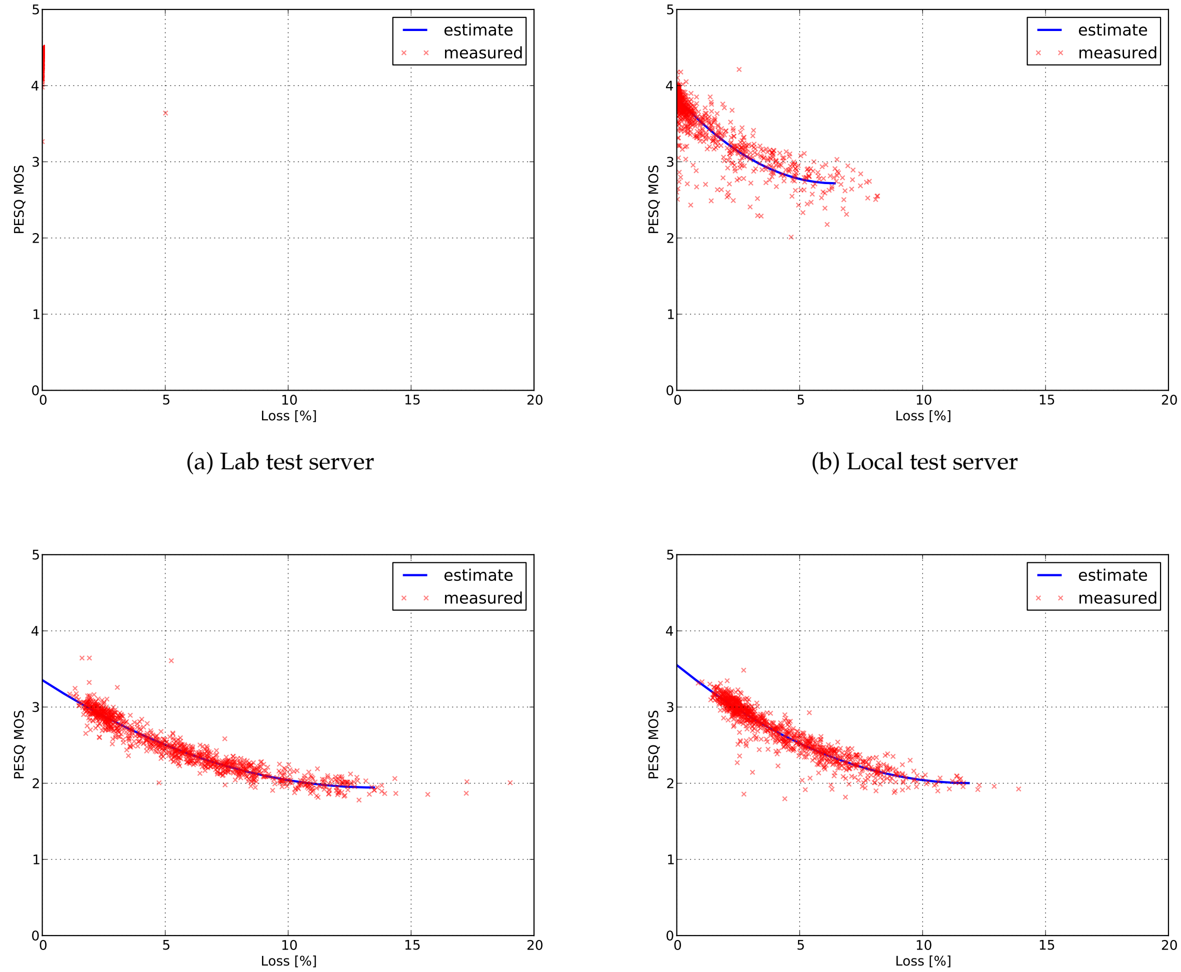
The lab test server In Figure 5.10a it is shown that, for the lab tests, there was no loss. Almost all the samples had 0% loss. One sample has packet loss and this is probably due to the test server's processing. The PESQ MOS scores are mostly high (above 4) as the reference and degraded signals are almost the same.

This test has not produced any statistically meaningful results, as it only shows that in the lab the network has no loss.

The local test server In Figure 5.10b, it is shown that, for the local test server, there is some packet loss. Most of the samples recorded had less than 1% packet loss. Hoewever, the difference between the reference signal and degraded signal can be seen in the PESQ MOS scores achieved. This shows that even with a nearby test server VoIP calls have low quality.

The distant test server The distant test server always has a loss of more than 1%. This is shown in Figure 5.10c. The quality of all the calls made to this server achieved a PESQ score of 3.8 and lower. There is also a bigger difference between the minimum and maximum percentage loss of the captured samples. This is mainly because there are more hops and therefore more places for the network to become congested.

The commercial test server As with the distant test server, the tests using the commercial server (Figure 5.10d) almost always had a loss of more than 1%. Most of the PESQ MOS scores measured lie between 3.5 and 2. This is a barely acceptable level of speech quality. The reason for the high loss of packets is the same as the case of the distant server. More hops generally mean more places for the network to become congested.



(a) Lab test server

(b) Local test server

(c) Distant test server

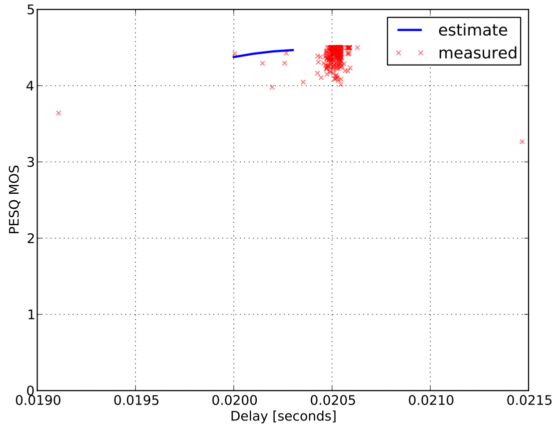
(d) Commercial test server

Figure 5.10: The impact of packet loss on PESQ MOS for the different test servers

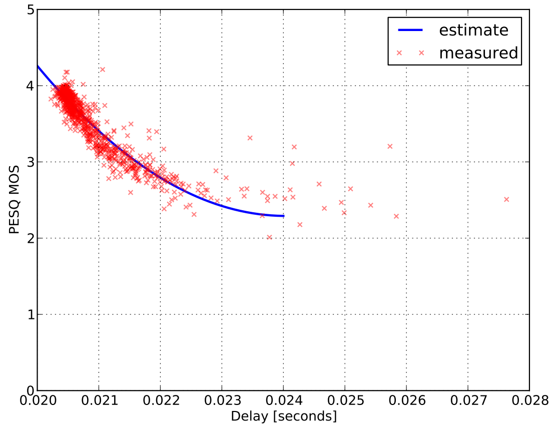
5.1.3.3 The Impact of Delay on Speech Quality

The impact that inter-packet delay† has on speech quality will be investigated. All four test servers mentioned in the previous section will be used. In some cases packet loss and inter-packet delay are correlated. This will be discussed for each of the tests seen in Figure 5.11. It is assumed that the correlation of packet loss and inter-packet delay is a result of congestion. This assumption is made by the theory that congestion on a network will delay packets before it starts to drop packets. The G.711 codec is used. As discussed in Section 2.2 the inter-packet delay will be at least 20 milliseconds for packetisation time (frame size).

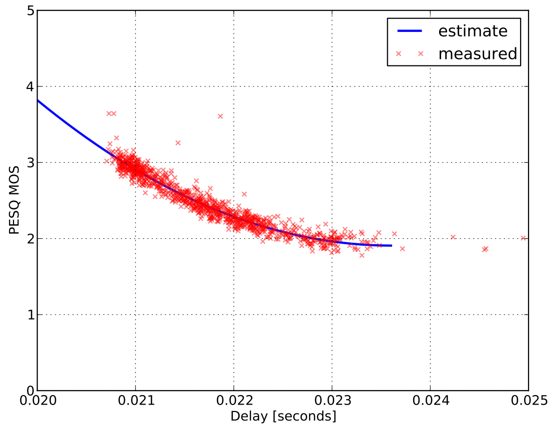
†This is inter-packet delay (the delay between receival of packets) not the trip delay (the delay between sender and receiver)



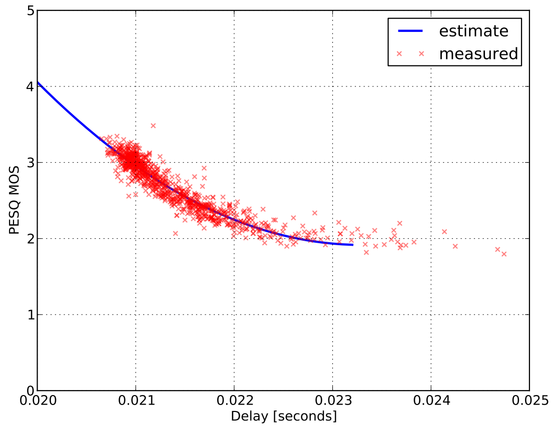
(a) Lab test server



(b) Local test server



(c) Distant test server

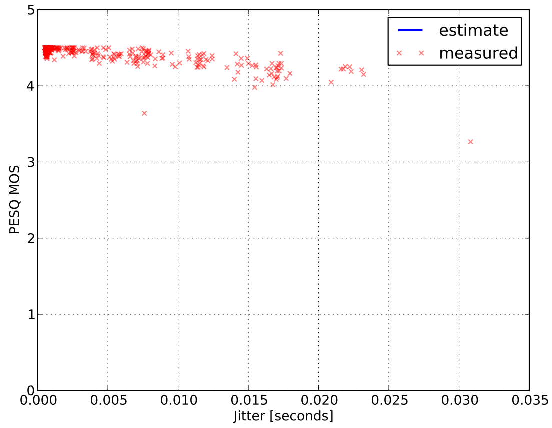


(d) Commercial test server

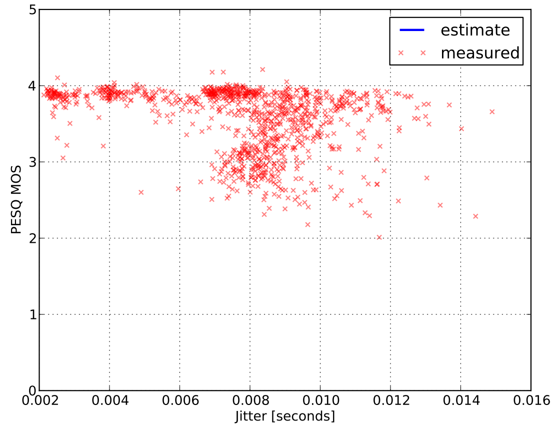
Figure 5.11: The impact of inter-packet delay on PESQ MOS for the four test servers

The lab test server As can be seen in Figure 5.11a, most of the packets captured had a inter-packet arrival time of 0.0205 seconds. This means that it took 0.5 milliseconds for most packets to travel through the lab network. There is very little to no congestion in this network. There was reversed correlation (-0.720) between loss and inter-packet delay. Again, as with packet loss tests, the lab tests for inter-packet delay did not produce statistically meaningful results.

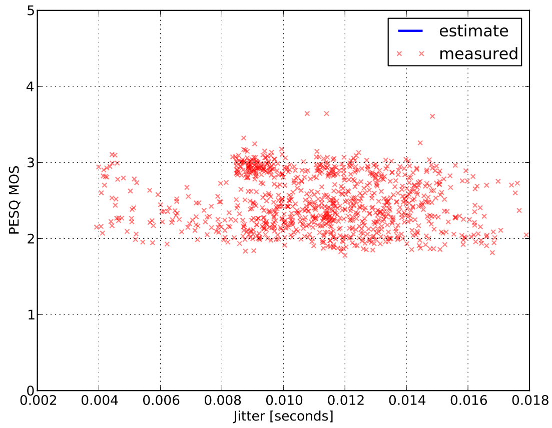
The local test server The results for the local test server (Figure 5.11b) provide much more statistically meaningful results than those of the lab test server. The correlation between the inter-packet delay and packet loss is much higher (0.686). This shows that, for the local test server there is very little congestion, especially



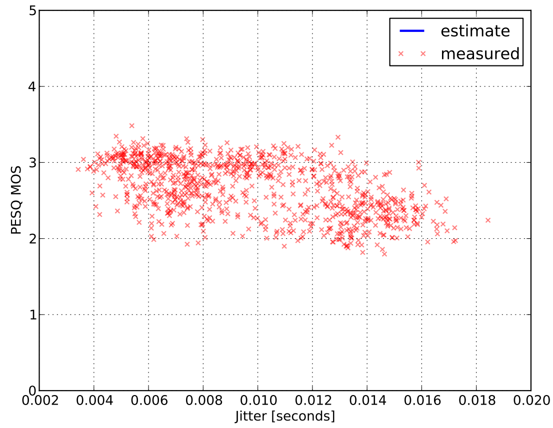
(a) Lab test server



(b) Local test server



(c) Distant test server



(d) Commercial test server

Figure 5.12: The impact of jitter on PESQ MOS for the different test servers

when considering the results for the following two test servers.

The distant test server Figure 5.11c indicates that in most cases the inter-packet delay was more than 21 milliseconds. This is much more than for the local test server. But this delay is normal when taking into account the physical distance between the sender and destination. The correlation between packet loss and inter-packet delay is 0.982. This complies with the theory that congestion adds to this correlation, as there is bound to be more congestion on the distant server than on the local or lab server.

The commercial test server As with the distant test server, the commercial test server also has, in most cases, an inter-packet delay of more than 21 milliseconds. This is again due to the physical distance. The correlation between loss and inter-packet delay for this case is 0.843. It is assumed (because it is for commercial use) that there is less congestion on this server than on the distant test server.

5.1.3.4 The Impact of Jitter on Speech Quality

The impact that jitter has on speech quality is shown in Figure 5.12. There is no clear linear relationship between jitter and speech quality, as was found with the other impairments. This probably because, as was researched in Section 2.4, jitter buffers are usually effective on delay variations of less than 100 ms. It is again necessary to note that the points shown on the graph are averages of the ten samples captured.

It is assumed that the clusters (dark spots in Figures 5.12b-5.12d) of data points are due to the different routes the packets take during different calls.The same variations can be seen when looking at jitter over time (Figure 5.14c). These clusters, as well as the widely scattered measurements, show that jitter does not give the statistically meaningful information required for this project. The standard deviation of the measured jitter is quite high as will be discussed in Subsection 5.2.3.

5.1.4 The Impact of Time of Day on Network Impairments and Speech Quality

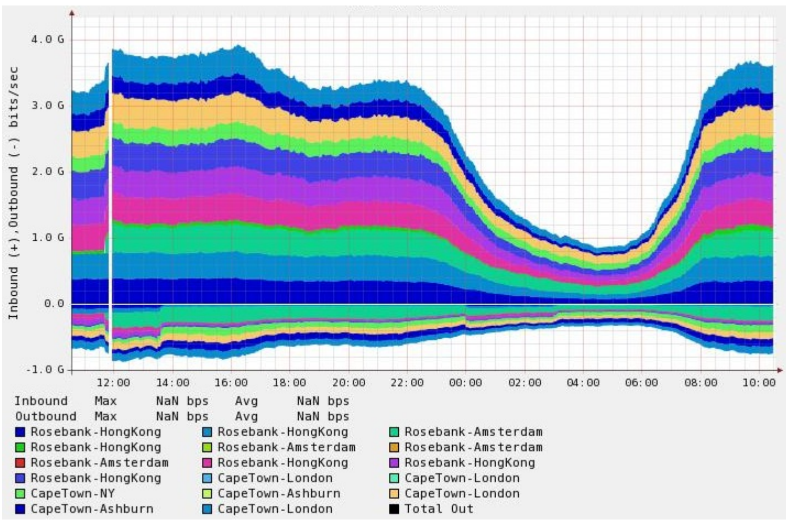


Figure 5.13: Normal international traffic usage for Telkom network

At different times of day, the usage of the internet and networks is different. This can be seen in Figure 5.13 ‡ .

It is well known that network traffic exhibits non-stationary (different network characteristics at different times of the day) behaviour [57]. It is, however, assumed that a repeatable network usage pattern exists. For example, the network was found to be used less between 00:00 - 07:00 than during the rest of the day. Therefore there is less congestion and packet loss and delay should be less. This was also found in when testing the network analysis tool, as shown in Figures 5.14a and 5.14b. Thus VoIP call quality should be better during these hours. This agrees with the results shown in Figure 5.14d.

In Figure 5.14c the results for jitter versus time is shown. This figure shows jumps in jitter measurements. It is assumed that these jumps are the result of the changing network paths taken by the packets.

When looking at the average (10 minute time band/frames) PESQ MOS score at a certain time we can observe that the PESQ score is influenced by the time of day. For the lab test (Figure 5.15a) the general PESQ score remains the same for the entire day, while the local test (Figure 5.15b) is influenced only between 7:00 and 22:00. This is when more users are active on the network.

Figures 5.15c and 5.15d show the results for international calls. Best call quality is achieved in the early hours of the morning. It is interesting to note that the call quality before 00:00 is higher than after 00:00. It is assumed that this is because of timed updates or network configuration changes. Figure 5.13 Figure 5.13 shows a drop in the upload speed at 00:00.

5.1.5 Other Tests

5.1.5.1 Longer Samples

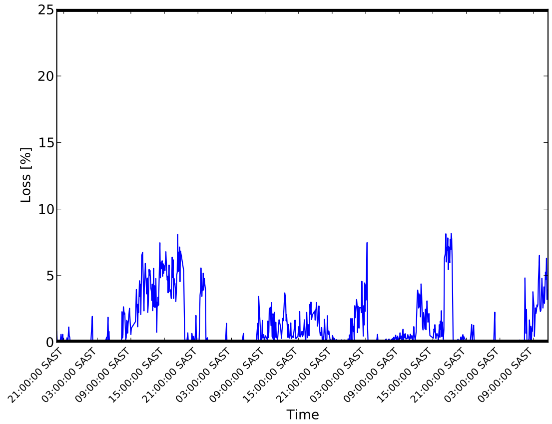
In order to test whether 10 seconds is the most efficient sample size, a test was done where 20 second samples were captured.

In Figure 5.16 the results are shown when using bigger (longer) samples for the distant server. These results indicate that when a 20 second capture sample size is used, the results are much more saturated than in the case of 10 second sample size, as shown in Figures 5.10c and 5.11c.

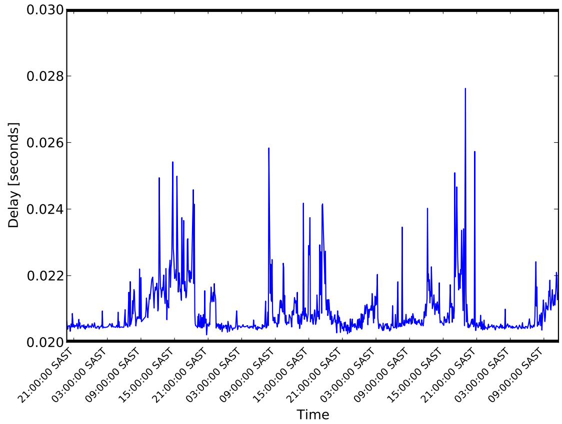
5.1.5.2 Different Codec

The network analysis tool can also be used on other codecs. A test was done using the GSM 06.10 codec. The results for the distant test server are shown in Figure

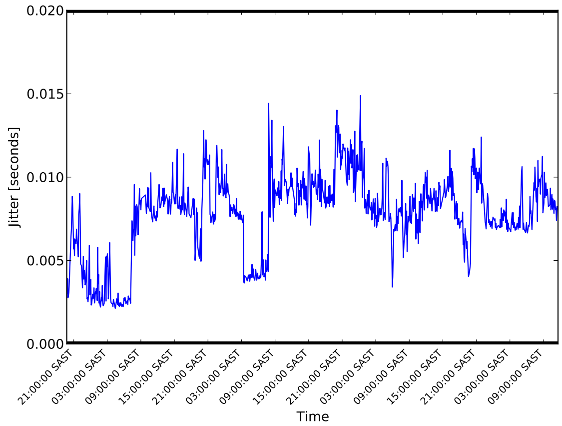
‡This image was received from a classified third party



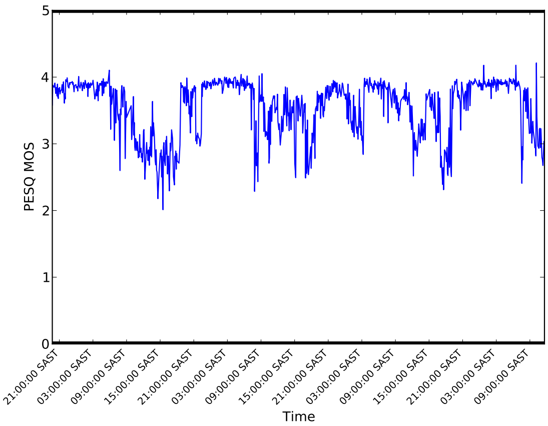
(a) Packet loss



(b) Inter-packet delay



(c) Jitter



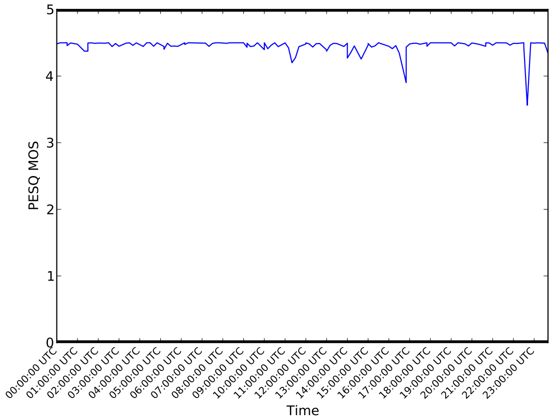
(d) PESQ Score

Figure 5.14: The impact the time of day has on network impairments and speech quality in a 3 day period

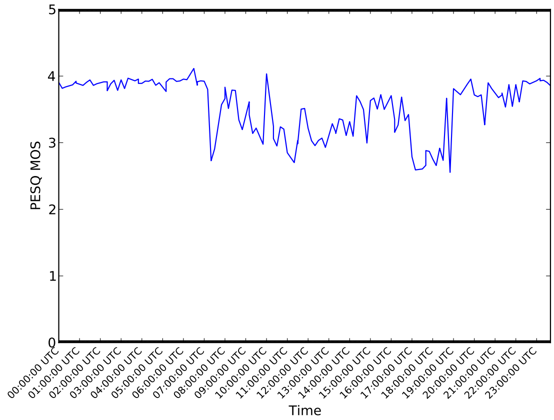
5.17. G.711 has small delay, as seen in Figure 5.11c, while coders that require more processing, e.g. GSM 06.10, introduce more delay as shown 5.17b.

5.2 Using Network Impairments to Predict Speech Quality

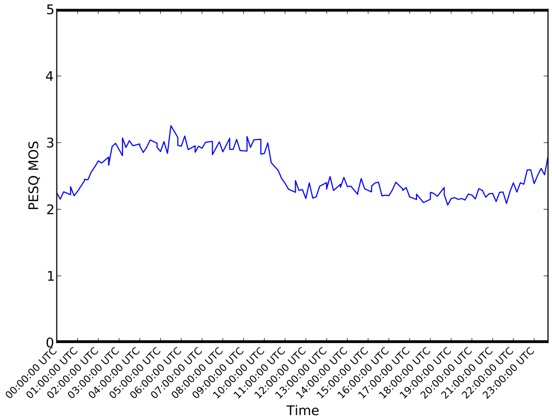
The data captured in the *Statistics table* can now be used to model the relationship between PESQ MOS scores and network impairments. These models can then be used to predict PESQ MOS scores by using quickly obtainable network characteristics. By this process a non-intrusive prediction tool for perceived VoIP quality is derived.



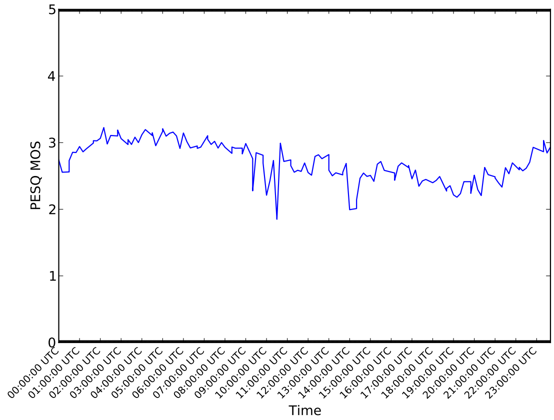
(a) Lab test server



(b) Local test server



(c) Distant test server

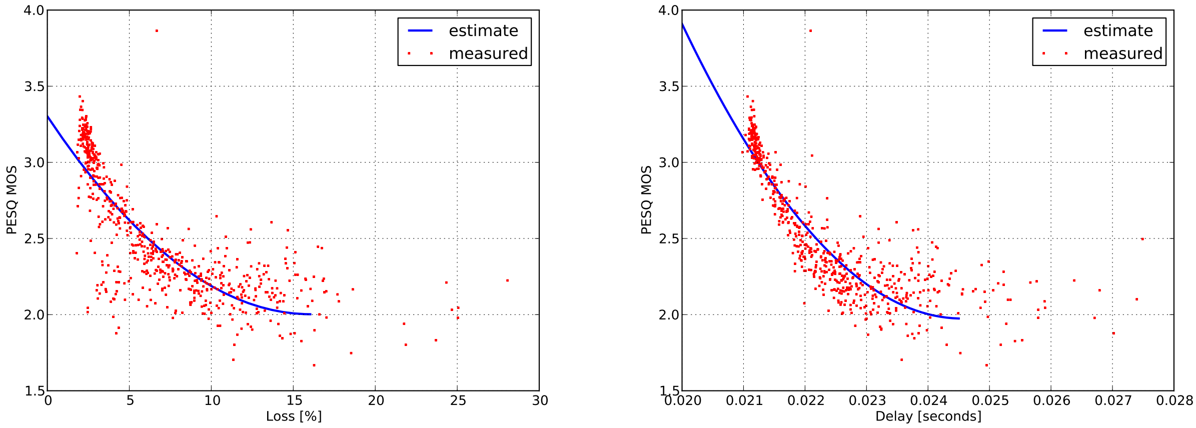


(d) Commercial test server

Figure 5.15: The impact time of day has on speech quality for the different test servers

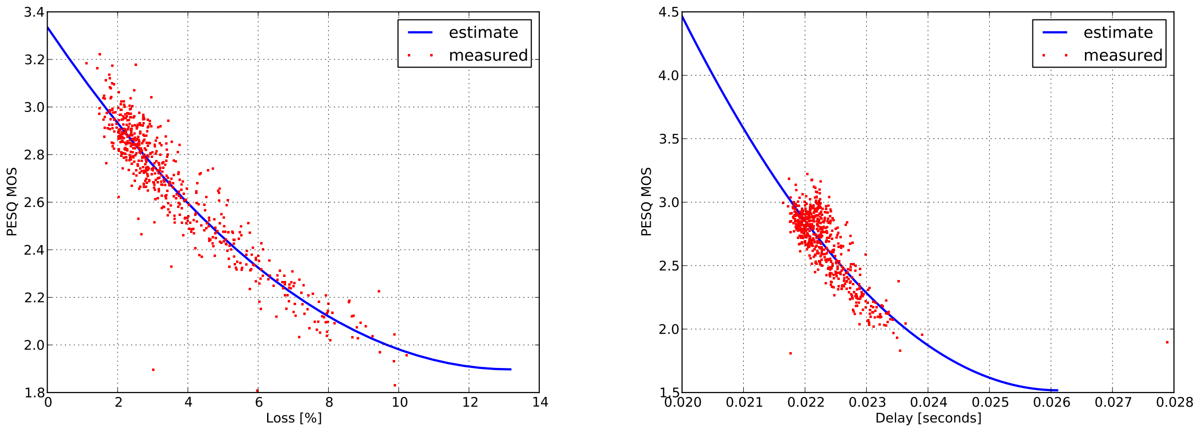
Two different approaches were taken to model the relationship between the impairments and PESQ MOS. First a straight line model was derived using the least squares method discussed in the previous chapter. The linear models derived for the various impairments of the commercial server can be seen in Figure 5.18. The the distributions of the estimation errors when using a straight line model are shown in Figure 5.19. The prediction error when using a straight line to model the relationship between packet loss, delay and jitter is visible in these distributions. The lab tests were done in near perfect network conditions (no loss and small delay), therefore results of the lab tests are not statistically meaningful.

Figure 5.20 shows the distributions of the estimation errors when using a sec-



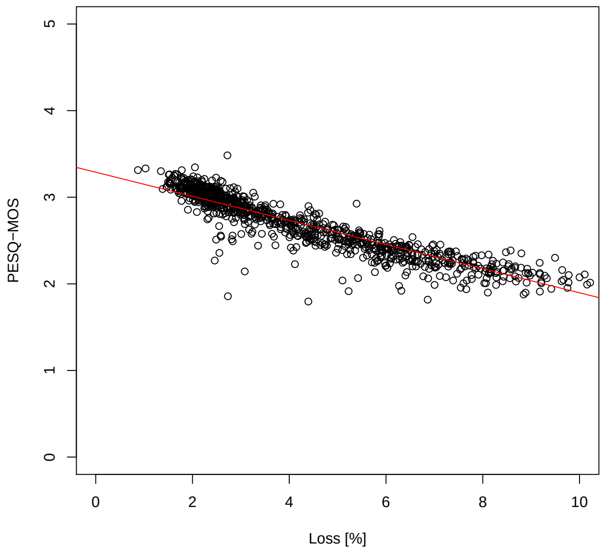
(a) Packet loss (b) Inter-packet delay

Figure 5.16: The relationship between different network impairments and PESQ MOS when capturing bigger samples

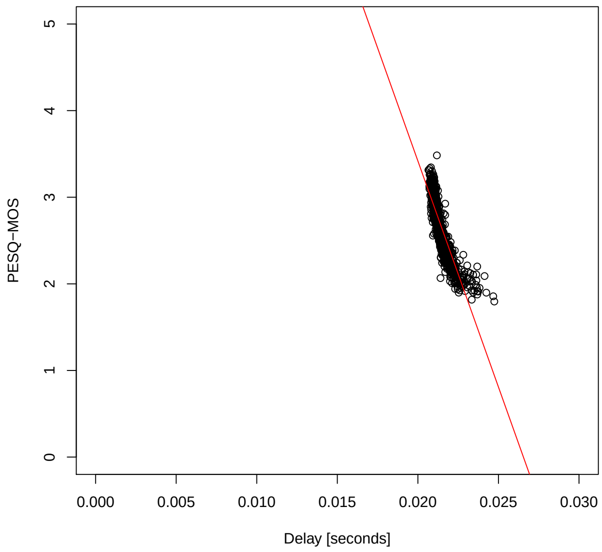


(a) Packet loss (b) Inter-packet delay

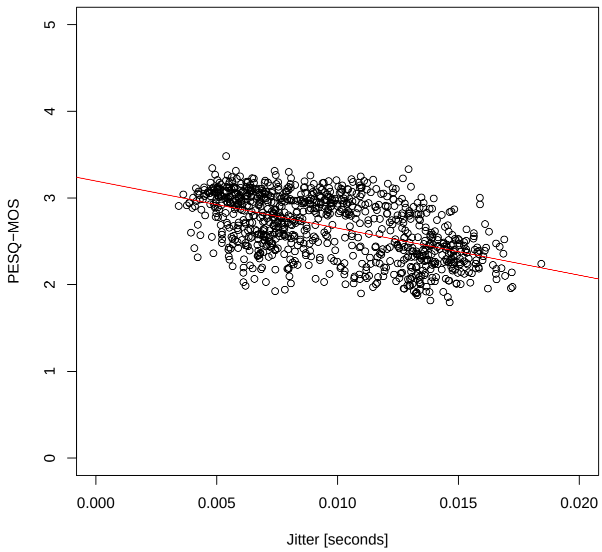
Figure 5.17: The relationship between different network impairments and PESQ MOS when using the GSM 06.10 codec



(a) Loss



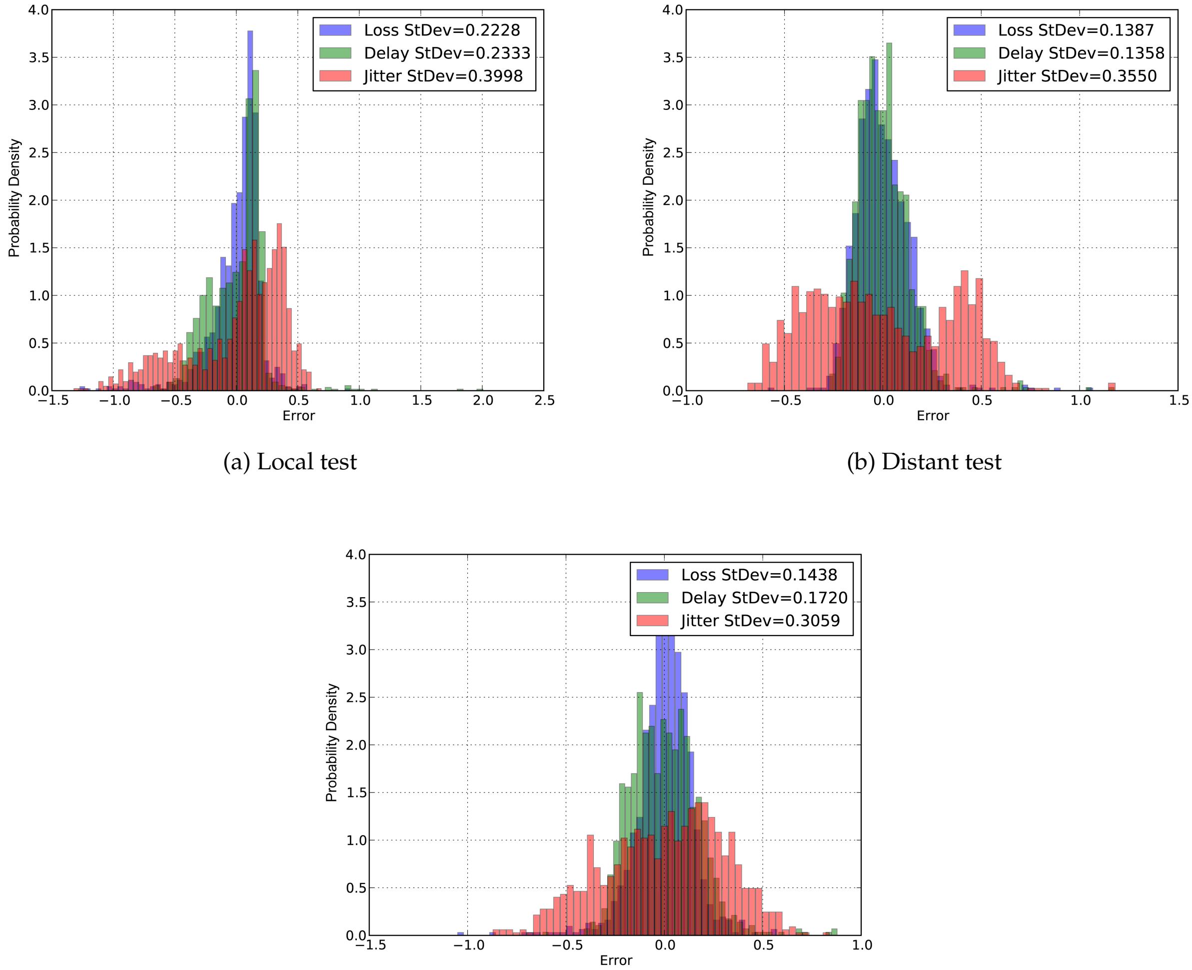
(b) Delay



(c) Jitter

Figure 5.18: Using a straight line to model the relationship between PESQ MOS and different network impairments on the commercial test server

ond order polynomial to model the relationship between speech quality and network impairments. Again the least squares model is used to derive the models. The derived models can be seen by the estimated line in Figures 5.10 - 5.12. The stan-



(a) Local test

(b) Distant test

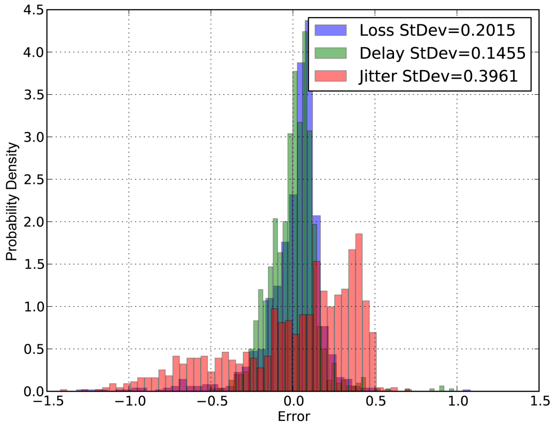
(c) Commercial test

Figure 5.19: Predicting speech quality using network characteristics - Straight line model

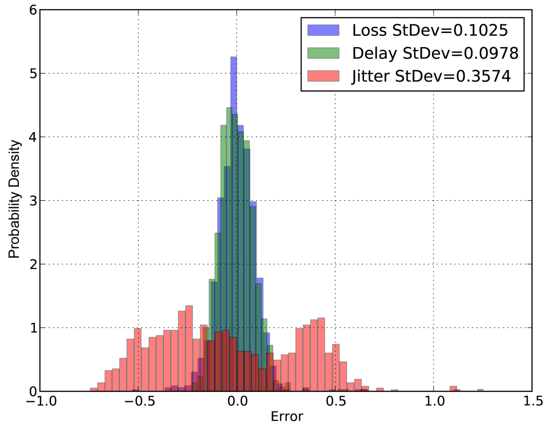
dard deviation of the distributions for second order polynomial model are smaller than for the linear models. Therefore the second order polynomial models will be used for prediction.

As stated in Subsection 4.4.2, it is assumed that the mean lies at 0. Therefore the standard deviation is used to determine the prediction error. For this project we will be looking at two standard deviations of the mean or where 95% of the prediction errors lie.

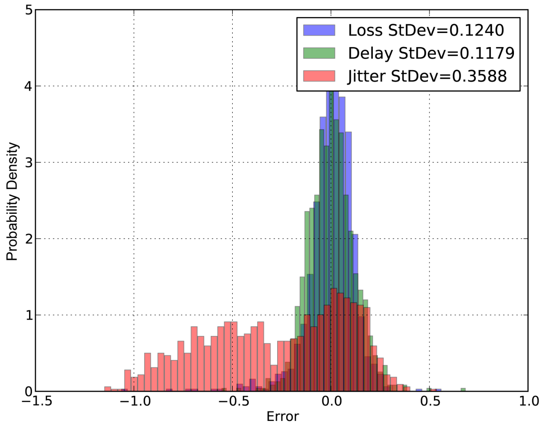
For the usefulness of each of the impairments for predicting perceived speech quality will now be discussed.



(a) Local test



(b) Distant test

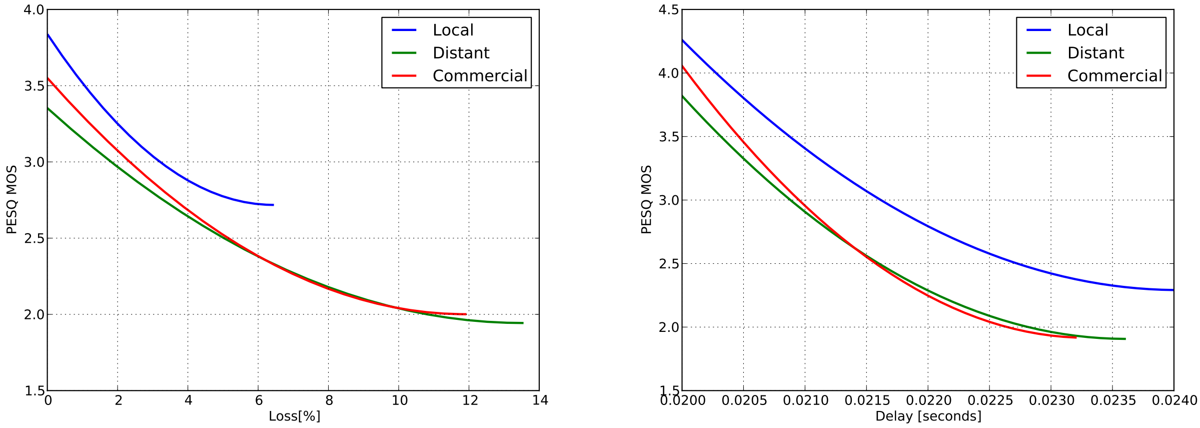


(c) Commercial test

Figure 5.20: Predicting speech quality using network characteristics - Second order polynomial model

5.2.1 Using Loss for the Prediction of Speech Quality

Packet loss is a characteristic which can be used for prediction as it has low standard deviations for the tests done. The worst case is for the local test (Figure 5.20a). This is because, as can be seen in Figure 5.10b, many of the loss samples' PESQ scores are saturated near 0%. Even in this worst case, 95% of the errors are smaller than 0.403. This is acceptable. In the best case, for the distant server (Figures 5.10c and 5.20b), 95% of the errors are smaller than 0.205. Taking into account that PESQ MOS is on a scale between -0.5 and 4.5 it can be calculated that an error of 0.205 is only 4%. The derived models for packet loss for each of the test servers can be seen in Figure 5.21a.



(a) Loss

(b) Delay

Figure 5.21: Second order polynomial models for the two network impairments

The equations for the derived packet loss/PESQ models in Figure 5.21a for the local (Equation 5.2.1), distant (Equation 5.2.2) and commercial (Equation 5.2.3) test cases follow.

For local test:

$$\text{PESQ MOS} = 0.0270496677072x^2 - 0.348224555857x + 3.83881485134 \tag{5.2.1}$$

For distant test:

$$\text{PESQ MOS} = 0.00770665044688x^2 - 0.208481485418x + 3.35342168892 \tag{5.2.2}$$

For commercial test:

$$\text{PESQ MOS} = 0.0109428292149x^2 - 0.260446367857x + 3.55082300924 \tag{5.2.3}$$

5.2.2 Using Delay for the Prediction of Speech Quality

Delay has has the lowest standard deviations of all the tests done. For even the worst case (Figure 5.20a) the standard deviation is only 0.1455. This means the that 95% of all the errors made, according to the model derived for the local test, are smaller than 0.291. Delay was therefore found to be an excellent variable to use for predicting speech quality.

The equations for the derived inter packet delay/PESQ models in Figure 5.21b are:

For local test:

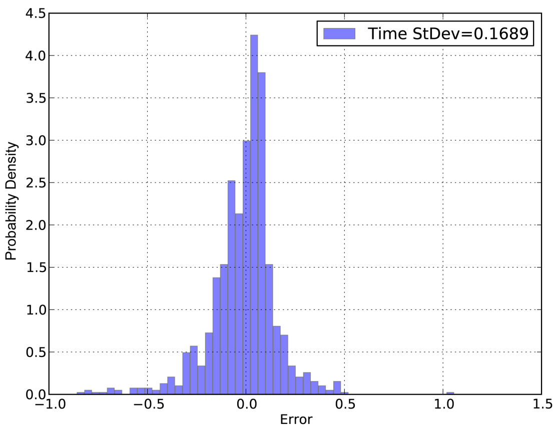
$$\text{PESQ MOS} = 120523.263407x^2 - 5795.05366874x + 71.9515415582 \qquad (5.2.4)$$

For distant test:

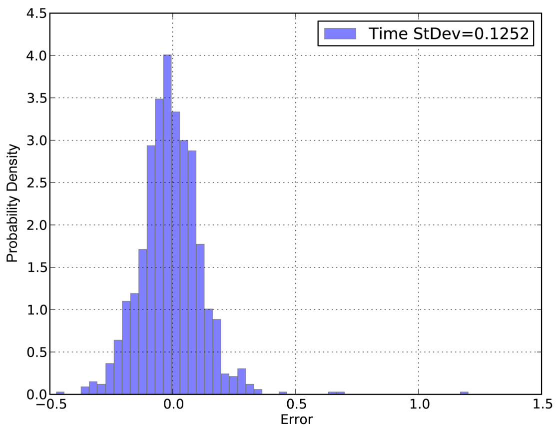
$$\text{PESQ MOS} = 146610.433815x^2 - 6923.38132768x + 83.6427148565 \qquad (5.2.5)$$

For commercial test:

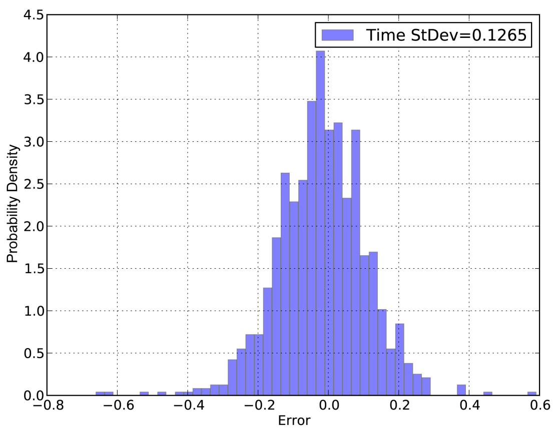
$$\text{PESQ MOS} = 196713.102074x^2 - 9166.12188713x + 108.692815841 \qquad (5.2.6)$$



(a) Local test



(b) Distant test



(c) Commercial test

Figure 5.22: Predicting speech quality using time of day information

5.2.3 Using Jitter for the Prediction of Speech Quality

Jitter is the only impairment whose relationship with speech quality is modelled more accurately by a straight line. None of the jitter distributions looks like a normal distribution, as had been assumed. Using this assumption, even if we look at the best case (in Figure 5.19c), the standard deviation is 0.3059. Therefore 95% of the errors made using that specific model are smaller than 0.6118. This is not acceptable, especially when looking at the standard deviations achieved using second order polynomial models for packet loss and delay.

Therefore, jitter cannot be used to predict perceived speech quality.

5.2.4 Using the Time of Day for the Prediction of Speech Quality

Time can also be used to predict the perceived call quality. Figure 5.15 indicates that the PESQ MOS scores change during the day. Therefore time of day is a variable which can be used to predict call quality. Figure 5.22 indicates the distributions of the estimation errors when using time to predict the error. As discussed in Subsection 5.1.4 in this case 10 minute bands (frames) were taken to get an average MOS score at a certain time. This may be improved by taking smaller time frames but, even with the 10 minute frames the standard deviation of these distributions, as seen in 5.22, is acceptable. Internet network usage differs from day to day, for example there is less usage on weekends than during weekdays. The time data can therefore be further improved by using specific statistics for different days of the week.

5.3 The Non-Intrusive Quality Prediction Algorithm (NIQPA)

We can use the results from the investigations to detemine the impact that the various network parameters have on the quality of perceived speech (Section 5.2), to set up a final prediction model. This prediction model or NIQPA combines network parameters which present the lowest standard deviations. This means that this simple algorithm uses the network parameters which best model of the relationship between perceived speech quality. This is not the best model, but can be used to predict speech quality when given network characteristics. It is used to prove that prediction is possible.

As shown in Section 5.2 packet loss, delay and time of day information are the network parameters which present the lowest standard deviations. The models for these parameters will be unique for every network as shown in Figures 5.14 and 5.21. The resulting algorithm for the NIQPA is shown in Equation 5.3.1.

As indicated earlier (Subsection 5.1.4), IP Networks characteristics are non-stationary [67], therefore, the NIQPA will become outdated. This is because network traffic exhibits non-repeatable or non-stationary behaviour [61].

$$\text{Predicted MOS} = \frac{LossMOS + DelayMOS + TimeMOS}{3} \tag{5.3.1}$$

where:

- $LossMOS$ = The MOS score calculated when using the least squares second order polynomial model derived for the relationship between packet loss and PESQ MOS. Examples of loss models are Equations 5.2.1, 5.2.2 and 5.2.3.

- $DelayMOS$ = The MOS score calculated when using the least squares second order polynomial model derived for the relationship between inter-packet delay and PESQ MOS. Examples of delay models are Equations 5.2.4, 5.2.5 and 5.2.6.

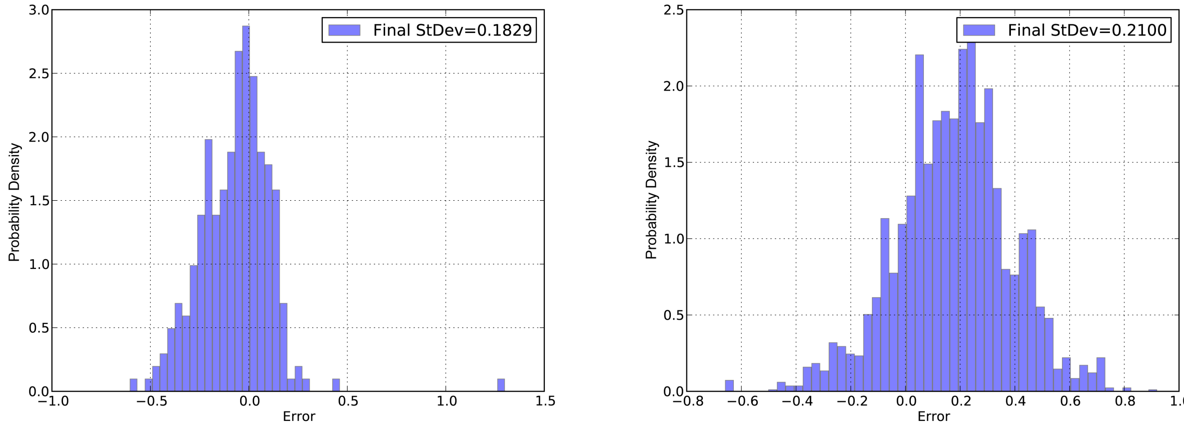
- $TimeMOS$ = The MOS score calculated when using the time of day information, as discussed in the previous subsection.

This is only valid for the models derived in this proof of concept, both the straight line and second order polynomial fitting were investigated. It was found that the second order polynomial was more suitable to model the relationship between speech quality degradation and network characteristics. However, third order polynomials were not investigated in this thesis. [10] found that using third order polynomials are more accurate for predicting speech quality. In this thesis however this is beyond the scope as only the concept of predicting was presented and not the investigation of which prediction model is best suited for prediction.

Figure 5.23 shows the distributions of the prediction errors when using the NIQPA to predict the MOS score.

Testing the NIQPA was done by capturing network traffic again, and thus generating prediction models for the various network characteristics versus PESQ, these models are then used by the NIQPA (the predictor model).

After the models have been set, network traffic was captured again, on the same network. Except this time the measured PESQ values were put against the predicted MOS via the NIQPA (Equation 5.3.1). Therefore, we can measure the prediction error.



(a) Distant test

(b) Commercial test

Figure 5.23: Predicting speech quality using the NIQPA

The error seen Figure 5.23a is the prediction error when the NIQPA was set up using 2652 measurements (captured between 2010-05-03 and 2010-05-14) and evaluated using 1378 measurements (captured between 2010-05-28 and 2010-05-30). This is for the distant server. The standard deviation of the prediction error in this case is 0.183.

The same was done for the commercial server, where the NIQPA was set up using 1270 measurements of PESQ and network characteristics captured between 2010-04-21 and 2010-04-23. This generated NIQPA was the tested by capturing 544 samples between 2010-04-15 and 2010-04-16, the PESQ measured here was then evaluated against the MOS predicted by NIQPA. The prediction error is shown in Figure 5.23b.

For the commercial server NIQPA test case the standard deviation is 0.21. Using the 68-95-99.7 rule (explained in 4.4.2 of this thesis), 95% of all the errors are smaller than 0.41. In this (worst) case 95% of the errors is, therefore, smaller than 12.2%. This is barely acceptable but still proves the concept, and viable speech quality prediction using network characteristics is achieved.

An advantage of the NIQPA is that it allows for the quality prediction of multiple VoIP calls. If a live VoIP call is to be analysed using an objective method, the speech needs to be extracted from the network traffic. Privacy implications arise here. Furthermore, the algorithms used by the objective methods also need processing power, which may be a problem when analysing the speech quality multiple VoIP calls.

Finally, a tool to measure network characteristics and predict speech quality using the NIQPA an as described in Section 4.4.4 can be implemented. An example of the output of such a tool is:

```
gshmaritz@floyd:~\$ python predict-simple.py
Capturing...

Total Packets = 345.0
Packets Lost = 11.0              MOS: 2.83165835264
Delay = 0.0231841594203          MOS: 1.9183094641
Time = 12:30                     MOS: 2.32576387683
Final Predicted MOS: 2.35857723119
```

The NIQPA is, therefore, a QoS implementation which uses an intrusive model, PESQ, to predict perceived speech quality non-intrusively.

5.4 Summary

The implementation of the network analysis and prediction tools can be considered successful, since stable functionality has been proven. As a result, the project objectives have also been met.

In Section 5.1, the impact of the network impairments, which were previously discussed in Section 2.4, on speech quality are studied and results are shown. The effect that network impairments have on speech quality are shown using simulation results. The difference between the speech quality measurement models (PESQ and 3SQM) can already been seen in these results. After the simulation results had been discussed the network analysis tool test set-up was discussed. Four test servers are used for the evaluation of this tool; The lab, local, distant and commercial servers.

The results from the different servers were compared and discussed. It was shown that the 3SQM measurement model does not give accurate results and it is therefore not used further for this project.

The network impairments (packet loss, inter-packet delay and jitter) are then discussed for each of the different servers. Plots show the effect that these impairments have on speech quality. The correlation between these impairments are also discussed, and it is assumed that the correlation between inter-packet delay and packet loss is an indicator of network congestion.

Another important parameter is time. Network usage differs at different times of the day. The variation of speech quality and the network impairments at various time is shown.

Results are shown for tests done under different test conditions (different sample size and codecs) are also shown.

In Section 5.2, the speech quality prediction tool as designed and implemented, is discussed. It is shown that straight line least squares fit models give more prediction error than using second order polynomial models. It was shown that the relationship between jitter and speech quality does not provide a statistically meaningful result.

Packet loss, inter-packet delay and time are shown be parameters that give the lowest prediction error. Therefore a combination of these three parameters are used to derive a final algorithm in Section 5.3. This final algorithm (NIQPA) is then implemented in an application, which quickly and efficiently predicts speech quality using only network traffic.

It is clear that the evaluation satisfies the project objectives set in Chapter 1.

Chapter 6

Conclusions and Future Work

The rapid way in which VoIP has been implemented internationally together with the value added services it brings makes it one of the fastest growing areas of telecommunication technology. Users expect to get the most out of their services. This makes it a rewarding area in which to conduct research and development

In this project, the goal was to use network traffic to gather network statistics indicative of VoIP quality, to enable fast and efficient prediction of perceived quality. To this end, the following conclusions have been reached:

6.1 Conclusions

This project required an in depth investigation of the appropriate technology, together with the subsequent development of the necessary tools. A good understanding of the background was very important.

After the necessary research had been done, two networking tools were developed; a network analysis tool and a speech quality prediction tool.

The *network analysis tool* probes the network by making an echo call to a set location and capturing the network traffic. The captured network traffic is then analysed by:

- Extracting the voice streams.
- Converting the voice streams to speech signals.
- Analyse the resulting speech signal by using standardised objective perceived speech quality measurement models.
- Extracting network parameters such as, packet loss, delay and jitter.

This data (gathered by the network analysis tool) is then used by the *prediction tool* to derive models, which model the relationship between network parameters and perceived speech quality. These models are then used for the speech quality prediction. The prediction tool is then used to non-intrusively predict speech quality using only easily obtainable network characteristics. This prediction is carried out by using the derived Non-Intrusive Quality Prediction Algorithm (NIQPA).

It was shown that jitter is not a statistically meaningful parameter for predicting speech quality. Packet loss and inter-packet delay, however, were very useful for this purpose. The correlation between loss and inter-packet delay was assumed to indicate network congestion, which is the main cause of speech degradation on packet networks.

The NIQPA uses the network parameters, which best model of the relationship between perceived speech quality, to predict a MOS score. This NIQPA assumes that packet loss, inter-packet delay and the time of day network parameters have equal impact on the degradation of perceived speech quality. There is no motivation for this. A weighted NIQPA needs to studied in future work.

The verification of the developed NIQPA algorithm is based on the variance of the error between the NIQPA predication and the MOS as measured by the flawed Perceptual Evaluation of Speech Quality (PESQ) algorithm. Subjective testing is the most reliable approach to assess perceived speech quality, because it measures the way humans perceive speech. However, in order to achieve a more standardised approach objective tests are needed.PESQ and 3SQM are accepted objective speech quality assessment standards. However, it was found (in Section 5.1) that PESQ is more accurate than 3SQM at assessing speech quality when network characteristics are varied. Therefore, PESQ was used to verify the NIQPA.

It was shown that the NIQPA generates different relationship for each of the networks tested. This shows that a generic model to predict speech quality using network statistics on all IP network can not be used. Therefore, a different prediction model will have to be generated for each network.

The results of this thesis demonstrates that intrusive methods can be used in a non-instrusive implementation if prediction is used to model the intrusive tests' results. This project provides proof that this can be used in real-world network monitoring.

The concept developed can be used to verify IP service level. Another area, which is beyond the scope of this project but is important for future work, is how this concept could be utilised to adapt the network to increase perceived speech quality.

The same project concept can also be used for other real-time applications that use packet networks; this includes images, music and video. The only requirement is that a PESQ-like application should exist for that media type. However, if the necessary resources (time and money) are available, subjective tests can also be used to gather the required statistics.

This project was challenging, but proved very interesting and its successful completion up to this point would seem to present subsequent network monitoring possibilities.

6.1.1 Compare and contrast

As mentioned in Chapter 1, this thesis is an extension and verification of the work done by Lingfen Sun in [10]. This subsection presents a comparison between previous studies (specifically [10]) and the work done in this study.

- This thesis only uses the straight line and 2nd order polynomial model for prediction models, while it was found in [10] that 3rd order polynomials are more suited to model the relationship between network characteristics and speech quality. To find the best possible prediction algorithm was not part of this study's objectives.

- This thesis includes a investigation into 3SQM. Although 3SQM has been investigated in other studies ([8], [64] and [65]), it has not been investigated for the predicting speech quality as it already measures the quality of speech on one end of the conversation.

- In [10] four codecs were investigated (G. 729, G723.1, AMR and iLBC), while in this thesis two different codecs (G.711 and GSM 06.10) were investigated. The reason for this is that the codecs used in this thesis are freely available, while the codecs used in [10] are proprietary.

- This thesis investigates the use of other statistics, such as time of day, to improve the prediction of speech quality. This was not investigated in [10].

- A tool to capture and analyse real world network data was designed in this study, while simulation results was used in [10]. A practical network analysis tool was not investigated in [10]. This is a important part of this thesis, as testing was done on real world networks. Thus we found, as was investigated in [10], that an intrusive model, such as PESQ, can indeed be used to predict perceived speech quality non-intrusively.

6.2 Future Work

Although the network analysis and perceived speech quality tools developed in this project can be used to monitor VoIP networks, various enhancements could be added. The following improvements can be considered:

- The speech quality measurement models, such as PESQ and Single Sided Speech Quality Measurement (3SQM), are effective but not perfect. A hybrid model which incorporates network parameters (E-model) and speech analysis (PESQ) is required.
- 3SQM needs to be improved, as there is a need for a non-intrusive voice quality measurement tool.
- The speech quality measurement models can be improved so as to use less processing power. This will allow for real-time speech quality analysis. Alternatively with the necessary processing power, more network segments could be monitored simultaneously.
- The algorithm derived can be improved by using rich statistical structures. These models would be self-configuring, self-learning and self-adaptive.
- As mentioned the weighting of the parameters NIQPA needs to studied.
- If more statistics are gathered the derived algorithms can further be improved by deriving different models for different days of the week. This can be further extended to cover different months.
- A deeper investigation into the security of VoIP is also very important, especially after capturing and analysing calls in this project that were without encryption.
- A graphical interface with current speech quality prediction information can be implemented. Service providers will find it useful for monitoring the quality of the VoIP service.
- The designed algorithm can be used for the automatic adaptation of network parameters and bandwidth to provide users with better perceived speech quality.

Bibliography

[1] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” *RFC 3550, IETF Network Working Group*, Jul 2003.

[2] H. Sinnreich and A.B Johnston, *Internet Communications Using SIP*, 2nd ed. Wiley Publishing, Inc., 2006.

[3] F. Ohrtman, *Softswitch Architecture for VoIP*. McGraw-Hill, 2004.

[4] I. Marsh, “Quality Aspects of Internet Telephony,” Ph.D. dissertation, School of Electrical Engineering, KTH, Stockholm, Sweden, Dec. 2009.

[5] L. Ding, “Learn about VoIP Quality Measurements,” 2009, http://www.eetindia.co.in/ad64153f.HTM.

[6] L. Yamamoto and J. Beerends, “Impact of Network Performance Parameters on the End-to-End Perceived Speech Quality,” *In Proceedings of Expert ATM Traffic Symposium*, 1997.

[7] L. Sun and E. C. Ifeachor, “Voice Quality Prediction Models and Their Application in VoIP Networks,” *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 809–820, 8 2006.

[8] M. Goudarzi and L. Sun, “Performance Analysis and Comparison of PESQ and 3SQM in Live 3G Mobile Networks,” 2009, http://www.tech.plym.ac.uk/spmc/staff/mgoudarzi/.

[9] V. Grancharov, D. Y. Zhao, J. Lindblom, and W. B. Kleijn, “Low Complexity, Non-Intrusive Speech Quality Assessment,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1948–1956, 2006.

[10] L. Sun, “Speech Quality Prediction For Voice Over Internet Protocol Networks,” Ph.D. dissertation, Jan 2004.

[11] Nortel Networks, *Real-Time Networking Technologies*. Nortel Press, 2008.

[12] M. Al-Akhras and I. A. Momani, *VoIP Quality Assessment Technologies*. InTech, 2011.

[13] ITU, "PESQ: An Objective Method for End-to-End Speech Quality Assessment of Narrow-band Telephone Networks and Speech Codecs," *Recommendation P.862*, Feb 2001.

[14] ITU, "Single-Ended Method for Objective Speech Quality Assessment in Narrow-band Telephony Applications," *Recommendation P.563*, May 2004.

[15] R.J. Bates, *Broadband Telecommunications Handbook*, 2nd ed. McGraw-Hill, 2002.

[16] Elotek Telecom, "VoIP Glossary," 2010, http://www.elotek.net/voip_glossary.asp.

[17] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," *RFC 2543, IETF Network Working Group*, Mar 1999.

[18] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," *RFC 3261, IETF Network Working Group*, Jun 2002.

[19] InteropNet, "What is SIP?" 2005, http://www.opus1.com/voip/2005ilabswhatissip.pdf.

[20] VoIP Think, "Codecs," 2010, http://www.voipforo.com/en/codec/codecs.php.

[21] ITU, "G.726: Adaptive Differential Pulse Code Modulation (ADPCM)," Dec 1990.

[22] Cisco, "Voice Over IP - Per Call Bandwidth Consumption," 2006, http://www.ciscosystems.sc/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml.

[23] ITU, "G.711 : Pulse Code Modulation (PCM) of Voice Frequencies," *Recommendation G.711*, Nov 1988.

[24] RJ Systems, "Asterisk: Minimal SIP Configuration," May 2010, http://www.rjsystems.nl/en/2100-asterisk.php.

[25] ETSI, "Digital Cellular Telecommunications System (Phase 2+) (GSM)," *European Telecommunication Standard*, Sep 1994.

[26] International Engineering Consortium, "Voice Over IP - Per Call Bandwidth Consumption," 2006, http://www.iec.org/online/tutorials/voice_qual/topic07.html.

[27] ITU-T, "One-way Transmission Time," *Recommendation G.114*, May 2003.

[28] Voip-Info.org, "QoS," Feb 2010, http://www.voip-info.org/wiki/view/QoS.

[29] Ditech Networks, "Voice Quality beyond IP QoS," Jan 2007, http://www.ditechnetworks.com/learningcenter/whitepapers/WP_VoIP_Voice_Quality.pdf.

[30] ITU, "P.800.1 : Mean Opinion Score (MOS) Terminology," Jul 2006.

[31] ITU, "G.107 : The E-model: A Computational Model for use in Transmission Planning," Jul 2006.

[32] ITU, "Mapping Function for Transforming P.862 Raw Result Scores to MOS-LQO," *Recommendation P.862.1*, Nov 2003.

[33] Opticom, "P.563 - Single Sided Speech Quality Measure," 2006.

[34] 3CX, "SIP phones/VoIP Phones types," 2009, http://www.3cx.com/PBX/VOIP-phones.html.

[35] Articlebase, "Which Popular VoIP Service Suits You?" 2010, http://www.articlesbase.com/voip-articles/which-popular-voip-service-suits-you-3301875.html.

[36] VOIP-Info.org, "Wideband VoIP," Feb 2010, http://www.voip-info.org/wiki/view/Wideband+VoIP.

[37] T. Berson, "Skype Security Evaluation," Oct 2005, www.skype.com/security/files/2005-031%20security%20evaluation.pdf.

[38] Skype, "What is Skype," Feb 2010, http://about.skype.com/.

[39] A. Ghaffar, "How Secure is VoIP?" Nov 2004, http://www.tmcnet.com/voip/1104/FeatureSecurity.htm.

[40] Defence Information Systems Agency, "Internet Protocol Telephony and Voice over Internet Protocol," Apr 2006, http://iase.disa.mil/stigs/stig/VoIP-STIG-V2R2.pdf.

[41] A. K. Sen and M. S. Srivastava, *Regression Analysis: Theory, Methods and Applications*. Springer, 1990.

[42] E. W. Weisstein, “Least Squares Fitting,” 2010, A Wolfram Web Resource http://mathworld.wolfram.com/LeastSquaresFitting.html.

[43] L. Ding and R. Goubran, “Speech Quality Prediction in VoIP Using the Extended E-Model,” *Global Telecommunications Conference*, vol. 7, pp. 3974–3978, 2003.

[44] A. Choudhury, “Correlation and Regression,” 2010, http://www.experiment-resources.com/correlation-and-regression.html.

[45] L. Bewick, V. Cheek and J. Ball, “Statistics Review 7: Correlation and Regression,” 2003, http://www.ncbi.nlm.nih.gov/pmc/articles/PMC374386/.

[46] pjsip.org, “PJSUA Manual Page,” 2010, http://www.pjsip.org/pjsua.htm.

[47] Joomla Connections, “Asterisk,” 2010, http://joomlaconnections.com/index.php/Asterisk/asterisk.html.

[48] Digium, Inc., “Asterisk,” 2009, http://www.asterisk.org/.

[49] TCPDUMP, “Manual Page,” 2009, http://www.tcpdump.org/tcpdump_man.html.

[50] Wireshark, “tshark Manual Page,” 2009, http://www.wireshark.org/docs/man-pages/tshark.html.

[51] SoX, “VoIP Industry Crowned by SIP Dialer,” 2009, http://sox.sourceforge.net/sox.html.

[52] Python Software Foundation, “Subprocess Management,” 2010, http://docs.python.org/library/subprocess.html.

[53] R. Oudkerk, “Processing Package,” 2010, http://pypi.python.org/pypi/processing.

[54] J. L. Rodgers and W. A. Nicewander, “Thirteen Ways to Look at the Correlation Coefficient,” *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[55] "Python-statlib", “Documentation for the Stats Module,” 2010, http://code.google.com/p/python-statlib/wiki/StatsDoc.

[56] J. Schafer, “Linear Regression Modeling,” 2010, http://www.stat.washington.edu/~handcock/506/Lectures/lec11.pdf.

[57] The Scipy Community, "Numpy and Scipy Documentation - numpy.polyfit," 2003, http://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html.

[58] Mathworks, "Documentation - Curve Fitting Toolbox," 2010, http://www.mathworks.de/access/helpdesk/help/toolbox/curvefit/bq_5ka6-1.html.

[59] SUNY Oswego, "The 68-95-99.7 Rule For Normal Distributions," 2010, http://www.oswego.edu/~srp/stats/6895997.htm.

[60] A. Bach, "Numerical Programming in Python," 2007, http://www.webdotdev.com/nvd/content/view/938/1/1/4/.

[61] J. Lv, X. Li, C. Ran, and T. He, "Network Traffic Prediction and Fault Detection Based On Adaptive Linear Model," *IEEE International Conference on Industrial Technology*, vol. 2, pp. 880–885, 2004.

[62] wav2rtp, "A Tool to Generate RTP Data Packets," 2008, http://wav2rtp.sourceforge.net/index.html.

[63] Ubuntu Manuals, "MTR - A Network Diagnostic Tool," 2010, http://manpages.ubuntu.com/manpages/gutsy/man8/mtr.8.html.

[64] M. Goudarzi and L. Sun, "Evaluation of Voice Quality in 3G Mobile Networks," Master's thesis, School of Computing, Communications and Electronics Faculty of Technology, University of Plymouth, 2008.

[65] J. Holub and L. Blaskova, "Transcoded Speech Contemporary Objective Quality Measurements Reliability," *Wireless Telecommunications Symposium*, pp. 106–109, 2008.

[66] C. Hoene, H. Karl, and A. Wolisz, "A Perceptual Quality Model Intended For Adaptive VoIP Applications," *International Journal of Communication Systems*, vol. 19, pp. 299–316, 2006.

[67] M. Masugi, "Applying a Recurrence Plot scheme to Analyze Non-stationary Transition Patterns of IP-network Traffic," *Communications in Nonlinear Science and Numerical Simulation*, pp. 1418–1430, 2009.

[68] E. W. Weisstein, "Vandermonde Matrix," 2010, A Wolfram Web Resource http://mathworld.wolfram.com/VandermondeMatrix.html.

[69] E.W. Weisstein, "Least Squares Fitting–Polynomial," 2010, A Wolfram Web Resource http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html.

Appendices

Appendix A

Least Squares Fitting

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. The most common form of linear regression is least squares fitting. This appendix is a discussion of the mathematics of the least squares method.

A.1 Straight Line Fitting

We want to find the best fitting straight line,

$$y = a + bx \tag{A.1.1}$$

when given a set of data, $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$, where $n \geq 2$. Therefore we want to find a and b when x_i and y_i are given.

The square error (vertical deviation from the line) R^2 is given by [42]:

$$\begin{aligned} R^2 &= \sum_{i=1}^{n} [y_i - f(x_i)]^2 \\ &= \sum_{i=1}^{n} [y_i - (a + bx_i)]^2 \end{aligned} \tag{A.1.2}$$

For the best fitting line, $f(x)$, we must find the least square error, therefore the unknown coefficients must have zero partial derivatives [42]:

$$\frac{\partial(R^2)}{\partial a_i} = 0 \tag{A.1.3}$$

From Equation A.1.2, the partial derivatives of the unknown coefficients, a and b, will be zero for the least square error:

$$\begin{aligned} \frac{\partial(R^2)}{\partial a} &= -2\sum_{i=1}^{n}[y_i - (a + bx_i)] = 0 \\ \frac{\partial(R^2)}{\partial b} &= -2\sum_{i=1}^{n}x_i[y_i - (a + bx_i)] = 0 \end{aligned} \tag{A.1.4}$$

If we expand the above equations:

$$\begin{aligned} \sum_{i=1}^{n} y_i &= a\sum_{i=1}^{n} 1 + b\sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i &= a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 \end{aligned} \tag{A.1.5}$$

Therefore the coeffiecients a and b (as shown in Equation A.1.1) can be obtained:

$$\begin{aligned} a &= \frac{\left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_i y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \\ b &= \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \end{aligned} \tag{A.1.6}$$

A.2 Polynomial Fitting

We want to find the best fitting n^{th} degree polynomial (curve),

$$y = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m \tag{A.2.1}$$

when given a set of data, $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$, where $n \geq 2$. Therefore we want to find $a_0, a_1, a_2, ..., a_m$ when x_i and y_i are given.

The residual or square error (vertical deviation from the curve) R^2 is given by:

$$\begin{aligned} R^2 &= \sum_{i=1}^{n}[y_i - f(x_i)]^2 \\ &= \sum_{i=1}^{n}[y_i - (a_0 + a_1 x_i + a_2 x_i^2 + ... + a_m x_i^m)]^2 \end{aligned} \tag{A.2.2}$$

As in Equation A.1.3, the minimum square error is found where the partial derivates of the residual is equal to zero, therefore:

$$\begin{aligned}
\frac{\partial(R^2)}{\partial a_0} &= -2\sum_{i=1}^{n}[y_i - (a_0 + a_1 x_i + a_2 x_i^2 + ... + a_m x_i^m)] = 0 \\
\frac{\partial(R^2)}{\partial a_1} &= -2\sum_{i=1}^{n} x_i[y_i - (a_0 + a_1 x_i + a_2 x_i^2 + ... + a_m x_i^m)] = 0 \\
\frac{\partial(R^2)}{\partial a_1} &= -2\sum_{i=1}^{n} x_i^2[y_i - (a_0 + a_1 x_i + a_2 x_i^2 + ... + a_m x_i^m)] = 0 \\
&\vdots \\
&\vdots \\
\frac{\partial(R^2)}{\partial a_m} &= -2\sum_{i=1}^{n} x_i^m[y_i - (a_0 + a_1 x_i + a_2 x_i^2 + ... + a_m x_i^m)] = 0
\end{aligned} \tag{A.2.3}$$

This lead to the equations:

$$\begin{aligned}
\sum_{i=1}^{n} y_i &= a_0\sum_{i=1}^{n} 1 + a_1\sum_{i=1}^{n} x_i + a_2\sum_{i=1}^{n} x_i^2 + ... + a_m\sum_{i=1}^{n} x_i^m \\
\sum_{i=1}^{n} x_1 y_i &= a_0\sum_{i=1}^{n} x_i + a_1\sum_{i=1}^{n} x_i^2 + a_2\sum_{i=1}^{n} x_i^3 + ... + a_m\sum_{i=1}^{n} x_i^{m+1} \\
\sum_{i=1}^{n} x_1^2 y_i &= a_0\sum_{i=1}^{n} x_i^2 + a_1\sum_{i=1}^{n} x_i^3 + a_2\sum_{i=1}^{n} x_i^4 + ... + a_m\sum_{i=1}^{n} x_i^{m+2} \\
&\vdots \\
&\vdots \\
\sum_{i=1}^{n} x_m^2 y_i &= a_0\sum_{i=1}^{n} x_i^m + a_1\sum_{i=1}^{n} x_i^{m+1} + a_2\sum_{i=1}^{n} x_i^{m+2} + ... + a_m\sum_{i=1}^{n} x_i^{2m}
\end{aligned} \tag{A.2.4}$$

or, in matrix form:

$$\begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_1 y_i \\ \sum_{i=1}^{n} x_1^2 y_i \\ \vdots \\ \sum_{i=1}^{n} x_m^2 y_i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} 1 & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \cdots & \sum_{i=1}^{n} x_i^m \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \cdots & \sum_{i=1}^{n} x_i^{m+1} \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 & \cdots & \sum_{i=1}^{n} x_i^{m+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_i^m & \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m+2} & \cdots & \sum_{i=1}^{n} x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \tag{A.2.5}$$

This is a Vandermonde matrix [68]. Therefore we can also obtain the matrix for the least squares fit by:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \tag{A.2.6}$$

or in matrix notation:

$$\mathbf{y} = \mathbf{X}\mathbf{a} \tag{A.2.7}$$

$$\mathbf{X}^\mathsf{T}\mathbf{y} = \mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{a} \tag{A.2.8}$$

The matrix, $\mathbf{a}$ and therefore $a_1, a_2, a_3, ..., a_m$ (as shown in Equation A.2.1) can be obtained by solving the following equation [69]:

$$\mathbf{a} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \tag{A.2.9}$$