Quality of Media Traffic over Lossy Internet Protocol Networks: Measurement and Improvement

Mousa Tawfiq AL-Akhras

A dissertation submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

Software Technology Research Laboratory
De Montfort University, United Kingdom

2007

Abstract

Voice over Internet Protocol (VoIP) is an active area of research in the world of communication. The high revenue made by the telecommunication companies is a motivation to develop solutions that transmit voice over other media rather than the traditional, circuit switching network.

However, while IP networks can carry data traffic very well due to their best-effort nature, they are not designed to carry real-time applications such as voice. As such several degradations can happen to the speech signal before it reaches its destination. Therefore, it is important for legal, commercial, and technical reasons to measure the quality of VoIP applications accurately and non-intrusively.

Several methods were proposed to measure the speech quality: some of these methods are subjective, others are intrusive-based while others are non-intrusive. One of the non-intrusive methods for measuring the speech quality is the E-model standardised by the International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T).

Although the E-model is a non-intrusive method for measuring the speech quality, but it depends on the time-consuming, expensive and hard to conduct subjective tests to calibrate its parameters, consequently it is applicable to a limited number of conditions and speech coders. Also, it is less accurate than the intrusive methods such as Perceptual Evaluation of Speech Quality (PESQ) because it does not consider the contents of the received signal.

In this thesis an approach to extend the E-model based on PESQ is proposed. Using this method the E-model can be extended to new network conditions and applied to new speech coders without the need for the subjective tests. The modified E-model calibrated using PESQ is compared with the E-model calibrated using

subjective tests to prove its effectiveness.

During the above extension the relation between quality estimation using the E-model and PESQ is investigated and a correction formula is proposed to correct the deviation in speech quality estimation.

Another extension to the E-model to improve its accuracy in comparison with the PESQ looks into the content of the degraded signal and classifies packet loss into either Voiced or Unvoiced based on the received surrounding packets. The accuracy of the proposed method is evaluated by comparing the estimation of the new method that takes packet class into consideration with the measurement provided by PESQ as a more accurate, intrusive method for measuring the speech quality.

The above two extensions for quality estimation of the E-model are combined to offer a method for estimating the quality of VoIP applications accurately, non-intrusively without the need for the time-consuming, expensive, and hard to conduct subjective tests.

Finally, the applicability of the E-model or the modified E-model in measuring the quality of services in Service Oriented Computing (SOC) is illustrated.

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the Software Technology Research Laboratory (STRL), School of Computing, Faculty of Computing Sciences and Engineering, at De Montfort University, United Kingdom.

No part of the material described in this thesis has been submitted for the award of any other degree or qualification in this or any other university or college of advanced education.

This thesis is written by me and produced using $\LaTeX\ 2_\epsilon$

This thesis was funded by The University of Jordan

Mousa Tawfiq AL-Akhras

Publications

1. M. AL-Akhras and H. Zedan, Quality Estimation for Streamed VoIP Services, in *The 2nd European Young Researchers Workshop on Service Oriented Computing*, 2007.

2. M. AL-Akhras, H. Zedan and R. John, Non-Intrusive Speech Quality Prediction in VoIP Networks using a Neural Network Approach, *Neurocomputing*, (Submitted).

3. M. AL-Akhras and H. Zedan, Policy-based QoS for VoIP Services, *Information & Software Technology*, (Submitted).

Dedication

To the sole of my mother... gone from this world, but never forgotten.
I know she would be delighted to witness my studying for my Ph.D.

To my father ... who sacrificed a lot for me to be what I am now. I owe everything I achieved or I am going to achieve to him. I hope by having my Ph.D. I can draw a smile on his face as this is the least I can give to him.

To my step mother for her care and pray for me all the time.

To my father-in-law, Dr Musa ALMomani, without him it was not possible to start my Ph.D. Without his support and wise advices it was going to be difficult to continue my journey in research.

To my mother-in-law for her love, prayers, endless love.

To my brothers and sisters, my brothers and sisters-in-law, and their families for their support, encouragement, and prayers.

To my long list of friends, the list includes but not limited to: AbdelAziz Haikal, Basem Morad, Mohammad Refai, Raed Tayseer, Qais el-Hinti. My current and past friends from DMU include: Ammar AlZahari, Mohammad AlShadadi, Raed Obaid, Etimad Fadel, Raed Kanaan, Mohammed AlAwairdhi, Omar Al-Dabbas, Ayad ALZobaydi, Mohammad Taye, Ali Hilal, Isam Salah, Abu Laith, Abu Ramzi, Abdul Aldobhani, Zhuopeng Zhang and Yang Xu, Dr Ashraf Saleem, Dr Mohammad Chaita, Dr Omar AlAni, and Dr Omar Daoud (Abu Al-Bara'a).

Most importantly I would like to thank my partner, my colleague and the most important person in my life, my wife ...

Iman

Without her being beside me, standing-by me during difficult moments and continuously encouraging me, everyday and every night, I was not going to be able to do my research or facing life's difficulties.

My thanks goes to my Basil, my son and gift from Allah. Seeing his smile gave me hope. Playing with him gave me relief. Teaching him a word meant a lot to me. My research took a lot of time he deserved to spent with. I hope he would appreciate that by me doing this research I will be able to give him a better life and I hope when I finish I am able to dedicate a lot of my time to him.

Acknowledgement

First and foremost, I would like to thank GOD for all his blessings and bounties he gave to me since I was born.

Here at De Montfort I have had a great experience with so many people supporting me, giving me their scientific support, critical comments, and reviewing my work.

I was very fortunate to have Professor Hussein Zedan as my first supervisor. His scientific support, insightful comments, ideas, visionary approach, support and guidance has greatly helped developing my thinking, technical writing, and widen my horizons. My meetings with him were enjoyable, discussions were stimulating. I am very grateful for his belief, and moral support during difficult moments. He is a great supervisor able to have the best of his students.

Also I would like to thank my supervisor Professor Robert John for his professional guidance, patience, critical comments, and insightful suggestions. His comments greatly improved my ability to analyse and my technical writing. Any comment he was giving, greatly improved this work.

Big thanks go to the examination committee for the the time they spent reading my thesis and giving constructive comments greatly improved the final output of the thesis. My thanks to my external examiner Dr Michael J. Pont for the depth of his questions, suggestions and comments which improved the thesis and gave me the opportunity to give parts of the work extra thought and improvement. Internal Examiners Professor Hongji Yang and Dr Aladdin Ayesh also gave very useful comments on technical writing and structure of the thesis which I think helped improving the readability of the thesis.

My thanks go to: Dr David Elizondo, Dr Amelia Platt, Professor Denis Anthony, Dr Ross Clement, Dr Mario Gongora and Professor Gwynne Evans for the useful discussions, technical review and encouragement. Their help and time are highly appreciated and will be remembered.

I also would like to thank Dr Omar AlAni, Dr Andrew Gelsthorpe, Dr John Gow and Dr Scott Linfoot, Brian Bramer, and Dr François Siewe for the useful discussions and hints they gave to me.

I would like to thank all the members of the STRL lab who made my days in the lab some of the most enjoyable times of my professional life. Special thanks go to Lynn Ryan and Lindsey Trent.

I would like to thank Professor Jeffrey Knight, DMU Associate pro vice-chancellor for his help and time.

Thanks for the research office and especially the head of the office, Mrs.Joanne Cooke for their huge efforts, commitment and dedication in doing their job in the best possible way.

I want to thank Dr Khalil el-Hindi, who taught me back in the University of Jordan, supervised my M.Sc. thesis, helped me starting my Ph.D. study and was influential and raw model for me through my professional career.

My big thanks goes to my wife, my colleague and my best friend, Iman ALMomani for the useful discussions, support, knowledge and experience sharing. She was the closest person to me during my research and the first one to discuss anything with.

Contents

List of Tables

List of Figures

List of Acronyms

AAA	Authentication, Authorisation and Accounting
AbS	Analysis by Synthesis
ACELP	Algebraic Code-Excited Linear Prediction
ACR	Absolute Category Rating
AD	Auditory Distance
ADPCM	Adaptive Differential PCM
AF	Assured Forwarding
AMR	Adaptive Multi Rate
ANN	Artificial Neural Network
ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation 1
Bpl	Packet-loss Robustness Factor
BT	British Telecom
BurstR	Burst Ratio
CAC	Call Admission Control
CCITT	International Telephone and Telegraph Consultative Committee
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CELP	Code-Excited Linear Predictive
CLP	Conditional Loss Probability
CODEC	coder/decoder

COPS	Common Open Policy Service
cRTP	compressed Real-time Transport Protocol
CS	Circuit Switching
CS-ACELP	Conjugate Structure-Algebraic Code Excited Linear Prediction
DAM	Diagnostic Acceptability Measure
DC	Direct Current
DCR	Degradation Category Rating
DiffServ	Differentiated Service
DMOS	Degradation Mean Opinion Score
DoD	Department of Defence
DoS	Denial of Service
DPCM	Differential PCM
DRT	Diagnostic Rhyme Test
DSP	Digital Signal Processing
DTX	discontinuous transmission
E-model	ETSI-model
EF	Expedited Forwarding
EFR	Enhanced FullRate
EIA	Electrical Industries Association
EMBAC	End-to-end Measurement Based Admission Control
ETSI	European Telecommunications Standards Institute
EVRC	Enhanced Variable Rate Coder
FEC	Forward Equivalence Class
FEC	Forward Error Correction
FF	Feedforward
FMNB	Frequency Measuring Normalising Blocks
FS-1015	Federal Standard-1015

FS-1016	Federal Standard-1016
GA	Genetic Algorithm
GSM	Global System for Mobile Communications
GSM AMR	GSM Adaptive Multi-Rate
GSM EFR	GSM Enhanced Full Rate
GSM-FR	GSM Full Rate
GSM-HR	GSM Half Rate
GW	GATEWAY
HMM	Hidden Markov Models
HTTP	HyperText Transfer Protocol
HVXC	Harmonic Vector eXcitation Coding
IDS	Intrusion Detection System
Ie	Equipment Impairment
IETF	Internet Engineering Task Force
iLBC	internet Low Bit-rate
IMBE	Improved Multi Band Excitation
Inf	Infinite
INMARSAT	INternational MARitime SATellite Corporation
IP	Internet Protocol
IPSec	IP Security
ISO	International Organisation for Standardisation
ITU	International Telecommunication Union
ITU-T	ITU-Telecommunication Standardisation Sector
kbps	kilo bit per second
LBG VQ	Linde, Buzo, and Gray Vector Quantization

LBR Low-bit Rate
LD-CELP Low-Delay CELP
LM Levenberg-Marquardt
LPC Linear Prediction Coding
LR Likelyhood Ratio

M2E Mouth to Ear
MC Multipoint Controller
MCU Multipoint Controller Unit
MDC Multi Description Coding
MELP Mixed Excitation Linear Prediction
MG Media Gateway
MGC Media Gateway Controller
MGCP Media Gateway Control Protocol
MLP Multilayer Perceptron
MNB Measuring Normalising Blocks
MOS Mean Opinion Score
MP Multipoint Processor
MP-MLQ Multi-Pulse Maximum Likelihood Quantisation
MPE Multi-Pulse excited
MPLS MultiProtocol Label Switching
ms millisecond

NB Narrow Band
NB-CELP Narrow Band CELP
ND Noise Disturbance
NGW Network GateWay

OSI Open System Interconnection
OSP Open Settlement Protocol

PAMS	Perceptual Analysis Measurement System
PCA	Principal Components Analysis
PCM	Pulse Code Modulation
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PESQ	Perceptual Evaluation of Speech Quality
PHB	Per-Hop Behavior
PKI	Public Key Infrastructure
PLC	Packet Loss Concealment
POTS	Plain Old Telephone Service
Ppl	Packet loss Probability
PSQM	Perceptual Speech Quality Measure
PSTN	Public Switched Telephone Network
RAS	Registration, Admission, and Status
RCELP	Relaxation Code Excited Linear Prediction
RESV	Reservation Request
RFC	Request For Comment
ROCCO	Robust Checksum-based header Compression
RPE	Regular-Pulse Excited
RPE-LTP	Regular Pulse Excited Long Term Prediction
	Signal-to-Noise Ratio
RSTP	Reliable Signalling Transport Protocol
RSVP	Resource Reservation Protocol
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RVP	Remote Voice Protocol
SAP	Session Announcement Protocol
SCCP	Skinny Client Control Protocol
SCTP	Stream Control Transmission Protocol

SDP	Session Description Protocol
SEGSNR	segmented SNR
Sigtran	Signalling Transport
SIP	Session Initiation Protocol
SNR	Signal-to-Noise Ratio
SOC	Service Oriented Computing
SS7	Signalling System 7
STANAG	STANdardisation AGreement
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
TFO	Tandem-Free Operation
TIA	Telecommunication Industries Association
TMNB	Time Measuring Normalising Blocks
UDP	User Datagram Protocol
ULP	Unconditional Loss Probability
UMTS	Universal Mobile Telecommunication System
UNO	United Nations Organisation
URI	Uniform Resource Indicator
URL	Uniform Resource Locator
UV	UnVoiced
V	Voiced
VAD	Voice Activity Detector
Vocoder	Voice coder
VoIP	Voice over IP
VPN	Virtual Private Network
VQ	Vector Quantisation
VSELP	Vector Sum Excited Linear Prediction

WB	Wide Band
WB-CELP	Wide Band CELP
WFQ	Weighted Fair Queue
Yle	listening effort score
Ylq	listening quality score

Chapter 1

Introduction

1.1 Motivation of the Research

Voice over Internet Protocol (VoIP) is an active area of research in the world of communication. The high revenue made by the telecommunication companies is a motivation to develop solutions that transmit voice over other media rather than the traditional, circuit switching network.

With the transmission of voice over packet switching networks such as IP networks, voice and data services can be integrated which makes creation of new and innovative services possible, IP networks are seen as long-term carriers for all types of traffic.

IP networks were originally designed to carry data traffic and they are doing this task very well. As IP networks are best-effort networks, they are not particulary fit to support real-time applications such as voice traffic in addition to data traffic.

Due to the best-effort nature of the IP networks, several challenges arise which prevent these networks from providing the high quality speech often provided by traditional telephony networks for voice services.

Among the challenges is sharing of resources in IP networks as no resources are dedicated to the voice call in contrast to what is happening in traditional circuit switching telephony where the required resources are allocated to the phone call from the start to the end.

With the absence of resource dedication, many problems are inevitable in IP networks. Among the problems is packet loss which occurs due to overflow in intermediate routers or due to the long time taken by packets to reach their destinations [24, 135].

Real-time applications are also sensitive to delay since they require the voice packets to arrive at the receiving end within a certain upper bound to allow interactivity of the voice call [81, 82].

Also due to their best effort nature, packets could take different routes from the same source to the same destination which makes the packets' interarrival time vary over time in what is known as jitter. Due to the problem of jitter, it is not easy to play packets in a steady fashion to the listener [133, 178, 179].

All the above problems affect the quality of the received speech signal and many solutions have been proposed to alleviate these problems. The quality of the received speech signal as perceived by the end user is greatly affected by the effectiveness of these solutions.

VoIP services often offer much cheaper solutions over their traditional circuit switching counterparts, but regardless of how the service is cheap, it is the user perception of the quality what matters. If the quality of the voice is poor, the user of the traditional telephony will not be attracted to the new system regardless how cheap the service is.

From this point of view, it is important to have solutions to enable accurate measurement of the speech quality. This importance comes from legal, commercial, and technical reasons.

One of driving forces in the world of communication is the International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T). ITU is the leading United Nations agency for information and communication technology. As the global focal point for governments and the private sector in developing telecommunication networks and services, ITU's role is to help the world communicate. ITU - Telecommunication Standardisation Sector (ITU-T) is a permanent organ of the ITU that plays a driving force role toward standardising and regulating international telecommunications world wide.

Toward this goal, ITU-T study technical, operating and tariff questions and produce standards under the name of Recommendations for the purpose of standardising telecommunications worldwide. ITU-T's Recommendations are divided into categories that are each identified by a single letter, referred to as the "series", and Recommendations are numbered within each series, for example "G.711". ITU-T has a formal recognition as it is part of International Telecommunication Union which is a United Nations Organisation (UNO). Prior to 1992 the ITU-T was known as International Telegraph and Telephone Consultative Committee (CCITT).

Part of the effort of ITU-T is developing standards in the world of communications, a series of these standards concerns the measurement of speech quality for voice services and many of these standards are considered in this thesis. Speech quality in ITU-T standards is expressed as Mean Opinion Score (MOS) which ranges between 1 and 5, with 1 corresponds to poor quality and 5 to excellent quality.

Some of the proposed standards measure the speech quality or the MOS subjectively by setting lab conditions and asking subjects to listen to the speech signal and give their estimation of the quality in terms of MOS. This method is standardised in ITU-T Recommendation P.800 [72].

Other methods depend on comparison of the received signal with the original signal to measure the perceived quality in terms of MOS, these methods are known as intrusive methods as they require the injection of the original signal to analyse the distortion of received signal. The most recent method for measuring the speech quality intrusively is known as Perceptual Evaluation of Speech Quality (PESQ). PESQ is standardised as ITU-T Recommendation P.862 [79].

Yet another category depends on networking conditions and the received signal to estimate the quality non intrusively without the need for the original signal. The most famous method in this category is called the E-model which is defined in ITU-T Recommendation G.107 [84].

While the above methods offer solutions for measuring the speech quality, their applicability to VoIP networks should take into consideration the nature of IP networks. Among the desired features of any solution for speech quality measurement are:

1. Provides measurement of speech quality online while the network is running.

2. It should be non-intrusive, i.e. it should depend on the received speech signal and not on the original speech signal.

3. Be accurate and reflects how the quality is perceived by the end-user.

4. With the changing world, it should be applicable to new and emerging applications and networking conditions. As such it should avoid the subjectivity in estimating parameters. The E-model for example depends on subjective tests to estimate packet loss parameters.

The applicability of different solutions for measuring the speech quality is the topic of this thesis.

1.2 Research Questions

The main research questions to be answered in this thesis are:

(1) What is the best method for measuring the speech quality in VoIP applications?
(2) Can this measurement be improved to make it suitable for this technology by satisfying the desired features?
(3) What is the effect of the inevitable packet loss on the speech quality?

Several research questions should be defined in order to be able to answer these questions:

1. What are the characteristics of different methods for measuring the speech quality?
   To answer this question, the desired characteristics of speech quality measurement in VoIP networks are identified and the existence of these features in different speech quality measurement methods is checked.

2. What are the relationships between different methods for measuring the speech quality?
   To answer this question, different methods for measuring the speech quality should be compared to find if different methods lead to the same outcome and whether they provide accurate measurement of speech quality.

3. What is the best method for measuring the speech quality in VoIP applications?
   Based on the desired features for VoIP speech quality measurement, the best method for this measurement is identified.

4. How can one or more of these methods be improved for the purpose of measuring the speech quality in VoIP networks?
   After identifying the best method, different deficits, if any, are identified and improvements are proposed to satisfy the desired features.

5. How can any proposed method be tested to make sure of its effectiveness?
   Any proposed method should be compared to a base measurement known to be accurate for the measurement the proposed method aim to measure to check its effectiveness and accuracy in measuring the speech quality.

1.3 Research Method

Different methods for measuring the speech quality are compared and their features are identified, for this purpose several standards from the ITU-T for speech quality measurement and speech coding are used.

Most of the simulation is implemented in MATLAB but the implementation of many standards is provided in C/C++ by the ITU-T. An interface is used to enable the use of these standards from within MATLAB.

When any standard is used, a set of conformance tests are executed to ensure the correct use of the standard. For comparison purposes several statistical approaches are used and visualisation of the results is provided when possible.

Careful selection of the terminology is used and differentiation between different terms used to describe the quality is clearly expressed. A qualifier is added to the terms used to make sure of no vagueness in the meaning of the term.

A list of acronyms is also provided for easy reference when needed and abbreviations are written in full at the beginning of each chapter, afterward the abbreviation is used in the rest of the chapter. If the abbreviation appears in a later chapter, the full name is given again then only the abbreviation is used.

1.4 Contribution of the Thesis

It is identified that the E-model is the most appropriate method for monitoring the speech quality in VoIP networks due to its non-intrusive nature. However the E-model suffers from several disadvantages hinders its applicability for the continuously changing world of communication.

To provide the best possible solution to the problem of measuring the speech quality in VoIP networks, the thesis makes the following contributions to the knowledge:

1. **E-model Extension Based on PESQ:** Based on the intrusive method for measuring the speech quality defined in PESQ method, the E-model is extended so that it does not depend on the time-consuming, expensive, hard to conduct subjective tests to calibrate its parameters. The extension is provided using 3 methods: linear regression, non linear regression and Artificial Neural Network (ANN). The accuracy of these 3 methods is compared with each other and with the original E-model calibrated using the subjective tests.

2. **Define a Correction Formula between the E-model and PESQ:** During the extension of the E-model to avoid the subjective tests, it was found that the E-model estimation and PESQ measurement do not perfectly map to each other, as such a correction formula is proposed to correct such deviation and the effectiveness of this formula in reducing the gap is tested.

3. **Improve the E-model Through Loss Classification:** The E-model is modified so that it gives a closer estimation of the speech quality as the measurement provided by the intrusive based method, PESQ. In doing so, packet loss in the E-model previously dealt with as overall loss, is categorised into

two classes: Voiced or Unvoiced loss with more emphasis on perception of different types of loss is drawn. The accuracy of the new model is evaluated by comparing it with the more accurate PESQ method.

4. **Combine the E-model Extension with the Voicing Classification:** The method for extending the E-model through PESQ is combined with loss voicing classification to provide a solution for measuring speech quality in VoIP networks accurately, non-intrusively, and without the need for the expensive subjective tests.

Based on the above contributions the modified E-model is applicable to estimating the speech quality in VoIP networks non-intrusively and accurately. Also, the applicability of the E-model or the modified E-model in Service Oriented Computing (SOC) community is illustrated.

1.5 Thesis Structure and Reading Guide

This thesis is organised into 9 chapters as follows:

Chapter 2. Describes different advantages, applications, and protocols associated with VoIP technology. Different challenges that face the technology and some of the proposed solutions are also explained. As speech coding is very important in this technology, speech production and coding is also explained in this chapter.

Chapter 3. Explains different solutions proposed to ensure certain quality level in IP networks and describes different available solutions provided for measuring the speech quality.

Chapter 4. Simulates some of the basic system components and presents the results obtained by the simulation. These materials are put into a separate chapter to make the reference to them in subsequent chapters easier and to avoid repetition of the same materials in subsequent chapters.

Chapter 5. Proposes an extension for the E-model based on PESQ to avoid the expensive and time-consuming subjective tests. Also the relation between the E-model estimation of the quality and PESQ measurement is investigated and a correction formula is proposed.

Chapter 6. Implements the extension proposed in the previous chapter in 3 different methods: linear regression, non linear regression and ANN. Compares between these methods and between their estimation and the E-model's estimation.

Chapter 7. Proposes a method to improve the E-model's accuracy by classifying packet loss into either Voiced or Unvoiced and compares the estimation with the measurement provided by the PESQ method.

Chapter 8. Combines the methods proposed in the previous chapters by presenting a method for estimating the speech quality accurately and non-intrusively without the need for the subjective tests.

Chapter 9. Summarises the work presented in this thesis, highlights the significance of the proposed contributions and discusses directions for possible future directions.

The reading order of the thesis is illustrated in Figure 1.1

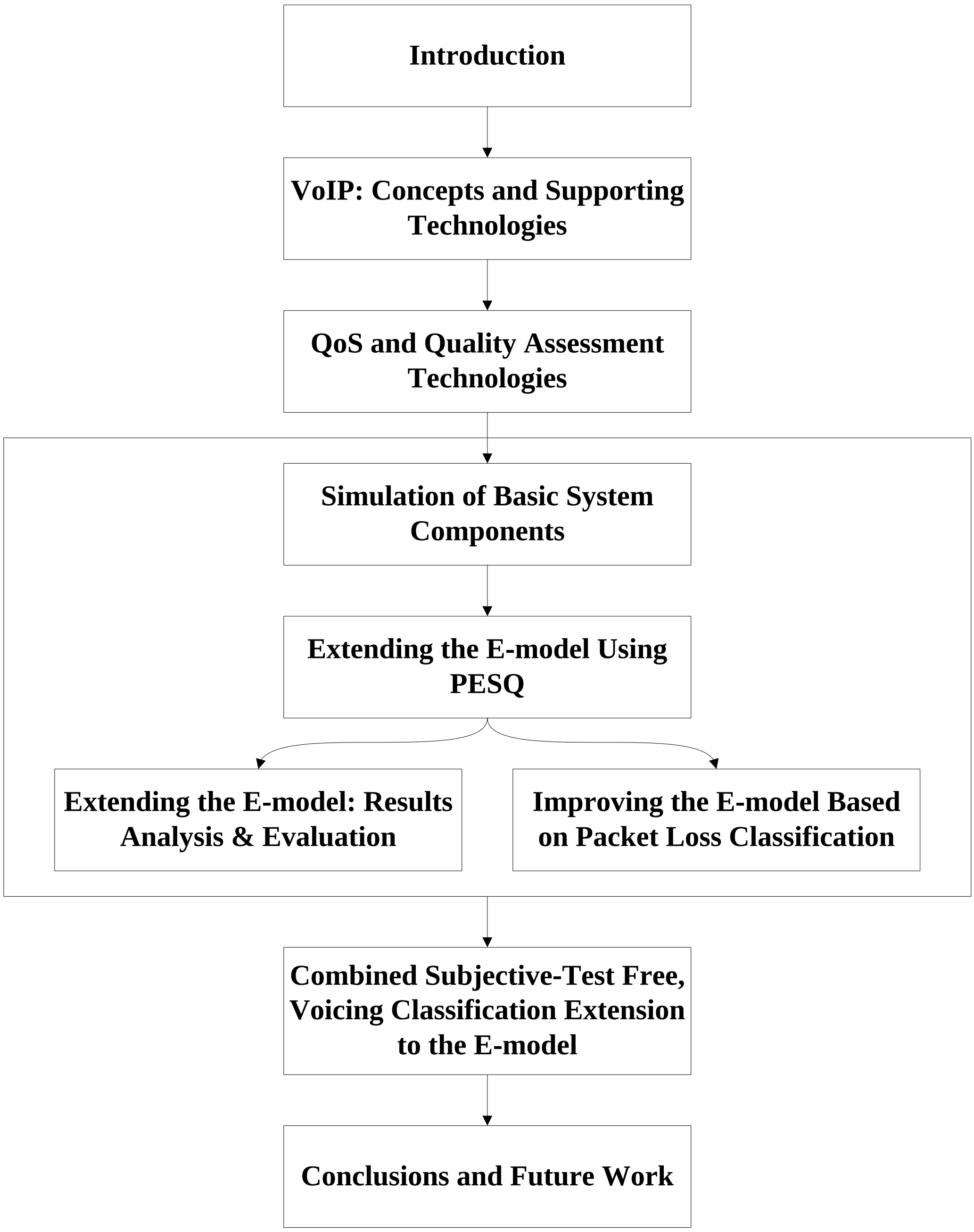

Figure 1.1: Reading's Outline

Chapter 2

VoIP: Concepts and Supporting Technologies

2.1 Introduction

Circuit Switching technology has been in use for long time by traditional Public Switched Telephone Network (PSTN) carriers for carrying voice traffic. In circuit switching network a dedicated channel or circuit is established between nodes and terminals before users may communicate. The established circuit cannot be used by other callers until the circuit is released, it remains unavailable to other users even when no actual communication is taking place.

Circuit switching builds a dedicated path from the sender to the receiver and that path is selected over the most efficient route. Accordingly, it is not necessary for a phone call, from the same sender to the same receiver, to take the same route every time a phone call is made. During call setup once the route is determined, that path or circuit stays fixed throughout the call and the necessary resources across the path are allocated to the phone call from the beginning to the end of the call, therefore circuit switching is carrying voice with high fidelity from source to destination [24, 132]. Circuit switching is like having a dedicated railroad track with only one train, the call, is permitted on the track at one time as illustrated in Figure 2.1.

Today's commercial telephone networks that based on Circuit Switching technology have a number of attractive features, including [24]:

- Availability: Availability of commercial telephone networks is 99.999 percent

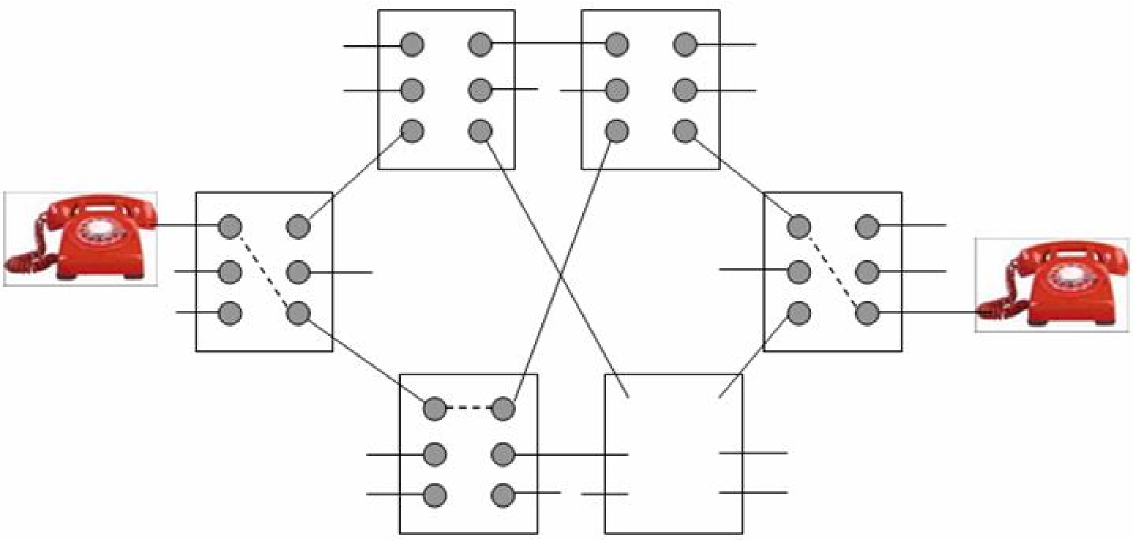

Figure 2.1: Circuit Switching

[24, 94]. This corresponds to a downtime of less than five minutes per year and this is known as five nines reliability.

- Capacity: Telephone networks also have the capability to support millions of subscribers and similar number of simultaneous calls.

- Fast Response: When one finishes dialling a number, the phone at the other end starts ringing within two to three seconds.

- High Quality: When someone answers the phone and once the conversation takes place, the speech quality is very high, without any perceptible echo, noticeable delay, or annoying noise on the line.

One alternative technology to circuit switching telephone networks for carrying voice traffic is to use data-centric packet switching networks such as Internet Protocol (IP) networks. In packet switching technology, no circuit is built from the sender to the receiver and packets are sent over the most effective route at time of sending that packet, consequently different packets may take different routes from the same sender to the same receiver as shown in Figure 2.2.

Given the fact that current circuit switching telephone networks are working in such a good way, why should anyone consider an alternative carrier? The answer lies in the following [24, 64, 210]:

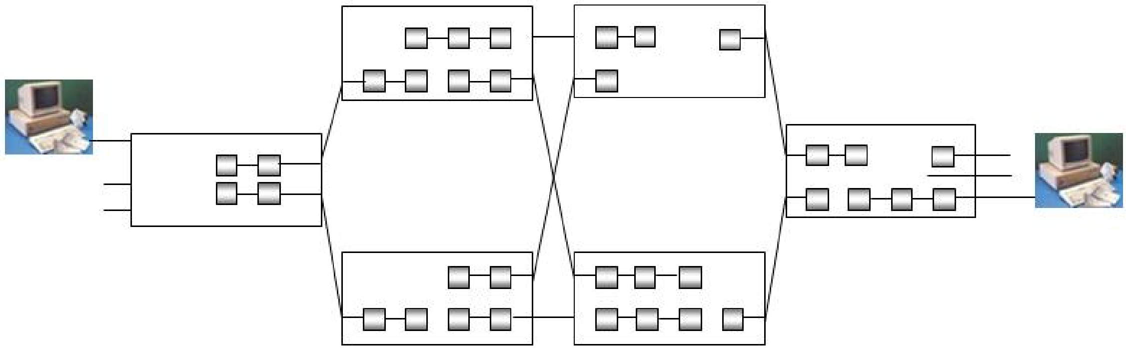

Figure 2.2: Packet Switching

- Circuit Switching networks are dedicated to carry voice. This dedication is its major strength and at the same time its major weakness because such dedication prevents it from doing anything else very well.

- The vast growth of interest of communication's applications worldwide has been accompanied with an increasingly growing interest in reducing the communication's costs for these applications.

- Integration of separate voice, fax and data resources using data-centric networks offers an opportunity for significant savings in expenses and innovation of new services.

The use of data-centric packet switching networks such as IP networks for carrying voice traffic is known as Voice over IP (VoIP). Some researchers refer to this technology by the names IP telephony or Internet Telephony. Although these terms can be used interchangeably, for the rest of this thesis the term VoIP shall be used.

VoIP is an active research area in the world of communication, as the high revenue achieved by the telecommunication companies is a motivation to develop solutions that transmit voice over other media rather than the traditional Circuit Switching network.

Several definitions have been proposed for VoIP. Some of these definitions include:

- The transport of voice traffic using Internet Protocol (IP) [24].

- Real time delivery of voice across networks using the Internet Protocols [132]

- Transmission of real-time telephone quality speech or voice signal-after digitisation and packetisation-over an Internet protocol (IP) based network-an Intranet or a VPN over the Internet-with or without sacrificing Plain Old Telephone Service (POTS)-like reliability, quality, and availability [95].

The above definitions lead to the same idea which is the use of IP networks to carry voice in addition to data rather than the common approach of using a dedicated telephone networks for carrying voice only and a dedicated data network for carrying data only. Such integration has many advantages and possible new application areas. In addition specific protocols are needed to manage voice stream in such data networks.

With such integration, voice should be digitised and packetised and carried as packets over packet switching networks rather than signals over circuit switching networks. Additionally when a user of VoIP network wants to interact with another user over the same IP network, all the communications goes through the data-centric IP network. When a VoIP user wants to interact with a normal phone user in the PSTN network, a gateway is needed to support interconnection between the call parties. The gateway should support translation functionality of the signalling protocols and media formats between the two networks [18].

The steps involved in carrying voice traffic over IP data networks from the source into the destination involve: digitising, compression, packetisation, transmission, depacketisation, decompression and playback as shown in Figure 2.3 [24, 53, 57, 165, 215].

1 Analogue to Digital Conversion: In this step the continuous analogue signal is sampled and each sample is represented by a number of bits.

2 Compression: In this step the samples of the previous step are compressed into a compact representation to reduce the bandwidth usage. Compression is done using a coding algorithm as explained in section 2.3.

3 Packetisation: Adding network protocols headers to allow the network determining the host node to deliver the voice packets to.

4 Transmission: Transmission over the IP network during which the packets traverse many domains and queued in a number of intermediate routers.

5 Depacketisation: Once the packet reaches its destination (if not lost during transmission), headers attached during step 3 are removed leaving the voice payload.

6 Decompression: This is a reverse step of compression where the voice contents are extracted for playout.

7 Playback: The decompressed speech is sent to the playout device.

In case a non-IP network node is involved in the communication, a gateway is placed between the IP network and the PSTN network to aid in signalling and media conversion.

Using data-centric networks such as packet-switching IP networks for transmitting voice as well as data seems lucrative solution as it provides promises of greater flexibility and advanced services than the traditional telephony with greater possibility for cost reduction in phone calls . The networks of the future will use IP network as the core transport network, and VoIP will become the main standard for third generation wireless networks. In fact these networks are seen as the long-term carriers for all types of traffic including voice and video. Even in the existing wireless networks it can be an alternative [15, 51, 62, 171].

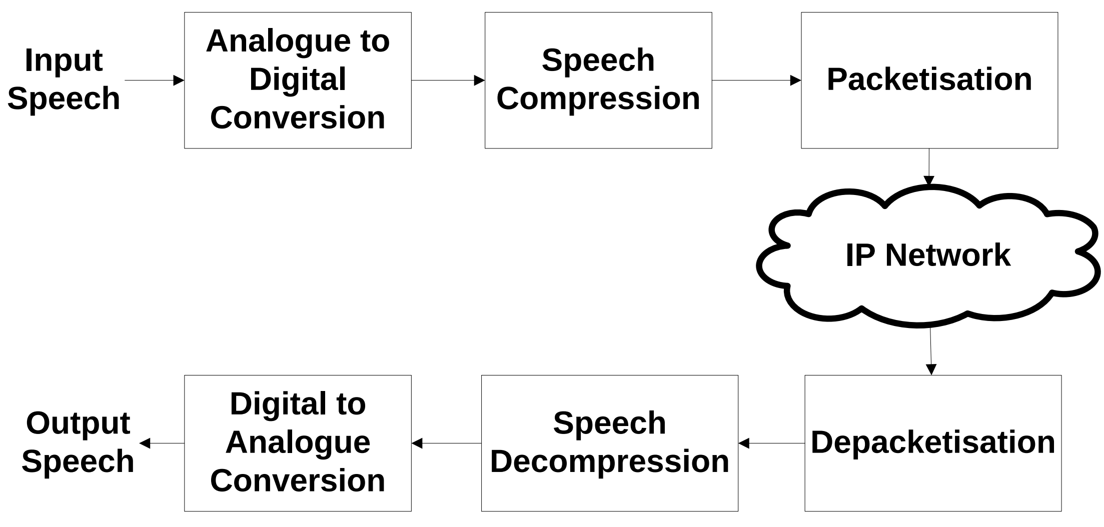

Figure 2.3: Voice over IP System

IP networks are characterised by being best effort networks with no guarantee of delivery as no circuit is established between the sender and the receiver. Also IP networks are designed originally to carry data rather than voice, the use of such networks for carrying voice is not straight forward as there are a number of technological issues that needs to be resolved in order to make data-centric networks suitable for carrying voice traffic as well as data traffic. These issues arise because of the time-varying characteristics (e.g. packet loss, delay, delay variation (jitter), sharing of resources) of IP networks. These characteristics which are normal to data traffic, cause serious deterioration to the real-time traffic. With the introduction of real-time application such as voice traffic in such networks, several parameters of these networks should be adjusted to make these networks suitable for voice traffic because no matter how low the price is, if the quality of voice over IP network is not capable of competing with the traditional telephony network, the customers are not expected to be attracted to the new solution. Table 2.1 lists some differences between traditional telephony and VoIP telephony [24, 132, 190].

Traditional Telephony	**VoIP Telephony**
Circuit-Switching Technology	Packet-Switching Technology
Synchronous Transmission - Low Channel Utilisation	Asynchronous Transmission - High Channel Utilisation
When congestion occurs, new calls will be blocked, but current calls maintain high quality	When congestion occurs, new calls may be blocked (admission control) or IP packets can be dropped which reduces quality for current calls
Standard Pulse Code Modulation voice-encoding scheme is used without compression which consumes 64 kbps	Voice compression encoding scheme is used which reduced bandwidth consumption considerably
Short end-to-end transfer delay and limited delay variation	Long end-to-end transfer delay and significant delay variation (jitter)
Guaranteed good voice quality	Voice quality cannot be guaranteed and greatly affected by network conditions
High operational costs with separate data and voice networks	Reduce operational costs due to the integration of two networks

Table 2.1: PSTN vs. VoIP

2.2 VoIP Advantages, Applications, Protocols and Challenges

Section 2.2.1 discusses the advantages of VoIP technology while section 2.2.2 gives an overview of some applications associated with VoIP. The main protocols created or used in this technology are reviewed in section 2.2.3. The main challenges that face the development of VoIP telephony services due to the nature of such networks are discussed in section 2.2.4.

2.2.1 VoIP Advantages

Using data-centric networks such as packet-switching IP networks for transmitting voice as well as data seems an attractive solution as it provides promises of greater flexibility and advanced services than the traditional telephony with greater possibility for cost reduction in phone calls. The advantages include: lower equipment cost, integration of voice and data applications, lower bandwidth requirements, lower operating and management expenses, widespread availability of IP, and other advantages [24, 62].

Lower Equipment Cost

Circuit switching systems use proprietary hardware, proprietary operating systems, and applications running on proprietary software. This closed nature of the circuit switching systems makes operators very careful in choosing the vendors for their systems because when they choose their vendors, they have to choose that vendor for the whole system due to the lack of inter-operability between products from different vendors and this result in monolithic architecture.

On the other hand in the IP world, the operating system is less tightly coupled to the hardware and much of the hardware is standardised. This offers greater choice between different vendors and allows customers to pick those companies that are the best in different areas to create the most advantageous solution in all aspects. Furthermore, IP networks tends to use a distributed client-server model rather than the monolithic systems, which means that it is easy to start small and grow as demand dictates.

Integration of Voice and Data

IP was originally designed to carry data ranging from e-mail to web browsing to e-commerce. When this capability is combined with the transport of voice, many new and advanced services can be offered. One example is a user browsing a web page and entering his/her location, and then when the user is pressing a button to talk to a customer representative, the user will be directed to the most appropriate customer representative based on the user's location [24, 112, 130, 150, 204, 210].

Lower Bandwidth Requirements

International Telecommunication Union - Telecommunication Standardisation Sector (ITU-T) standard G.711 [66] is used in traditional telephony for coding voice according to Pulse Code Modulation (PCM). While G.711 requires 64 kbps, more efficient and sophisticated voice coding algorithms exist, which enable speech to be transmitted at different rates ranging from 5.3 kbps to 32 kbps as will be discussed in section 2.3. These more advanced coding schemes could be applied to the current telephone system but this would require that these coding schemes be implemented in practically every telephone switch in the world. Advanced techniques such as silence suppression (discontinuous transmission during the periods of silence) could also be applied easily in IP telephony. Silence suppression can lead to a big saving in transmission requirements as statistics show that 50% of a conversation is silence [24, 27, 56, 215].

Lower Operating and Management Expenses

Today's enterprises need to manage two types of networks: the telephone networks, and the data (IP) networks. Implementing VoIP simplify the life for the enterprises and to their network administrators by merging these two networks into one unified network that can carry both voice and data. This unification results in a lower operational and technical effort, consequently this leads to a reduction in expenses [15, 24, 50, 215].

Widespread Availability of IP

IP networks are widely spread nowadays. The wide spread of these networks helps in fast adoption and success of VoIP systems, due to the fact that minimal change and cost results from such adoption, consequently this has a very cheap financial implications. This also takes the reachability of users to a new level [24, 50, 215].

Other Advantages of VoIP

Other advantages of VoIP networks over the traditional telephony include: adjustable voice quality according to the available bandwidth, possible application of security techniques such as encryption of messages, provision of a graphical user interface, and others [24, 27, 132, 210].

2.2.2 VoIP Applications

VoIP can be used in many applications, including:

Call Centre Integration

Integration of one unified network for carrying both voice and data reduces the infrastructure and administration costs. This integration may also serve as a way of integrating the Internet and IP networks with the PSTN and cellular networks all together. Such integration can be used by companies with multi-branches to reduce costs such that all the traffic (voice and data) is sent through the same network. Additionally by integrating voice and data over one network, various types of messages can be retrieved on a single device which enable applications such as an IP-based call centre [127, 155].

Directory Services over Telephones

Ordinary telephones can be used as Internet access devices, also, directory services could be implemented by submitting a name and receiving a reply. This serves as a search facility within a phone book, where the user searches the phone book by providing the name s/he is looking for, then after the search process, the user will have the option if s/he wants to ring that number [7, 155].

IP Video Conferencing

Using VoIP, the cost of long distance calls has been greatly reduced. Additionally, video conferences could be held over IP networks which reduce the cost for running business [24, 28]. An example application for making long distance calls is Skype which is a tool that enables Skype users to speak to other Skype users for free, call traditional telephone numbers for a fee (SkypeOut), receive calls from traditional phones for a fee (SkypeIn), and receive voicemail messages for a fee. It is estimated there are more than 100 million Skype subscribers around the world. [138, 177, 192].

Fax over IP

Facsimile services could be used over IP networks by converting the data into IP packets and send them over IP network which lead to a great saving especially for long distance destinations. Fax services are more delay tolerable than voice services as explained later in this chapter.

Radio/ TV Broadcasting

Using IP networks to carry multimedia signals enabled live broadcasting of radio/TV channels and this help in faster and wider transmission of these channels. The reachability of TV stations is now taken to a new level.

2.2.3 VoIP Protocols

VoIP traffic is carried over the data-centric network which uses Internet Protocol (IP), therefore this section will discuss IP protocol and its relation with VoIP. VoIP-specific protocols will also be discussed.

Voice and TCP/IP Protocol Suite

Open System Interconnection (OSI) model as developed by the International Organisation for Standardisation (ISO) has seven layers while Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite implements five-layer protocol as shown in Table 2.2.

In the TCP/IP protocol either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) is used in the transport layer. TCP offers reliability, ensures ordered delivery without loss of data, while UDP is an inherently unreliable protocol with no mechanism for ensuring ordered delivery and with no mechanism to retransmit packets in case of loss of packets.

Surprisingly, when voice is to be carried over IP, it is the unreliable UDP protocol that is used rather than TCP protocol. The reason is that VoIP is a delay sensitive

	OSI Model	TCP/IP Protocol suite
Layer 7	Application	Applications and Services
Layer 6	Presentation	
Layer 5	Session	
Layer 4	Transport	TCP or UDP
Layer 3	Network	IP
Layer 2	Data Link	Data Link
Layer 1	Physical	Physical

Table 2.2: OSI Model Vs. TCP/IP Protocol suite

application and TCP introduces delay to setup a connection, and acknowledgment. Even worse, in case of lost packets, TCP will cause retransmission and thereby introduces even more delay. By the time the retransmitted packets reach the listener, their value will be lost. However TCP could be a better alternative for fax, because fax is not as sensitive to delay as voice, lost packets can be retransmitted.

One problem remains with the usage of UDP namely, the order of delivery, as UDP offers no mechanism for ordering packets and as packets traversing IP network could experience different delay times due to them taking different routes as discussed in section 2.2.4, consequently they may arrive out-of-order. To solve this problem, Real-time Transport Protocol (RTP) is used above UDP protocol in the protocol stack.

Real-time Transport Protocol

Real-time Transport Protocol (RTP) is used above UDP protocol to provide additional functionality such as providing a sequence number to detect lost packets and present the packets in the correct order as packets may arrive out-of-order due to the nature of the IP networks. In addition, a time stamp is included in the RTP protocol. This time stamp ensures synchronisation on play-out, and accurate calculation of delay and jitter (described in section 2.2.4). RTP has a companion protocol, RTP Control Protocol (RTCP). RTCP provides statistical information about the session and includes such information as the number of lost packet to provide feedback regarding the quality of the session and aid in quality estimation.

Based on the above discussion the protocol stack usually used for VoIP sessions is RTP/UDP/IP. The number of bytes used for the above protocols is 40 bytes with 12, 8, and 20 bytes used for RTP, UDP and IP respectively. This makes the size of the headers large in comparison with the size of the voice frame, i.e. high overhead. To overcome this problem and reduce the protocol overhead, header compression techniques could be used or multiple voice frames can be packed together before being encapsulated as one RTP packet. A trade-off exists between the number of voice frames that could be packed and the amount of overhead delay due to large packet size. One of the header compression techniques is compressed Real-time Transport Protocol (cRTP) which can compress the 40-byte RTP/UDP/IP header to the order of 2 to 4 bytes most of the time. Such compression reduces the transmission delay and at the same time incurs better bandwidth utilisation, consequently this minimises the effect of congested network on voice quality. Another header compression scheme that provides a high degree of compression and is robust for cellular usage is Robust Checksum-based header Compression (ROCCO) [53, 103, 132, 205].

One other alternative to header compression is to multiplex multiple voice packets into one IP packet at the gateway. In this approach the gateway identify VoIP streams going to the same destination and then multiplex them into one IP packet. At the receiving gateway, VoIP packets are demupltiplexed again before delivery to their final destinations. In this way bandwidth efficiency is achieved which allows more calls to be admitted [55, 63, 98, 172, 191].

Voice over IP Signalling Protocols

As described earlier voice packets in IP networks are transmitted using RTP/UDP/IP protocol stack. But how to control issues such as the start/end of the session, determination of the coding technique to be used, issues such as: authentication to make a call, etc. To achieve this, a specialised signalling protocol is needed.

In the market there are two competing protocols: the ITU-T H.323 Recommendation and the Internet Engineering Task Force (IETF) SIP Protocol. These protocols and other related protocols are described next.

H.323

H.323 is an ITU-T H series Recommendation and it serves as a standardised signalling protocol for generic packet-based multimedia communication. H.323 is the most widely deployed standard in VoIP networks today [24, 111, 132]. H.323 architecture involves four components:

- Terminal: Terminal is a H.323 endpoint, typically an end user that offers real-time communication and supports one or more voice compression/decompression algorithms known as coder/decoder (codec).
- Gateway (GW): Gateway is a H.323 endpoint that provides interface between H.323 network and other networks such as PSTN. GW provides translation functionality of the signalling protocols and media formats between the two networks.
- Multipoint Controller Unit (MCUs): Multipoint Controller (MC) is a H.323 endpoint that resides inside a Multipoint Controller Unit (MCU) and manages multipoint conferences. MC transmits the capabilities of the endpoints (e.g. the supported codecs) between the call's participants. In cooperation with MC there is Multipoint Processor (MP), which is controlled by the MC. MP performs functions that require heavy processing such as media translation (from one media format to another).
- Gatekeeper: Gatekeeper is an optional entity that controls set of terminals, gateways, and MCUs in what is known as Gatekeeper's zone.

H.323 Consists of a set of protocols:

- H.225.0: which is a two-part protocol and consists of the following protocols
  - Registration, Admission, and Status (RAS): RAS used between the gatekeeper and the endpoints in its zone to enable the gatekeeper to control its endpoints. RAS messages runs over UDP protocol.
  - Call-Signalling (Q.931): Q.931 is used between endpoints and it enables the **setup** and **release** of calls. Q.931 messages can be sent over either UDP or TCP.
- H.245: H.245 runs over TCP and is used to manage the media streams between the call participants such as agreement on the media format to be used, agreement on the bandwidth to be used.

- H.450.X Services: H.450.x series defines H.323 supplementary services that are similar to the features available in the PSTN such as call transfer, call diversion, and call hold.

All H.323 messages are specified using the joint ISO and ITU-T Abstract Syntax Notation 1 (ASN.1) standard. As H.323 is a complex protocol with a lot of overhead, consequently IETF developed a new protocol for VoIP signaling called Session Initiation Protocol (SIP).

Session Initiation Protocol(SIP)

SIP is an IETF standard that offers a powerful alternative to H.323. SIP is an application-layer protocol claimed to be more flexible, simpler, and easier to implement than H.323. SIP is part of the IETF multimedia data and control architecture; therefore it can be used with other IETF's protocols such as Session Announcement Protocol (SAP), Session Description Protocol (SDP), Real Time Streaming Protocol (RTSP), and Resource Reservation Protocol (RSVP) [24, 65, 154, 156, 212, 214].

When SIP is used in conjunction with SDP, SIP handles the communication between session participants while it relies on SDP for exchanging media capabilities (describing the session). SAP is used for advertising multimedia sessions and conferences by multicasting the session description (defined by SDP) to a multicast address and port (default port number is 9875). The announcement has the same scope as the session it is announcing [24, 132, 154, 155, 212, 214].

RTSP allows clients to have control over media servers by instructing commands to record and playback multimedia sessions including functions such as seek, fast forward, rewind, and pause. A user can use SIP to invite media server to a multimedia session, and then use RTSP to control operation during the session [132, 155].

SIP does not care about the type of media to be exchanged or the type of transport. By doing this, SIP provides more flexibility than other protocols. This flexibility can be exploited to enable custom services and features. Also, SIP messages can contain optional fields that can carry user-specific information; these fields enable the creation of many intelligent and customised features.

SIP is based on a client-server model with clients (called user agents) sending requests and servers responding to these requests. A device that implements both user-agent client and user-agent server can be used as a phone as it can send and receive messages.

SIP messages are text-based messages which are very similar to HyperText Transfer Protocol (HTTP) which are more readable to the user and allows the reuse of the programs designed for HTTP such as HTTP parsers, but such text-based protocols consume more bandwidth in comparison with binary-based protocols.

Similar to H.323, SIP provides a mechanism for PSTN-interworking. A Network GateWay (NGW) is used between the SIP network and the PSTN network to provide translation functionality from the SIP protocol to the signalling protocol used in the PSTN such as Signalling System 7 (SS7). NGW also provides translation functionality of the media formats between the two networks.

In addition to the interface needed between SIP network and the PSTN network, another interface is needed between SIP network and the H.323 network. Although SIP is considered to be the standard of the future for VoIP networks nevertheless, H.323 was the first standard developed for signalling of VoIP networks and currently large number of products are deployed or are being deployed using H.323 protocol, therefore, an interface between the two types of network is needed. A gateway is used between the SIP's network and the H.323's network to provide translation functionality between the SIP's messages and the H.323's messages. In the SIP side, the gateway appears as a user-agent client or server. In the H.323 side the gateway appears as a H.323 endpoint. This gateway is known as SIP-H.323 Interworking Function [132].

Comparison of H.323 and SIP

As H.323 was the first protocol developed for VoIP, consequently it has lot of deployments today, but SIP is becoming the choice of all new products being developed and it has adopted in Universal Mobile Telecommunications System (UMTS) [132], As H.323 has a large share in the market, these two competing protocols will coexist together for now. Table 2.3 lists some of the differences between H.323 and SIP. Technical comparison between SIP and H.323 can be found at [6, 95, 190].

H.323	SIP
Binary representation for its messages	Human-readable textual representation
Requires full backward compatibility	Does not require full backward compatibility
Complex signalling	Comparatively simpler
A lot of overhead - Hundreds of elements	Only 37 headers
Not very scalable due to complexity	Highly scalable due to simplicity
Large share of market	Backed by IETF

Table 2.3: Comparing H.323 with SIP

Gateway Control Protocols

Gateways are needed to provide interconnectivity between different types of networks. In H.323 networks, gateways are needed to connect to PSTN networks and to SIP networks. In SIP networks, the gateway provides connectivity to PSTN and to H.323 networks as well. These gateways are providing translation functionality of the signalling and media format between the two networks. It is not necessary for the signalling to take the same path as the media and also, the bandwidth needed for signalling is tiny in comparison with the bandwidth needed for the media. Taking these factors into consideration, the logical separation between the two functionalities of the gateway can be noticed. This logical separation can be turned into physical one. This physical separation of the gateways into Media Gateway Controller (MGC) or call-agent and Media Gateway (MG) will allow many advantages over the monolithic architecture of the gateway. Among the advantages are: scalability, distribution of media conversion, and quicker addition of new features.

These advantages of the physical separation need also a standardised protocol to allow the MGC that handles signalling conversion to control MG that handles media conversion. MGC controls the MG in a master-slave relationship while another protocol such as SIP can be used as the call-signalling protocol in the VoIP network. This architecture is known as softswitch architecture. The name softswitch is used because the switching functions originally handled by large monolithic system in circuit-switched networks are instead handled by software systems in this configuration. The most common protocols for MGC-MG relation is Media Gateway Control Protocol (MGCP) and MEGACO/H.248. MEGACO/H.248 can be considered as a successor of the MGCP.

MGCP: In Media Gateway Control Protocol (MGCP), the master MGC or call-agent gives instructions to the slave MG to perform. The commands generated from the call-agent are generally related to the establishment and teardown of connections from one side of the MG to the other side (e.g from the SIP network to the PSTN network). Each command can come with one or more parameters related to that command. Also, a set of responses are defined for MGCP messages. MGCP's commands, parameters, and responses use US-ASCII character set. For a complete list of MGCP messages, parameters, and responses one can refer to [24].

MEGACO/H.248: It is known as MEGACO within IETF and H.248 within ITU-T and is a result of cooperation between the two organisations. Megaco/H.248 is very similar to MGCP in that it has MGC and MGs with a master-slave relationship. In MEGACO, MGC sends TransactionRequest and the MG replies by a TransactionReply. A TransactionRequest comprises a number of commands and a TransactionReply comprises a corresponding number of responses. For a complete list of MEGACO commands, parameters, and responses one can refer to [24].

Protocols Summary

The main protocols used in VoIP networks are reviewed. Mainly the use of H.323, SIP, and MGCP/MEGACO Protocols was investigated. Other protocols are also in use, among those protocols are [138, 177, 192]:

- Remote Voice Protocol (RVP): RVP is MCK proprietary Communications' protocol for transporting telephony sessions over data networks. RVP is used primarily in MCK's product family.

- Skinny Client Control Protocol (SCCP): SCCP is a Cisco protocol used to connect Skinny clients to H.323 servers such as H.323 proxy.

- Skype: Skype is a proprietary protocol used to connect Skype clients. It is estimated there are more than 100 million Skype subscribers around the world. Skype users can speak to other Skype users for free, call traditional telephone numbers for a fee (SkypeOut), receive calls from traditional phones for a fee (SkypeIn), and receive voicemail messages for a fee.

2.2.4 Challenges of Voice Over IP

VoIP technology offers a number of attractive features and greater flexibility over traditional telephony networks. But as mentioned earlier IP networks are originally designed to carry data rather than voice and IP networks offer a hostile environment to voice traffic, therefore the most important issue in VoIP is ensuring high speech quality comparable to the quality of traditional telephony. To ensure good voice quality, a number of issues should be addressed, these issues vary from those that should be considered before the VoIP session (voice call) is even started to those should be considered during the VoIP session and still more issues arise even after the VoIP is finished. The focus will be on the challenges that occur during the VoIP session, those occur before or after the session are mentioned briefly next.

Challenges Before VoIP Session

Among the challenges that arise before the VoIP session is started or during call setup attempt are:

- **Availability of Dial Tone:** Users who are familiar with circuit switching systems are used to hear a dial tone, because it gives the user the impression that the called host is ready to start the session (voice call). Circuit Switching users expect the availability of the dial tone in the VoIP networks as well.

- **Availability of Resources to Start the Call:** In case a managed IP network is used, some resources should be available to proceed with the call. Such resources include the availability of a mechanism to determine the called user's location and determining the best route to reach the user. Mechanisms are needed to determine whether to accept a call request if it is possible to allocate the required resources (bandwidth) and maintain the given QoS target for all existing calls, or otherwise to reject the call. Such mechanism is known as Call Admission Control (CAC) [95] (For further information see section 3.1). Additionally if the call is admitted, new mechanisms are needed to reserve the needed resources to make a call, such resources are vital for the provisioning of good voice quality. One such mechanism is RSVP protocol which is discussed with other similar mechanisms in section 3.1.

- **Total Amount of Time Required to Set up a Call:** This call setup time measured from the moment the last digit of the called number is entered to the moment the ring back tone is heard. This time should be bounded to certain limits comparable with those of traditional telephony.

Challenges After VoIP Session

Among the challenges arise after the VoIP session is finished, for commercial, technical and may be legal reasons as well, is maintaining detailed call records for the purpose of billing, testing, diagnosis, network capacity planning and traffic engineering [95]. The major issue in accounting is the selection of the suitable billing model. A number of billing models have been proposed [7]:

- **Time-based:** The billing is metered by flow duration, time-of-day, day-of week with possible flat price regardless of the destination or the offered quality [17, 24].

- **Distance-based:** The billing is based on the distance between the caller and the called users. The current IP protocol (IPv4) is not designed to support region-based IP. The newer version of IP protocol (IPv6) is designed in a manner that the user is assigned an IP address based on his/her geographical location [44].

- **Quality of Service-based:** The billing is based on the service quality offered.

- **Congestion-Based:** The billing depends on the congestion at the gateway which is measured as the percentage of trunks in use [17].

A hybrid of the above accounting models is expected to be dominating in the future. IETF's Authentication, Authorisation and Accounting (AAA) can also be used for billing of users based on their accounting information. Additionally, AAA can be used for QoS purposes. To provide end-to-end QoS for a phone call, authentication has to be provided between devices (e.g. SIP phones) to service provider, end user to service provider. Additionally, users need to be authenticated when they try to reserve resources. The Open Settlement Protocol (OSP) can be used for authorisation and accounting by service providers. IP Security (IPSec) is used for IP telephony gateway authorisation. The policies for admission control can use Common Open Policy Service (COPS) for policy administration.

Challenges During VoIP Session: Packet Loss

During transmission of packets in IP networks, speech stream may suffer from packet loss due to different reasons. These include: excessive bit errors, or congestion in the IP network, or simply excessive delay that causes the receiver to ignore the late frames in the decoding process [135], these reasons are illustrated in Figure 2.4.

In IP networks (especially in public -nonmanaged- networks), queues might experience overflow in network nodes between the sender and the receiver, resulting in loss of packets. Retransmission mechanisms can be applied in case of lost data packets that use the reliable TCP protocol to compensate the lost packets as discussed in section 2.2.3. These retransmission mechanisms cannot be applied to real-time applications (such as voice) because the time needed to detect the lost packets and retransmit them is long enough to the degree that lost packets become useless for decoding in such real-time application [24].

As the problem of packet loss is inevitable in IP networks, appropriate mechanisms are required to ensure packet loss is eliminated or reduced from the origin and if presented, the bad effect of that loss is minimised. For example some complex coding techniques deal with the problem of lost packets through what is known as Packet Loss Concealment (PLC) algorithms. Several PLC algorithms exist today and they range from simple algorithms to complex ones.

Another approach is to reserve resources across the path from the sender to the receiver. This is difficult and very expensive proposition as it requires changes to all routers across the network, additionally it is inapplicable in the non-managed

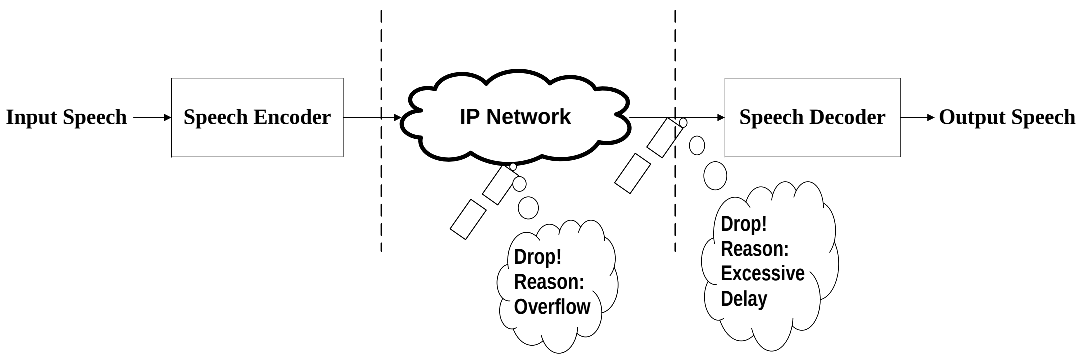

Figure 2.4: Reasons of packet loss of real-time applications in IP networks

networks such as the Internet.

Challenges During VoIP Session: Delay

Voice as a real-time application is extremely intolerable of delay. Mouth to Ear (M2E) delay is the delay measured from the moment a noticeable voice signal appears at the sending end (speaker's mouth) of a connection to the moment the same voice signal appears at the receiving end (listener's ear). ITU-T Recommendation G.114 [81, 82] states that the M2E delay should be 150 ms or even less for national calls in the range of 5000 km or less because users expect such calls to be completely delay-transparent. For international calls delay between 150-400 ms would be acceptable, while any delay longer than 400 ms is not acceptable for general network planning. In contrast to delay requirements for voice traffic, delay requirements for data packets are much less restricted. It hardly matters if sending data such as sending an e-mail took 3 seconds or 3 minutes, provided it arrives correctly [139].

Having 150 ms or more delay may incur several problems. One is talker overlap (double talk), which is the same as when one talker is stepping on the other talker's speech thinking the speaker does finish his/her talk due to the large amount of delay. This problem starts to appear when the one-way delay is above 150 ms [24, 95]. But further experiments performed by James et al. in [88] showed that delay between 150 and 200 ms will not cause a potential problem. Nachiappan, N. and F. Sjoqvist [132] gone even further to state that 250 ms delay will not cause a major degradation in voice quality. But as a preventive procedure, 150 ms is considered as a guideline for the maximum tolerable delay for regional calls.

Another problem associated with the delay is echo. Echo can be defined as a second and delayed version of what just came out of one's mouth [88]. Echo is caused by the signal reflection of the speaker's voice from the far-end back into the speaker's ear. This reflection becomes annoying when the delay is greater than 50 ms.

In conventional circuit switching networks the problem of echo is present, but because of the small round-trip delay, therefore, the speaker can not differentiate the echo from the original speech. In VoIP networks because the delay may be significant, echo-cancellation techniques should be in place and ITU standard G.165 gives design guidelines and defines the performance requirements of the echo cancellers [69]. To summarise, delay should be 150 ms or even less but if it exceeds 50 ms, echo cancellation techniques should be in place as well.

To understand the delay and reduce its effect on the quality, different components compromising the transmission delay of the voice signal should be analysed and these components are:

- **Voice coding (algorithmic) delay and voice accumulation delay:** The speech coder needs to collect speech data in order to process a packet of encoded speech; this delay depends on the type of speech coder used and consists of the following components [23]:
  - Analogue to Digital conversion.
  - Accumulation delay.
  - Algorithmic (look-ahead) delay
  - Processing delay which include error-correction mechanism.
- **Voice framing delay:** Before sending packets in IP network RTP/UDP/IP headers are attached to the packets as explained in section 2.2.3.
- **Transmission Delay:** Packet transmission over the physical medium is another source of delay and it depends on:
  - Buffer sizes at the ingress (network traffic that originates from outside of the network's routers and proceeds toward a destination inside of the network [99]) and egress (network traffic that begins inside of a network and proceeds through its routers to a destination somewhere outside of the network [99]) networks. These depend on link transmission capacity.
  - Packet propagation time. This depends on the physical length of the transmission link.
  - Bandwidth of the link.
  - Packet storing and header processing delays at the intermediate links.

  This delay is difficult to quantify. A popular estimate of 10 microseconds/mile or 6 microseconds/km [81] is widely used.

- **Jitter Delay:** Buffers are used at the receiver side to compensate the effect of jitter (described in the next section). These buffers add to the overall delay and it can be a significant part of the overall delay [24, 132].

Challenges During VoIP Session: Jitter

As mentioned earlier in section 2.1, circuit switching networks builds an open, dedicated path from the sender to the receiver for the duration of the conversation. Thus, all speech follows the same path using dedicated resources, experiencing no or very small variation in delay.

On the other hand, packets in IP networks may need different times to reach their destination. The differences in the time needed may result from two reasons. First, packets can take different routes from the sender to the receiver (this is a normal behaviour of the IP Protocol) and consequently packets can experience different delay times. Second, packets may experience different queuing times even if they took the same route [133, 178, 179].

If the variation of delay in arrival time between successive packets-also known as jitter- keeps changing, then it is difficult to adjust the delay for smooth and natural play-out of the speech signal. To minimise the effect of jitter, buffers are used to collect the speech packets before play out, so that speech can be played-out to the receiver in a steady fashion. Packets that did not arrive before the end of buffering time are considered lost. Jitter buffer is illustrated in Figure 2.5.

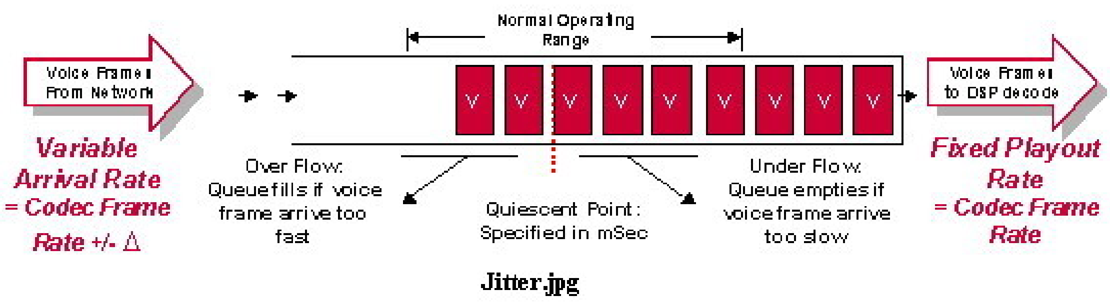

Figure 2.5: Jitter Buffer [23]

The use of jitter buffers adds to the delay, there are two conflicting goals of minimising delay by reducing the buffering time and minimising the packet loss by increasing the buffering time, and the trade off between them is decided upon the VoIP network design. In networks where delay is dominant, buffering time is reduced while in networks where packet loss is more apparent, buffering time is increased so that the effect of packet loss is reduced. Several studies have been going on to improve the trade-off between buffering delay and packet loss by adjusting the buffering time adaptively to minimise the packet loss with an acceptable level of delay [23, 92, 102, 133, 146, 178, 179].

While conventional jitter buffer playout algorithms modify the play out time during the silence period, Ramjee et al. [146] and Aziz et al. [159] propose quick modification to buffering delay as a quick response to delay spikes.

James et al. [88] and Tseng and et al. [178, 179] introduce two factors to determine the amount of buffering time: the type of codec, and the mode of communication which could be either interactive or non-interactive. For loss-sensitive codec, large buffering time could be chosen, and for delay-sensitive codec, small buffering time could be chosen. For non-interactive communication such as radio broadcasting, in order to improve the voice quality, buffering time is increased because the listener is not aware of such delay.

Challenges During VoIP Session: Other Challenges

In addition to the above challenges, there are other challenges need to be handled, including:

- **Ease of Use:** Users who are familiar with the circuit switching systems are expecting the same ease of use and the same set of features in the new VoIP system as these of circuit switching systems. Such services include call-forwarding, call hold, call-muting, toll-free charges, multi-party call, and user mobility as what is exist in mobile networks. Several efforts have been going on to implement such services in VoIP networks [49, 52, 93, 108, 156, 210].

- **Reliability (Availability):** The five nines (99.999 %) availability requirement mentioned earlier must be met if VoIP is to be a commercial challenge to the telephone network. To meet such condition, then redundancy of major system components should be applied and balance should be made between network cost and network quality.

- **Scalability:** It should be possible for the network to handle millions of simultaneous calls. It is easy with VoIP to start on a small scale and then expand as needs dictates.

- **Security:** Telephone calls can be secured in VoIP networks by making use of the services available in the TCP/IP environment. Among the issues that need attention in relation with VoIP security are [7, 187]:

  - **Authentication of users:** Make sure that users are really who they say they are.
  - **Integrity:** Validate that the data is indeed is an unchanged representation of the data.
  - **Privacy:** Hiding the data from eavesdroppers.
  - **Non-Repudiation:** Protect against someone falsely denying that they had participated in a call.
  - **Protection against Denial of Service (DoS):** DoS attacks flood the network, thereby preventing legitimate network traffic.
  - **Protection against several types of attacks such as**:
    - **Snooping:** In snooping an attacker tries to gain information on users' identifiers, services, media, and network topology.
    - **Modification:** Attackers in IP networks try to intercept the signalling path and modify the messages.
    - **Spoofing:** Attackers in IP networks impersonate the identity to gain some information s/he is not allowed/ unauthorised normally to get.

  To combat security threats, three categories of techniques have been developed based on their functionality [187]:

  - **Security Enabling:** Such as Public Key Infrastructure (PKI) aims at ensuring that messages cannot be intercepted or read by anyone other than the intended person and guaranteeing the authenticity of a message.
  - **Security Protection:** Such as firewall focus on protecting from external threats.
  - **Security Violation Detection Technique:** Such as Intrusion Detection System (IDS). IDSs concentrate on monitoring the events in a computer system or network and analysing them for signs of intrusion. There are two methods of intrusion detection:

  * **Anomaly Detection:** Where normal system behaviour is maintained and any divergence from that behaviour will produce an alarm. This system may produce false alarms as a result of users (non-attackers) inconsistently doing their job.
  * **Misuse Detection:** Where the system uses information about well-known vulnerabilities and attacks and compares them with the current system activities, upon a match is found, an alarm is produced. This system should maintain information about large number of possible attacks and may be hard to maintain.

  IDSs could also be classified as host-based (installed on a single host) or network-based (consists of a collection of agent applications strategically placed within a network). The characteristics of VoIP application such as time-sensitivity make it important to have VoIP-specific IDS. These IDS must be time-sensitive, scalable, and have VoIP-protocol decoding ability.

- **Electricity Consumption:** Deploying VoIP requires much more electricity than the current circuit switching system due to the power consumption of the additional communication and network equipments needed in VoIP network. Consequently when an enterprise decide to move from traditional telephony to VoIP, a feasibility study should be performed to determine if such movement has significant benefit in terms of cost reduction while taking into consideration the new expenses due to the electricity consumption of VoIP telephony equipments [21, 122].

In this section the advantages, applications, protocols and challenges of VoIP are discussed. As speech coding plays a vital role in VoIP technology, the next section discusses speech coding technology, philosophies and standards.

2.3 Speech Coding Technology

With the vast growth of interest in communication's applications world wide, such applications require that the speech signal is in digital format so that it can be processed, stored, or transmitted under software control. This digital representation offers ease of regeneration and signalling, flexibility, opportunities for encryption, and integration of voice and data applications. When such a digital representation of speech is uncompressed, it requires high data rate and high requirements for storage and high bandwidth for transmission. With the limitation of bandwidth and especially in the physical spectrum for wireless services a compact representation is required to optimise bandwidth efficiency [162].

Speech coding is the field concerned with obtaining compact digital representation of voice signals for the purpose of efficient transmission and storage which is something can be achieved through the use of Digital Signal Processing (DSP) techniques and recent advances in digital hardware [100, 162].

The objective of speech coding is to represent the speech signal with a minimum number of bits while maintaining its perceptual quality. The stages of speech coding involve sampling the analogue signal and amplitude quantisation or representation. Low-bit Rate (LBR) codecs are able to code speech using very low bit rate requirements, in these codecs speech information is coded into parameters which are then coded for transmission. Any corruption or loss of these parameters often results in severely annoying distortion which is propagated to the time even after the loss or corruption due the dependency of these codecs [100].

Speech production is described in section 2.3.1. Section 2.3.2 describes some speech coding principles. Some of the speech coding design issues are discussed in section 2.3.3. Section 2.3.4 describes the main categories of speech coding while section 2.3.5 describes some speech coding standards.

2.3.1 Speech Production

Speech is produced by a cooperation of lungs, glottis (with vocal cords) and articulation tract (mouth and nose cavity). The vocal tract extends from the opening in the vocal cords (the glottis) to the mouth. When the lungs press the air through the vocal cords and along the vocal tract, the vocal cords vibrate and interrupt the air stream and produce a quasi-periodic pressure wave. The pressure impulses are commonly called pitch impulses and the frequency of the pressure signal is the pitch frequency or fundamental frequency as shown in Figure 2.6. When we speak with a constant pitch frequency, the speech sounds monotonous but in normal cases the pitch frequency varies slowly as depicted Figure 2.7 [43, 200].

The pitch impulses stimulate the air in the mouth and for certain sounds (nasals) also the nasal cavity. As the shape of the vocal tract vary with time, we are able to pronounce different sounds, and it introduces short-term correlations into the speech signal, and can be thought of as a short term filter with both cavities act as resonators with characteristic resonance frequencies, called formant frequencies. The frequencies of these formants are controlled by varying the shape of the tract for example by moving the position of the tongue. Speech sounds can be broken into three classes depending on their mode of excitation:

- **Voiced Sound:** This is produced when the vocal cords vibrate open and closed, interrupting the flow of air generated from the lungs to the vocal tract and producing quasi-periodic pulses of air as the excitation. The rate the vocal tract opens and closes gives the pitch of the sound which can be adjusted by varying the shape of the vocal cords, and the pressure of the air behind them. Voiced sounds show a high degree of periodicity at the pitch period, which is typically between 2 and 20 ms.

- **Unvoiced Sound:** These result when the excitation is a noise-like turbulence (can be modelled by a white noise generator) produced when the air is forced at high velocities through a constriction in the vocal tract while holding the glottis open and the vocal cords do not vibrate. Unvoiced sounds show little long-term periodicity, although short-term correlations due to the vocal tract are still present.

- **Plosive Sound:** These occur when a complete closure is made in the vocal tract while the air pressure is built up behind this closure and then released suddenly.

Some sounds cannot be considered to fall into any one of the three above classes, but are a mixture. For example voiced fricatives result when both vocal cord vibration and a constriction in the vocal tract are present. An important part of many

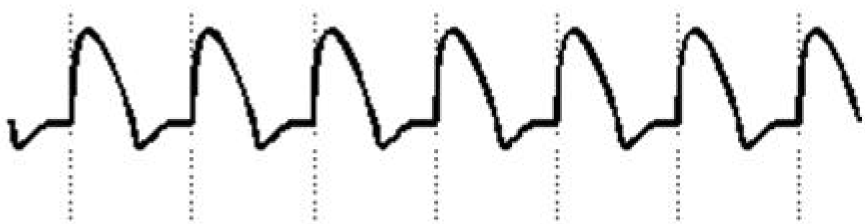

Figure 2.6: Typical impulse sequence

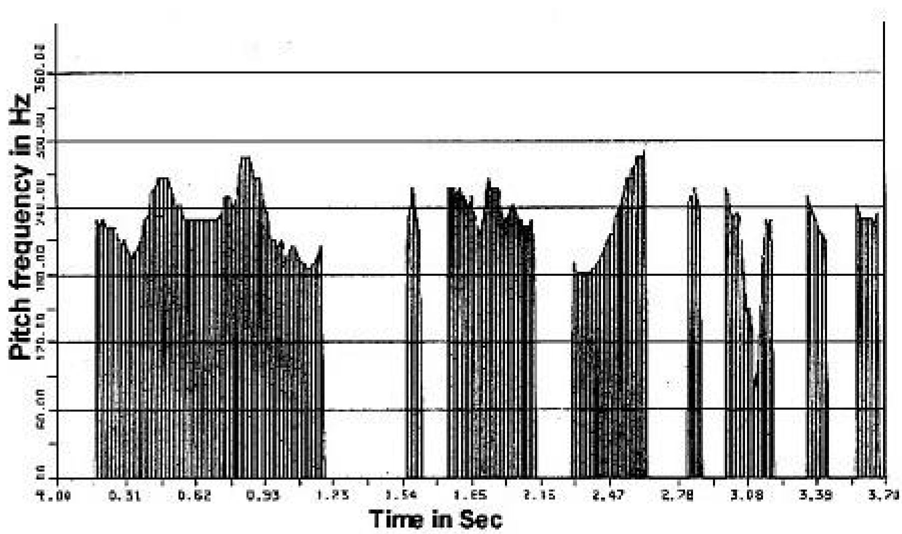

Figure 2.7: Variation of the pitch frequency

speech codecs is the modelling of the vocal tract as a short term filter. As the shape of the vocal tract and its mode of excitation change relatively slowly, so speech can be considered to be quasi-stationary over short periods of time which allow it to show high degree of predictability and the transfer function of its modelling filter needs to be updated only relatively infrequently (in the order of 20 ms) [200]. Speech coders attempt to exploit this predictability in order to reduce the data rate necessary for good quality voice transmission.

2.3.2 Speech Coding Principles

The term codec is an acronym used to reference for coder/decoder chip or compression/decompression. In either case, the term codec or compression refers to the computer program or computer-like chip to perform sampling; quantisation; and associated processing of the speech signal with the objective of digitising them to reduce the number of bytes consumed by large files and programs. On the other hand the decoder performs the reverse process to regenerate the analogue signal [95, 100, 162].

During the speech coding process several factors should be taken into consideration, such factors include: bit rate selection, good voice quality, knowledge of the characteristics of the source signal, and selection of the coding technique.

In speech coding there are two conflicting goals of minimising the bandwidth while maintaining good voice quality. Minimising the bandwidth is necessary for two reasons: first is to increase the revenue. Second, to provide services which are not possible otherwise. Minimising bandwidth is not enough if the speech still sounds synthetic, however, the relation is not linear as the bandwidth can be reduced considerably while maintaining acceptable voice quality. Bandwidth is easily quantified, but how voice quality can be quantified? Voice quality is subjective rather than objective, however methods exist for measuring voice quality in a subjective manner and in an objective manner as well [175]. Quality Assessment techniques are described in more details in section 3.2 as they are the main theme of this thesis.

The process of speech coding consists of representing the spoken signal into a sequence of bits. Speech coding involves sampling and amplitude quantisation. Different speech coders treat the speech signal in different ways resulting in different performance for each coder. However, regardless of the speech coder selected, it is necessary to sample the speech signal with a sampling rate of a minimum twice the maximum frequency of the source analogue signal according to Nyquist theorem. Human speech is in the frequency range of 300 to 3400 Hz, assuming maximum of 4000 Hz, usually a sampling rate of 8000 Hz is adequate. Any sampling less than the Nyquist Frequency will cause distortion to the signal, where the high frequencies overlap low frequencies. This distortion is called aliasing [60, 198].

After the sampling process is finished, the sampled values should be quantised (represented in binary). The quantisation can be direct or parametric, uniform or non-uniform. In direct quantisation the speech sample (or a vector of speech samples) itself is quantised while parametric quantisation involves binary representation of speech model and/or spectral parameters [162].

Sampling

In the sampling process, the continuous analogue speech signal is discretised by selecting some samples from the continuous speech signal. The sampled waveform can be represented by:

$$s(n) = s_a(nT_s), \qquad -\infty < n < \infty \tag{2.1}$$

where

s_a	The original analogue signal
n	An integer number
T_s	The sampling interval

According to the Nyquist theorem discussed earlier, T_s is determined according to the highest frequency of the input signal in such a way the sampling frequency equals to at least double the highest frequency [24, 60, 100, 128, 144, 198].

Quantisation

During the quantisation process, the amplitude of the continuous-amplitude analogue signal (digitised usually using 16 bits representation) is mapped into one of a finite set of discrete-amplitude signal values (quantisation levels). This quantisation can be done separately or for a set of samples. The former known as scalar quantisation and the later is known as vector quantisation.

Scalar Quantisation : In scalar quantisation, the digitised amplitude of the analogue signal is compared against a finite set of amplitudes, separated uniformly. or non uniformly, and the closest value of the set of amplitudes is selected to represent the sample. The distance between the finite set of amplitude values is called the step size which is usually represented by Δ (delta). Each amplitude level is represented by a symbol which is transmitted to the de-quantiser. When the same number of bits used to represent each quantisation level, the required bit rate will be:.

$$BitRate = B * F_s \tag{2.2}$$

where

B	The number of bits necessary to represent each sample.
F_s	The sampling frequency.

When non-uniform quantisation is used, the step size is finely quantised for frequently occurring amplitudes and coarsely quantised for rarely occurring amplitudes. Alternatively log quantiser can be used for non-uniform quantisation.

Vector Quantisation : In Vector Quantisation (VQ) a number of samples in the continuous signal are quantised at the same time leading to a reduction of the required quantisation space in comparison with scalar quantisation. Consequently VQ is a data compression method where a continuous signal is approximated by a digital representation (quantisation) using a fewer number of bits [162]. The design of vector quantisation is as follows, a number of N-dimensional (N samples) training sequences is divided into M regions or partition based on some distortion measure as shown in Figure 2.8. The selection of the M regions is done is such a way that the vector at the centroid of partition m (m=1, 2, . . . , M) becomes representative to all the N-dimensional vectors in that partition and all these vectors are closer to this centroid vector (codeword m) more than any other codeword k (k≠m) in any other partition. The set of all M codewords form what is known as a codebook. When a new vector is presented to the vector quantiser, the codeword where the vector lies (this is determined using a distortion measure) replaces the original codevector and only the index (called channel symbol) for that codeword is used to encode the vector. The number of bits necessary to represent the index are $\log^{M}{}_{2}$. The novel work of Linde, Buzo, and Gray (LBG) allows design of the VQ codebook according to the above procedure [109].

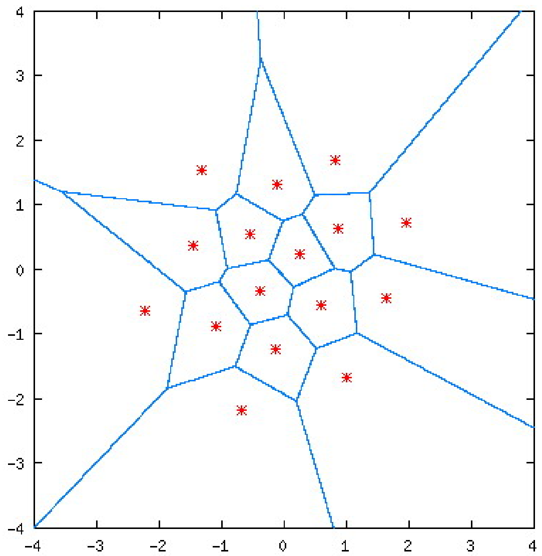

Figure 2.8: Vector Quantisation [29]

2.3.3 Speech Coder Design Issues

.

Many issues should be taken into consideration for successful design of a speech coder. Some of these factors include:

- Coding delay contributes to the overall M2E delay and is a major factor in speech coder design. Coding delay may be algorithmic (buffering speech for analysis or look-ahead), computational (Processing time). Some coders such as Pulse Code Modulation (PCM) introduce little delay due to the nature of the algorithm, while most low-rate algorithms introduce substantial delay.

- Mobile connections suffer greatly from random and burst errors. Forward Error Correction (FEC) mechanism or a coder built-in PLC is important for acceptable performance. Using built-in robustness reduces the need for FEC, thus more capacity is available to give better speech quality. In fixed link systems such as fibre-optics links the channels are more robust, so the need for error correction mechanism is thus reduced [144].

- Statistics show that 50% of a conversation is silence; therefore silence suppression techniques (discontinuous transmission during silence periods) were suggested because of the a big bandwidth saving implications during silence periods. A silence suppression technique requires the use of a voice activity detector (VAD) [88, 184]. Several algorithms have been proposed for voice detection. The simpler ones take account of the zero-crossings and the amplitude of the signal to detect the lower frequencies and higher amplitudes of speech segments when compared to noisy frames. Other algorithms are based on spectral characteristics of the signal such as the amount of energy contained at a certain frame. VAD techniques may cause a speech clipping which causes degradation to the voice quality. Also, silence suppression techniques try to play a comfort noise during the periods of silence [12, 88, 113, 184]. Some studies claim that this noise might sound artificial and will be annoying rather than comfortable [88] and they suggest the use of LBR coding techniques rather than the use of silence suppression.

- The performance of the speech coder is not dependent only on the selection of the speech coder, this is because a VoIP call normally traverses more than one network and each network may implement a different speech coder than the others. Therefore, speech is processed each time speech packets cross network borders. Because speech coding standards may differ in bit rate, frame size, and number of parameters, appropriate mapping between different speech coders should be done. Such processing is called coder tandeming or transcoding. Having multiple decompression/compression cycles can cause distortion to the speech, also this mapping introduces extra delay [48]. Several Tandem-free

Operation (TFO) strategies have been proposed, also other studies [51, 88, 91] state that in order to maintain the quality of the speech as high as possible, it is recommended to use one of the following speech coders: G.711, G.726, G.728, or G.729E because of the easy mapping between these coders.

- Another issue other than transmitting voice is the transmission of tones. Two approaches exist for transmitting tones. The first approach is, using an external signalling system that is separated from the media. The second approach is using the same media path to transmit RTP packets to convey information about the tone type and duration.

2.3.4 Categories of Speech Coders

Several standards have been proposed for coding speech, each one of these standards is built based on a philosophy or paradigm on how the speech signal should be represented or modelled. Such differences yield several categories of speech coders and each has its own performance. One of the most common classifications categorises speech coders into: waveform coders, vocoders, and hybrid coders.

Waveform Coders

Waveform codecs attempt faithful representation of the speech signal, without using any knowledge of how the signal to be coded was generated, to produce a reconstructed signal whose waveform is as close as possible to the original. This means that in theory they should be signal independent and work well with non-speech signals as well as with speech signals. Waveform coders accept the continuous analogue signal, sample the signal and encode these samples into digital form before transmission. Waveform coders are in general low complexity codecs, have high speech quality, but the main drawback of this type of coders is that they require high bit rate. Trials to reduce the bit rate for waveform coders have failed to produce high speech quality [24, 193, 203].

Voice coders (Vocoders)

In contrast to waveform coder, Voice coders (Vocoders) do not attempt to preserve the shape of the speech signal, instead to represent speech, they model how the source speech is produced . The parameters of this model are extracted from the speech and all other redundant information are removed. These parameters are transmitted and in the receiver side speech is synthesised back from parameters. Although the bit rate for this type of coders is very much reduced which may be as low as 2.4 kbps or even below which leads to what is called Low-bit Rate (LBR) coders, the quality of the synthesised speech is generally very low and does not sound natural although it may still be intelligible [24, 197].

The human speech production (see section 2.3.1) can be illustrated by a simple model as shown in Figure 2.9. To model speech production, lungs are replaced by a DC source, the vocal cords by an impulse generator. The vocal tract is represented as time-varying linear filter and is excited through the impulse generator for voiced speech or with a white noise source, for unvoiced speech segments as shown in Figure 2.10 [43].

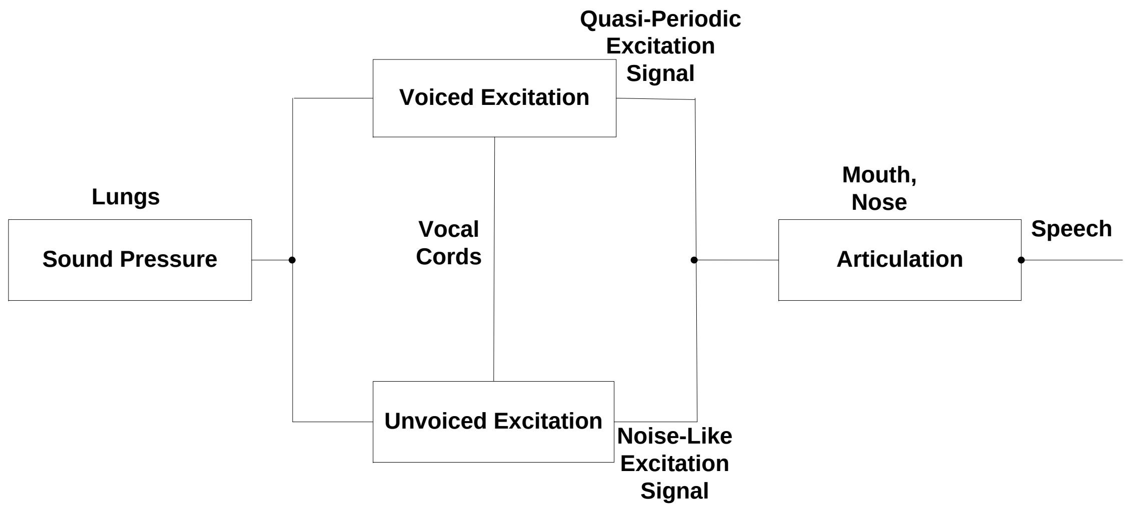

Figure 2.9: Human Speech production [43]

The information which must be sent to the decoder is the filter specification, a voiced/unvoiced flag, the necessary variance of the excitation signal, and the pitch period for voiced speech. This is updated every 10-20 ms to follow the non-stationary nature of speech [197].

The filter, representing the articulation tract, is a simple recursive digital filter; its resonance behaviour (frequency response) is defined by a set of filter coefficients. These parameters are used to describe the speech signal and are computed based on the mathematical optimisation procedure of Linear Prediction Coding (LPC)

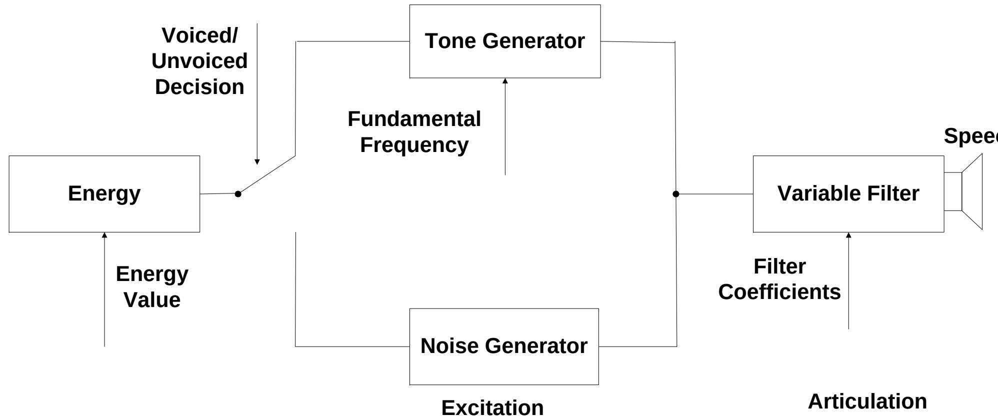

Figure 2.10: Machine Speech production [43]

they are called Linear Prediction Coding Coefficients or LPC coefficients and the complete model is the so-called LPC Vocoder [43].

Hybrid Coders

Hybrid coders combine features of waveform coders and vocoders. Many hybrid codecs employ vocoding techniques to derive some of the codec parameters and code the residual between the original and synthesised speech using waveform coding techniques.

One of the most common forms of hybrid coding is Analysis by Synthesis (AbS) which is used by many speech coding standards to allow producing good quality speech at Low-bit rate. In the analysis stage, the speech is represented by a compact set of parameters which are encoded efficiently. In the synthesis stage these parameters are decoded and used within a construction mechanism to form speech. Analysis can be open-loop or closed-loop. In closed-loop analysis, the parameters are extracted and encoded by minimising explicitly a measure of error (usually the mean square error) between the original and the reconstructed speech. Hence, closed-loop analysis of the input speech at the encoder incorporates synthesis and different excitation signals are attempted and the one that result closest match between the reconstructed speech waveform and the original one is selected. Because the original speech signal is used, this process is called Analysis by Synthesis (AbS).

AbS codecs use linear prediction filter model of the vocal tract as found in LPC vocoders. AbS coders work by dividing the input speech to be coded into frames and for each frame parameters are determined for a synthesis filter, and then instead of applying two-state, voiced/unvoiced model -as in the case of LPC- to find the necessary input to this filter, different excitation signals are attempted and the closest match between the reconstructed speech waveform and the original one is selected. The encoder transmit for each frame information representing the synthesis filter parameters and the excitation to the decoder, and at the decoder the given excitation is passed through the synthesis filter to allow the reconstruction of the signal [196].

AbS coders have been used in different types of coders such as: Multi-Pulse excited (MPE) codecs, Regular-Pulse Excited (RPE), Code-Excited Linear Prediction (CELP) and Vector Sum Excited Linear Prediction (VSELP) [24]. The differences between these types lies in the representation of the excitation signal used:

- **Multi-Pulse Excited (MPE):** Is the first form of AbS coder and it was introduced in 1982 by Atal and Remde [9]. In MPE codecs the excitation signal is given by a fixed number of non-zero pulses for every frame of speech. The positions of these non-zero pulses within the frame, and their amplitudes, must be determined by the encoder and transmitted to the decoder. To find the very best values for all the pulse positions and amplitudes, it would entail excessive complexity. In practice some sub-optimal method of finding the pulse positions and amplitudes must be used. Typically about 4 pulses per 5 ms are used, and this leads to good quality reconstructed speech at a bit-rate of around 10 kbps [9, 196].

- **Regular Pulse Excited (RPE):** Regular Pulse Excited (RPE) is another form of AbS coders. Like MPE coders, RPE coder uses a number of non-zero pulses to give the excitation signal. However, in RPE codecs pulses are regularly spaced at some fixed interval and the encoder needs only to determine the position of the first pulse and the amplitude of all the pulses [196]. Therefore less information needs to be transmitted about pulse positions in RPE codec than MPE codecs and this allows RPE codecs to give slightly better quality reconstructed speech quality than MPE codecs by using larger number of non-zero pulses for a given bit rate. However they also tend to be more complex.

- **Code-Excited Linear Prediction (CELP):** Both MPE and RPE can provide good quality speech but they require transmitting large amount of information, consequently they are not suitable for rates much below 10 kbps. The most commonly used algorithm for producing good quality speech at rates below 10 kbps is Code-Excited Linear Prediction (CELP) proposed by Schroeder and Atal in 1985 [10, 153]. CELP is different from MPE and RPE in that the excitation signal is efficiently vector quantised.

  The speech-synthesiser in CELP consists of two time-varying linear recursive filters each with a predictor a feedback loop. The short-delay predictor has 16 coefficients updated every 10 ms and determined using the weighted stabilised covariance method of LPC analysis. In this method of LPC, the instantaneous prediction error is weighted by 20 ms-long Hamming window, and the prediction coefficients are determined by minimising the energy of the weighted error. The long-delay (pitch) predictor has 3 coefficients which are determined by minimising the mean-squared prediction error after pitch prediction over a 5 ms interval.

  Several speech coding standards have been defined based on the CELP principle such as: Department of Defence (DoD) 4.8 kbps codec and ITU-T's G.723.1, G.728, and G.729 coders as described in the next section.

- **Vector Sum Excited Linear Prediction (VSELP)**: Introduced in 1990 by Gerson and Jasluk [47] and it was inspired by the idea of CELP coding family. VSELP coders address the major drawback of the CELP coders which is their large computational requirements during the search for the best excitation within the codebook. VSELP uses highly structured excitation codebooks such that the computational complexity required for codebook search is significantly reduced and at the same time the robustness to channel error is increased. VSELP coder was selected by the Telecommunication Industries Association (TIA) as the standard for use in North American digital cellular telephone systems.

  The excitation in VSELP comes from 3 different codebooks. The first excitation comes from the long term (pitch) predictor state or adaptive codebook, the second and third ones come from two VSELP fixed excitation codebooks. Each of these 3 excitations is multiplied by their corresponding gain and summed to form the overall excitation sequence for the synthesis filter. After each sub frame, the value of this excitation signal is used to update the adaptive codebook. Each codebook contains 128 vectors which require 7 bits for the index. The excitation for the filter need to be updated every 5 ms sub frame, the process of selecting the vectors from the codebook is sequential, first the adaptive codebook is searched, then after determining the excitation from the adaptive codebook, the excitation from the first fixed codebook is determined, then the excitation from the second codebook is determined.

The synthesis filter is a 10th order LPC filter whose coefficients are computed once every 20 ms frame and updated from one sub frame to another through interpolation. The three excitation gains are effectively vector quantised to 8 bits per 5 ms sub frame. An energy term which represents the average speech energy per frame is also computed once per frame and needs 5 bits.

2.3.5 Speech Coding Standards

Different standards for speech coding are released by different standardisation bodies. The evaluation of the speech coder depends on several factors, these include: the required bit rate, quality of the reconstructed speech, complexity of the algorithm, delay introduced, and robustness to channel errors and acoustic interference [162]. Some of the most famous standard speech codecs are described next:

ITU-T Standards

Several speech coding algorithms were standardised by the ITU-T such as G.711 [66, 194], G.726 [67, 195], G.728 [68, 201], G.723.1 [70, 87] and G.729 [7, 71, 151]. Some of the ITU-T standards are explained next

- **G.711:** Often called Pulse Code Modulation (PCM) codec defined in ITU-T Recommendation G.711 [66]. It is the simplest form of waveform codecs and the most common coding technique used in today's networks, and it is the coding technique used in circuit switching telephone networks all over the world. As previously explained G.711 is typically sampled at sampling rate of 8 kHz according to Nyquist theorem. Uniform quantisation requires 12 bits per sample leading to a bit rate of 96 kbps [194]. In non-uniform quantisation two variations are exist: A-law which has been standardised in Europe and μ-law which has been standardised in America [194]. Both A-law and μ-law are based on logarithmic scaling of each sampled pulse amplitude modulation, with 8 bits are used to represent each sample. In both A-law and μ-law, every quantised value is associated with a specific uniform PCM value. The slight difference between A-law and μ-law lays in the length of the quantisation level and the selection of the decision value that separate one quantisation level from another [193, 194].

Non-uniform quantisation leads to the well-known 64 kbps rate that we are familiar with and happy with in today's telephone networks. G.711 as a waveform coder does not imply any algorithmic delay, while its main drawback is the high bandwidth requirements.

- **(Adaptive) Differential PCM:** As voice signal changes its value relatively slowly and the difference between successive voice samples is small. It is possible to minimise the bandwidth requirements by quantising the difference between the current sample and the next sample rather than quantising the sample itself, and that is what Differential PCM (DPCM) is all about.

  DPCM has two variations, in the first variation only the difference between sample N and sample N+1 is transmitted, therefore the far end add this difference (positive or negative) to sample N in order to recreate sample N+1 exactly and as voice signal is changing slowly, therefore a fewer bits are needed to represent the difference than representing the actual sample value. In the second variation, some intelligent prediction is made about the sample values based on the past samples keeping in mind the knowledge of how speech varies with time. The difference between the actual sample and the prediction is quantised, and this difference is sent to the far end. The far end is doing the same prediction and uses the difference it received from the sender to retrieve the actual value. The performance of the codec is aided by adapting the prediction and the difference quantiser to the changing characteristics of the speech signal, Adaptive Differential PCM (ADPCM) [195]. Consequently more accurate prediction can be achieved and fewer bits are needed to represent the difference between the actual value and the predicted value. DPCM, and ADPCM as waveform coders has no algorithmic delay, still both of them require relatively high bandwidth. Examples of ADPCM are ITU-T Recommendation G.721 operating at 32 kbps which is a significant reduction from the 64 kbps in case of PCM, and the more advanced ITU-T Recommendation G.726 operating at 40, 32, 24, or 16 kbps using 31, 15, 7 or 4 quantisation levels respectively [67].

- **G.729:** Conjugate Structure-Algebraic Code Excited Linear Prediction (CS-

ACELP) or G.729 finalised in 1996 is the ITU-T's officially recommended CODEC for all wide area networking applications [7, 71, 151]. It operates on frames of 80 samples at a time (sampled at 8kHz) which represent 10 ms. G.729 also utilises a look-ahead of 5 ms. This will result in 15 ms algorithmic delay. The standard specifies a code-excited linear predictive coder that uses an algebraic codebook to code the excitation signal.

The speech signal is analysed to extract the parameters of the CELP model (linear prediction coefficients, excitation codebook indices, and gain parameters). These parameters are transmitted to the far end. The bit allocation for each parameter is shown in Table 2.4. This information is 80-bit per frame producing an 8 kbps transmission rate. At the decoder, these parameters are used to retrieve the excitation through the short-term 10th order Linear Prediction (LP) synthesis filter. The adaptive codebook is used to implement the long term or pitch analysis filter [193].

G.729 has a number of annexes suggesting many modifications and enhancements to the basic G.729 algorithm. For example for the purpose of discontinuous transmission (DTX), annex B suggests addition of a silence suppression feature. With the low rate of 8 kbps, G.729 is the lowest bit rate ITU-T standard with toll quality (quality comparable to the analogue speech) [71].

Parameter Name	Required Number of Bits
LSP	18
Pitch Prediction filter	14
Codebook indicies	34
Gains	14
Total	80

Table 2.4: Bit allocation for 8 kbps CS-ACELP

ETSI Standards

Global System for Mobile Communications (GSM) is digital cellular radio communications system built in order to create a common European mobile telephony standard. GSM was designed to be compatible with the ISDN and it is extensively used in Europe, and also in other parts of the world [157, 193].

Since 1989 GSM become the responsibility of the European Telecommunication Standards Institute (ETSI). ETSI has standardised many speech coding standards for the digital cellular telephony, including: GSM Full Rate (GSM-FR) [39, 202], GSM Half Rate (GSM-HR) [38], GSM Enhanced Full Rate (GSM EFR) [37], and GSM Adaptive Multi-Rate (GSM AMR) [36, 100, 143, 158].

Department of Defence Standards

In military application security and voice intelligibility are major concerns rather than the natural reproduction of voice as is the case for telephony and cellular communications. The Department of Defence (DoD) in the USA has standardised many speech coders for secure communications systems. Parametric coders such as Mixed Excitation Linear Prediction (MELP) are used widely in secure communication due to their intelligible speech quality at very low bit rates. Among the DoD standards are: Federal Standard-1015 (FS-1015), Federal Standard-1016 (FS-1016), DoD 2.4 , and STANAG (NATO) [100, 199].

2.3.6 Speech Coding Summary

Different speech coders operating at different rates and may result in different voice qualities. Also, different speech coders introduce different coding delays, additionally, some complex speech coders deal with the problem of packet loss through what is known as Packet Loss Concealment (PLC) algorithms. Consequently, the selection of speech coding technique is crucial to the speech quality. For the purpose of VoIP many speech coders are candidate and many others are excluded. Among the excluded coders is G.711 or PCM coding standard which is used in standard telephony due to its high bandwidth requirements. Among the candidates is G.729 as it operates at low bit rate with an acceptable quality. The topic of quality and how the quality is measured is discussed in next chapter.

2.4 Summary

In this chapter background information about VoIP technology was presented. Although a wide range of topics are covered and many of those will not elaborated upon in the coming chapters and not part of the main theme of this thesis, it was felt necessary to cover such topics for the purpose of full understanding of the VoIP technology and the interaction of different parameters. This will give a full picture that will put the work of the coming chapters into context.

As IP networks are best effort networks with no guarantee of speech quality, some impairments such as packet loss are inevitable in such networks. As such, it is important for legal, commercial, and technical reasons to measure the speech quality in VoIP technology. The topic of how to measure the quality of a speech signal is discussed in the next chapter. Section 3.1 discusses QoS issues. Section 3.2 describes different technologies used to measure the quality of voice in IP networks.

The reason for dedicating chapter 3 to cover such topics in addition to their importance is that some of the work discussed in this chapter is related to development done during this research and not purely work from the literature.

Chapter 3

QoS and Quality Assessment Technologies

As the main theme of this thesis is how the quality of the speech signal can be measured, this chapter is dedicated to discuss this topic. Section 3.1 discusses QoS issues. Section 3.2 describes how the quality of speech in IP networks can be measured using specialised technologies to assess the successfulness of a voice transmission system.

Some of the topics covered in this chapter present some advances in the technology from the literature, additionally some of work that was done during this research in relation to these topics is also included.

3.1 Quality of Service

Several definitions have been proposed for the term Quality of Service (QoS) in the context of VoIP networks depending on the user's perspective:

- A collective measure of the level of service delivered to a customer [24].
- QoS is providing applications with a mean of managing and predicting network resources such as available bandwidth and latency to achieve the more efficient use out of these resources and allowing preferential treatment for certain subsets of data [126].
- A network with quality of service has the ability to deliver data traffic with a minimum amount of delay in an environment in which many users share the same network [110].

QoS includes several areas. The first important area is what voice quality the new system is able to achieve. The second important area to investigate is the network that will carry speech packets around, a congested network with burst packet loss, delay, and jitter all of which will contribute in degrading the voice quality. Increasing the bandwidth can compensate for the degradation in voice quality due to unpredicted traffic in the network. Another possibility is to carefully design and manage the network in order to suit the needed applications. Several techniques have been proposed in order to control network resources and provide the required level of QoS, these solution are described in this section.

Several criteria can characterise QoS. Among these criteria are: availability (low downtime), call setup time (less than two seconds for local calls), voice delay (less than 150 ms one way), minimal echo and disturbing sounds, and percentage of successful transmissions [95, 132]. Quality of the speech can be measured either subjectively or objectively using quality assessment techniques as discussed in section 3.2.

3.1.1 QoS Solutions

Some of the QoS solutions should be applied end-to-end (at the sender side and/or at the receiver side). Among the end-to-end solutions are Packet Loss Concealment (PLC) and buffering of packets to reduce the effect of jitter. On the other hand several solutions could be applied in the transmission path only in private, managed networks and most probably not in the current open public Internet. Hence VoIP will remain to be a best effort only in the public Internet [114, 115].

As most IP networks are likely to be non-managed, shared networks resources among many types of traffic, then the lack of resources and mainly the needed bandwidth in the IP networks, causes the unwanted delay and packet loss in the intermediate routers between the source and the destination. Therefore, the most direct and simple way of addressing the issue of providing good voice quality is by over provisioning the network with greater bandwidth than the heaviest possible traffic would require, but this extra bandwidth needed during the traffic bursts and which is extremely costly to the enterprise, would remain unused for most of the time [182]. Hence, over provisioning the bandwidth although could provide the necessary level of QoS for voice traffic, but additional mechanisms are needed to be in place to manage the available bandwidth more effectively.

A mechanism called Call Admission Control (CAC) is necessary to determine whether to accept a call request if it is possible to allocate the required resources (bandwidth) and maintain the given QoS target for all existing calls, or otherwise to reject the call [116]. Among the solutions that have been proposed to implement CAC and to manage the available bandwidth efficiently are: Resource Reservation Protocol (RSVP), Differentiated Service (DiffServ), MultiProtocol Label Switching (MPLS), and End-to-end Measurement Based Admission Control (EMBAC). All the above techniques are applied in case the IP network is a managed network. When the IP network is non-managed, the above techniques can not be applied. Also VoIP signalling protocols (section 2.2.3) such as SIP should comply with one or more of the aforementioned resource management protocols. Signalling in VoIP such as in SIP would require reservation of resources during call setup time, and once the reservation has taken place, normal VoIP signalling can continue as usual [19, 116].

Resource Reservation Protocol (RSVP)

RSVP enables resources to be reserved prior to any attempt to exchange media between session's participants. Consequently, VoIP sessions should run smoothly as all the required resources are reserved in advance for the session. Although RSVP is a complex protocol, but it comes closest to circuit emulation within the IP network.

In RSVP, the sender issues a *path* message to the far end via a number of routers. The receiver of the *path* message responds with a reservation request (RESV) message which travels back to the sender along the same route that the *path* message took but in the reverse direction. At each router, the requested resources are allocated, assuming these resources are available and the receiver has the authority to make such a reservation. If reservation of such resources was not successful, the session will not start due to the lack of resources. Based on the previous sequence, resources reservations in RSVP are made by the receiver, not by the sender. This approach accommodates multicasting situations, where there may be a large number of receivers and only one sender. RSVP also supports different levels of reservation guarantee.

RSVP deallocation can be done through a set of messages, but these deallocation messages are unreliable and could be lost in the network. Due to this fact, situations may arise where resources are reserved when they are no longer needed, while new reservations fail due to the lack of available resources. To avoid such situations, RSVP introduces the concept of implicit deallocation, where reservation of resources should be refreshed regularly in what is known as soft state and if this refreshment does not occur, these resources are automatically deallocated. One of the main drawbacks of RSVP is that it introduces a significant overhead and does not scale well in large systems with millions of simultaneous calls. Some improvements on the basic RSVP protocol can be applied to improve its scalability. These improvements suggest keeping track of aggregate statuses of the major traffic types rather than keeping track of every single flow status. One major disadvantage of such approach is per-flow information is not isolated from other flows of the same class, therefore, it may not guarantee QoS for each flow [19, 116, 164].

One major disadvantage of RSVP is that for the right behaviour of RSVP, most routers in the network should be replaced to implement the new protocol. This is an expensive proposal and its implementation will come gradually [1].

Differentiated Service (DiffServ)

DiffServ is simpler than RSVP. Diffserv is based on Per-Hop Behavior (PHB) where traffic is offered different services based on the type of traffic, in this case a queuing strategy such as Weighted Fair Queue (WFQ) can be used. Therefore, DiffServ can be classified as traffic prioritisation technique rather than true resource reservation technique. Two types of PHBs are currently defined [19, 24, 139]:

1 **Expedited Forwarding (EF):** In expedited forwarding specific amount of bandwidth is reserved to allow the rate of departure of packets to be larger than the arrival rate. The objective of EF is to minimise queuing time (which contributes effectively to the overall transmission delay) and to provide service with low loss, low delay, and low jitter [131, 180].

2 **Assured Forwarding (AF):** Assured forwarding defines different classes and for each class certain amount of resources are allocated. Within each class, different drop rates are applied. If there is a congestion, the packets are dropped according to their drop rates. Asosheh and Bahaei [8] proposed distributing

of drop of packets evenly among flows to reduce the possibility of continuous packet loss in one stream

One other possible use of DiffServ is to give different users different priorities or precedence classes based on the importance of the call. This is especially important in military networks where different priorities could be assigned to different military personnel [181]. Also several studies investigate the effect of different parameters of the queues such as the queue size on the performance of the system especially in catastrophic conditions [116, 206].

A hybrid approach between RSVP and DiffServ can also be used in such a way RSVP is used in the access network while DiffServ is used in backbone networks where it can scale well [33].

MultiProtocol Label Switching (MPLS)

RSVP is powerful at session level, but does not scale well in large networks while DiffServ is simpler and scales well, but it is a priority assigning technique rather than resource-guarantee mechanism. MPLS tries to offer the best of both worlds.

MPLs marks traffic similar to the DiffServ marking. The difference is that traffic marking in MPLS known as labelling does not allocate priority according to the traffic type. Instead this label is used to determine the next hop in the path according to the sender, receiver and the current traffic in the network. MPLS is trying to distribute traffic over the network in a fair way such that the possibility of traffic being heavy on specific links while other links are in an idle situation is being minimised. In MPLS, packets from the same data stream are identified to have the same Forward Equivalence Class (FEC). This FEC is mapped to the label. As packets from the same data stream have the same FEC value, consequently the same label is assigned for them. Therefore, these packets will have the same treatment and will consequently follow the same path. An example of MPLS is shown in Figure 3.1.

In Figure 3.1, the packet has an FEC of F. At ingress router (R1), the FEC of F means that the packet must be sent to router R2 and that it should have a label value of L1. When the packet and label arrive at R2, R2 knows that the label value L1 means an FEC of F for packets arrived from R1. R2 then proceeds to lookup the next hop and label value. It determines that the packet should be forwarded to

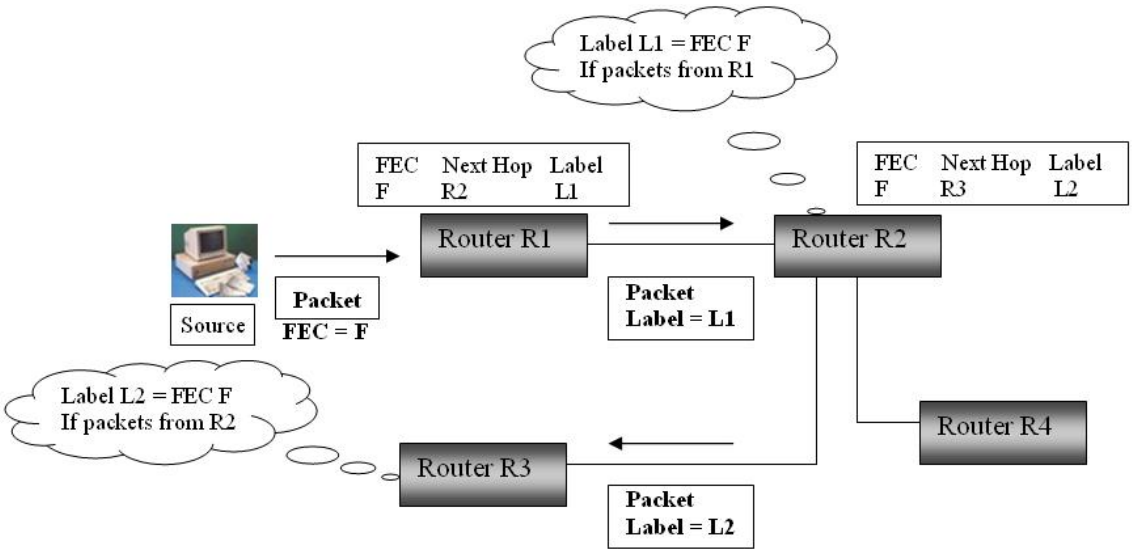

Figure 3.1: Label value and FEC relationships in MPLS [24]

R3 with a label value of L2. At R3, the packet and label are received. R3 knows in advance that the label value L2 means an FEC of F for packets from R2 and it uses this information for its routing decision. This process continues until the packet arrives to its destination [19, 24].

Providing the necessary resources on that path will allow us to provide the required QoS with minimum delay, packet loss, and jitter. At the same time assigning different classes for different streams, allows us better control the network. The major disadvantage of MPLS is that it requires significant changes to all routers that want to use it.

End-to-end Measurement Based Admission Control (EMBAC)

In this method end-points estimate the network congestion by measuring some transmission parameters such as delay, packet loss, and jitter over a measurement period. There are two types of control schemes in EMBAC:

1 **Passive method:** In this method actual data flows are monitored in order to estimate the network status, and if the sum of measured existing load with the load of the new flow is not exceeding the network capacity limit, the new flow will be admitted, otherwise, the new flow will be rejected. In measuring the network load, sampling techniques over a timescale is applied in order to estimate the network load. One advantage of the passive method is that actual

flow is used rather than generating probe flow which consumes part of network resources as in the case of the active method [40, 117].

2 **Active method:** In this method, a short-period probe flow is generated between the end-points and QoS of the probe flow is measured to estimate the network load. Depending on the measured QoS of the probe flow, the new flow is either admitted or rejected. Active method is simpler than the passive method in design but it adds extra delay to the call setup time and extra overhead load on the network [13, 40, 117, 118, 207, 208].

Estimation of the network load can also be used to dimension the network [183].

3.1.2 QoS Policies

Protocols such as RSVP, DiffServ, MPLS and EMBAC provide the mechanism to differentiate traffic and allocate resources to specific types of traffic. In addition to these protocols QoS policies are also needed to specify how these mechanisms are used. These policies are also used to provide authentication functions to identify users and ensure that a given user is who he or she claims to be. In addition, a given user may be entitled to a given level of QoS under certain circumstances but not under other circumstances. These policies can also be used, to protect from Denial of Service (DoS) attacks. The Internet Engineering Task Force (IETF) has developed Common Open Policy Service (COPS) protocol. COPS includes a Policy Enforcement Point (PEP), such as a router that enforces certain rules. PEP queries a Policy Decision Point (PDP) that makes a decision [24]. PEP works like a policeman, while PDP works like a Judge.

3.2 Assessment Technologies for Measuring VoIP Perceptual Quality

As mentioned earlier, quality is not guaranteed in non-managed IP networks such as the Internet, therefore, it is important to monitor the speech quality in telecommunication systems and take appropriate actions when necessary. It is also important to measure the quality even in managed networks. This is important for commercial, technical and may be legal reasons. Also this allows service providers to evaluate their own and their competitors' service using a standard scale [215]. It is also a strong indicator of user's satisfaction of the service provided. In doing so, a specialised mechanism is required for measuring the quality [175].

The primary criterion for voice and video communication is subjective quality, the user's perceptions of service quality. A subjective quality assessment method is used to measure the quality. Subjective quality factors affect the quality of service of VoIP, among those factors are: packet loss, delay, jitter, loudness, echo, and codec distortion. To measure the subjective quality, a subjective quality assessment method is used, the most widely accepted metric is the Mean Opinion Score (MOS) as defined by ITU-T Recommendation P.800 [72].

However, although subjective quality assessment is the most reliable method, it is also time-consuming and expensive as any other subjective test. Thus other methods to automatically estimate subjective quality objectively should be considered. This can be done intrusively by comparing the reference signal with the degraded signal or non-intrusively utilising physical quality parameters without using the reference signal. This process is called objective quality assessment and there exist methods for measuring voice quality objectively either intrusively or non-intrusively. Subjective quality assessment is discussed in section 3.2.1. Section 3.2.2 describes intrusive objective quality assessment technologies and non-intrusive objective quality assessment technologies are described in section 3.2.3. Section 3.2.4 determine the desired features for a VoIP speech quality solution and lay a ground for coming chapters. To avoid ambiguity, different qualifiers used to distinguish between different quality measurement methods are presented in section 3.2.5.

3.2.1 Subjective Assessment of Speech Quality

The primary criterion for voice and video communication is subjective quality, the user's perceptions of service quality. Subjective quality factors affect the quality of service of VoIP, among those factors are: packet loss, delay, jitter, loudness, echo, and codec distortion. To measure the subjective quality, a subjective quality assessment method is used. The most widely used subjective quality assessment methodology is opinion rating defined in ITU-T Recommendation P.800. Such subjective tests could be conversational or listening-only tests. In conversational test, two subject share a conversation while in listening tests, one subject is listening to pre-recorded sentences [72].

Opinion rating methodology rate the performance of the system either directly (Absolute Category Rating, ACR) or relative to the subjective quality of a reference system (Degradation Category Rating, DCR) [72, 173, 175].

The most common metric in opinion rating is Mean Opinion Score (MOS) which is an ACR metric with five point scale: (5) Excellent, (4) Good, (3) Fair, (2) Poor, (1) Bad [72]. MOS is internationally accepted metric as it provides a direct link to the quality as perceived by the user. A MOS value is obtained as an arithmetic mean for a collection of MOS scores for a set of subjects. MOS scores should be measured under strict lab conditions as stated in ITU-T Recommendation P.800. When the subjective test is listening-only, the results are in terms of listening subjective quality, i.e. Mean Opinion Score - Listening Quality Subjective or MOS_{LQS}. When the subjective test is conversational, the results are in terms of conversational subjective quality, i.e. Mean Opinion Score - Conversational Quality Subjective or MOS_{CQS} [72, 83].

In DCR, the rating is performed in comparison with a reference signal and the subjects are instructed to rate the conditions according to the following five-point degradation category scale: degradation is (5) inaudible, (4) audible but not annoying, (3) slightly annoying, (2) annoying, and (1) very annoying. The mean value of the results is called the Degradation Mean Opinion Score (DMOS). This is especially useful when the impairment is small and a sensitive measure of the impairment is required [72, 175].

To conduct a subjective experiment according to the ITU-T Recommendation P.800, strict lab conditions should be in place. Such conditions concerns the room size, noise level, and the use of sound-proof cabinet in a room with a volume not less than 20 m^3. Also the sound pressure level should be measured from a vertical position above the subject's seat while the furniture in place. In case of recording, the microphone is positioned between 140 mm and 200 mm from the talker's lips [72].

Recommendation P.800 [72] also specifies other conditions regarding the subjects who participate in the test such as they have not been directly involved in work connected with assessment of the performance of telephone circuits, or related work such as speech coding, also they have not participated in any subjective test whatever for at least the previous six months, and not in a conversational/listening test for at least one year. Also in case of listening-test they have never heard the same sentence lists before.

Results of MOS scores should be dealt with care as results may vary depending on the speaker, hardware platform, listening groups and test data and slight variation between different subjective tests should be expected although the above rigid conditions should guarantee minimisation of such cases.

Although opinion rating methods are the most famous subjective quality assessment methodology, but other methods have also been proposed. Diagnostic Rhyme Test (DRT) is an intelligibility measure where the subject task is to recognise one of two possible words in a set of rhyming pairs (e.g. meat-beat). Diagnostic Acceptability Measure (DAM) scores are based on results of test methods evaluating the quality of a communication system based on the acceptability of speech as perceived by a trained normative listener [162]. Li [105] proposed the use of intelligibility index as an additional parameter that can be used along with the commonly used MOS score. But even with these methods opinion rating methods are still the most famous and widely used method.

Although the primary criterion for voice and video communication is subjective quality, the user's perceptions of service quality and subjective measurement is the most accurate assessment method to measure the subjective quality, but there are few problems associated with subjective tests.

It is apparent from the strict conditions associated with opinion rating methods as mentioned above that the inherent problems in subjective MOS measurement are that it is: time-consuming, expensive, lacks repeatability , and inapplicable for monitoring live traffic as commonly needed for VoIP applications, which if not addressed appropriately may result in legal disputes and technical and commercial problems. This has made objective methods very attractive to estimate the subjective quality for meeting the demand for voice quality measurement in communication networks. However as subjective methods are the most accurate methods for measuring speech quality, they are used to calibrate objective methods

3.2.2 Intrusive Objective Assessment of Speech Quality

As discussed in section 3.2.1 speech quality is often a subjective matter and it is related to the users' perceptions of service quality. A subjective quality assessment method is used to measure the quality such as MOS_{LQS} or MOS_{CQS} [72, 83]. The inherent problems of subjective quality measurement are that it is time-consuming, expensive, lacks repeatability, and inapplicable in monitoring live traffic which is necessary for legal, technical and commercial purposes. This motivated the use of objective measurements to estimate the subjective quality. Objective quality assessment methodologies can be categorised into two groups: Intrusive speech-layer objective models and Non-Intrusive opinion models. Intrusive methods are described in this section while non-intrusive methods are described in the next section.

Intrusive objective assessment of speech quality or speech-layer objective models are full-reference, intrusive methods of measuring the quality. They provide more accurate method for measuring speech quality as they require the original or reference speech signal as input and produce measurement of listening MOS by comparing the post-transmitted signal with the original one. However, such methods are inapplicable in monitoring live traffic because it is difficult or impossible to obtain actual speech samples as the reference signal is not available at the receiver.

Some intrusive algorithms are used in time-domain, others in frequency-domain. More recent algorithms are in perceptual domain. Several such methods have been proposed and they vary from simple ones to complex algorithms.

- **Signal-to-Noise Ratio:**

  The performance of the waveform speech codecs such as Pulse Code Modulation (PCM)-coded speech can be measured in term of Signal-to-Noise Ratio (SNR) as the encoding is generally done on a sample by sample basis. SNR is defined as the ratio of a signal power to the noise power corrupting the signal. The SEGmented SNR (SEGSNR) computes the SNR for each N-points segment of speech as such it can detects the temporal variations. The problem of SNR is that they cannot be applied to Low-bit Rate (LBR) codecs as in these coders the shape of the signal is not preserved and they become meaningless [100].

Other objective quality assessment methods are also proposed and some of them make use of spectral distortion to evaluate the performance of LBR codecs. None of these standards were accurate enough to be adopted by ITU-T. Later, perceptual domain measures were introduced and standardised. Perceptual domain measures are based on models of human auditory perception. These measures transform the speech signal into a perceptually relevant domain such as bark spectrum or loudness domain, and incorporate human auditory models [165].

In perceptual measure the original and degraded signal are both transformed into a psychophysical representation that approximates human perception. Then the difference between the original and the degraded signal is mapped into estimation of perceptual difference as perceived by the listener. Among perceptual domain measures include: Measuring Normalising Block (MNB), Perceptual Analysis Measurement System (PAMS), Perceptual Speech Quality Measure (PSQM), and Perceptual Evaluation of Speech Quality (PESQ) [74, 79, 165, 185, 186].

- **Measuring Normalising Block (MNB):**

  In this algorithm, both the input and output speech signals are perceptually transformed and a distance measure that consists of a hierarchy of Measuring Normalising Blocks (MNB) is then calculated. Each MNB integrates two perceptually transformed signals over some time or frequency interval to determine the average difference across the interval. This difference is then normalised out of one signal to provide one or more measurements.

  MNB algorithm starts by estimating the delay between the input speech signal and output speech signal due to the device or system (possibly IP network) under test. This is done using cross-correlation of speech envelops because highly compressed speech coders do not preserve the speech waveform, therefore waveform cross correlation give misleading estimation for the delay. Once the delay is estimated and compensated for, MNB proceed to the next step which is perceptual transformation.

  In perceptual transformation, the representation of the audio signal is modified in such a way it is approximately equivalent to the human hearing process and only the perceptual information is retained. The following steps are performed by Voran in [185, 186] on the speech signal sampled at a rate of 8000 samples/s before perceptual transformation. The speech signal is divided into frames of size 128 samples with 50% frame overlap. As Voran pointed out, the nonuniform ear's frequency resolution on the Hertz scale and nonlinear relation between loudness perception and signal intensity are the most important perceptual properties to model [185].

For modelling the nonuniform frequency resolution, the Hertz frequency scale is replaced by a psychoacoustic frequency scale such as the bark frequency scale using the relation:

$$b = 6.\sinh^{-1}\left(\frac{f}{600}\right) \tag{3.1}$$

where

b	Bark frequency scale variable.
f	Hertz frequency scale variable.

Figure 3.2 shows the transformation from Hertz to Bark scale. In bark scale, roughly equal frequency intervals are of equal importance. From the figure it can be seen that on the band 0-1 kHz in Hertz scale (corresponding to 0-7.703 Bark) is given equal importance by Bark scale as the band 1-4 kHz. It is worth noting that bark scale is used recently for measuring speech quality for wideband speech coding [58]

To model the nonlinear relation between loudness perception and signal intensity, the logarithmic function is used to convert signal intensity to perceived loudness.

The distance between the two signals is calculated using a hierarchy of Time Measuring Normalising Blocks (TMNB) and Frequency Measuring Normalising Blocks (FMNB). The hierarchy structure works from larger time and frequency scales down to smaller time and frequency scales. Each block integrates the perceptually transformed signals over time or frequency to determine the average difference between the two signals. Once all the measurements of the

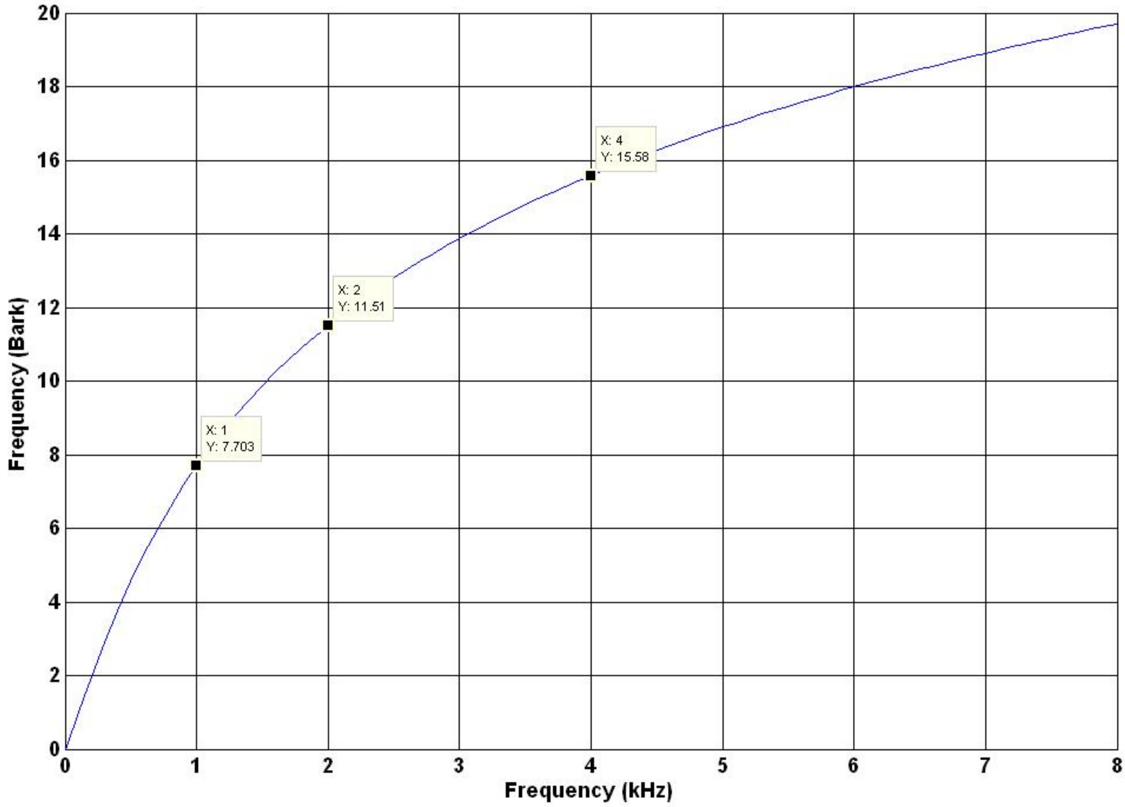

Figure 3.2: Transformation from Hertz scale to Bark scale

hierarchy are calculated on different levels, these measurement are linearly combined to calculate the Auditory Distance (AD) between the two signals. Finally the AD can be mapped using a logistic function into a finite set of values from 0 to 1 to increase correlation with subjective tests. [185, 186].

- **Perceptual Analysis Measurement System (PAMS):**

  Developed by British Telecom (BT). The PAMS process uses an auditory model that combines a mathematical description of the psychophysical properties of human hearing with a technique that performs a perceptually relevant analysis taking into account the subjectivity of the errors in the received signal. The PAMS process compares the original and received signal and produces two scores, listening quality score (Ylq) and listening effort score (Yle). Both scores are in the range 1 to 5 and MOS score can be estimated using a linear combination of both scores [32, 215].

- **Perceptual Speech Quality Measure (PSQM):**

  ITU-T Recommendation P.861 originally developed by KPN, Netherlands and then standardised by ITU-T as Recommendation P.861 to measure the speech quality objectively. PSQM transforms the speech signal into the loudness domain, applies a nonlinear scaling factor to the loudness vector of distorted speech. The scaling factor is obtained by calculating the loudness ratio of the reference and the distorted speech. The difference between the scaled loudness of the distorted speech and loudness of the reference speech is called Noise Disturbance (ND). The final estimated distortion is an average ND over all the frames processed where a small weight is given to silence portions during calculations. PSQM computes the distortion frame by frame, with the frame length of 256 samples with 50% overlap. The result is shown in noise disturbance as a function of time and frequency. The average ND is directly related to the quality of coded speech. There are two meaningful scores in the PSQM measure: one is a distortion measure and the other is a mapped number such as MOS. PSQM scores can be mapped into MOS scores using a non-linear mapping [74, 165, 215].

  PSQM designed to work under error-free coding conditions, therefore it is inapplicable for VoIP environment which suffers from packet loss especially in mobile communications that suffers from bit errors. Consequently a new standard, P.862 "Perceptual Evaluation of Speech Quality (PESQ)" was released. P.862 is a compromise between PAMS, and PSQM+ which is an extension of PSQM and is a result of co-operation between BT and KPN.

- **Perceptual Evaluation of Speech Quality (PESQ):**

  PESQ is the latest ITU-T standard for objective evaluation of speech quality in narrowband telephony network and codecs. It was a result of a collaboration project between KPN, Netherlands and BT, UK by combining the two speech quality measures PSQM+ and PAMS. Later it was standardised by ITU-T as Recommendation P.862 [79, 147].

  It was specifically developed to be applicable to end-to-end voice quality testing under real network conditions, such as VoIP, ISDN etc. The results obtained by PESQ was found to be highly correlated with subjective tests with correlation factor of 0.935 on 22 ITU benchmark experiments, which cover 9 languages (British English, American English, Swedish, French, Italian, German, Finnish, Dutch and Japanese). Upon its standardisation, PSQM in Recommendation P.861 was withdrawn by ITU-T [79, 86, 137, 149, 215].

Real systems may include filtering and variable delay, as well as distortions due to channel errors and LBR codes. PSQM was designed to assess speech codec and is not able to take proper account of filtering, variable delay, and short localised distortions.

In PESQ the original and the degraded signals are time-aligned, then both signals are transformed to an internal representation that is analogous to the psychophysical representation of audio signals in the human auditory system, taking account of perceptual frequency (Bark) and loudness (Sone). After this transformation to the internal representation, the original signal is compared with the degraded signal using a perceptual model. This is achieved in several stages: level alignment to a calibrated listening level, compressive loudness scaling, and averaging distortions over time as illustrated in Figure 3.3 [79, 147].

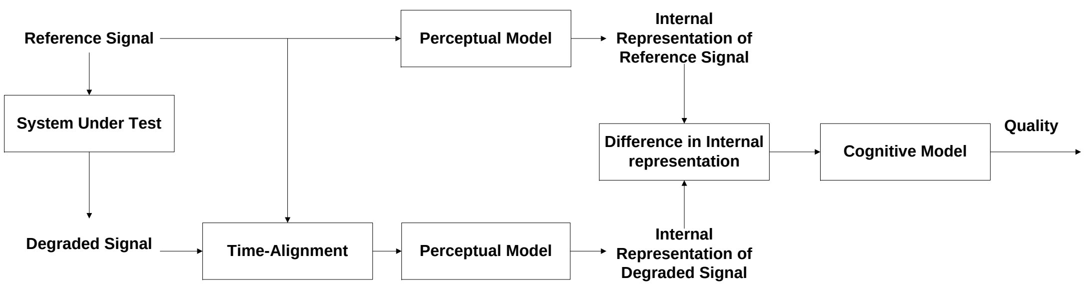

Figure 3.3: Conceptual diagram of PESQ philosophy [79]

PESQ score lie in the range -0.5 to 4.5, to make such score comparable with ACR MOS score, a function is provided in Recommendation P.862.1 to map these values to the range 1 to 5. The function in equation (3.2) do the conversion from a PESQ score to Mean Opinion Score - Listening Quality Objective or MOS_{LQO} which makes the comparison with other MOS_{LQO} results very convenient independent of the implementation of ITU-T Recommendation P.862 [86].

$$MOS_{LQO} = 0.999 + \frac{4.999 - 0.999}{1 - e^{(-1.4945*PESQ+4.6607)}} \tag{3.2}$$

The relation between the PESQ score and MOS_{LQO} is also shown in Figure 3.4.

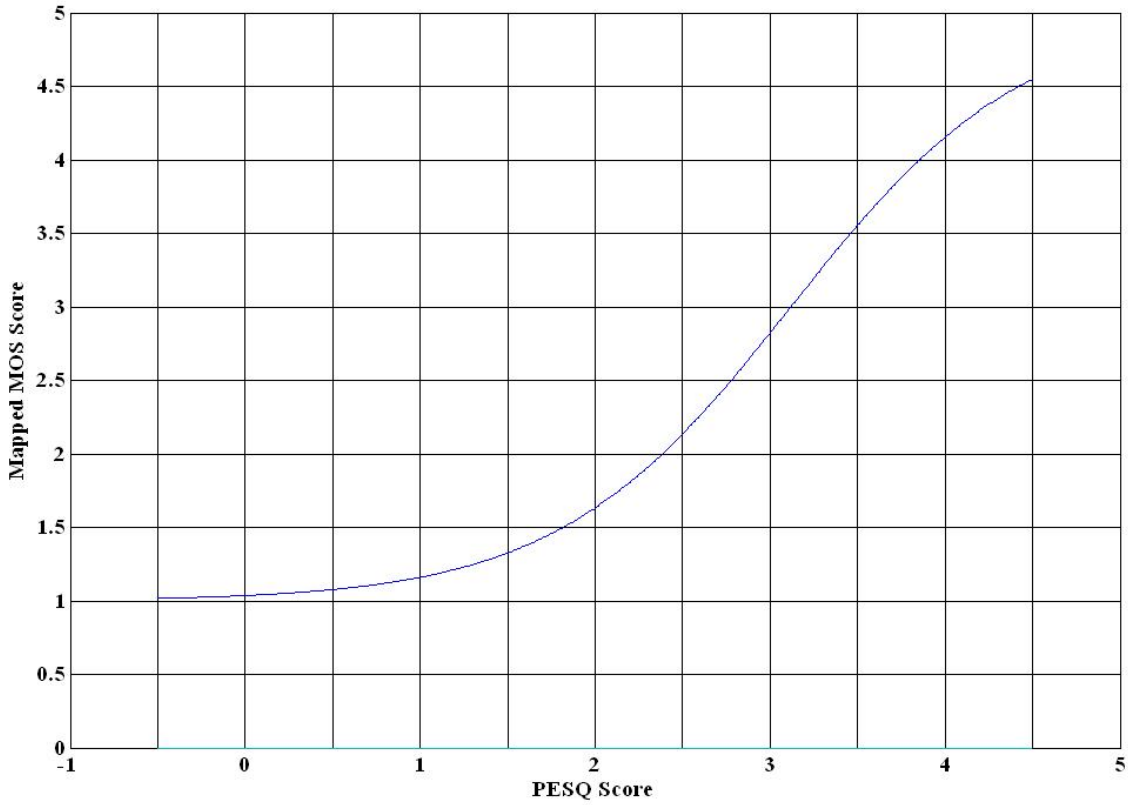

Figure 3.4: Mean Opinion Score (MOS) vs. PESQ Score

ITU-T Recommendation P.862.1 [86] also provides the formula to move back to PESQ score from an available MOS_{LQO} score. The equation is:

$$PESQ = \frac{4.6607 - \ln\left(\frac{4.999 - MOS_{LQO}}{MOS_{LQO} - 0.999}\right)}{1.4945} \tag{3.3}$$

The relation between MOS_{LQO} and PESQ score is shown in Figure 3.5.

Chong et al. studied the accuracy of PESQ in measuring the speech quality in Chinese language [20] by studying the the correlation between PESQ

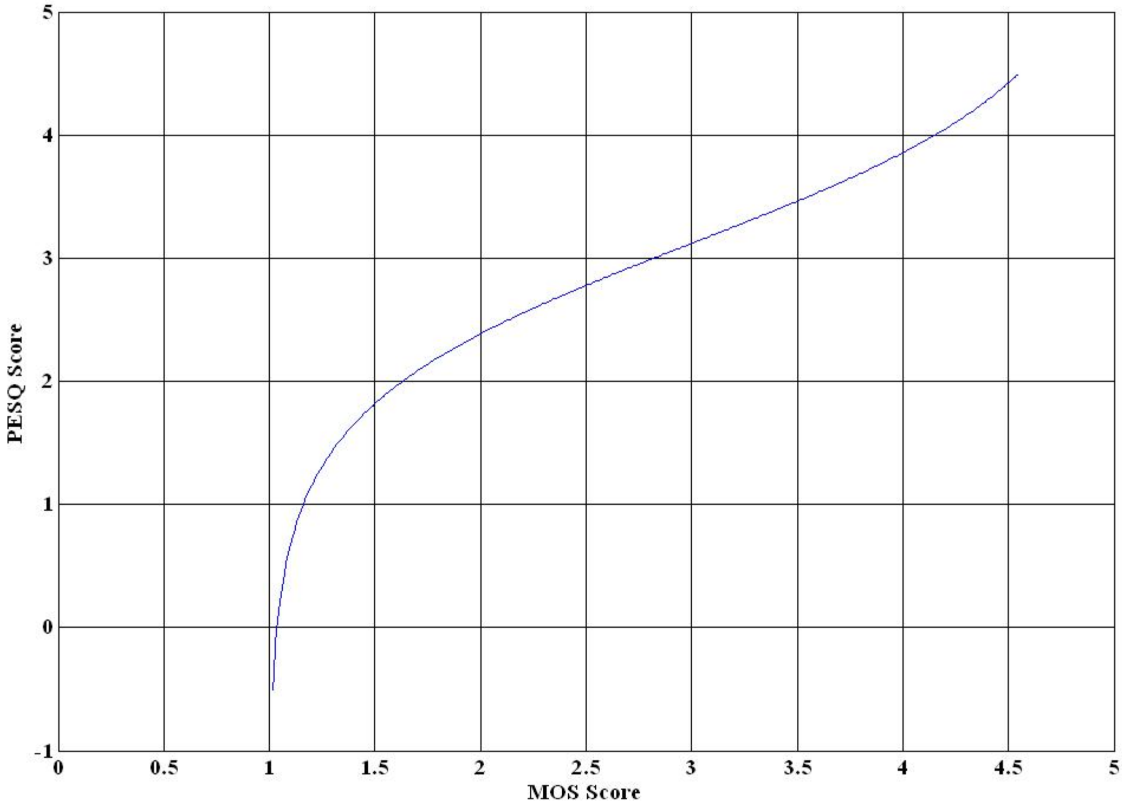

Figure 3.5: PESQ Score vs. Mean Opinion Score (MOS)

scores and Chinese subjective test. PESQ was found to not accurately predict the quality for Chinese speech which is something they referred to the characteristics of this language. A method is proposed to pre-process the input and output speech by consonant amplification before applying PESQ. This method was found to improve the accuracy of PESQ in relation to subjective intelligibility.

- **Intrusive ANN method:**

  During estimation of objective quality, distortion measures are usually used to determine the level of degradation in the received signal. Then the level of distortion is used to estimate the quality in some measure depending on the used objective test. This is a two-step procedure. For example using PESQ, the distortion measurement method output PESQ score. Then the output of the objective measure is mapped into an equivalent MOS score.

  Fu et al. proposed using an ANN model where the feature vector of the input vector and output vector corresponding to the original and degraded signal respectively are fed to the ANN model in addition with MOS subjective score as calculated using set of subjects as a target for the ANN. Utilising the proposed method and using the the error between the original input signal and output signal as input to the ANN model, MOS score can be estimated directly using one-step rather than the usual approach of estimating the distortion and then mapping the quality [45]

Several intrusive methods have been discussed in this section, PESQ still the most famous method used in intrusive measurement of speech quality as it is the latest ITU-T standard and it shows high degree of correlation with subjective tests as reported in ITU-T Recommendation P.862 [79]. Due to their intrusive nature, these methods can not be applied in monitoring live traffic. Non-intrusive methods for estimating the speech quality are described next.

3.2.3 Non-Intrusive Objective Assessment of Speech quality

Different methods have been proposed for objectively estimating the speech quality non-intrusively. These methods vary in complexity and accuracy from very simple techniques to very complex ones. The most famous method is the E-model as defined in ITU-T Recommendation G.107 [84]. This section review several such methods and then focuses on the E-model as it is the focus of the coming chapters.

- Duysburgh et al. [32] utilised the PESQ's ancestor, PAMS (section 3.2.2), to measure the effect of distortion due to different speech coders on the speech quality. The resulted quality is called MOS_{opt} as it represents the quality without any impairment. Then a series of experiments were performed with different speech coders to derive an equation of the form:

$$MOS_{pred} = MOS_{opt} - C.ln(Ppl + 1) \tag{3.4}$$

where

MOS_{pred}	Predicted MOS value
MOS_{opt}	Optimal MOS value without impairment
C	Speech coder constant factor (e.g. 0.25 for G.729)
Ppl	Packet loss Probability

In the above equation MOS score decreases proportional to the percentage of packet loss. In this effort, the authors used the superseded PAMS method which is now replaced by PESQ to measure MOS value under different packet loss conditions namely, 0%, 1%, 2%, 5%, and 10% and then a relation was derived to relate the resultant MOS value with the percentage of packet loss.

- Conway [25, 26] used information in RTP header (time-stamp, sequence number) at an intermediate node to analyse information about how the signal is degraded such as the location of packet loss in the signal. These information is then used to construct a packet loss pattern. Then the original signal is replaced by a reference signal and packet loss pattern extracted from RTP packets is imposed on the reference signal to obtain a pseudo-received signal. Then the pseudo-received signal is compared with the reference signal using an intrusive method such as PESQ to obtain PESQ score which can then be mapped into MOS score. Using this method speech quality can be estimated non-intrusively utilising PESQ, but as a reference signal is used instead of the original signal to estimate the quality, it is expected to have a deviation in quality estimation due to different characteristics of speech signals.

- Several methods have also been proposed to measure the speech quality over network links at different times rather than measuring the speech quality of a specific speech stream (voice call):

  – Wakahara et al. used a high quality speech recognition system to aid in quality estimation. They proposed the use of a voice synthesiser and voice recogniser at the receiver side. Speech parameters are sent over the network and then speech is synthesised and speech recognition is performed using the high quality recogniser. Then the quality of the received signal is estimated based on the ratio of correct recognition [188].

  – Hammer et al. [57] proposes a method to measure the performance of network in transmitting speech in general. They argue that subjective listening tests do not reflect the huge variability in packet loss patters. Objective tests results are irreproducible and measurements are bounded by the length of speech samples which can not characterise the behaviour of the network in general. They proposes employing arbitrary packet traces obtained from an IP network and then matching the trace fragments against the bitstream of encoded bitstream by sliding the bitstream through the trace fragment and imposing the losses locations of the trace on the bitstream. Then the quality is measured by comparing the original stream with the degraded one using PESQ.

  – In one study [101] several traces for local and international connections were run and delay and packet Loss were recorded. Based on the measurements, the impairments in speech samples were simulated. With loose delay requirements present, it is expected to have high quality. When we have medium or tight delay requirement or when packet loss increases we can expect degradation to the quality. This way used to characterise the network conditions on specific routes rather than characterising a specific voice stream (e.g. voice call).
  – Similarly, other studies [2, 11] attempted assessing the quality of the network in general not a specific call to test whether the network is ready for IP telephony traffic. This is determined by automatically discovering the network topology, monitor devices, generate probe calls and analyse the performance of the calls to see whether the quality is good based on experienced delay and packet loss.

- It was proposed in one study [148] to use PESQ in a non-intrusive way in the sender side by asking the received side to check if frame erasure occurs to the received stream and if this is the case, the error pattern is sent back to the sender. In the sender side where the original stream is available, the received error pattern is imposed on the original stream to retrieve the degraded stream which is then decoded and compared against the original speech signal to calculate the PESQ score which can then be mapped into MOS score.
  Although using the above proposal the speech quality can be estimated non-intrusively, but it requires extra overhead in sending the error pattern back to the sender which may cause extra overhead in the network. Additionally, the error pattern may get lost in the way back to the sender if sent unreliably using UDP protocol and if sent reliably using TCP it causes extra overhead during session establishment.

- Kim and Tarraf from Lucent Technologies proposed a model for non-intrusive perceptual assessment of speech quality, although perceptual assessment is usually done in an intrusive way. In the proposed model, the functional role of human auditory system in judging the quality of speech is modelled. It

consists of critical-band filters, modulation filterbank, articulation analysis, and compensation stages. In contrast to the conventional intrusive objective speech quality methods, the proposed model estimates the quality of speech without the need for the reference speech information. Doing this is closer to real subjective MOS tests where the listener estimate the quality by listening to the received signal in case of MOS without listening to the original signal [96, 97].
Although the proposed scheme seems promising in assessing the speech quality non-intrusively, but it is not established yet as a mature standard and is still in development stages.

- A Genetic Algorithm (GA) approach was proposed by Raja et al. [145] to automatically estimate the speech quality non-intrusively. During simulation the reference of the quality was PESQ measurement of the quality. Different parameters that can be extracted from the receive signal are also fed into a square error equation to be optimised. The simulation is performed with different packet loss conditions and with the use of GA techniques the equation is optimised to make its estimation of the quality using extracted parameters as close as possible to PESQ. After the simulation the produced equation can be used to estimate the quality of the received signal non-intrusively by feeding the equation with the extracted parameters.

- Several studies estimate the speech quality non-intrusively by constructing a model of the behavior of speech or features (coding parameters) of undegraded signal and compare this model with the degraded signal, the degree of degradation can be estimated and mapped into MOS score. Several attempts were made to construct such a model, some attempts are based on Hidden Markov Model (HMM) [106, 176], other are based on Vocal tract modelling [54], while others are based on Gaussian Mixture Models [41, 42].

- Several methods have also been proposed to measure the speech quality using an Artificial Neural Network (ANN) approach by feeding the network to be trained with different input parameters and target MOS score. Then after the training process is finished, the resulted ANN will be used to estimate the quality by feeding it with different input parameters to produce estimation of the quality, such efforts include:

    - Sun and Ifeacher [165, 166] simulate several values for packet loss for sev-

eral speech coders and measure the quality using PESQ which is mapped into MOS score. This value is used as a target for an ANN model and different parameters include: speech coder, Gender, Unconditional Loss Probability (ULP), and Conditional Loss Probability (CLP) are used as input for ANN. The output of the PESQ after mapping into MOS is compared against the output of the ANN to train the ANN.
After the training process, the used speech coder and the gender of the talker are both extracted at the receiver side and used as input parameters to the ANN in addition to ULP and CLP as extracted from the received speech stream. The ANN is then used to produce estimation of the quality based on these four parameters [166].

- Masugi [119] proposed the use of self-organising neural network to map several input QoS-related parameters into quality level. QoS-related parameters such as end-to-end delay, packet loss, PSQM+, and PSQM+ under no load conditions are used as input to a self-organising neural networks to map this high-dimensional data into low-dimensional display (cartesian coordinate) where each quality level is concentrated in a specific area of the cartesian coordinates.
During the training phase end-to-end delay, packet loss rate and PSQM+ were measured for three different packet sizes and two speech coders (G.711 and G.729) and the self-organising network is used to map these values into clusters based on the combination of input variables. Using this method QoS-related parameters can be projected into a two-dimensional space, so that the position of QoS level can be determined for each condition composed of several variables. i.e. using this system the QoS input parameters can be automatically mapped into a QoS evaluation output.

- To estimate the speech quality intrusively, the reference signal is needed and as this signal is not available the receiver side, a method is proposed to find a replacement for this signal. A codebook is first appropriately constructed by clustering speech parameter vectors, extracted from an undistorted clean speech database using self-organising maps. After constructing the codebook, the codebook entries become a reference for computing objective auditory distance measures for distorted speech, i.e. the distortion between the original and degraded signal as normally performed in intrusive methods. The measure is performed by comparing the output speech to the closest match from the codebook as representative of the original signal. The median minimum distance is then used as a measure of the objective auditory distance [141, 142].

- Mohamed et al. [129] developed an ANN model to take account of the effect of different parameters on the speech quality concurrently. This approach is similar in principle to the previous three approaches that uses ANN as it aims at estimating the quality accurately. Mohammed et al. pointed to possible applications such as for control purposes, for pricing applications, etc. Six input parameters were considered which affect the perceived speech quality. Two of the parameters are network-related parameters namely, loss rate and mean size of loss burst, the effect of jitter is considered as part of network loss. The remaining four parameters concerning the encoding schemes used and they are:

  1 Coding algorithm used.

  2 Secondary encoding which is used to aid in Forward Error Correction (FEC).

  3 Redundancy offset which is the offset, in packets, between the original encoding and the redundant encoding.

  4 Packetisation interval (PI) which is the length (in ms) of audio contained in each packet (Usually called packet size).

  By taking different parameters into account as input to the ANN, the proposed method allows studying the impact of several source and network parameters on the quality. By doing this, the combined effect of different parameters can be studied.

  For training the ANN, a database of subjective test results (MOS scores) for different speech samples transmitted under different conditions was developed. One possible advantage of this approach is the possibility of adding another parameter(s) to the input of the ANN in case they proved to be relevant to the speech quality measurement.

- **Opinion Models - The E-model:**

With the wide variety of non-intrusive speech quality measurement, one of the most widely used methods for objectively evaluating the speech quality non-intrusively is opinion modelling. In opinion models the subjective quality factors are mapped into manageable network and terminal quality parameters to automatically produce an estimate of subjective quality and could be used as a network planning tool. The most famous standard for opinion modelling is the E-model which is defined according to ITU-T Recommendation G.107 [84, 175].

The E-model, abbreviated from the European Telecommunications Standards Institute (ETSI), was developed by a working group within ETSI during the work on ETSI Technical Report ETR 250 [35]. It is a computational tool originally developed as a network planning tool, but it is now being used for objectively estimating voice quality for VoIP applications using network and terminal quality parameters. In the E-model, the original or reference signal is not used to estimate the quality as the estimation is based purely on the terminal and network parameters. As such, the E-model is non-intrusive method of measuring the quality as it does not require the injection of the reference signal [84, 165, 175].

The network parameters such as packet loss rate can be estimated from information contained in the headers of Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP).

Several studies also investigated the accuracy of the E-model in estimating the quality which is investigated further in subsequent chapters [34, 88, 173, 174, 175].

An example of such studies is the work of Takahashi et al. in a series of studies where the performance of the E-model in estimating the speech quality in Japanese experiments in comparison to subjective tests is investigated. The experimental results showed that the E-model prediction sometimes diverges from the actual subjective quality in evaluating delay, talker echo and the interaction between delay and speech distortion although it accurately predicts subjective quality in evaluating loudness. A new model was proposed which model the interaction between the contributions of delay and speech distortion and remodelled the effects of delay, noise floor, and talker echo. Experimental results verified that the proposed model achieves better performance than the

E-model in terms of correlation coefficients between subjective and estimated quality in Japanese langauge [173, 174, 175].

The E-model was used in enormous number of studies for the purpose of network planning or to help the operator design or to live monitor the network. In one study [160] the E-model was used as part of fault monitoring system where the network was continuously monitored and if the E-model's rating is less than 70% out of 100% for a period of time, an alarm is raised and the network managers are notified of the failure. Having the E-model's rating less than 70% indicates degradation to the quality due to some network or terminal parameters and if the degradation persists for some time, network administrators should by notified to attempt to solve the problem.

Galiotos et al. [46] used the E-model to estimate the quality based on measurement of delay and packet loss. Based on the calculated R-Rating Factor (to be explained shortly) the used compression algorithm is selected. When the network congestion is low, the speech compression with highest possible quality is selected regardless of its bit rate as extra packets are not highly possible to degrade the network. With high level of congestion, speech coder with low bit rate and degraded quality is selected in order to mitigate the effect of network congestion.

Wang et al. [189] utilised a set of relay servers to route traffic. The best route or set of relay servers is selected based on the characteristics of different routes (packet delay and loss). These characteristics are then mapped into the E-model's R-Rating Factor and the route that offer the best R-Rating Factor is then selected for traffic. This route could differ for incoming from outgoing traffic and with time-varying Internet statistics the selected route could be changed dynamically during a call.

In the E-model, the subjective quality factors are mapped into manageable network and terminal quality parameters. Among the network quality parameters are: network delay and packet loss. Among the terminal quality parameters are: jitter buffer overflow, coding distortion, jitter buffer delay, and echo cancellation. Example of mapping is the mapping of delay subjective quality parameter into network delay and jitter buffer delay.

The fundamental principle of the E-model is based on a concept established by J. Allnatt around 20 years ago [5]:

> "Psychological factors on the psychological scale are additive"

It is used for describing the perceptual effects of diverse impairments occurring simultaneously on a telephone connection. Because the perceived integral quality is a multidimensional attribute, the dimensionality is reduced into one-dimension so-called transmission rating factor, R-Rating Factor. Based on Allnatt's psychological scale all the impairments are - by definition - additive and thus independent of one another.

In the E-model all factors responsible for quality degradation are summed on the psychological scale. Due to its additive principle, the E-model is able to describe the effect of several impairments occurring simultaneously.

The E-model is a function of 20 input parameters that represent the terminal, network, and environmental quality factors (quality degradation introduced by speech coding, bit error, and packet loss is treated collectively as an equipment impairment factor).

The E-model starts by calculating the degree of quality degradation due to individual quality factors on the same psychological scale. Then the sum of these values is subtracted from a reference value to produce the output of the E-model which is a single scalar value called the R-Rating Factor. The R-Rating Factor can lie in the range of 0 and 100 to indicate the level of estimated quality where R=0 represents an extremely bad quality and R=100 represents a very high quality. The R-Rating Factor can be mapped into a MOS score based on the G.107 ITU-T's Recommendation [84, 88] as explained later in this section. The reference model that represents the E-model is depicted in Figure 3.6 [84]. The input parameters to the E-model, beside their default values and permitted range are listed in Table 3.1.

By following the additive principle, the E- model is able to describe the effect of several impairments occurring simultaneously, the R-Rating Factor combines

Parameter	**Default value**	**Permitted range**
Send Loudness Rating	8	0...+18
Receive Loudness Rating	2	-5...+14
Sidetone Masking Rating	15	10...20
Listener Sidetone Rating	18	13...23
D-Value of Telephone, Send Side	3	3...+3
D-Value of Telephone, Receive Side	3	-3...+3
Talker Echo Loudness Rating	65	5...65
Weighted Echo Path Loss	110	5...110
Mean one-way Delay of the Echo Path	0	0...500
Round-Trip Delay in a 4-wire Loop	0	0...1000
Absolute Delay in echo-free Connections	0	0...500
Number of Quantisation Distortion Units	1	1...14
Equipment Impairment Factor	0	0...40
Packet-loss Robustness Factor	1	1...40
Random Packet-loss Probability	0	0...20
Burst Ratio	1	1 2
Circuit Noise referred to 0 dBr-point	-70	-80...-40
Noise Floor at the Receive Side	-64	
Room Noise at the Send Side	35	35...85
Room Noise at the Receive Side	35	35...85
Advantage Factor	0	0...20

Table 3.1: Default values and permitted ranges for the E-model's parameters

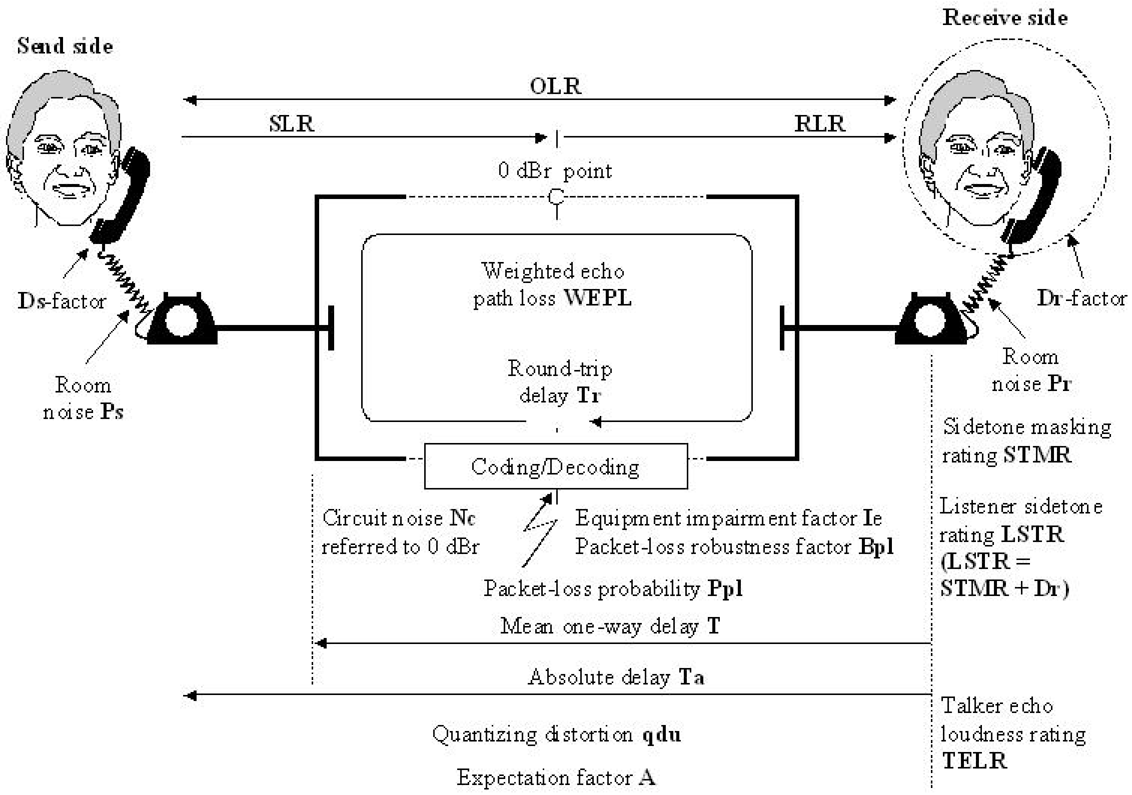

Figure 3.6: Reference connection of the E-model

the effects of various transmission parameters such as (packet loss, jitter, delay, echo, noise). The R-Rating Factor is calculated according to the following formula which follows the previous summation principle:

$$R = R_0 - Is - Id - Ie\text{-}eff + A \tag{3.5}$$

where

R_0	Basic signal-to-noise ratio (groups the effects of noise)
Is	Impairments which occur more or less simultaneously with the voice signal e.g (quantisation noise, sidetone level)
Id	Impairments due to delay, echo
$Ie\text{-}eff$	Impairments due to codec distortion, packet loss and jitter
A	Advantage factor or expectation factor (e.g. 10 for GSM)

The advantage factor captures the fact that users might be willing to accept some degradation in quality in return for the ease of access, e.g. users may find the speech quality is acceptable in cellular networks because of its access advantages. The same quality would be considered poor in the public circuit-switched telephone network. In the former case A could assign the value 10, while in the later case A would take the value 0 [34, 114, 115].

Each of the parameters in equation (3.5) except the Advantage factor (A) is further decomposed into a series of equations as defined in ITU-T Recommendation G.107 [84]. When all parameters set to their default values (Table 3.1), R-Rating Factor as defined in equation (3.5) has the value of 93.2 which is mapped to an MOS value of 4.41.

When the effect of delay is considered, the estimated quality according to the E-model is conversational, i.e. Mean Opinion Score - Conversational Quality Estimated MOS_{CQE}. When the effect of delay is ignored and Id is set to its default value the estimation is listening only, i.e. Mean Opinion Score - Listening Quality Estimated MOS_{LQE}.

The focus of this thesis is the quality measurement and quality degradation due to packet loss as defined in equation (3.5) which is characterised by packet loss dependent Effective Equipment Impairment Factor (*Ie-eff*), the effect of other parameters will not be considered and as such the default values for all the parameters except *Ie-eff*-related parameters will be used from Table 3.1). For example Id will be set to zero.

Packet loss dependent Effective Equipment Impairment Factor (*Ie-eff*) is calculated according to the following formula [84]:

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{3.6}$$

where

Ie	Codec-specific Equipment Impairment Factor
Bpl	Codec-specific Packet-loss Robustness Factor
Ppl	Packet loss Probability
$BurstR$	Burst Ratio (BurstR-to count for burstiness in packet loss)

The packet-loss dependent Effective Equipment Impairment Factor (*Ie-eff*) -as defined in equation (3.6) - is derived using codec-specific values for the Equipment Impairment Factor (Ie) and Packet-loss Robustness Factor (*Bpl*) at zero packet-loss. The values for Ie and *Bpl* for several codecs are listed in ITU-T Recommendation G.113 Appendix I [80] and they are derived using subjective MOS test results. For example for the speech coder defined according to the ITU-T Recommendation G.729 [71], the corresponding Ie and *Bpl* values are 11 and 19 respectively. On the other hand Packet loss Probability (*Ppl*) and Burst Ratio (*BurstR*-to count for burstiness in packet loss) depend on the packet loss presented in the system.

BurstR as defined by the latest version of the E-model represents

$$BurstR = \frac{\text{Average length of observed bursts in an arrival sequence}}{\text{Average length of bursts expected for the network under "random" loss}} \tag{3.7}$$

Based on equation (3.7), when packet loss is random (i.e., independent) $BurstR = 1$ and when packet loss is bursty (i.e., dependent) $BurstR > 1$.

The impact of packet loss in previous versions of the E-model (prior to the current version, 2005) was characterised by Equipment Impairment (Ie) factor. Specific impairment factor values for codec operating under random packet loss have been previously tabulated to be packet-loss dependent. In the current version of the E-model (2005), Packet-loss Robustness Factor (*Bpl*) is defined as codec-specific value and Ie is replaced by the packet-loss dependent Effective Equipment Impairment Factor *Ie-eff*.

Prior to the release of the current version of the E-model, Zhang et al. [211] proposed a modification to the E-Model (prior to 2005 version) to take account of packet loss burstiness. The proposal was to calculate burstiness level from the packet loss pattern in the stream, then this is converted into an equivalent random packet loss and used as usual in the E-Model calculations. This idea was realised by the ITU-T and thus they included $BurstR$ in the current release.

A simplified version of the R-Rating Factor as defined in (3.5) considers only the effect of packet loss. The simplified version is:

$$R = R_0 - Ie\text{-}eff \tag{3.8}$$

Using equation (3.8), the relation between *Ie-eff* and the R-Rating factor is shown in Figure 3.7.

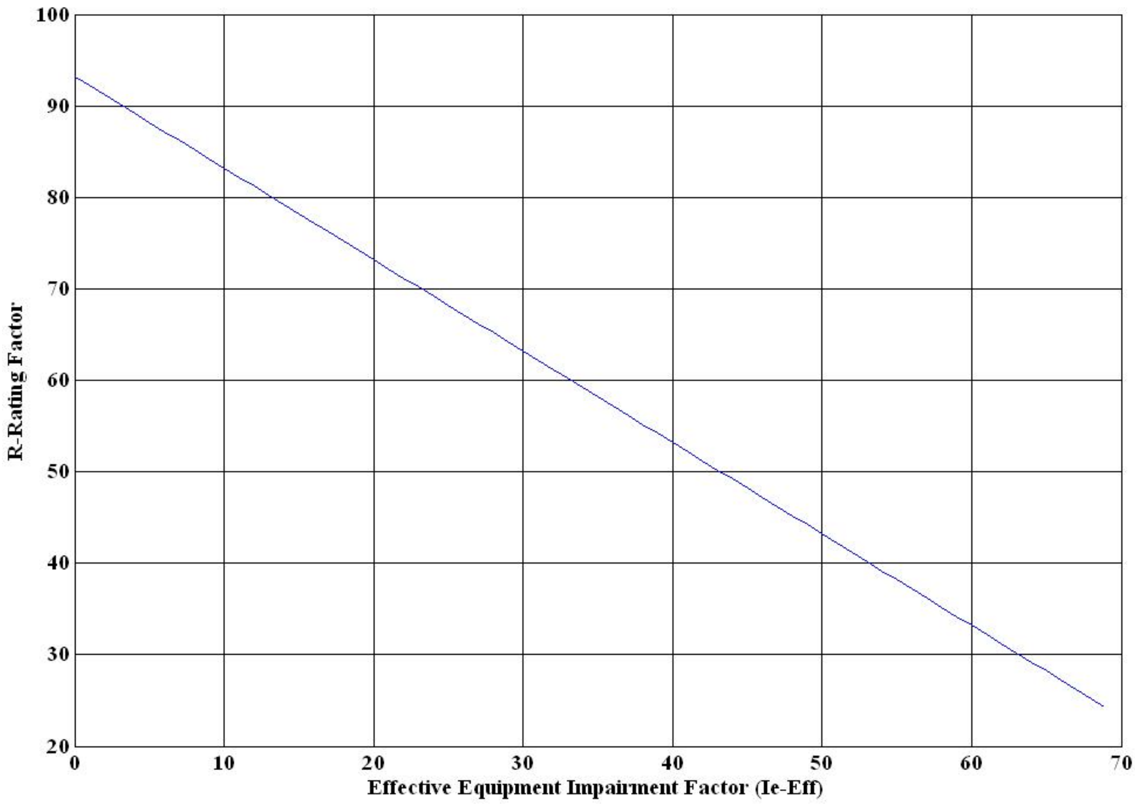

Figure 3.7: R-Rating Factor vs. *Ie-eff*

Mapping R-Rating Factor into MOS Value: The computed R-Rating Factor from equation (3.8) reflects the impairment caused by the packet loss only. This can be mapped into an MOS value. Equation (3.9) [84] gives the mapping function between the computed R-Rating Factor (in either the simplified or the original version of R-Rating Factor) and the MOS value. The relation between the R-Rating Factor and MOS is also shown in Figure 3.8.

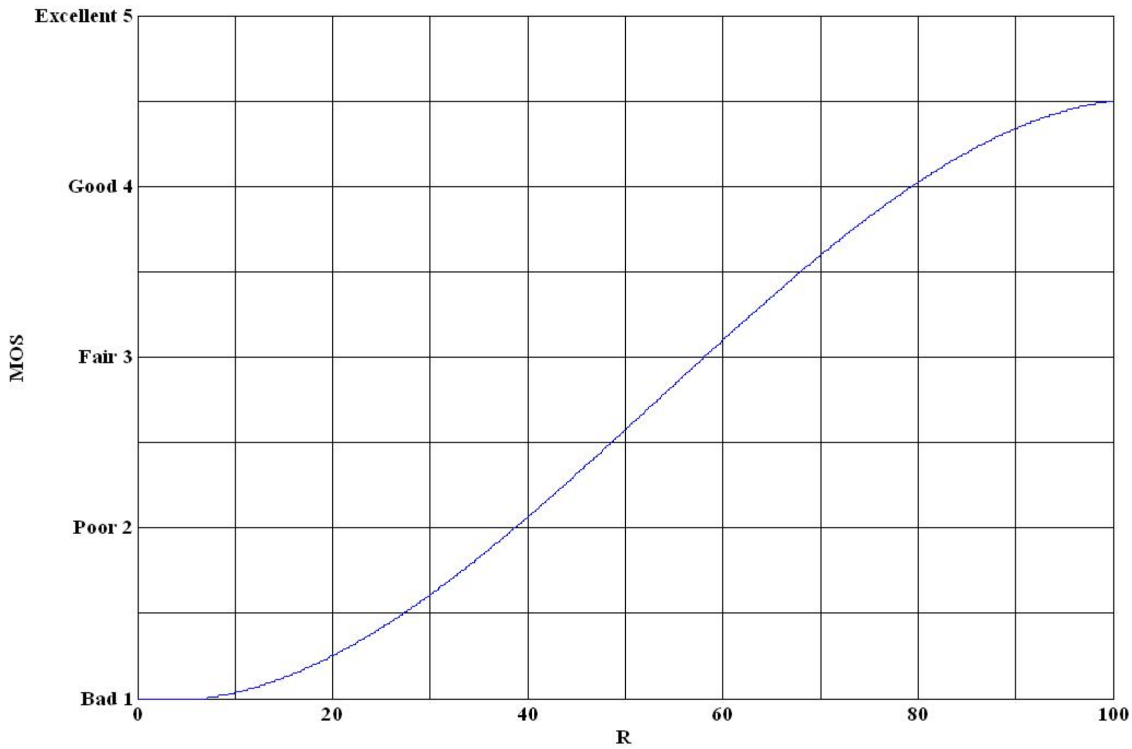

Figure 3.8: Mean Opinion Score (MOS) vs. R-Rating Factor

$$MOS = \left\{ \begin{array}{cc} 1 & R < 0 \\ 1 + 0.035R + R(R-60)(100-R).7.10^{-6} & 0 < R < 100 \\ 4.5 & R > 100 \end{array} \right\} \tag{3.9}$$

ITU-T Recommendation G.107 [84] also provides the formula to move back to R-Rating Factor from an available MOS score. The equation is:

$$R = \frac{20}{3}\left(8 - \sqrt{226}\left(h + \frac{\pi}{3}\right)\right) \tag{3.10}$$

with

$$h = \frac{1}{3}atan2\left(18566 - 6750MOS, 15\sqrt{-903522 + 1113960MOS - 202500MOS^2}\right) \tag{3.11}$$

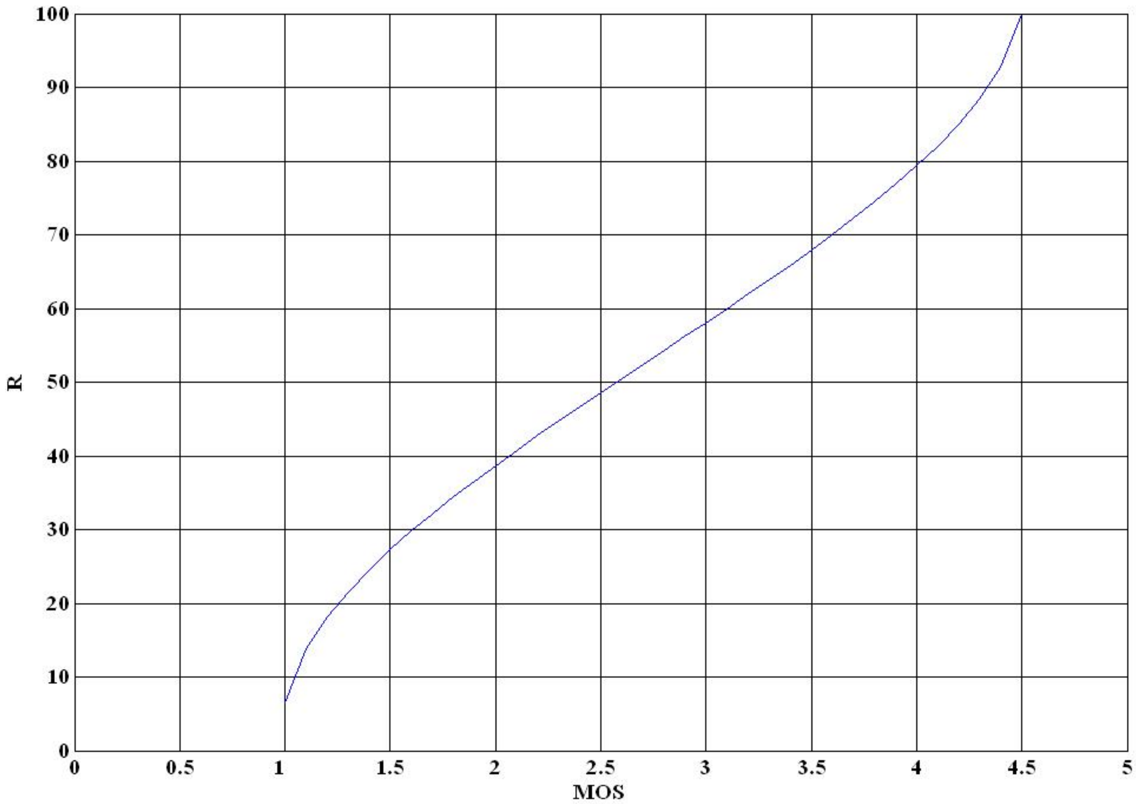

Figure 3.9: R-Rating Factor vs. Mean Opinion Score (MOS)

where

$$atan2(x,y) = \begin{Bmatrix} atan\left(\frac{x}{y}\right) & for\ x \geq 0 \\ \pi - atan\left(\frac{y}{-x}\right) & for\ x < 0 \end{Bmatrix} \tag{3.12}$$

Figure 3.9 shows the relation between the MOS score and the R-Rating factor.

E-model Tool: During this research a tool to measure the performance of the system using the latest version of the E-model (2005) [84] was developed. This tool can be used as an aid to check the effect of different parameters on the overall quality as estimated by the E-model. Mapping the R-Rating Factor to an equivalent MOS score is also provided. The Input window of the simulation tool is shown in Figure 3.10 where the user can enter the different input values while an example of the calculations of the E-model is shown in Figure 3.11. In both cases the calculated R-Rating Factor and the estimated MOS values are shown at the top of the window.

According to the E-model and as stated earlier, when all the parameters are

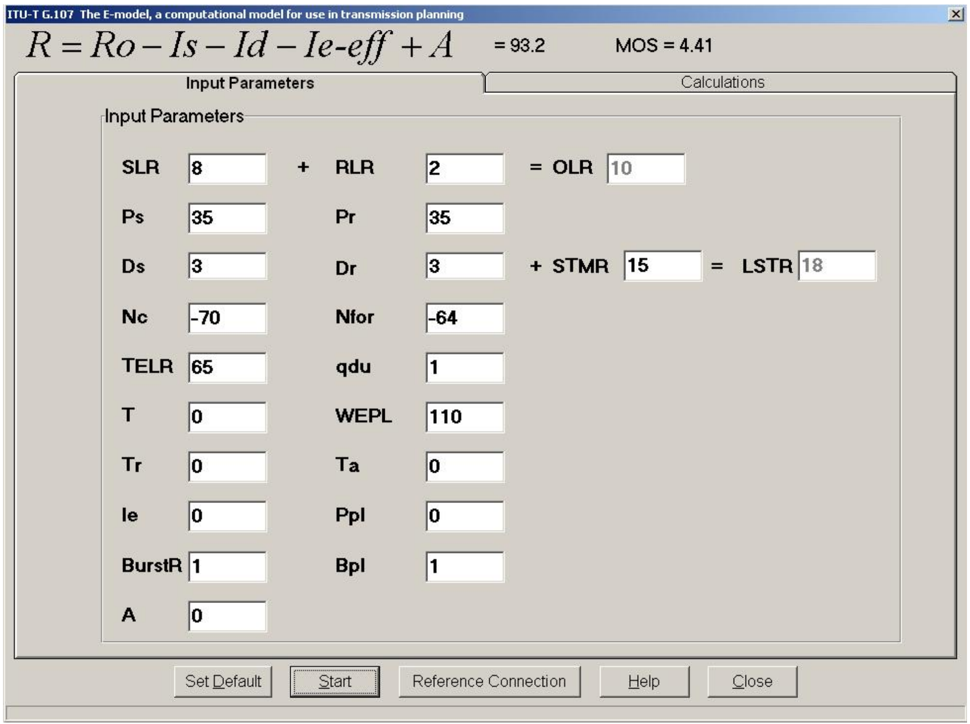

Figure 3.10: E-model Input Parameters

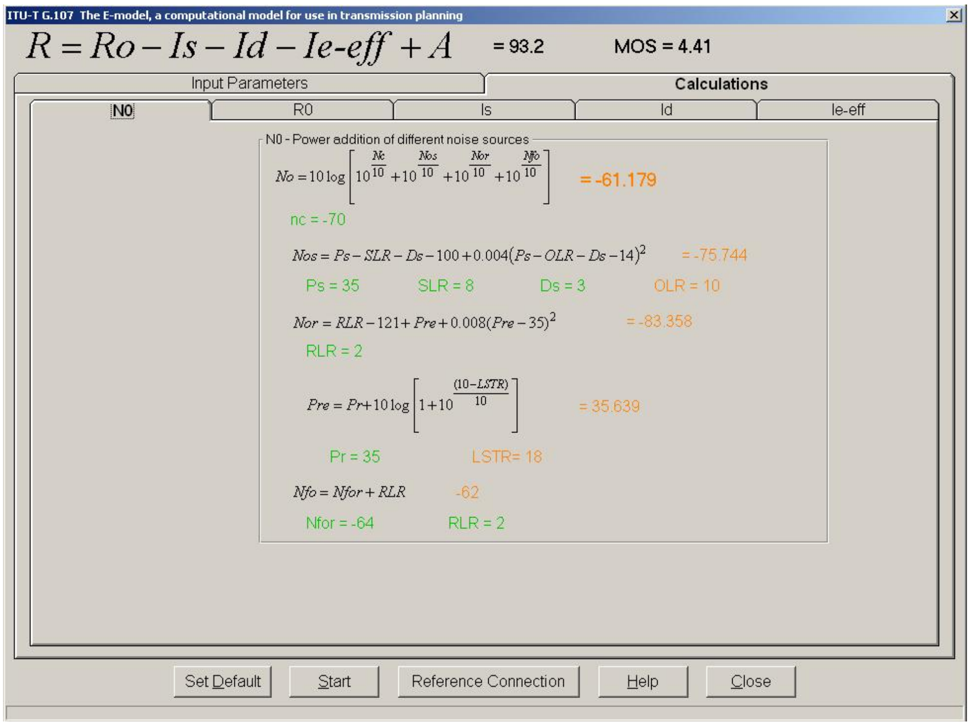

Figure 3.11: E-model Calculations

set to their default values in Table 3.1, the R-Rating factor has the value of 93.2 and the MOS is 4.41 which are both shown at the top of both figures.

R-Rating Factor and User Satisfaction: The calculated R-Rating Factor and the mapped MOS value can be translated into a user satisfaction as defined by ITU-T Recommendation G.109 [75] and listed in Table 3.2. Connections with R values below 50 are not recommended. Understanding the degree of user's and having a direct measurement of user's satisfaction is important for commercial reasons as a network that does not satisfy user's expectations is not expected to be a commercial success. If the quality of the network is continuously low, more percentage of users are expected to look for a another network with a consistent quality.

R-Rating factor	**MOS**	**Quality**	**User Satisfaction**
$90 \leq R < 100$	$4.34 \leq MOS < 4.50$	Best	Very Satisfied
$80 \leq R < 90$	$4.02 \leq MOS < 4.34$	High	Satisfied
$70 \leq R < 80$	$3.60 \leq MOS < 4.02$	Medium	Some users dissatisfied
$60 \leq R < 70$	$3.10 \leq MOS < 3.60$	Low	Many users dissatisfied
$50 \leq R < 60$	$2.58 \leq MOS < 3.10$	Poor	Nearly all users dissatisfied

Table 3.2: User satisfaction as defined by ITU-T Recommendation G.109

The E-model is good choice for non-intrusive estimation of voice quality, but it has some drawbacks. It is based on empirical formulae and as such it is applicable to a restricted number of codecs and network conditions (because subjective tests are required to derive model parameters) and this hinders its use in new and emerging applications. This feature is further discussed through out the thesis.

3.2.4 Requirements for a VoIP Speech Quality Assessment Solution

Measuring the speech quality for VoIP networks is important for commercial, technical and legal reasons. Several solutions for measuring the quality in VoIP networks have been discussed in this section. These solution varied from subjective solutions, to intrusive-based objective solutions to non-intrusive solutions.

Based on the nature of IP networks and the characteristics of voice traffic, it is determined that any such solution should satisfy the following requirements:

1 Automatic: It should be able to provide measurement of the speech quality in live networks.

2 Non-Intrusive: It should be able to provide measurement of the speech quality non-intrusively without the need for the original signal.

3 Accurate: It should provide accurate measurement of speech quality.

4 Non-Subjective based: It should be applicable to new and emerging applications without the need to run subjective tests.

Based on the above requirements and from the discussion in the previous sections, the non-intrusive solutions that have been discussed in section 3.2.3 are candidates for such task. The subjective and intrusive solutions discussed in sections 3.2.1 and 3.2.2 respectively can not be used for such task due to their nature.

The most famous and a widely used non-intrusive subjective solution for measuring the speech quality is the E-model as defined in ITU-T Recommendation G.107 and described in details in section 3.2.3.

However, the E-model as standardised by the ITU-T satisfies the first two requirements but does not satisfy the other two requirements from the above list as it depends on subjective tests to calibrate its parameters and due to its non-intrusive nature, it also does not consider the content of the signal in its calculations.

In the coming chapters of the thesis attempts will be made to avoid such deficits in the E-model utilising the intrusive-based PESQ solution as a base criteria. By utilising PESQ, the subjectivity in estimating the E-model's parameters is avoided as will be discussed in chapters 5 and 6. Also the contents of the received signal is analysed using PESQ as a base criteria and packet loss of the received signal is broken into Voiced and Unvoiced loss as will be discussed in chapter 7. Finally the above ideas are combined to offer a complete solution that can be used for measuring the speech quality objectively, non-intrusively, accurately and without the need for the subjective tests to calibrate its parameters. In other words a solution that satisfies all the above requirements.

Several efforts have been going on to extend the E-model based on the intrusive-based $PESQ$ speech quality prediction methodology [30, 31, 165, 168, 167, 169].

Ding and Goubran attempted to relate the E-model with PESQ [30, 31]. Their study relied on a previous version of the E-model, 2000 [78]. In older versions of the E-model specific impairment factor values for codec operating under random packet loss have been previously tabulated to be packet-loss dependent.

Several studies [14, 107] have shown dependency in packet loss based on Internet statistics. Based on these and similar studies and for the importance of taking burstiness into account, in the current version of the E-model, 2005 [84], Packet-loss Robustness Factor (*Bpl*) is defined as codec-specific value and Ie is replaced by the packet-loss dependent Effective Equipment Impairment Factor (*Ie-eff*) to take burstiness into account.

As Ding and Goubran did not consider the burstiness in packet loss which is something they pointed out in their study as a proposal for future work is something taken into consideration in this study.

Also in the extension proposed by Ding and Goubran, when the subjectivity in estimating E-model's parameters was avoided, they relied on the following formula:

$$Ie = Ie_{opt} + C1.ln(1 + C2.Ppl) \tag{3.13}$$

where

Ie_{opt}	Ie when packet loss is zero from ITU-T G.113
Ppl	Packet loss Probability
$C1, C2$	Curve fitting parameters

For example for the speech coder defined according to ITU-T Recommendation G.729 [71], $Ie_{opt} = 11$, $C1 = 25.21$ and $C2 = 0.150$ [30]. It should be noted that as this work depends on the published Ie values for the speech coder (11 in this case) from ITU-T Recommendation G.113 [80]. Consequently, it may be able to extend to new packet loss values but it is not able to extend to new speech coders as the work in this thesis is able to do.

On the other hand, the work proposed by Sun and Ifeachor [165, 168, 167, 169] avoided the need for the optimal Ie value (Ie_{opt}) as it was needed in Ding and Goubran's study. However, they still used the old E-model, 2000 [84] which does not consider the effect of burstiness on the speech quality.

Additionally, as the results of this work were not compared against the original E-model to prove its validity, some difference in quality estimation may arises as discussed in section 6.6 in chapter 6.

During their work, Sun and Ifeacher used a fitness curve to find the following relation

$$Ie = a.ln(1 + b.Ppl) + c \tag{3.14}$$

where

Ppl	Packet loss Probability
a, b, c	Curve fitting parameters

For the speech coder defined according to ITU-T Recommendation G.729 [71], they reported that the values of a, b and c to be 21.14, 0.1273 and 22.45, respectively [165, 168, 167, 169].

The selection of this fitness curve to find the relation between packet loss and Ie was not justified and other fitness curves could provide better extension of the E-model. Most importantly the relation between speech quality prediction according to the E-model and speech quality prediction according to PESQ which is used as a base criteria was not investigated at all as the assumption was made that the MOS values of the two models are equal. As any differences between the two models could lead to significant changes in the results and any possible the conclusions, this relation is investigated thoroughly in this thesis.

These limitations are avoided in this study as the latest E-model, 2005 [84] is considered to study the effect of burstiness. The accuracy of the prediction is also evaluated by comparing the quality of prediction using the proposed technique against the quality of the prediction using the current E-model. Also the relation between the quality according to the E-model and the quality according to PESQ is fully investigated which led us to some important and interesting findings. These extensions and measures are discussed in details in the coming chapters.

3.2.5 MOS Quality Qualifiers

ITU-T Recommendation's P.800.1 gives a clear terminology distinction among different MOS terms whether the test is listening or conversational and whether it a result of subjective or objective test [83]. In the recommendation it is stated that the identifiers in Table 3.2.5 are to be used:

LQ	Listening Quality
CQ	Conversational Quality
S	Subjective
O	Objective
E	Estimated

Table 3.3: MOS Qualifiers

It is recommended to use these identifiers together with the MOS to avoid confusion and distinguish the area of application. The result of such qualification is [72, 79, 83, 84]:

- **Subjective Tests**
  - **Listening Quality:** For the score collected by calculating the arithmetic mean of listening subjective tests conducted according to Recommendation P.800, the results are qualified as Mean Opinion Score - Listening Quality Subjective or MOS_{LQS}.
  - **Conversational Quality:** For the score collected by calculating the arithmetic mean of conversational subjective tests conducted according to Recommendation P.800, the results are qualified as Mean Opinion Score - Conversational Quality Subjective or MOS_{CQS}.

- **Network Planning Estimation Tests**

  - **Listening Quality:** For the score calculated by a network planning tool to estimate the listening quality according to Recommendation G.107 and then transformed into mean opinion score, the results are qualified as Mean Opinion Score - Listening Quality Estimated or MOS_{LQE}.

  - **Conversational Quality:** For the score calculated by a network planning tool to estimate the conversational quality according to Recommendation G.107 and then transformed into mean opinion score, the results are qualified as Mean Opinion Score - Conversational Quality Estimated or MOS_{CQE}.

- **Objective Tests**

  - **Listening Quality:** For the score calculated by an objective model to predict the listening quality according to Recommendation P.862 and then transformed into mean opinion score, the results are qualified as Mean Opinion Score - Listening Quality Objective or MOS_{LQO}.

  - **Conversational Quality:** For the score calculated by an objective model to predict the conversational quality according to Recommendation P.562 and then transformed into mean opinion score, the results are qualified as Mean Opinion Score - Conversational Quality Objective or MOS_{CQO}.

The relation between different MOS qualifiers is depicted in Figure 3.12 where the related speech signal and the MOS from the subjective tests, E-model and PESQ are related together.

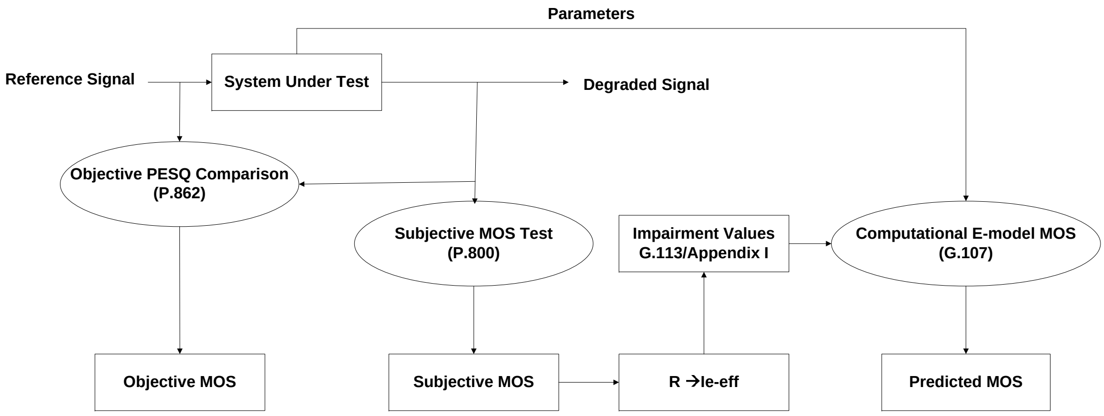

Figure 3.12: Relationship between MOS qualifiers

3.3 Summary

In this chapter preliminary information about QoS and different methods for measuring the speech quality is presented. In section 3.1 the term QoS is introduced with various definitions to it along with several QoS network solutions and QoS policy protocols. Section 3.2 discusses different subjective and objective speech quality measurement methods. Among objective measurement methods intrusive and not-intrusive methods are discussed and the existing approaches are discussed.

Chapter 4

Simulation of Basic System Components

This chapter discusses simulation of some basic system components and verification of correct implementation and conformance of the developed simulation modules to the specification stated for them. As the developed tools and obtained results in this chapter are common for the subsequent chapters, therefore putting them in a separate chapter avoids the need to repeat them in every related chapter and at the same time makes referring to them when needed an easy task.

Section 4.1 explains some of the main simulation modules used in this thesis, this include: speech codec as defined by the ITU-T Recommendation G.729, objective measurement of speech quality according to ITU-T Recommendation P.862, and objective estimation of speech quality according to Recommendation G.107.

Section 4.2 describes the speech materials used in the experiments and on which basis they were selected and their characteristics. Also this section investigates the effect of speech encoding according to ITU-T Recommendation G.729 on the quality of this speech data set measured using Recommendation P.862.

Section 4.3 explains how packet loss is modelled in this thesis and what assumptions are made. It is important to understand this section as it simplifies the description of many sections in subsequent chapters.

4.1 Simulation Modules

In this section some important simulation modules are described.

Most of the simulation used in this thesis is written in MATLAB while most of the reference implementation for the used Recommendations as provided by the standardisation bodies is written in C/C++. In order to avoid rewriting the C/C++ code in MATLAB in the simulation, the C/C++ implementation provided by the ITU-T is invoked from within MATLAB. The decision to do so comes from three reasons:

- To save the time required to re-implement these standards in MATLAB.
- To make sure the correct implementation of Recommendations is used in the simulation.
- Some of these standards state explicitly that if another implementation is to be used for the Recommendation in question, a set of very rigid conformance tests are required to make sure the new implementation is inline with the standard. This is the case specially with ITU-T Recommendation P.862 standardising Perceptual Evaluation of Speech Quality (PESQ) algorithm [79]

However, to make sure the results of the MATLAB call to the Recommendations' reference implementation is the same as the output of the C/C++, a series of tests are performed. These conformance tests are different depending on the Recommendation in question and this is described over the next few sections.

The call to the C/C++ is implemented using MEX calls [61, 121].

Section 4.1.1 discusses the implementation of the G.729. Section 4.1.2 discusses the implementation of PESQ. Section 4.1.3 describes the implementation used for the E-model.

4.1.1 G.729 Codec

The original ANSI C code for the G.729 Encoder/Decoder as implemented by the ITU-T Recommendation G.729 consists of a set of C and header files [71]. The two main programs that simulates the encoder and the decoder are called as follows:

```
coder inputfile bitstreamfile
```

The inputfile is a sampled data file containing 16 bit Pulse Code Modulation (PCM) input signal. The bitstreamfile is the output of the coder and it contains the encoded parameters to be sent over the network.

The main program which simulates the decoder is called as follows:

```
decoder bitstreamfile outputfile
```

The bitstreamfile contains the encoded set of parameters transmitted over the network in order to be decoded. The output file is sampled data file containing 16 bit PCM decoded signal, the same format as the original inputfile. If the output of the encoder is directly fed into the input of the decoder and the inputfile is compared with the outputfile, any degradation in the quality is a pure result of the coder distortion.

Usually in a lossy network where packets could be lost or corrupted, the output of the encoder is not the same as the input to the decoder and the received bitstream would be a corrupted version of the sent bitstream. The loss in IP network is simulated by introducing the loss to the bitstream as this bitstream is what is transmitted over the network between the call parties. The degraded bitstream is then decoded to get the degraded speech signal. This is shown in Figure 4.1.

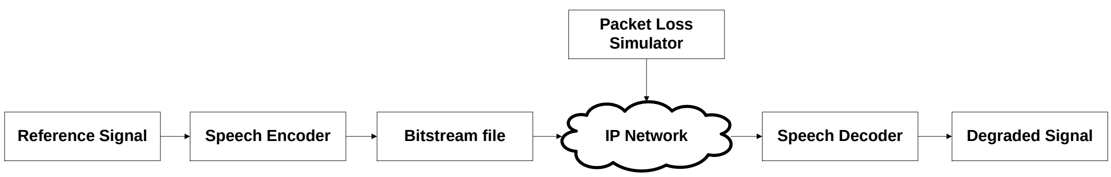

Figure 4.1: Loss Simulation for the bitstream

In order to integrate the code for G.729 standard with the MATLAB simulation, the following changes are made to the standard implementation:

- The main function in the encoder and decoder is renamed to mexFunction function as required by MEX calls.
- The input to the decoder is a matrix containing the bitstream values to be

decoded and a matrix where the decoded speech signal should be placed. In the original decoder the input was the filename where the bitstream values are stored and the file name where the decoded speech signal should be stored.

The main program which simulates the decoder is called as follows:

```
outputMatrix=decoder(bitstreamMatrix)
```

The bitstreamMatrix is the name of the matrix containing the bitstream values to be decoded. The outputMatrix is a matrix where the 16 bit PCM decoded signal should be placed.

Using 32 speech files (American English and British English each with 8 male and 8 female) taken from ITU-T Recommendation P.50, Appendix I [73] (as described in section 4.2), a set of tests are performed to make sure the call to the G.729 decoder from MATLAB gives the same results as the C implementation provided with the Recommendation for G.729 Codec, the performed tests are:

- The bitstream for all 32 files are decoded using the C code and decoded again using the MATLAB call to the decoder. The two set of files are compared by size and content (using Total Commander v6.53 [22]) and they both were the same for all cases.

- For all 32 files the decoded file using the C code is compared against the decoded file using MATLAB call to the decoder in terms of the PESQ. In all 32 cases the quality was 4.5 (the maximum possible value for PESQ) which indicates there is no degradation between the two files. i.e. the contents of the two files are the same.

- For all 32 files, the original file before encoding is compared against the file decoded using the C implementation from the standards. Again the original file before encoding is compared against the file decoded using the MATLAB call to the decoder. In all 32 cases the quality was equal which indicates the two decoded files are the same.

4.1.2 Perceptual Evaluation of Speech Quality

The original ANSI C code for PESQ as implemented by the ITU-T Recommendation P.862 consists of a set of C and header files. The main program that simulates the PESQ program is called as follows from the command prompt:

```
pesq +samplingfrequency OriginalSpeechSignal DegradedSpeechSignal
```

The sampling frequency could be 8000 or 16000 depending on whether narrow band or wide band signal is used. OriginalSpeechSignal is the original speech signal without degradation and before transmission. DegradedSpeechSignal is the degraded speech signal after coder distortion and/or network impairments are introduced. The output would be the quality of the degraded signal in comparison with the original signal in terms of PESQ score and mapping of this quality into Mean Opinion Score - Listening Quality Objective (MOS_{LOQ}).

If the name of the OriginalSpeechSignal is the same as the name of the DegradedSpeechSignal, meaning the speech file is compared to its self, the resulting PESQ quality should be 4.5 which is the maximum possible PESQ score due to the absence of degradation as the degraded signal is the same as the original signal.

In order to integrate the code for PESQ standard with the MATLAB simulation, the following changes are made to the standard implementation:

- The main function in the PESQ C code is renamed to mexFunction function as required by MEX calls.
- The input to the PESQ algorithm is a scalar value representing the sampling frequency preceded by + sign as required by PESQ code, a matrix containing the reference or the original speech signal values before degradation and a matrix containing the degraded speech signal values after coder and/or network degradation. In the original code the input is a scalar value representing the sampling frequency preceded by + sign as required by PESQ code, the filename where the original signal is stored, and the filename where the degraded signal is stored.

Using the MATLAB simulator, the main program which simulates the PESQ is called as follows:

```
[PESQ_Score,MOS_Score]  =  PESQ('+8000',OriginalSpeechMatrix,
                                DegradedSpeechMatrix)
```

The OriginalSpeechMatrix is a matrix containing the original speech signal values while DegradedSpeechMatrix is a matrix containing the degraded speech signal values. PESQ_Score and MOS_Score are the PESQ and MOS scores when the degraded speech signal is compared with the original speech signal.

Using 40 pairs of speech signals provided with the PESQ standard for the purpose of conformance test, the MATLAB code for PESQ is called and the resulted quality is compared against the reported values in the Recommendation [79, 85]. The result of this comparison is listed in Table 4.1 and Table 4.2.

The following comments are made about this conformance test quality

- For the 40 pairs the absolute difference in PESQ score between the Recommendation and from MATLAB is less than 0.01.
- For 35 pairs out of 40 the absolute difference in PESQ score between the Recommendation and from MATLAB is less than 0.001. For the remaining 5 files the difference is ± 0.001 which could be due to different rounding operations between different C compilers.

Recommendation P.862 [79] states that an implementation passes the conformance test when the absolute difference in PESQ score compared to the reference implementation is less than 0.05 in 39 of the 40 file pairs. A single file pair is allowed to have an absolute difference in PESQ score of less than 0.5. This may be any one of the 40 file pairs. This condition is satisfied in the MATLAB call as appears in the tables.

The above tests give enough confidence that the MATLAB code for PESQ gives the same results with a very good approximation as the results obtained from the C code provided by the Recommendation and the results provided by this version of PESQ are good enough to rely on.

In addition to the above implementation of the PESQ algorithm, two additional functions are implemented to perform the mapping between the PESQ score and MOS_{LQO}.

MOS_{LQO}=`mostopesq(PESQ)`

Reference Signal	Degraded Signal	C PESQ	MATLAB PESQ	Difference
or105.wav	dg105.wav	2.237	2.237	0.000
or109.wav	dg109.wav	3.180	3.180	0.000
or114.wav	dg114.wav	2.147	2.147	0.000
or129.wav	dg129.wav	2.680	2.680	0.000
or134.wav	dg134.wav	2.365	2.365	0.000
or137.wav	dg137.wav	3.670	3.670	0.000
or145.wav	dg145.wav	3.016	3.016	0.000
or149.wav	dg149.wav	2.558	2.558	0.000
or152.wav	dg152.wav	2.768	2.769	0.001
or154.wav	dg154.wav	2.694	2.694	0.000
or155.wav	dg155.wav	2.606	2.606	0.000
or161.wav	dg161.wav	2.608	2.608	0.000
or164.wav	dg164.wav	2.850	2.851	0.000
or166.wav	dg166.wav	2.527	2.527	0.000
or170.wav	dg170.wav	2.452	2.452	0.000
or179.wav	dg179.wav	1.828	1.828	0.000
or221.wav	dg221.wav	2.774	2.774	0.000
or229.wav	dg229.wav	2.940	2.940	0.000
or246.wav	dg246.wav	2.205	2.205	0.000
or272.wav	dg272.wav	3.288	3.288	0.000
u_am1s01.wav	u_am1s01b1c1.wav	3.483	3.483	0.000
u_am1s01.wav	u_am1s01b1c7.wav	2.420	2.420	0.000
u_am1s02.wav	u_am1s02b1c9.wav	4.042	4.042	0.000
u_am1s01.wav	u_am1s01b1c15.wav	3.179	3.180	0.001
u_am1s03.wav	u_am1s03b1c16.wav	2.872	2.872	0.000
u_am1s03.wav	u_am1s03b1c18.wav	2.806	2.806	0.000
u_am1s01.wav	u_am1s01b2c1.wav	4.300	4.300	0.000

Table 4.1: Comparison between Recommendation PESQ results and MATLAB PESQ results

Reference Signal	Degraded Signal	C PESQ	MATLAB PESQ	Difference
u_am1s02.wav	u_am1s02b2c4.wav	3.634	3.634	0.000
u_am1s02.wav	u_am1s02b2c5.wav	3.369	3.369	0.000
u_am1s03.wav	u_am1s03b2c5.wav	3.911	3.911	0.000
u_am1s03.wav	u_am1s03b2c6.wav	2.905	2.906	0.001
u_am1s03.wav	u_am1s03b2c7.wav	3.579	3.579	0.000
u_am1s01.wav	u_am1s01b2c8.wav	2.198	2.199	0.001
u_am1s03.wav	u_am1s03b2c11.wav	3.276	3.276	0.000
u_am1s02.wav	u_am1s02b2c14.wav	3.316	3.316	0.000
u_af1s01.wav	u_af1s01b2c16.wav	3.307	3.307	0.000
u_af1s03.wav	u_af1s03b2c16.wav	3.592	3.592	0.000
u_af1s02.wav	u_af1s02b2c17.wav	2.614	2.614	0.000
u_af1s03.wav	u_af1s03b2c17.wav	2.806	2.806	0.000
u_am1s03.wav	u_am1s03b2c18.wav	2.540	2.540	0.000

Table 4.2: Comparison between Recommendation PESQ results and MATLAB PESQ results

This function implements the mapping function between the computed PESQ score into an MOS_{LQO} according to equation (3.2).

`PESQ=mostopesq(`MOS_{LQO}`)`

This function implements the mapping function between an MOS_{LQO} value into a PESQ score according to equation (3.3).

4.1.3 The E-model

The E-model as defined in the ITU-T's Recommendation G.107 [84] implements a set of complex equations to estimate the quality in terms of R-Rating Factor which can be mapped into MOS value.

As the main theme of this thesis is speech quality and the effect of packet loss on it, the other factors are set to their default values in Table 3.1. This simplifies the E-model into:

$$R = R_0 - Ie\text{-}eff \tag{4.1}$$

Recall that packet loss dependent Effective Equipment Impairment Factor (*Ie-eff*) is calculated according to the following formula [84]:

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{4.2}$$

where

Ie	Codec-specific Equipment Impairment factor
Bpl	Codec-specific Packet-loss Robustness Factor
Ppl	Packet loss Probability
$BurstR$	Burst Ratio (BurstR-to count for burstiness in packet loss)

The main program which simulates the E-model is called as follows:

```
[R,MOS]=Emodel(Ie,Ppl,BurstR,Bpl)
```

This program takes the packet loss input parameters and set the rest of parameters to their default values and calculate an estimate of the quality in terms of R-Rating Factor which is also mapped into MOS value using equation (3.9).

In addition to the above implementation of the E-model, two additional functions are implemented to perform the mapping between the R-Rating Factor and MOS.

```
MOS=rtomos(R)
```

This function implements the mapping function between the computed R-Rating Factor into an MOS score according to equation (3.9).

```
R=mostor(MOS)
```

This function implements the mapping function between an MOS value into R-Rating Factor according to equation (3.10).

To study the effect of packet loss on the speech quality measured by the E-model on speech coded using the G.729 speech coder, experiments with different network scenarios are conducted. Different values for Ppl and $BurstR$ are attempted as the other parameters namely: Ie and Bpl have fixed values for a specific speech coder.

According to the E-model, the permitted range for Ppl is 0 to 20 and for $BurstR$ is 1 to 2. For G.729 speech coder the values of Ie and Bpl are 11 and 19 respectively. R-Rating Factor and MOS values for G.729 speech coder corresponding to each possible combination of Ppl and $BurstR$ were calculated. The result of such calculations is listed in Table 4.3.

Figure 4.2 also shows the functional form of the relation between speech quality in terms of MOS (as defined by the E-model) with both Ppl and $BurstR$.

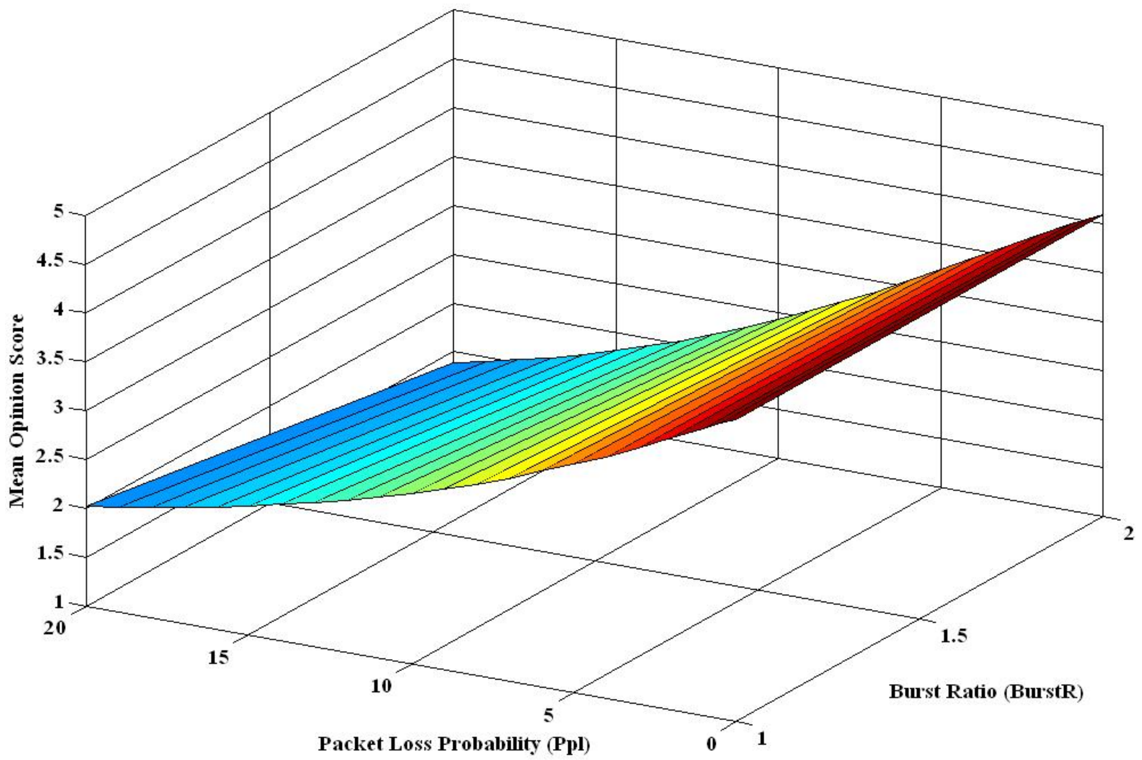

Figure 4.2: MOS vs. Packet Loss Probability and Burst Ratio

BurstR	Ppl	Ie-eff	R	MOS	BurstR	Ppl	Ie-eff	R	MOS
1	0	11.000	82.20	4.10	2	0	11.000	82.20	4.10
1	0.5	13.154	80.10	4.03	2	0.5	13.182	80.00	4.02
1	1	15.200	78.00	3.95	2	1	15.308	77.90	3.94
1	2	19.000	74.20	3.79	2	2	19.400	73.80	3.77
1	3	22.455	70.80	3.63	2	3	23.293	69.90	3.59
1	4	25.609	67.60	3.48	2	4	27.000	66.20	3.41
1	5	28.500	64.70	3.34	2	5	30.535	62.70	3.24
1	6	31.160	62.00	3.21	2	6	33.909	59.30	3.06
1	7	33.615	59.60	3.08	2	7	37.133	56.10	2.89
1	8	35.889	57.30	2.96	2	8	40.217	53.00	2.73
1	9	38.000	55.20	2.85	2	9	43.170	50.00	2.58
1	10	39.966	53.20	2.75	2	10	46.000	47.20	2.43
1	11	41.800	51.40	2.65	2	11	48.714	44.50	2.29
1	12	43.516	49.70	2.56	2	12	51.320	41.90	2.16
1	13	45.125	48.10	2.47	2	13	53.824	39.40	2.03
1	14	46.636	46.60	2.40	2	14	56.231	37.00	1.92
1	15	48.059	45.10	2.32	2	15	58.547	34.70	1.81
1	16	49.400	43.80	2.25	2	16	60.778	32.40	1.71
1	17	50.667	42.50	2.19	2	17	62.927	30.30	1.62
1	18	51.865	41.30	2.13	2	18	65.000	28.20	1.54
1	19	53.000	40.20	2.07	2	19	67.000	26.20	1.46
1	20	54.077	39.10	2.02	2	20	68.931	24.3	1.39

Table 4.3: Ie-eff,R-Rating Factor and MOS for different possible values of Ppl and $BurstR$ for speech coder G.729

4.2 Speech Material

In this section, the speech sources used in the experiments are described. It is important that the used test signals are representative of the real speech signals carried by communications networks as networks treat speech and silence differently and coding algorithms are often highly optimised for speech - and so may give meaningless results if they are tested with signals that do not contain the key temporal (including silent intervals) and spectral properties of speech.

Also when the signals are processed by the system in hand which is an IP network in this case, it is important to be able to measure the effect the system has on the speech signal, therefore the used signal should be undistorted, with high-quality. In this way any distortion in the output signal can be referred to the system and any further processing performed on the signal rather than due to the choice the signal or way the signal was recorded.

In selecting the test signals there were two choices:

1 To record the speech signals to be used according to ITU-T Recommendation P.830 in strict lab conditions regarding setting the recording environment and selecting the recording material. OR

2 To use pre-recorded signals, that known to satisfy the above conditions, which are recorded by a trusted party.

The first option allows us to control the content of the recorded speech signal and to experiment with the recording environment. However, this approach was not chosen due to two reasons:

1 The needed recording environment is unavailable except in few Labs in the UK such as BT, which requires special arrangement to access.

2 Although the test signals are important in this research but the recording procedure is not the main theme of this thesis and as such spending any unnecessary time on recording such signals will not add value to this research.

Due to the above reasons, it was decided to choose the second option which is using pre-recorded signals. Such signals should satisfy the necessary conditions of being representative of real speech signals carried by communications networks that contain the temporal and spectral properties of speech. Also the recorded signals should be noise-free similar to the recording conditions in absence of any interference or distortion in order to be able to study the behaviour of the system accurately.

One obvious choice for the test signals is to use artificial voice signals provided by the ITU-T in Recommendations P.50 and P.50 Appendix I [73, 77].

Artificial voice as described in Recommendation P.50 is a signal that is mathematically defined to reproduce the time and spectral characteristics of human speech to characterise telecommunication systems intended for speech transmission. Two kinds of artificial voices are defined, reproducing respectively the characteristics of female and male speech. Among the characteristics reproduced by the artificial voices is voiced and unvoiced structure of speech waveform and long and short-term spectrum [77].

This data set contains speech signals spoken in different languages, these languages include: English (British), American English, Arabic, Chinese, Danish, Dutch, Finnish, French, German, Greek, Hindi, Hungarian, Italian, Japanese, Norwegian, Polish, Portuguese, Russian, Spanish, and Swedish. For each language there are 16 speakers, 8 Female and 8 Male.

As the purpose of this study was not to study language dependency, only English signals are studied in this research. The applicability of other signals spoken by other languages was not studied although the approaches followed were general and their applicability to other languages is possible but not tested in this thesis.

Although artificial speech signals are artificial in nature and does not sound natural but they represent the temporal structure and phonetic structure of real speech signals. Also the use of the artificial voice instead of real speech has the advantage of both being more easily generated and having a smaller variability than samples of real voice.

Next are 3 examples of English sentences from Recommendation P.50, Appendix I.

"I was away for nine weeks. The dining-room was lit by gas. There were no vegetables left."

"It's human nature to blame another. He had completely forgotten his hat. I did not wish him to know."

"The act was a deliberate murder. He was attracted by her face. The timber fell across the road."

PESQ of G.729 Encoded Artificial Voice:

For each of 16 speakers, 8 Female and 8 Male in British English and American English the speech is encoded using G.729 codec to study the effect of encoding on quality at zero packet loss. For each of the 16 files the extension for the input file is .16p as provided by ITU-T Recommendation P.50. Each one of these files is encoded and then the result is decoded again, then the two files were compared using PESQ algorithm provided in ITU-T's Recommendation P.862 and implemented in the previous section to get the PESQ score. This PESQ score is then mapped into MOS_{LQO} using the equation provided in ITU-T Recommendation P.862.1 as discussed in section 3.2.2.

Table 4.4 lists the filename with the PESQ scores and the mapped MOS_{LQO} values for each of the 32 files included. The file names (speech signal) is the basic filename without any extension.

4.3 Packet Loss Model

As the problem of packet loss is inevitable in IP networks, it is needed to model the behaviour of a network with packet loss. For this purpose and to simulate packet loss in the speech signal stream, a 2-state Markov model is constructed. The 2-state Markov model has transition probabilities p between a "Found" and a "Loss" state, and q between a "Loss" and a "Found" state as depicted in Figure 4.3. The system suffers from bursty packet loss if it remains in "Loss" state [84].

Prior to this model packet loss was assumed to be random and modelled by a bernoulli model, but many studies revealed that packet loss can exhibit temporal dependency or bursts, which degrade the effectiveness of packet loss recovery techniques and raise the need to model such burstiness in packet loss [19, 134, 140, 152,

File Name	PESQ	MOS	R
American English - Female			
A_eng_f1	3.7228	3.8453	75.5496
A_eng_f2	3.7488	3.8769	76.3017
A_eng_f3	3.4729	3.5166	68.3064
A_eng_f4	3.7053	3.8237	75.0426
A_eng_f5	3.5689	3.6477	71.0901
A_eng_f6	3.5825	3.6658	71.4839
A_eng_f7	3.6086	3.7003	72.2417
A_eng_f8	3.5262	3.5900	69.8509
Average	**3.6170**	**3.7083**	**72.4834**
American English - Male			
A_eng_m1	3.6589	3.7654	73.7004
A_eng_m2	3.3173	3.2938	63.7874
A_eng_m3	3.2571	3.2053	62.0453
A_eng_m4	3.6862	3.7999	74.4902
A_eng_m5	3.5147	3.5743	69.5177
A_eng_m6	3.5483	3.6201	70.4944
A_eng_m7	3.6190	3.7138	72.5410
A_eng_m8	3.6400	3.7412	73.1535
Average	**3.5302**	**3.5892**	**69.9662**

File Name	PESQ	MOS	R
British English - Female			
B_eng_f1	3.7535	3.8826	76.4388
B_eng_f2	3.3037	3.2740	63.3955
B_eng_f3	3.2666	3.2193	62.3193
B_eng_f4	3.7307	3.8549	75.7767
B_eng_f5	3.4335	3.4611	67.1592
B_eng_f6	3.3745	3.3769	65.4476
B_eng_f7	3.7126	3.8327	75.2532
B_eng_f8	3.5406	3.6096	70.2692
Average	**3.5145**	**3.5639**	**69.5074**
British English - Male			
B_eng_m1	3.3261	3.3067	64.0435
B_eng_m2	3.4280	3.4533	66.9992
B_eng_m3	3.2628	3.2137	62.2096
B_eng_m4	3.5041	3.5597	69.2093
B_eng_m5	3.2046	3.1275	60.5317
B_eng_m6	3.4346	3.4628	67.1941
B_eng_m7	3.4124	3.4312	66.5476
B_eng_m8	3.3719	3.3731	65.3711
Average	**3.3681**	**3.3660**	**65.2633**

Table 4.4: Basic PESQ and MOS scores due to only the coder distortion at 0 packet loss for G.729 Encoded Artificial Voices

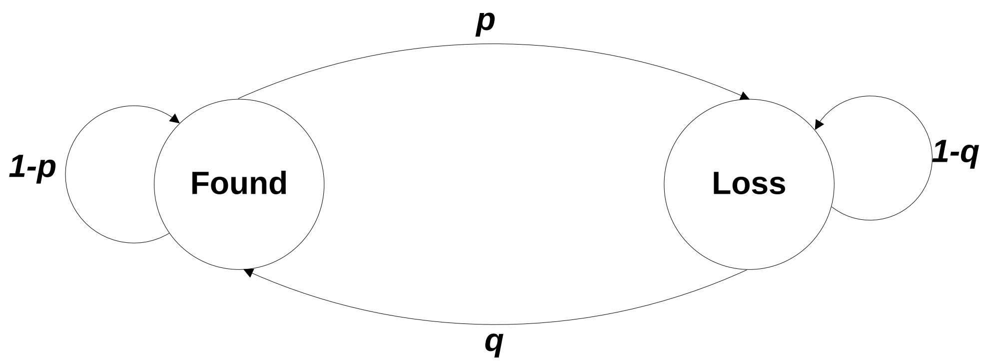

Figure 4.3: Packet loss distribution using 2-state Markov model

165].

Appropriately the term $BurstR$ is introduced [84, 125] to represent burstiness in packet loss and it is defined as:

$$BurstR = \frac{\text{Average length of observed bursts in an arrival sequence}}{\text{Average length of bursts expected for the network under ``random'' loss}} \tag{4.3}$$

With the 2-state markov model proposed by E.N. Gilbert and named after him, the burstiness in packet loss can be modelled. This model is used in large number of studies discussing the problem of packet loss [16, 89, 90, 123, 124, 125, 152, 205, 209, 213].

Using this model packet loss can be simulated in IP networks where speech signal is divided into packets and sent packet by packet over the network. In this model there are four cases for a sequence of two packets:

1 Found-Found: When the system stays is state Found, this means the previous packet was received correctly and the current packet is received as well.

2 Found-Loss: When the system moves from state Found to state Loss, this

means the previous packet was received correctly and the current packet is Lost.

3 Loss-Loss: When the system stays is state Loss, this means the previous packet was lost and the current packet is lost as well.

4 Loss-Found: When the system moves from state Loss to state Found, this means the previous packet was Lost and the current packet is received correctly.

The movement between different states of the Gilbert Model can be represented by pseudocode like representation in Figure 4.4.

```
R=RandomNumber

If R<=Ppl Then
    Initial state=Loss
Else
  Initial state=Found
End If

State=Initial state

For each speech frame
    If state==found
          {Move to state loss with possibility of p
            Put 0 in the packet loss indicator for the current frame}
          {Remain in state found with possibility 1-p
             Put 1 in the packet loss indicator for the current frame}
    Else
          {Move to state found with possibility of q
            Put 1 in the packet loss indicator for the current frame}
          {Remain in state loss with possibility 1-q
            Put 0 in the packet loss indicator for the current frame}
    End
End
```

Figure 4.4: Pseudo code for the packet loss simulator

In this model, the selection of an initial state is constrained by the packet loss probability in the system, Ppl. If Ppl=20% for example, this means that the percentage of packet loss is 20%. In this case a uniformly distributed random number between 0 and 1 is generated. If this number is less than or equal 0.2, then "Loss" is selected as start state, otherwise "Found" is selected as start state. Because a uniformly distributed random number is used there is a 20% chance of being in state "Loss", and 80% chance of being in state "Found".

After the initial state, a random number corresponding to each speech packet is generated and based on the current state and the generated number, movement between different states occurs according to the corresponding probabilities.

Using this model, packet loss in a speech signal can be simulated. If state "Found" is represented by 1 and State "Loss" by 0, packet loss can be illustrated as in Figure 4.5.

1	1	0	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1

Figure 4.5: An example of packet loss in a speech signal

In this figure packets 3, 4, 8, 9, 10, and 13 are lost while the rest are received.

Derivation of Model Parameters

One obvious and very important question to answer is how to construct such a model? i.e. how the values of transition probabilities p and q can be determined? The derivation of p and q utilises the values of Packet loss Probability (Ppl) and Burst Ratio ($BurstR$). ITU-T Recommendation G.107 [84] relates p and q with Ppl and $BurstR$ through the following equation:

$$BurstR = \frac{1}{p+q} = \frac{Ppl}{p} = \frac{1-Ppl}{q} \tag{4.4}$$

Depending on the values of Ppl and $BurstR$, the values of p and q can be derived as follows:

$$
\begin{aligned}
BurstR &= \frac{1}{p+q} && (4.5)\\
\Rightarrow p+q &= \frac{1}{BurstR} && (4.6)\\
\Rightarrow p &= \frac{1}{BurstR} - q && (4.7)
\end{aligned}
$$

Also from Equation (4.4) we have:

$$
\begin{aligned}
\frac{Ppl}{p} &= \frac{1-Ppl}{q} && (4.8)\\
\Rightarrow Ppl.q &= (1-Ppl).p && (4.9)\\
\Rightarrow p &= \frac{Ppl.q}{1-Ppl} && (4.10)
\end{aligned}
$$

(4.10)-(4.7) yields:

$$
\begin{aligned}
0 &= \frac{Ppl.q}{1-Ppl} - \frac{1}{BurstR} + q && (4.11)\\
\frac{1}{BurstR} &= \frac{Ppl.q}{1-Ppl} + q && (4.12)\\
\frac{1}{BurstR} &= q.\left(\frac{Ppl}{1-Ppl} + 1\right) && (4.13)\\
\frac{1}{q} &= BurstR\left(\frac{Ppl}{1-Ppl} + 1\right) && (4.14)\\
q &= \frac{1}{BurstR\left(\frac{Ppl}{1-Ppl} + 1\right)} && (4.15)\\
q &= \frac{1}{BurstR\left(\frac{Ppl+1-Ppl}{1-Ppl}\right)} && (4.16)\\
q &= \frac{1}{\frac{BurstR}{1-Ppl}} && (4.17)
\end{aligned}
$$

Consequently:

$$
q = \frac{1-Ppl}{BurstR} \tag{4.18}
$$

Substituting q in Equation (4.7) to calculate p

$$p = \frac{1}{BurstR} - \frac{1 - Ppl}{BurstR} \tag{4.19}$$

$$p = \frac{1 - (1 - Ppl)}{BurstR} \tag{4.20}$$

Consequently

$$p = \frac{Ppl}{BurstR} \tag{4.21}$$

From the above derivation the values of q, p to be used in characterisation of the Gilbert model can be calculated based on the values of Ppl and $BurstR$ using Equation (4.18) and Equation (4.21) respectively. Using these two relations, the values of p and q were calculated for each possible combination of Ppl in the range 0 to 20 and $BurstR$ in the range 1 to 2. These values are listed in Table 4.5. Having this table it can be noticed that the value of p increases with the increase of Ppl while the value of q decreases with the increase in Ppl.

BurstR	**Ppl**	**P**	**Q**	**BurstR**	**Ppl**	**P**	**Q**
1	1	0.010	0.990	2	1	0.005	0.495
1	2	0.020	0.980	2	2	0.010	0.490
1	3	0.030	0.970	2	3	0.015	0.485
1	4	0.040	0.960	2	4	0.020	0.480
1	5	0.050	0.950	2	5	0.025	0.475
1	6	0.060	0.940	2	6	0.030	0.470
1	7	0.070	0.930	2	7	0.035	0.465
1	8	0.080	0.920	2	8	0.040	0.460
1	9	0.090	0.910	2	9	0.045	0.455
1	10	0.100	0.900	2	10	0.050	0.450
1	11	0.110	0.890	2	11	0.055	0.445
1	12	0.120	0.880	2	12	0.060	0.440
1	13	0.130	0.870	2	13	0.065	0.435
1	14	0.140	0.860	2	14	0.070	0.430
1	15	0.150	0.850	2	15	0.075	0.425
1	16	0.160	0.840	2	16	0.080	0.420
1	17	0.170	0.830	2	17	0.085	0.415
1	18	0.180	0.820	2	18	0.090	0.410
1	19	0.190	0.810	2	19	0.095	0.405
1	20	0.200	0.800	2	20	0.100	0.400

Table 4.5: Values of p and q in 2-state Markov model for different combinations of Ppl and $BurstR$

Chapter 5

Extending The E-model Using PESQ

5.1 Introduction

As discussed in section 3.2.3 the E-model is used for objectively estimating the voice quality non-intrusively. As it is a non-intrusive method, it is suitable for monitoring live traffic in a productive network.

Recall from chapter 3 that the output of the E-model is called the R-Rating Factor which is calculated according to the following formula:

$$R = R_0 - Is - Id - Ie\text{-}eff + A \tag{5.1}$$

Each of the parameters in equation (5.1) (except the Advantage factor (A)) is further decomposed into a series of equations as defined in ITU-T Recommendation G.107 [84]. Quality impairment due to packet loss as defined in equation (5.1) is characterised by packet loss dependent Effective Equipment Impairment Factor (*Ie-eff*) which is calculated according to the following formula [84]:

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{5.2}$$

where

Ie	Codec-specific Equipment Impairment Factor
Bpl	Codec-specific Packet-loss Robustness Factor
Ppl	Packet loss Probability
$BurstR$	Burst Ratio (BurstR-to count for burstiness in packet loss)

Specific impairment factor values for codec operating under random packet-loss have formerly been treated using tabulated, packet-loss dependent Ie-values. Now, the packet-loss dependent Effective Equipment Impairment Factor *Ie-eff* -as defined in equation (5.2) - is derived using codec-specific values for the Equipment Impairment Factor (Ie) and Packet-loss Robustness Factor (Bpl) at zero packet-loss. The values for Ie and Bpl for several codecs are listed in ITU-T Recommendation G.113, Appendix I [80], these values are not related to other input parameters in equation (5.1) but derived using subjective mean opinion score test results. For example for the speech coder defined according to ITU-T Recommendation G.729 [71], the corresponding Ie and Bpl values are 11 and 19 respectively. On the other hand Packet loss Probability (Ppl) and Burst Ratio ($BurstR$-to count for burstiness in packet loss) depend on the packet loss characteristics of the system.

Consequently, an obvious problem of the E-model is that it is based on subjective tests to calibrate model parameters [84]. The inherent problems of subjective tests are that they are hard to conduct (as they require strict lab conditions), time-consuming, expensive, and lack repeatability. This makes the E-model applicable just for limited number of cases, specifically to those cases where the corresponding subjective tests are already performed. Additionally, as the E-model is defined over a specific range of parameters as defined in Table 3.1. Consequently, the E-model cannot be applied to any new codec or even for new network conditions (outside the permitted range) before conducting a series of subjective tests corresponding to the new coder and the new network conditions to derive model parameters and this hinders its use in new and emerging applications.

Based on the above, it would be useful and practical if a methodology to extend the E-model applicability range, without the need for the time-consuming and expensive subjective tests needed to calibrate the E-model's parameters, can be found as this will remove one of the major obstacles in the applicability of the E-model for new coders and new network conditions. The proposed extension will use the latest version of the Intrusive-based speech quality prediction methodology - Perceptual Evaluation of Speech Quality (PESQ)- as a reference criteria for the accuracy of the prediction of the E-model parameters instead of performing the expensive subjective tests.

While the proposed method is described in details in this chapter, however, the exact methods used for deriving the extension are described in the next chapter. This chapter is structured as follows: Section 5.2 describes the proposed technique. In section 5.3 the exact derivation steps of the extended model is described in details.

5.2 The Proposed Technique

In this section the proposed technique for E-model extension is described. The setup for the system is depicted in Figure 5.1.

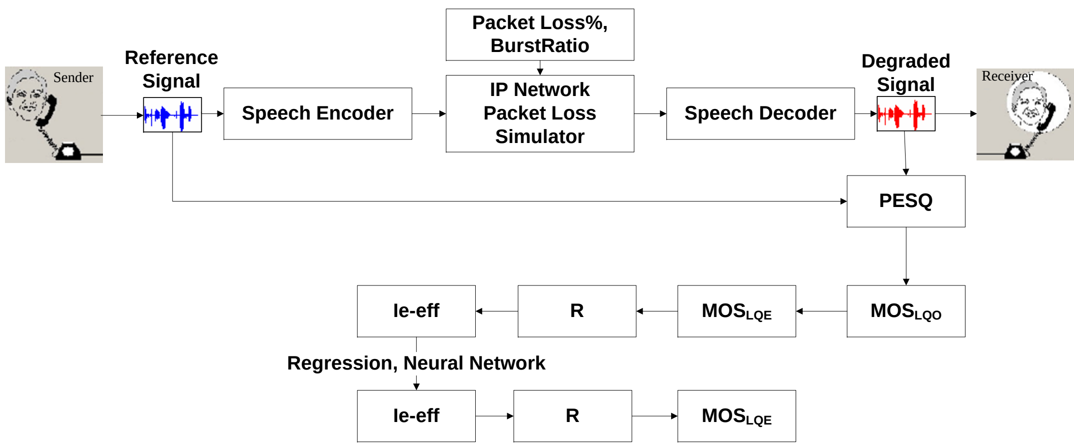

Figure 5.1: System setup for the E-model Extension

It should be noted that in the calculations of this chapter the accuracy of the E-model is not questioned as the assumption is made that the equations of the E-model are accurate and the aim of the chapter is to extend the E-model.

In the system setup shown in Figure 5.1, the reference speech signal is encoded and then packet loss is simulated. The simulation of packet loss is performed using 2-state Gilbert model as defined in section 4.3, where the input to the Gilbert model are *Ppl* and $BurstR$ to calculate the probabilities of Gilbert model (p and q). Then the received degraded stream is decoded to retrieve the degraded speech signal.

Using PESQ, the speech quality is predicted by comparing the reference signal with the degraded signal as defined in ITU-T's Recommendation P.862 and discussed earlier in section 3.2.2. Based on the $PESQ$ score retrieved from the $PESQ$ prediction of speech quality, the Mean Opinion Score - Listening Quality Objective (MOS_{LQO}) is derived using equation (3.2) which represents the speech quality as predicted by PESQ. Then this quality is mapped into (MOS_{LQE}) which represents the speech from the E-model's point of view. Then the R-Rating R-Rating Factor is derived from MOS_{LQE} using equation (3.10).

A simplified version of the R-Rating Factor as defined in (5.1) is used to consider the effect of packet loss only. The simplified version is:

$$R = R_0 - \textit{Ie-eff} \tag{5.3}$$

By manipulating equation (5.3), *Ie-eff* can be calculated from an R-Rating Factor:

$$\textit{Ie-eff} = R_0 - R \tag{5.4}$$

As mentioned in section 3.2.3 when all E-model's parameters are set to their default values, R_0 has the value of 93.2 [84].

The above sequence of derivations can be summarised as follows: under specific *Ppl* and $BurstR$, packet loss is simulated and speech quality is measured using $PESQ$ algorithm to retrieve the $PESQ$ score which can be then mapped into an MOS score. From the MOS score the R-Rating Factor can then be derived and then *Ie-eff* is calculated.

From the above sequence, a relation between (*Ppl* and $BurstR$) and *Ie-eff* is constructed by simulating packet loss for several values of *Ppl* and $BurstR$ and performing the above derivation. This three dimensional relation is then used to derive a model to relate (*Ppl* and $BurstR$) with *Ie-eff*. Such relation is derived using Regression methods and Artificial Neural Network (ANN) methods.

By deriving such a relation, *Ie-eff* is defined in terms of *Ppl* and *BurstR* in absence of *Ie* and *Bpl* which both depend on subjective tests while in the original equation (equation (5.2)) *Ie-eff* was defined according to these four terms *Ie*, *Bpl*, *Ppl*, and *BurstR*. In this way the E-model becomes extendable to new network conditions and to new coders as soon as the relation is re-derived for these new coders and new network condition utilising PESQ. This will by-pass the time consuming, expensive subjective tests which are a major obstacle toward the generalisation of the E-model.

It should be noted, however, that the derived model is applicable for the speech coder in use. If a new speech coder is to be used, a new derivation is required. Therefore before applying the derived model, the used speech coder should be determined and the model derived for that coder should be used. However, requiring objective tests to derive such model is much simpler, cheaper, and faster than requiring subjective tests which is the aim here.

The derived model can be integrated with the E-model in monitoring live traffic non-intrusively as depicted in Figure 5.2.

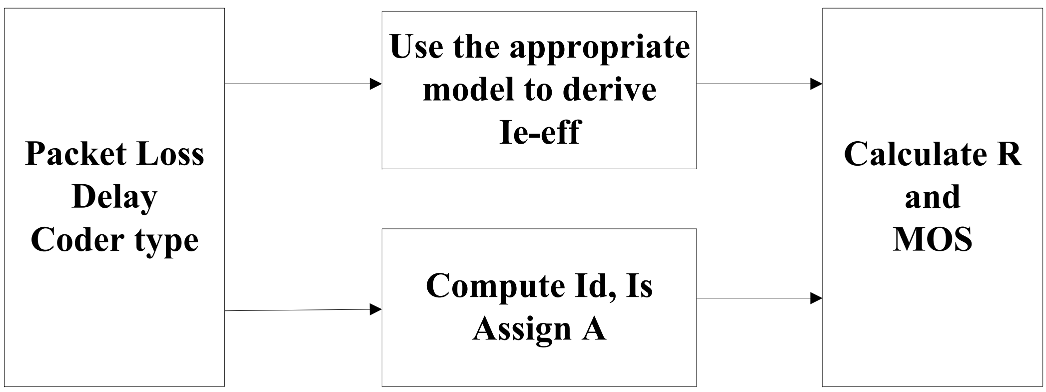

Figure 5.2: Conceptual diagram of E-model Extension used to monitor live traffic

IP packets can be captured at some appropriate point in the IP network (possible at an ingress gateway) and the information about packet loss, delay, and the coder type are extracted from Real-time Transport Protocol (RTP) header as explained in section 2.2.3.

By identifying the speech coder, then the appropriate model for that speech coder as derived above is used and the information about packet loss are fed into that model to calculate the corresponding *Ie-eff* value. Similarly the information about the delay and other parameters in the E-model are used to calculate *Id*, and *Is* to use them in equation (5.1). The value of the advantage factor is added according to the characteristics of the system in-hand (wired, wireless, etc).

Ie-eff value computed from the model and the results from the computation of *Id*, *Is* are combined with A to calculate an overall R-Rating Factor which can then be mapped into an MOS score to give an estimation for the overall conversational quality.

Using the proposed technique, if a new speech coder emerges , it is readily applicable to the E-model as soon as the required objective tests are performed to derive a relation between *Ie-eff* and *Ppl* and $BurstR$.

5.3 Derivation of the Model

Using speech signal encoded according to the speech coder defined in ITU-T Recommendation G.729 [71], this section details the steps toward deriving a relation between (*Ppl* and $BurstR$) with *Ie-eff*. This will provide an alternative formula to equation (5.2) and thus will by-pass the subjective tests required to calculate Ie and *Bpl* values for the speech coder.

The basic steps for the derivation are:

1 Simulate packet Loss using 2-state Gilbert model, for each combination compare the resulted degraded signal with the original signal to compute the PESQ score.

2 Convert PESQ scores into MOS_{LQO} values.

3 Convert MOS_{LQO} values into MOS_{LQE} values.

4 Map MOS_{LQE} values into R-Rating Factor values.

5 Calculate *Ie-eff* from R-Rating Factor using the simplified E-model.

6 Derive a model to relate *Ie-eff* with Ppl and $BurstR$ using:

   (a) Linear Regression Models.

   (b) Non-Linear Regression Models.

   (c) Artificial Neural Network Model

7 Use the above models to calculate the R-Rating Factor.

8 Map the results into MOS score.

The above steps are described in more details next where steps 1-5 are described in this chapter while the remaining steps are explained in the next chapter.

Step 1: Calculate PESQ Score

Using the speech coder defined in ITU-T Recommendation G.729 [71], packet loss is simulated using 2-state Gilbert model (section 4.3) for each combination of Ppl and $BurstR$. Then the reference signal is compared with the degraded signal to calculate the PESQ score.

When packet loss is simulated using 2-state Gilbert model with specific values for Ppl and $BurstR$, the exact locations of packet loss are unknown in advance and as such the speech quality could differ between two runs of the simulator even with the same combination of Ppl and $BurstR$ due to the fact that loss during voiced period of the signal has different effect on the quality in comparison with loss during unvoiced periods of the speech signal [165]. To remove the effect of randomisation on the result, a pilot study was conducted to figure out the required number of times the simulation need to be repeated for each combination of Ppl and $BurstR$ to have enough confidence in the results.

Step 1.1: Pilot Study

The purpose of this pilot study is to determine the required number of iterations the simulation need to be repeated in order to accurately predict the quality in the presence of packet loss in random locations. The goal is to have an accurate estimation of the quality with $\pm$ 0.01 MOS because any difference of 0.01 or less is indistinguishable to the ear. The equation used to calculate the error in the estimation is:

$$Error = \frac{z * \sigma}{\sqrt{N}} \tag{5.5}$$

where

$Error$	The error in estimating the quality between the actual and the prediction
z	A number to indicate how confident we want to be
σ	The standard deviation of the population
N	The size of the sample

As the size of the sample increases, the accuracy of the prediction increases and the error decreases, but as an infinite sample size can not be used, therefore a sample of limited size should be used and calculations can be made with a margin of error. In this case the standard deviation of the population σ is replaced with the standard deviation of the sample S and a modified version of equation (5.5) is used as follows:

$$Error = \frac{z * S}{\sqrt{N}} \tag{5.6}$$

Manipulation of equation (5.6) yields:

$$N = \left(\frac{z * S}{Error}\right)^2 \tag{5.7}$$

In equation (5.7) the required level of error is $\pm$ 0.01 MOS, as the MOS scores are normally is given with up to two digits, and the required confidence level is 95% (95% is high confidence level and if the required confidence level is as high as 99% confidence level, a much higher number of experiments is needed in this case).

The corresponding value of z for the 95% confidence level is 1.96. To determine the required number of iterations (N) to achieve this confidence level with this error margin, the sample's standard deviation, S is still to be determined which will be computed using the pilot study.

In the pilot study, one value of Ppl is used, specifically 5 which lies in the range 0 to 20 which are the boundaries of the permitted range for Ppl as defined in Table 3.1). The selected value for $BurstR$ is 1.5 (middle point between the permitted range 1 to 2). For this combination of *Ppl* and *BurstR* (5, 1.5), p and q parameters for the 2-state Gilbert model are computed (using equations (4.21) and (4.18)) and this model is used to simulate the packet loss behavior. The degraded signal is compared against the reference signal (without loss) to compute the resultant PESQ score. This was repeated for 10000 times to have a large enough sample to represent the population. The standard deviation for this sample (S) was 0.0249. When this value fed into equation (5.7), the required number of iterations is 23.755. The speech material used in this study is the speech file named (b_eng_f1) from the speech data set defined in ITU-T Recommendation P.50 as discussed in section 4.2. The pseudocode for this pilot study is listed below in Figure 5.3.

As a result it was concluded that to achieve 95% confidence level of having $\pm$ 0.01 MOS, at least 24 iterations should be performed for each combination of *Ppl* and *BurstR*. In order to have a normal distribution (according to the Central Limit Theorem) together with having a 95% confidence level, 30 iterations are used to satisfy both requirements.

Step 1.2: Calculate PESQ Score

From the previous step it was determined that for each combination of *Ppl* and *BurstR* the simulation need to be repeated for 30 times to have 95% confidence of the results with $\pm$ 0.01 MOS.

The permitted range according to the E-model is 0 to 20 for *Ppl* and is 1 to 2 for *BurstR*. For the purpose of comparing the results with those of the E-model within the permitted range of the E-model and the expected quality outside the range, packet loss was simulated in the range 0 to 30 for *Ppl* and the range 1 to 5 was used for *BurstR*.

The first step in the derivation of the model is the calculation of the corresponding PESQ score for each combination of *Ppl* and *BurstR*. The selected values for *Ppl* are $\{0, 0.5, 1 \ldots 30\}$ and for *BurstR* are $\{1 \ldots 5\}$, consequently there are 160 possible combination. For each possible combination, the parameters for the 2-state Gilbert model (p and q) are calculated using equations (4.21) and (4.18). Then 30 iterations are performed (as discussed in the previous section) and for each iteration, packet loss is simulated using the values of p and q. In each iteration the original

```
StartTime=Now%Record the start time
BurstR=1.5
Ppl=5
z=1.96
ErrorLevel= 0.01
p=calculate_p(BurstR,ppl)%calculate p using values of BurstR and Ppl
q=calculate_q(BurstR,ppl)%calculate q using values of BurstR and Ppl
z=1.96%Value of Z-score to have 95% confidence
reference_speech=b_eng_f1%the reference speech signal
%Encode the speech for transmission
enocded_signal=g729encode(reference_speech)
MOSList=[ ]
for iteration=1:10000
  Begin
    %simulate Loss using 2-state Gilbert model with (p,q) parameters
    degraded_signal=simulateloss(enocded_signal,p,q)
    %Decode the received stream
    degraded_speech=g729decode(degraded_signal)
    %Compute PESQ by comparing the original signal with the degraded one
    PESQ_Score=PESQ(reference_speech,degraded_speech)
    %Map the value of PESQ to an MOS value
    MOS_Score=pesqtomos(PESQ_Score)
    %Add the current score to the list
    MOSList=[MOSList MOS_Score]
  End

S=compute_standard_deviation(MOSList)
RequiredIterations=(z*S/ErrorLevel)^2
FinishTime=Now%Record the finish time
output RequiredIterations
```

Figure 5.3: Pseudo code for the pilot study

speech file is compared against the degraded speech file using PESQ algorithm (see section 3.2.2) to compute the PESQ score. The overall PESQ score for each combination of *Ppl* and *BurstR* is computed by taking the average over the 30 iterations. The pseudocode for this process is listed in Figure 5.4

As a result of this experiment, the PESQ score is calculated for each possible combination of *Ppl* and *BurstR*. This forms a 3-dimensional relation that relates *Ppl* and *BurstR* with PESQ score as shown in Figure 5.5. All the PESQ scores for all combinations of *Ppl* and *BurstR* for all the 30 iterations are listed in Tables

```
StartTime=Now%Record the start time
nIterations=30%Number of iterations for each combination
PESQ=double(5,32)%Array to hold PESQ scores
reference_speech=b_eng_f1%the reference speech signal
%Encode the speech for transmission
enocded_signal=g729encode(reference_speech)

for BurstR in (1, ... ,5)
  Begin%BurstR
    for Ppl in (0, 0.5, 1, 2, 3, ...,30)
      Begin%Ppl
        %calculate p for the current combination of BurstR and Ppl
        p=calculate_p(BurstR,Ppl)

        %calculate q for the current combination of BurstR and Ppl
        q=calculate_q(BurstR,Ppl)

        PESQList=[ ]%Initialise the list of scores

        for iteration=1:nIterations
          Begin
            %simulate Loss using 2-state Gilbert model with (p,q) parameters
            degraded_signal=simulateloss(enocded_signal,p,q)
            %Decode the received stream
            degraded_speech=g729decode(degraded_signal)
            %Compute PESQ by comparing the reference speech with the degraded
            PESQ_Score=PESQ(reference_speech,degraded_speech)
            %Add the current score to the list
            PESQList=[PESQList PESQ_Score]
          End

          %Calculate the average PESQ score for the current combination
          PESQ(BurstR,Ppl)=Average(PESQList)
      End%Ppl
  End%BurstR
FinishTime=Now%Record the finish time
output PESQ
```

Figure 5.4: Pseudo code for the calculation of the $PESQ$ score

A.1-A.5 in Appendix A.

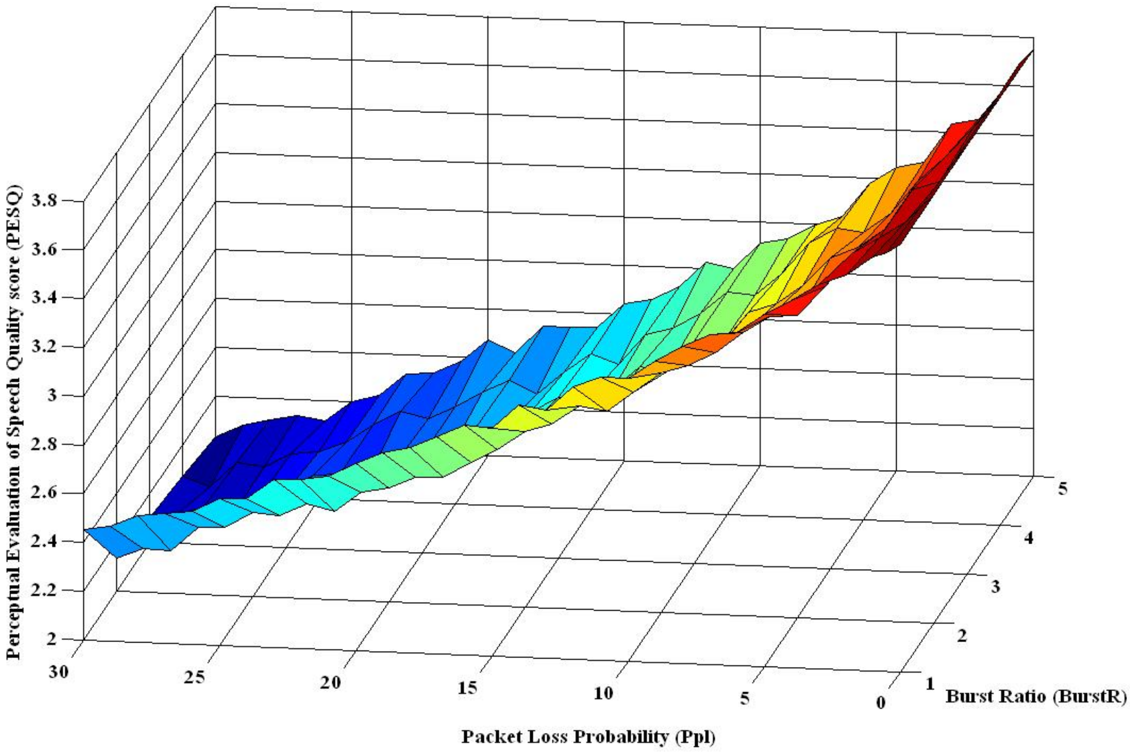

Figure 5.5: PESQ (Experimental) vs. Packet Loss Probability and Burst Ratio

As expected from the experiment, the quality of the signal expressed as a PESQ score is inversely proportional to both Ppl and $BurstR$. In other words the PESQ score decreases with any increase in either Ppl or $BurstR$. This can be noticed from Figure 5.5 where the maximum PESQ achieved with the minimum values of Ppl and $BurstR$ namely 0 and 1 respectively. With any value of $BurstR$ such as 1, PESQ score decreases monotonically with the increase in Ppl. Similarly, with any value of Ppl such as 10, PESQ score decreases with the increase in $BurstR$. It should be noted that with $Ppl = 0$, PESQ score remains the same, 3.7535, which is the maximum possible value of PESQ for the used speech file. This is due to the fact that when packet loss equals 0, there is no degradation in the quality due to packet loss. The only degradation to the quality in this case is due to the distortion of the coder. For the used speech signal (b_eng_f1), the basic quality (in terms of PESQ score) due to the coder's distortion is 3.7535 as discussed earlier in section 3.2.2 and listed in Table 4.4.

This is consistent with the expected quality under the E-model (see section 5.1). According to the E-model the impairment due to packet loss is characterised by the packet loss dependent *Ie-eff* as defined in equation (5.2). When *Ppl* equals 0, this equation reduces to:

$$Ie\text{-}eff = Ie \tag{5.8}$$

where Ie is the impairment due to the coder (due to the absence of packet loss).

Step 2: Convert PESQ Score to MOS_{LQO} *Score*

As shown in the system setup in Figure 5.1, the next step of the derivation is to convert the $PESQ$ score into an MOS_{LQO} score. This can be achieved using the following formula [86] (see section 3.2.2):

$$MOS_{LQO} = 0.999 + \frac{4.999 - 0.999}{1 - e^{(-1.4945*PESQ+4.6607)}} \tag{5.9}$$

The graph shown in Figure 5.5 which demonstrates the relation between (*Ppl* and $BurstR$) and PESQ score is then developed into Figure 5.6 that shows the relation of *Ppl* and $BurstR$ with the MOS_{LQO} score. All the MOS_{LQO} scores for each combination of $BurstR$ and *Ppl* are listed in Tables A.1-A.5 in Appendix A.

As the MOS_{LQO} score is proportional to the PESQ score as illustrated in Figure 3.4, Figure 5.6 is expected to have similar characteristics to Figure 5.5 and the same comments can be said here. The quality of the signal expressed as an MOS_{LQO} score in Figure 5.6 is inversely proportional to both *Ppl* and $BurstR$, i.e. the MOS_{LQO} score decreases with any increase in either *Ppl* or $BurstR$. This can be noticed from Figure 5.6 where the maximum MOS_{LQO} is achieved with the minimum values of *Ppl* and $BurstR$ namely 0 and 1 respectively. With any value of $BurstR$ such as 1 the MOS_{LQO} score decreases monotonically with the increase in *Ppl*. Similarly, with any value of *Ppl* such as 10, the MOS_{LQO} score decreases with the increase in $BurstR$. The MOS_{LQO} score in Figure 5.6 remains the same when *Ppl* $= 0$ which is in this case equals to 3.8826 (the maximum possible value of MOS_{LQO} for the used speech file). This is due to the fact that when packet loss equals 0, there is no degradation in the quality due to the packet loss. The only degradation to the

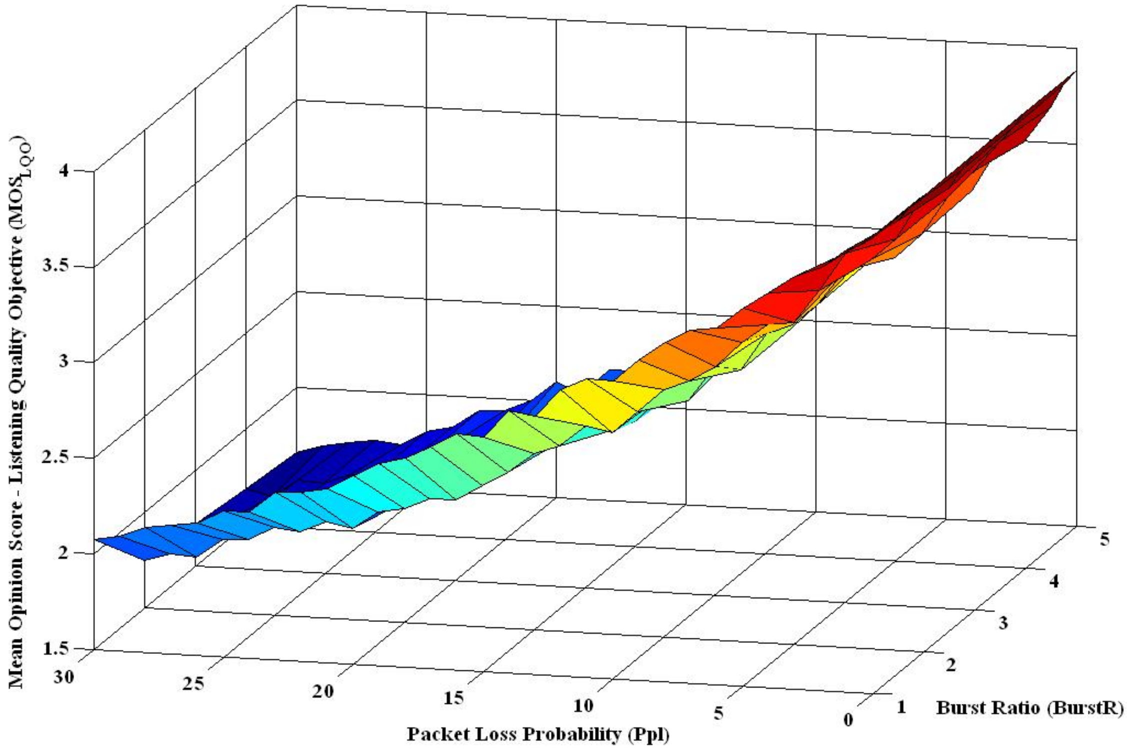

Figure 5.6: MOS_{LQO} (Experimental) vs. Packet Loss Probability and Burst Ratio

quality in this case is due to the distortion of the coder. For the used speech signal (b_eng_f1) the basic quality (in terms of the MOS_{LQO} score) due to the coder's distortion is 3.8826 as discussed earlier in section 3.2.2 and listed in Table 4.4.

This is consistent with the expected quality under the E-model (see section 5.1). According to the E-model the impairment due to the packet loss is characterised by (*Ie-eff*) as defined in equation (5.2). When *Ppl* equals 0, this equation reduces to Ie (impairment due to the coder at 0 packet loss).

Step 3: Convert MOS_{LQO} *Score to* MOS_{LQE} *Score*

Speech quality as defined by PESQ yields an objective measurement of listening quality (MOS_{LQO}) while speech quality as defined by the E-model yields an estimation of listening quality (MOS_{LQE}) (if delay is not taken into consideration) or an estimation of conversational quality MOS_{CQE} (if delay is taken into consideration) as defined in section 3.2.5.

Previous studies [30, 31, 165, 168, 167, 169] assumed equivalence of MOS_{LQO} as predicted by PESQ and MOS_{LQE} from the E-model, and they directly substitute between the two terms for the purpose of deriving the R-Rating Factor.

This assumption is investigated in this study as a clear distinction is made between the two terms in ITU-T's Recommendation P.800.1 [83]. Figure 5.7 shows both the quality of speech in terms of MOS_{LQO} as empirically measured in the previous step and the MOS_{LQE} as calculated from the E-model and listed in Table 4.3.

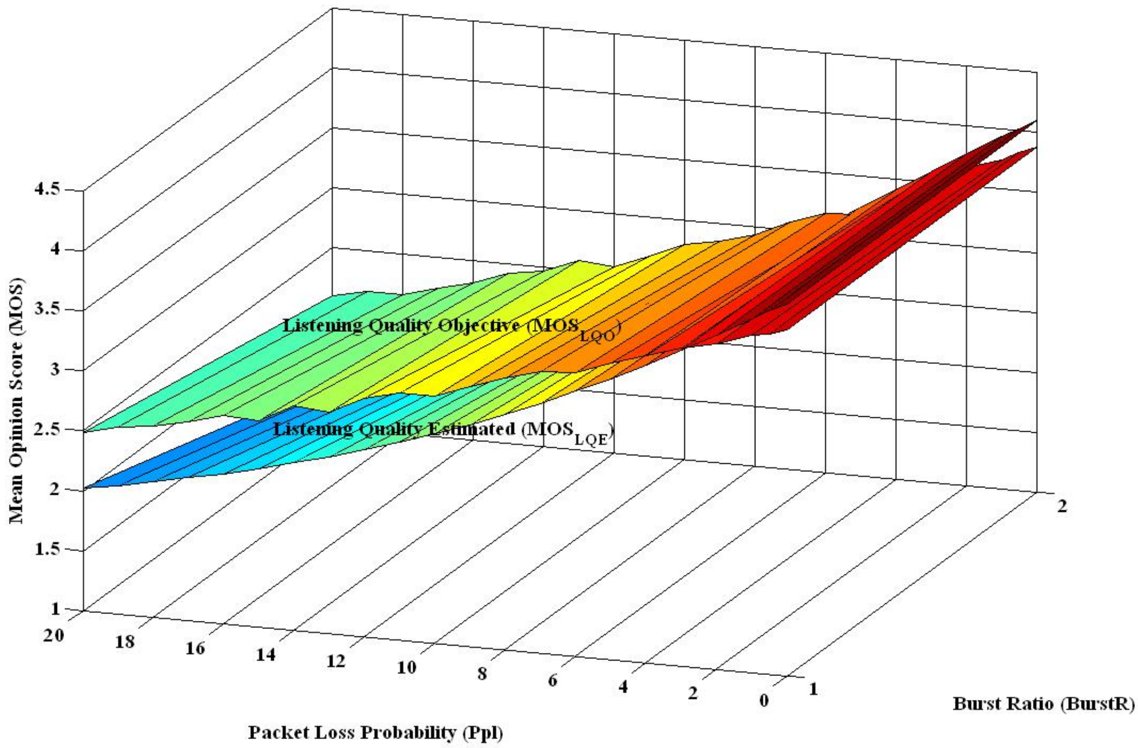

Figure 5.7: MOS_{LQO} and MOS_{LQE} vs. Packet Loss Probability and Burst Ratio

Although the correlation factor between MOS_{LQE} and MOS_{LQO} is 0.9937 which is quite high value but the two surfaces do not fit. It is clear from Figure 5.7 that direct substitution will not lead to accurate results. The maximum absolute difference between the two surfaces is 0.7071 MOS and the average absolute difference is 0.3275 MOS which are quite high values.

A conversion formula is needed to convert from MOS_{LQO} to MOS_{LQE} so that the results of the quality measurement according to $PESQ$ (MOS_{LQO}) can be converted to quality estimation according to the E-model (MOS_{LQE}). Thus R-Rating Factor can be calculated from MOS_{LQE}.

The conversion formula will be retrieved by plotting MOS_{LQO} vs MOS_{LQE} and finding the best curve that can represent the relation. This was done using the MATLAB's curve fitting tool as depicted in Figure 5.8. In this figure, MOS_{LQE} as defined by the E-model is plotted against MOS_{LQO} as calculated empirically in the previous section. Then polynomials with different degrees are derived to fit the data. The simplest polynomial was a linear function and the most complex was of degree 10. As can be shown, it seems that no advantage is achieved with very complex polynomials.

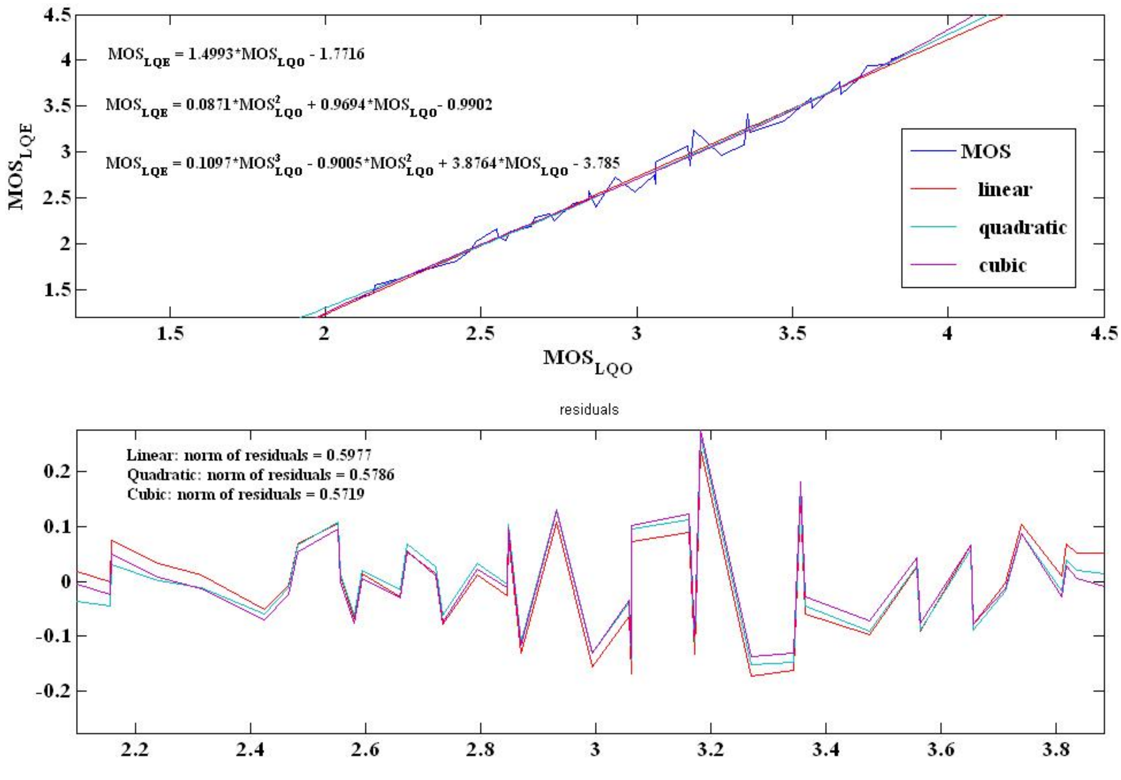

Figure 5.8: MOS_{LQE} vs. MOS_{LQO} and different fitting functions

For each polynomial the norm of residuals is calculated. Also, the correlation between MOS_{LQE} and new MOS_{LQO} after applying each polynomial is calculated in addition to the maximum and average absolute difference between MOS_{LQE} MOS_{LQO} after applying the polynomial (attempted polynomials are from degree 1 to degree 10).

The first five polynomials together with their norm of residuals, correlation coefficients, max absolute difference and average absolute difference between MOS_{LQE} and the new MOS_{LQO} after applying the formula are listed in Table 5.1. Higher degree polynomials offer little gain in terms of norms, correlation coefficients, maximum difference or average difference to justify using them.

Degree	**Norms**	**Correlation**	**Maximum Difference**	**Average Difference**
Linear	0.5977	0.9937	0.2425	0.0716
Quadratic	0.5786	0.9940	0.2654	0.0677
Cubic	0.5719	0.9941	0.2755	0.0653
4th degree	0.5715	0.9941	0.2741	0.0652
5th degree	0.5714	0.9941	0.2833	0.0655

Table 5.1: Goodness of different possible fitting functions for MOS_{LQO} to MOS_{LQE} conversion

It is clear from Table 5.1 that the conversion formula offers greater improvement in terms of the degree of closeness of MOS_{LQO} to MOS_{LQE}. While the original MOS_{LQO} was far from MOS_{LQE} with maximum absolute difference of 0.7071 MOS and average absolute difference of 0.3275 MOS, the new MOS_{LQO} after applying the polynomial is much closer to MOS_{LQE}.

Figure 5.9 shows MOS_{LQO} after applying the linear degree polynomial as an example. The new MOS is almost indistinguishable from the MOS_{LQE} calculated from the E-model.

It seems from Table 5.1 that the linear polynomial is a good enough to convert MOS_{LQO} into MOS_{LQE}. The linear relation is:

$$MOS_{LQE} = 1.4993 * MOS_{LQO} - 1.7716 \tag{5.10}$$

The value of 1.4993 is called the **correction coefficient** and the name **correction constant** is used for the value of -1.7716. The reverse relation that links MOS_{LQO} with MOS_{LQE} is:

$$MOS_{LQO} = 0.6670 * MOS_{LQE} + 1.1816 \tag{5.11}$$

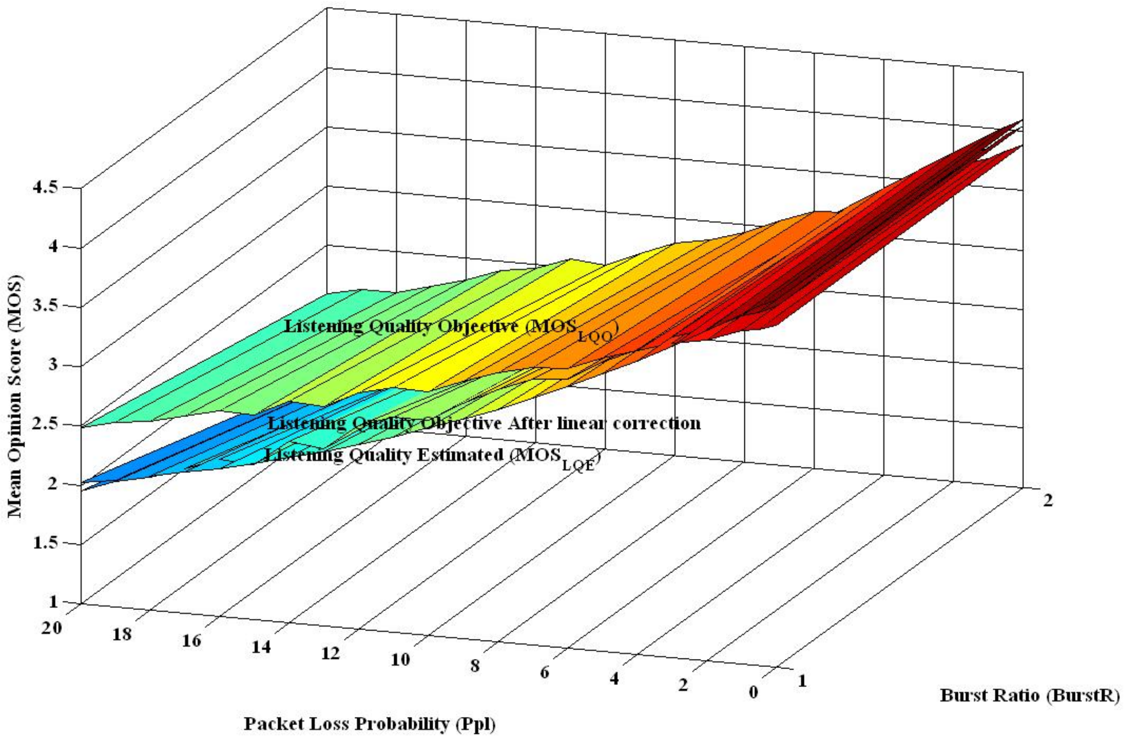

Figure 5.9: MOS_{LQO}, MOS_{LQE}, and MOS_{LQO} (Linearly corrected) vs. Packet Loss Probability and Burst Ratio

Using equation (5.10), Figure 5.10 relates *Ppl* and $BurstR$ with MOS_{LQE} after linear correction which will be used for subsequent derivations in the coming sections. All the MOS_{LQE} scores for each combination of *Ppl* and $BurstR$ are listed in Tables A.1-A.5 in Appendix A.

Step 4: Mapping MOS_{LQE} *Score to* R*-Rating Factor*

The next step of the derivation (Figure 5.1) is to map the MOS_{LQE} scores retrieved in the previous step into their corresponding R-Rating Factor values. Equation (3.10) in section 3.2.3 is used to convert the MOS_{LQE} scores into an R-Rating Factor value. Figure 5.11 shows the relation between the R-Rating Factor with *Ppl* and $BurstR$. All the values of the R-Rating Factor for each combination of *Ppl* and $BurstR$ are listed in Tables A.1-A.5 in Appendix A.

Again as R-Rating Factor is proportional to the MOS_{LQE} score as illustrated in Figure 3.9, the graph shown in Figure 5.11 has similar characteristics to the graph

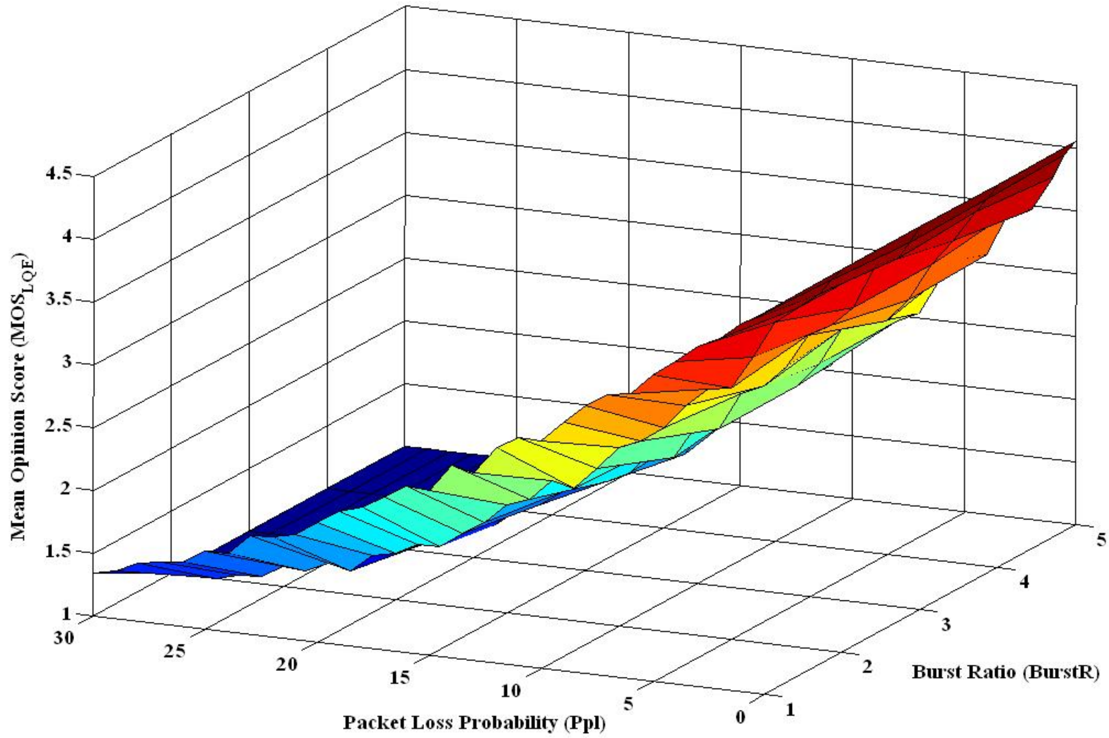

Figure 5.10: MOS_{LQE} (Experimental) vs. Packet Loss Probability and Burst Ratio

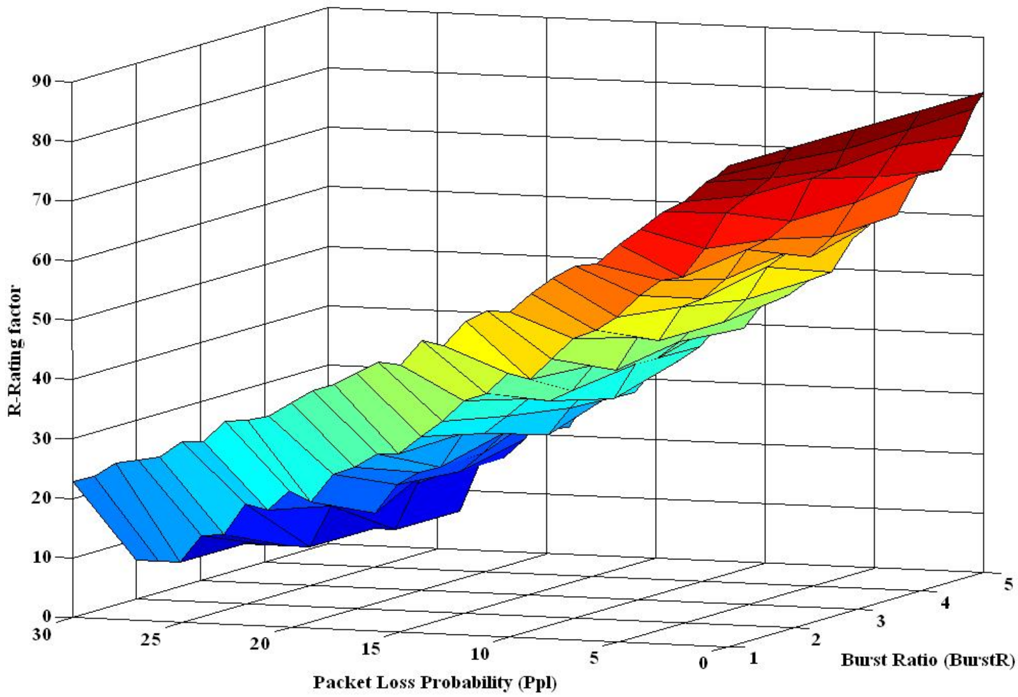

Figure 5.11: R-Rating Factor (Experimental) vs. Packet Loss Probability and Burst Ratio

in Figure 5.10 and the same comments can be said here. The quality of the signal expressed as R-Rating Factor score in Figure 5.11 is inversely proportional to both *Ppl* and $BurstR$, i.e. the R-Rating Factor score decreases with any increase in either *Ppl* or $BurstR$. This can be noticed from Figure 5.11 where the maximum R-Rating Factor is achieved with the minimum values of *Ppl* and $BurstR$ namely 0 and 1 respectively. With any value of $BurstR$ such as 1 the R-Rating Factor decreases with the increase in *Ppl*, the same can be said about any *Ppl*. The R-Rating Factor in Figure 5.11 remains the same when $Ppl = 0$ which is in this case equals 80.6834 (the maximum possible value for R-Rating Factor for the used speech file). This is due to the fact that when Packet loss equals 0, there is no degradation in the quality due to Packet loss. The only degradation to the quality in this case is due to the distortion of the coder.

This is consistent with the expected quality under the E-model (see section 5.1). According to the E-model the impairment due to the packet loss is characterised by *Ie-eff* as defined in equation (5.2). When *Ppl* equals to 0, this equation reduces to Ie (impairment due to the coder at 0 packet loss).

Step 5: Calculate Ie-eff *from R-Rating Factor*

From the R-Rating Factor derived in the previous section, the value of *Ie-eff* can be derived using equation (5.1). To consider the effect of only the packet loss, a simplified version of equation (5.1) is used. The simplified version is:

$$R = R_0 - \textit{Ie-eff} \tag{5.12}$$

From equation (5.12), *Ie-eff* can be computed using the following equation.

$$\textit{Ie-eff} = R_0 - R \tag{5.13}$$

As mentioned in section 3.2.3 when all parameters set to their default values, R_0 has the value of 93.2 [84]. When equation (5.13) is used to derive the *Ie-eff* from the R-Rating Factor values retrieved from the previous step, Figure 5.12 is produced to show the relation of *Ie-eff* against *Ppl* and $BurstR$. All the *Ie-eff* values for each combination of *Ppl* and $BurstR$ are listed in Tables A.1-A.5 in Appendix A.

As *Ie-eff* is inversely proportional to the R-Rating Factor values as illustrated in

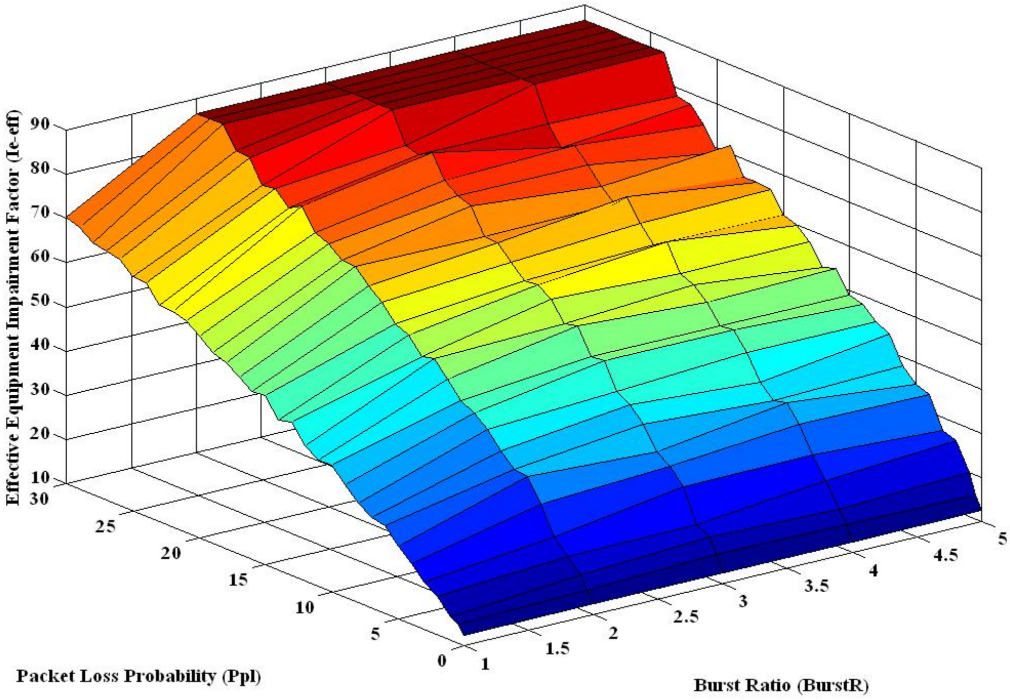

Figure 5.12: *Ie-eff* (Experimental) vs. Packet Loss Probability and Burst Ratio

Figure 3.7, *Ie-eff* is proportional to both the Ppl and $BurstR$, i.e. *Ie-eff* increases with any increase in either Ppl or $BurstR$, as can be seen in Figure 5.12. The maximum *Ie-eff* value is achieved with the maximum values of Ppl and $BurstR$. *Ie-eff* reaches its minimum when $Ppl = 0$ as a result of applying equation (5.13). This is due to the fact that when packet loss equals 0, there is no degradation in the quality due to the packet loss. The only degradation to the quality in this case is due to the distortion of the coder. For the used speech signal (b_eng_f1) the basic quality (in terms of the R-Rating Factor) due to coder's distortion is 80.6834. When the R-Rating Factor reaches its maximum, applying equation (5.13), the basic degradation (in terms of *Ie-eff*) is produced which is equal 12.5166 in this case.

5.4 Summary

In this chapter a methodology for extending the scope of the E-model based on PESQ was proposed. The basic steps for deriving a model to extend the E-model were detailed. Next chapter linear regression, non linear regression and ANN models will be used to illustrate such extension.

Chapter 6

Extending the E-model: Results Analysis & Evaluation

6.1 Introduction

In the previous chapter a 3-dimensional figure that relates Effective Equipment Impairment Factor (*Ie-eff*) with Packet loss Probability (*Ppl*) and Burst Ratio ($BurstR$) was derived starting from an empirical study through a sequence of equations and derivations. The derived relation reproduced in Figure 6.1 for easy reference.

This chapter will attempt to derive a relation or formula from the set of data in order to relate *Ie-eff* with packet loss conditions of the network (*Ppl* and $BurstR$). By deriving such a potential relation, the E-model becomes extendable to new network conditions and to new coders provided the relation is re-derived for these new coders. This will bypass the time consuming, expensive subjective tests which are a major obstacle toward the generalisation of the E-model. The derivation in this chapter will be done using regression models (both linear and non-linear) and Artificial Neural Network (ANN) Model. The results of these three ways of derivation will then be compared to determine the most successful way for extending the E-model.

Section 6.2 discusses the Linear Regression analysis while the Non Linear Regression analysis is explained in section 6.3. The ANN derivation is discussed in section 6.4. In section 6.6 the results of the 3 proposed methods are compared. In section 6.7 a description is provided on how the proposed model can be used in conjunction

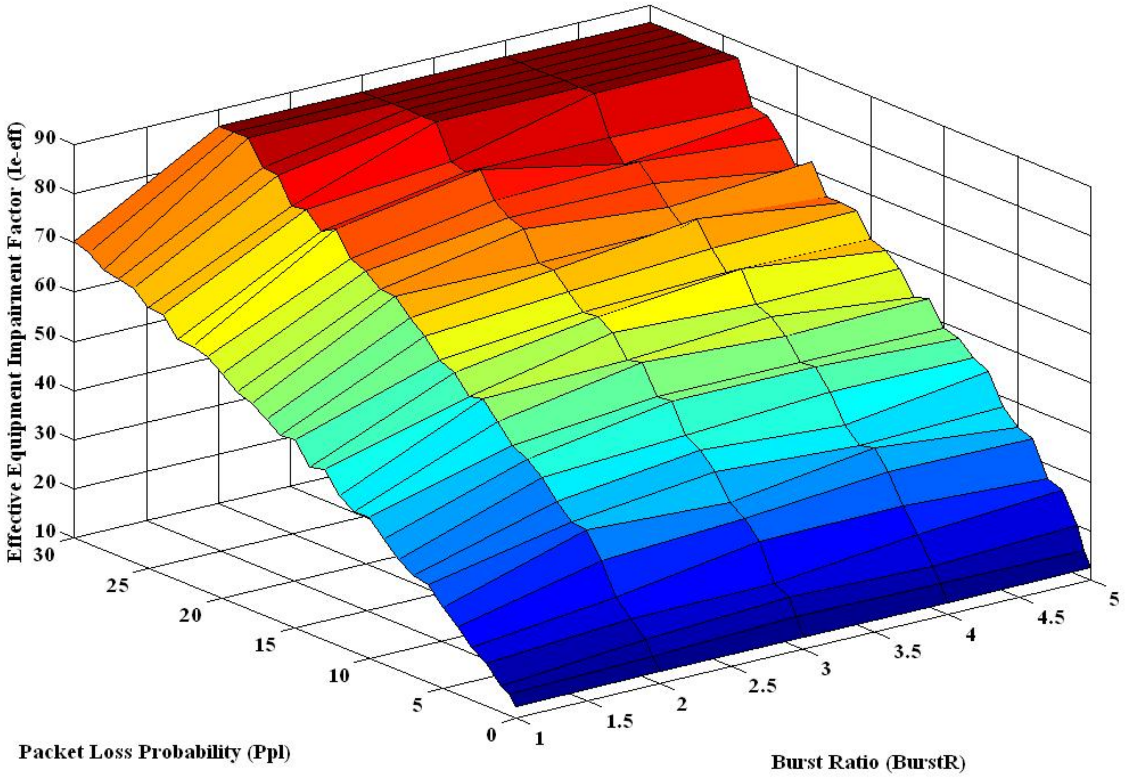

Figure 6.1: *Ie-eff* (Experimental) vs. Packet Loss Probability and Burst Ratio

with the E-model to provide an estimation of the conversational quality. Section 6.8 summarises the chapter.

6.2 Linear Regression

Using the SPSS statistical package version 12.0 [163], and the data for *Ie-eff* shown in Figure 6.1 and listed in Tables A.1-A.5 in Appendix A, a multivariate linear model is derived to relate *Ie-eff* with both *Ppl* and *BurstR*. The choice of a linear model to derive a relation between the *Ie-eff* and the *Ppl* and *BurstR* is based on the observation that the graph in Figure 6.1 is not far from being a plane with an underlying linear relation that can be used to relate *Ie-eff* with *Ppl* and *BurstR*. The resultant equation from the SPSS's linear regression analysis is:

$$\textit{Ie-eff} = 3.080 * BurstR + 2.331 * Ppl + 10.886 \tag{6.1}$$

Equation (6.1) relates *Ie-eff* to both *Ppl* and *BurstR*. If you compare this to the original *Ie-eff*'s equation as defined by the E-model and presented in equation

(6.2):

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{6.2}$$

It can be noticed that the speech coder dependent parameters (Ie and Bpl) have disappeared. These parameters resulted from subjective tests. As such *Ie-eff* as it is defined in equation (6.1) does not depend on the time-consuming, subjective tests to calibrate its parameters. That is exactly was the aim of deriving this linear regression model, to avoid these subjective tests.

Using equation (6.1) a new 3-dimensional graph can be drawn to relate *Ie-eff* against *Ppl* and *BurstR*. This new graph is shown in Figure 6.2. All the *Ie-eff* values shown in Figure 6.2 are listed in Tables A.6-A.10 in Appendix A.

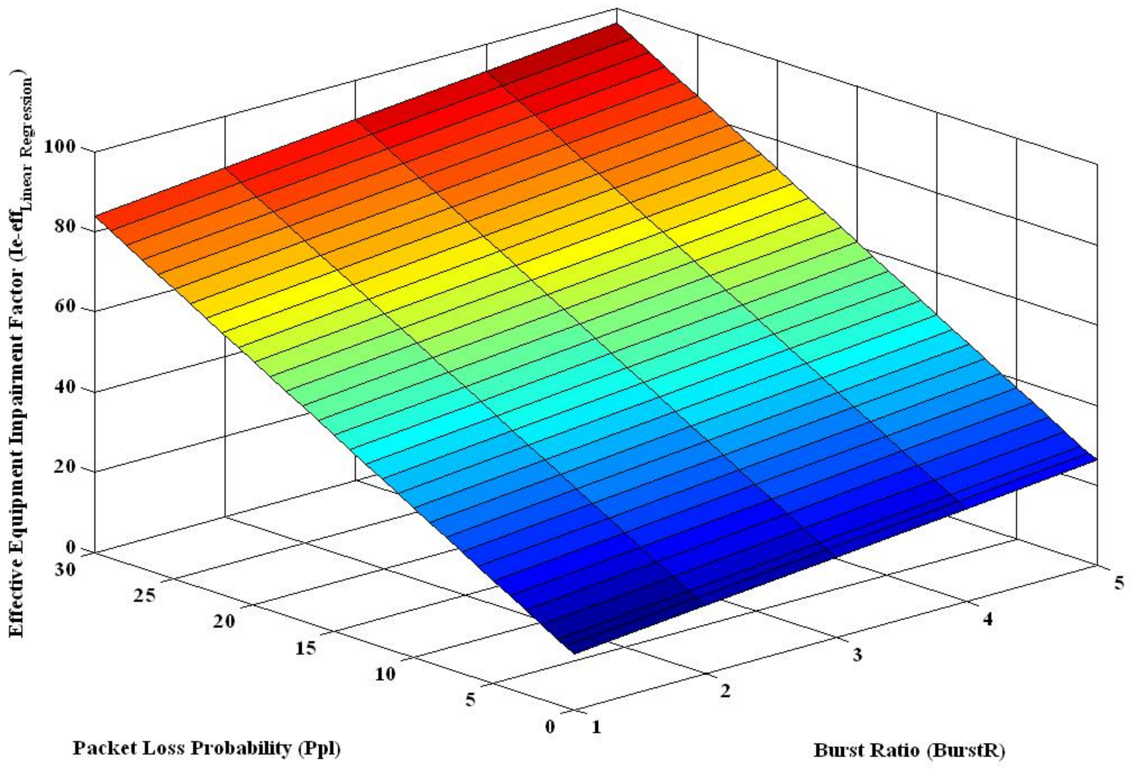

Figure 6.2: Expected *Ie-eff* (Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

By comparing Figure 6.1 and Figure 6.2, some similarities and correlations are noticed between the two graphs. Both graphs have their minimum point at *Ppl* and $BurstR$ combination of 0 and 1, respectively. Both have their maximum value at *Ppl* and $BurstR$ values of 30 and 5, respectively, also both of them have the same slope.

To visually compare the *Ie-eff* from the experiments as shown in Figure 6.1 with the *Ie-eff* from the linear regression model in Figure 6.2, both surfaces are drawn in Figure 6.3. The figure was rotated from its original angle to offer the best possible view for comparison. Visually, the fit is good although some differences can appear specially in both ends of the surface as the two surfaces do not perfectly overlap.

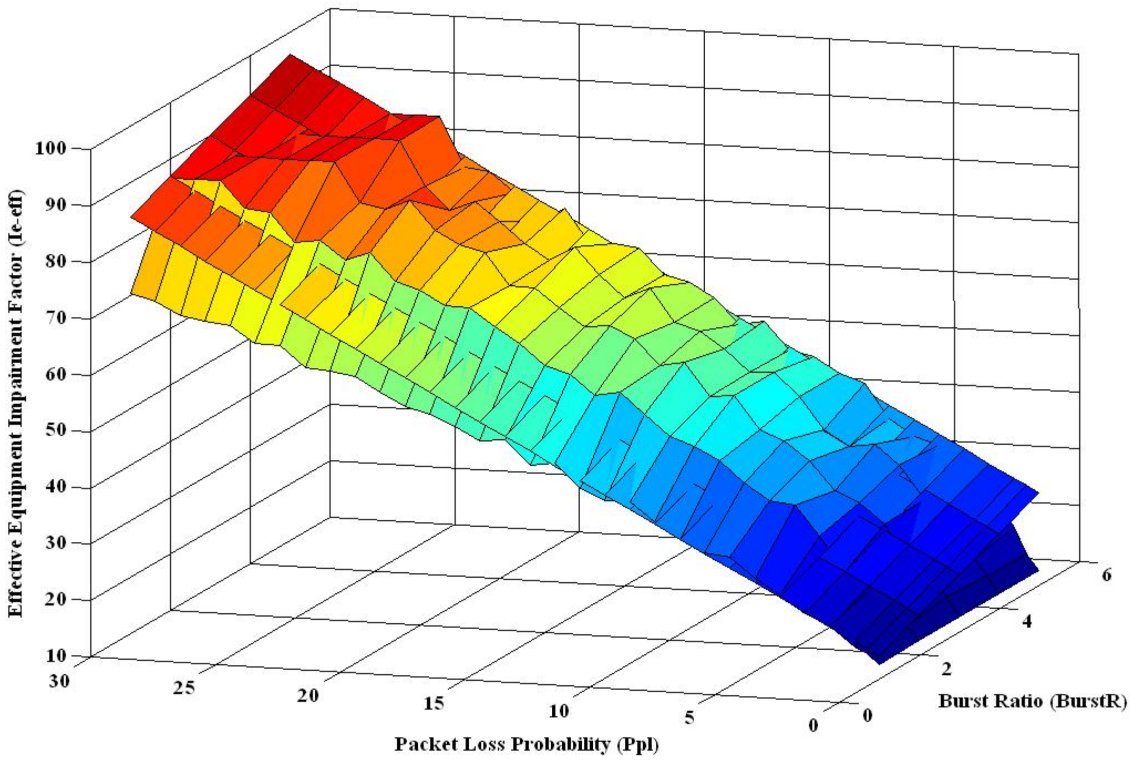

Figure 6.3: *Ie-eff* (Experimental and Linearly Regression) vs. Packet Loss Probability and Burst Ratio

To compute the goodness of the fit, the statistics that resulted from SPSS are used. The model summary Table that gives information about the strength of the relationship between the model and the predicted variable, reported that the multiple correlation coefficient (R) between the observed and the model-predicted values for the dependent variable (*Ie-eff*) has the value of 0.976 which indicates quite strong relation and a quite good fit. The R^2, the coefficient of determination, has the value of 0.953 which indicates that 95.3% of the time the variation in the independent variable is explained by the model.

Having derived the linear relation between *Ie-eff* and *Ppl* and *BurstR*, a reverse process to that described in the previous chapter can be done. i.e. MOS can be derived from *Ie-eff*. The purpose of this reverse process is to check the effect of the model on the predictive accuracy of E-model in terms of MOS.

The process start by using *Ie-eff* values obtained through linear regression analysis to calculate R-Rating Factor. In the original process *Ie-eff* was derived from the R-Rating Factor using equation (5.13), now the R-Rating Factor is derived from *Ie-eff* using equation (5.12). Based on this equation, Figure 6.4 is produced and the R-Rating Factor data is listed in Tables A.6-A.10 in Appendix A.

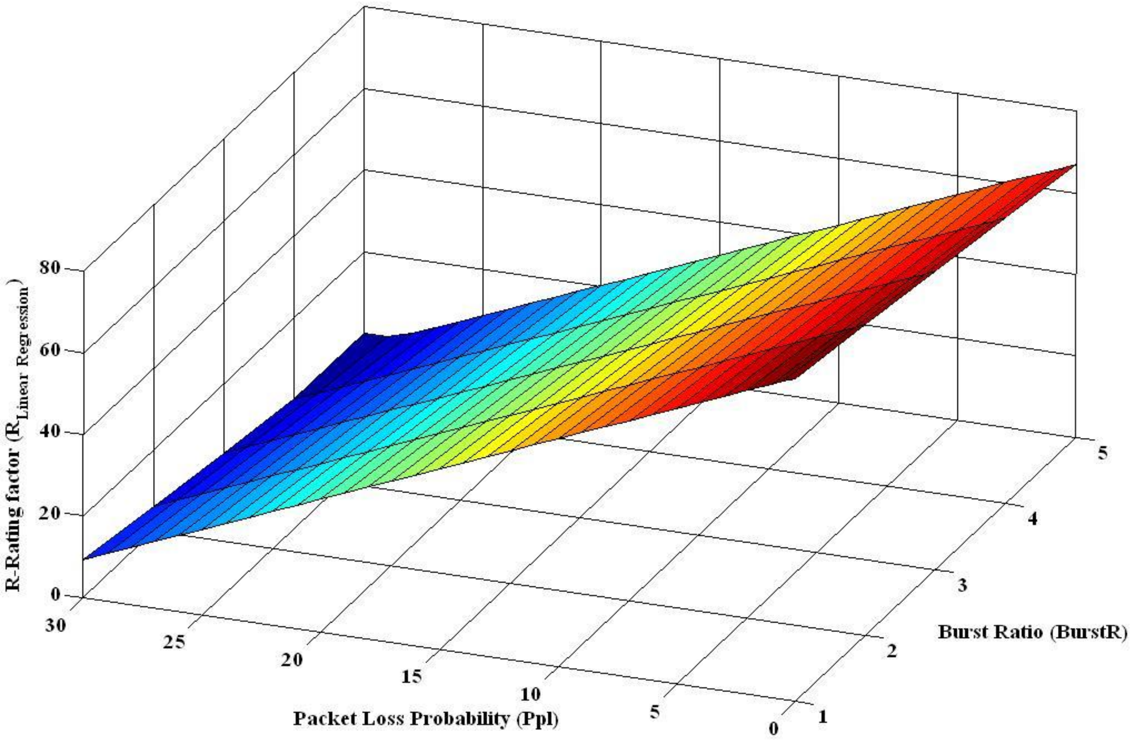

Figure 6.4: R-Rating Factor (Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

Opposite to the *Ie-eff*, the R-Rating Factor is inversely proportional to *Ppl* and *BurstR*. The maximum value achieved with the combination of $Ppl = 0$ and $BurstR = 1$ while the minimum value achieved with the combination $Ppl = 30$ and $BurstR = 5$.

Similarly, MOS_{LQE} can be derived from the R-Rating Factor using equation (3.9) to produce Figure 6.5. Again the MOS_{LQE} data is listed in Tables A.6-A.10

in Appendix A.

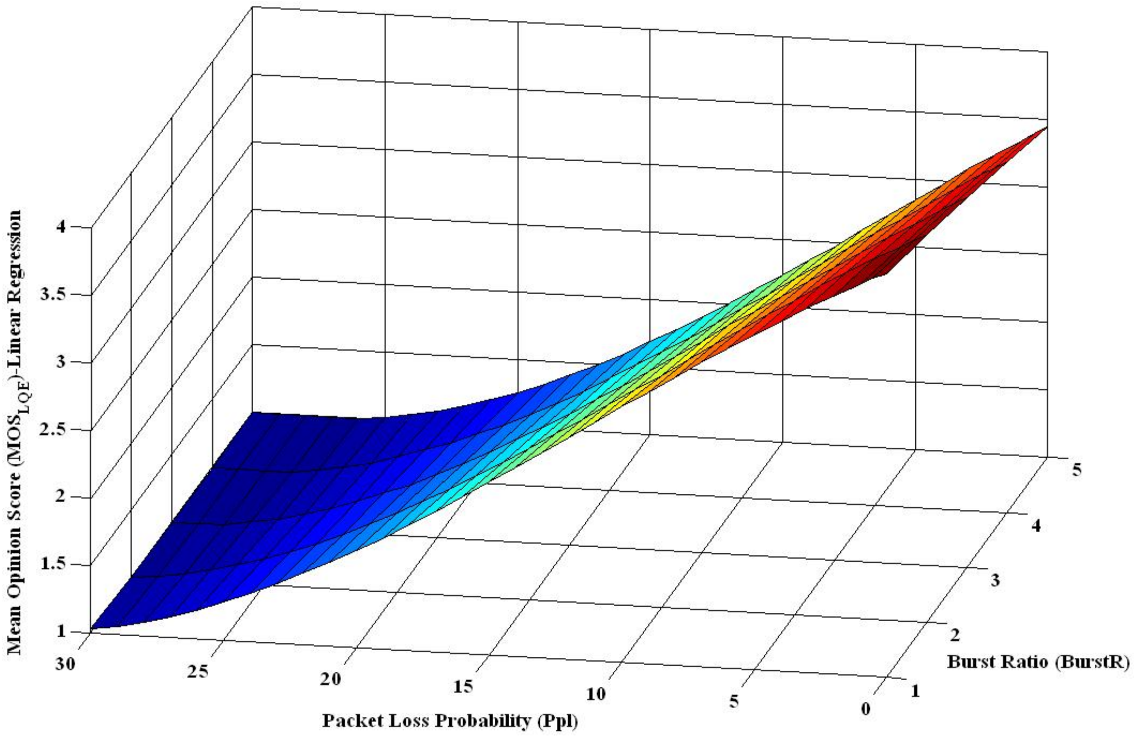

Figure 6.5: MOS_{LQE} (Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

With the generation of MOS_{LQE} values based on the linear regression, the performance of the linear regression model in comparison with the original E-model can be measured. Also MOS_{LQE} values can be used to derive MOS_{LQO} values using equation (5.11) which is the reverse equation for the correction formula developed in chapter 5 to correct the deviation between quality prediction between the E-model and PESQ. Utilising this derivation, Figure 6.6 is produced to show the values of MOS_{LQO} with Ppl and $BurstR$. MOS_{LQO} data is listed in Tables A.6-A.10 in Appendix A. MOS_{LQO} values in this figure can be compared against MOS_{LQO} obtained empirically.

In the same way $PESQ$ scores can be derived from MOS_{LQO} values using equation (3.3) to produce Figure 6.7 which relates $PESQ$ scores to Ppl and $BurstR$. Also $PESQ$ data is listed in Tables A.6-A.10 in Appendix A. These $PESQ$ scores can be compared with $PESQ$ retrieved empirically.

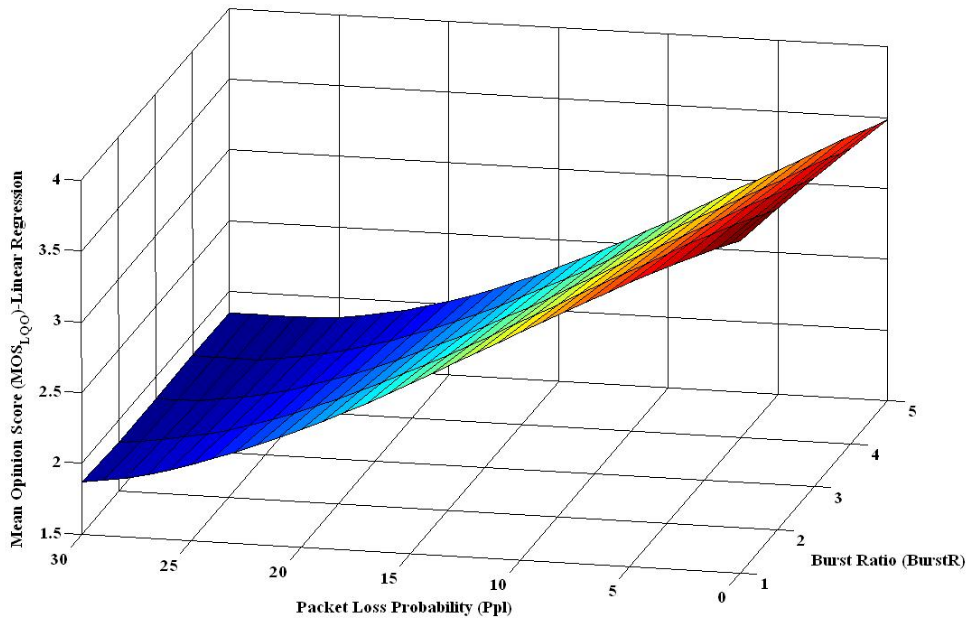

Figure 6.6: MOS_{LQO} (Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

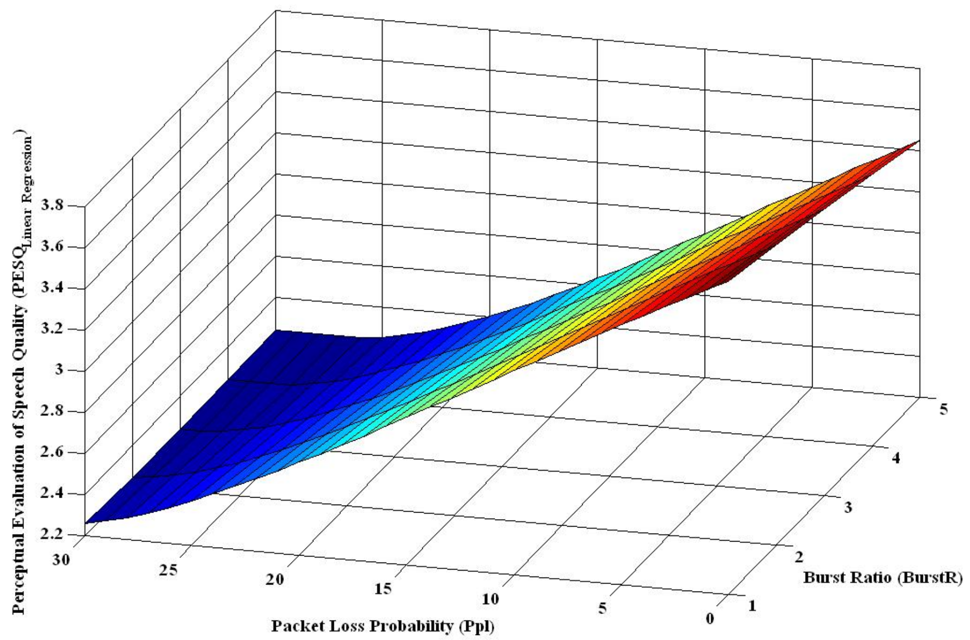

Figure 6.7: $PESQ$ (Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

Comparison Between Quality Prediction in Linear Regression Analysis and E-model:

The derived MOS_{LQE} values in Figure 6.5 represent quality prediction using the linear regression model. These values should be compared against the E-model's predicted values to determine the successfulness of the proposed model in extending the E-model to new network conditions and new speech coders.

To study the effectiveness of the linear regression in modelling *Ie-eff* and ultimately predicting the speech quality in terms of MOS score, a comparison is performed between model-predicted MOS_{LQE} (as shown in Figure 6.5 and listed in Tables A.6-A.10 in Appendix A) and E-model prediction of MOS_{LQE} (as shown in Figure 4.2 and listed in Table 4.3 in chapter 4).

As the parameters of the E-model are defined over a specific range (Table 3.1), specifically Ppl lies in the range 0 to 20 and $BurstR$ lies in the range 1 to 2. Consequently the comparison will be constrained to the MOS_{LQE} values corresponding to these ranges. For comparison between the linear regression values and the E-model predicted values outside the above ranges, the E-model range need to be extended outside these ranges using the original method of performing subjective so that the E-model covers wider range.

Correlation coefficient between MOS_{LQE} values calculated using the E-model and MOS_{LQE} values calculated using the linear regression model is calculated according to the following equation:

$$corr = \frac{\sum_m \sum_n \left(MOSE_{mn} - \overline{MOSE}\right)\left(MOSLR_{mn} - \overline{MOSLR}\right)}{\sqrt{\left(\sum_m \sum_n \left(MOSE_{mn} - \overline{MOSE}\right)^2\right)\left(\sum_m \sum_n \left(MOSLR_{mn} - \overline{MOSLR}\right)^2\right)}} \tag{6.3}$$

where

m	number of possible values of $BurstR$ (2 values)
n	number of possible values of Ppl (22 values)
$MOSE_{mn}$	The MOS (E-model) for $BurstR_m$ and Ppl_n
$\overline{MOSE}$	Average MOS (E-model) over all possible values of Ppl and $BurstR$
$MOSLR_{mn}$	The MOS (Linear Regression) for $BurstR_m$ and Ppl_n
$\overline{MOSLR}$	Average MOS (Linear Regression) over all possible values of Ppl and $BurstR$

MOS_{LQE} values for both the E-model and the linear regression model in the above ranges are listed in Tables 6.1-6.2. The reason for including these values and putting the rest of the values in the appendices is due to their importance as they reflect the quality as received by the user. In Tables 6.1-6.2 the second column represents the quality prediction using the E-model (E-model MOS_{LQE}) while the third column represents the quality prediction according to the linear regression model (Linear Regression MOS_{LQE}).

By applying equation (6.3) on the MOS_{LQE} values in Tables 6.1-6.2, the resulted correlation coefficients value is found to be 0.9762 which indicates strong relation between the Linearly regressed MOS_{LQE} values and the E-model MOS_{LQE} values.

To visually compare the MOS_{LQE} values from the E-model with the MOS_{LQE} values from the linear regression model, both graphs (in the specified range) are shown in Figure 6.8.

It appears from Figure 6.8 that the two relations for the MOS (E-model based and linear regression) both have similar characteristics but the E-model appears to be more pessimistic about the quality for large packet loss percentages and with increased burstiness in the degraded signal than the linear regression model. The average absolute difference is 0.1705 MOS while the maximum absolute difference is 0.3300 MOS_{LQE} and the standard deviation is 0.0968.

It is worth noting that when the correction formula discussed in section 5.3 is not used and the MOS_{LQE} values retrieved from the PESQ experiment are used directly to calculate the R-Rating Factor, the absolute difference was 0.9000 MOS_{LQE} which indicates that the correction formula helped in improving the accuracy of the

Ppl	**E-model** MOS_{LQE}	**Linear Regression** MOS_{LQE}
0	4.10	3.99
0.5	4.03	3.95
1	3.95	3.90
2	3.79	3.80
3	3.63	3.70
4	3.48	3.59
5	3.34	3.48
6	3.21	3.37
7	3.08	3.25
8	2.96	3.13
9	2.85	3.01
10	2.75	2.89
11	2.65	2.76
12	2.56	2.64
13	2.47	2.52
14	2.40	2.40
15	2.32	2.28
16	2.25	2.16
17	2.19	2.04
18	2.13	1.93
19	2.07	1.82
20	2.02	1.72

Table 6.1: E-model MOS_{LQE} and Linear Regression MOS_{LQE} for different possible values of Ppl for speech coder G.729. $BurstR = 1$

Ppl	E-model MOS_{LQE}	Linear Regression MOS_{LQE}
0	4.10	3.87
0.5	4.02	3.82
1	3.94	3.77
2	3.77	3.67
3	3.59	3.56
4	3.41	3.45
5	3.24	3.33
6	3.06	3.21
7	2.89	3.09
8	2.73	2.97
9	2.58	2.85
10	2.43	2.72
11	2.29	2.60
12	2.16	2.48
13	2.03	2.36
14	1.92	2.24
15	1.81	2.12
16	1.71	2.01
17	1.62	1.90
18	1.54	1.79
19	1.46	1.69
20	1.39	1.59

Table 6.2: E-model MOS_{LQE} and Linear Regression MOS_{LQE} for different possible values of Ppl for speech coder G.729. $BurstR = 2$

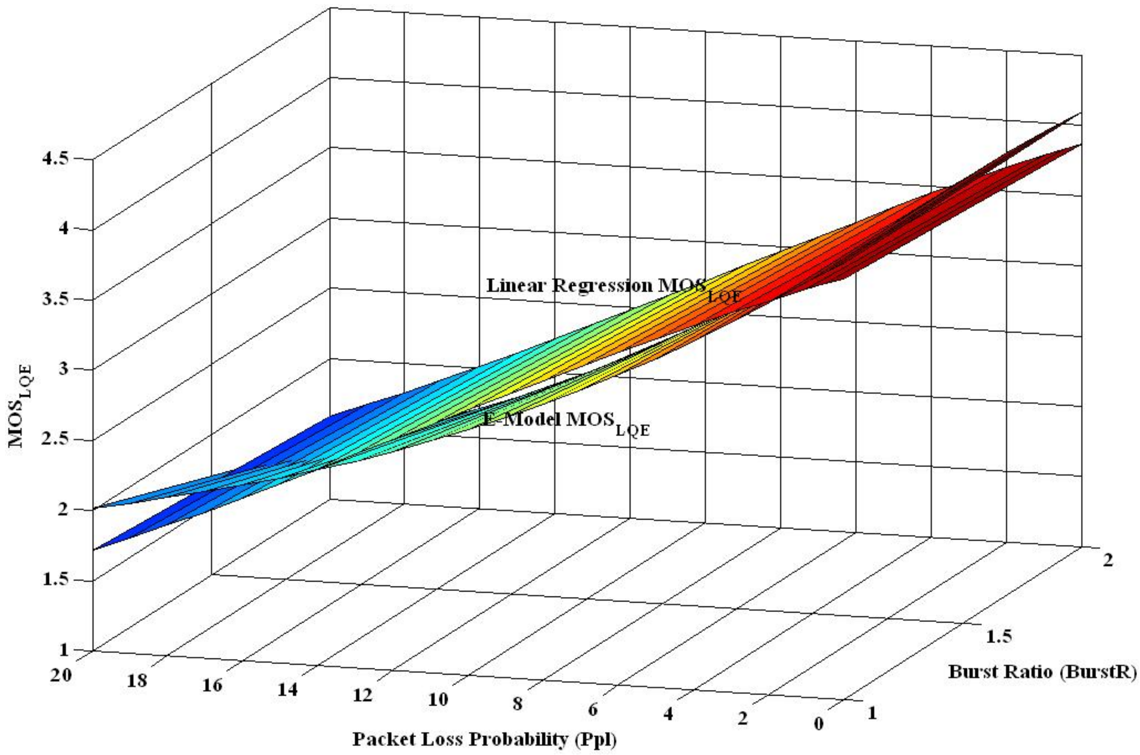

Figure 6.8: MOS_{LQE} (E-model and Linearly Regression) vs. Packet Loss Probability and Burst Ratio

prediction considerably and decreased the maximum difference to 0.3300.

In Figure 6.9 a scatter diagram is shown between the E-model-based prediction and the Linear Regression prediction. From the figure it is noted that most of the points are close to the perfect fit line which indicates close approximation by the linear regression analysis in comparison with the E-model .

Figure 6.10 shows the box plot of difference in quality prediction between the E-model and the linear regression model. From the figure it appears that the values of error in prediction are evenly distributed in the range 0 to 0.33 MOS. The first quartile lie in the range 0 to 0.09 MOS, the two middle interquartiles are in the range 0.09 to 0.25 MOS, and the last quartile between 0.25 and 0.33 MOS. The median value is 0.165 MOS.

Figure 6.11 shows the Cumulative Distribution Function (CDF) of the difference in quality prediction between the E-model and the linear regression model. The upper bound for the error was 0.33 MOS, with 15.91% below or equal to 0.05 MOS,

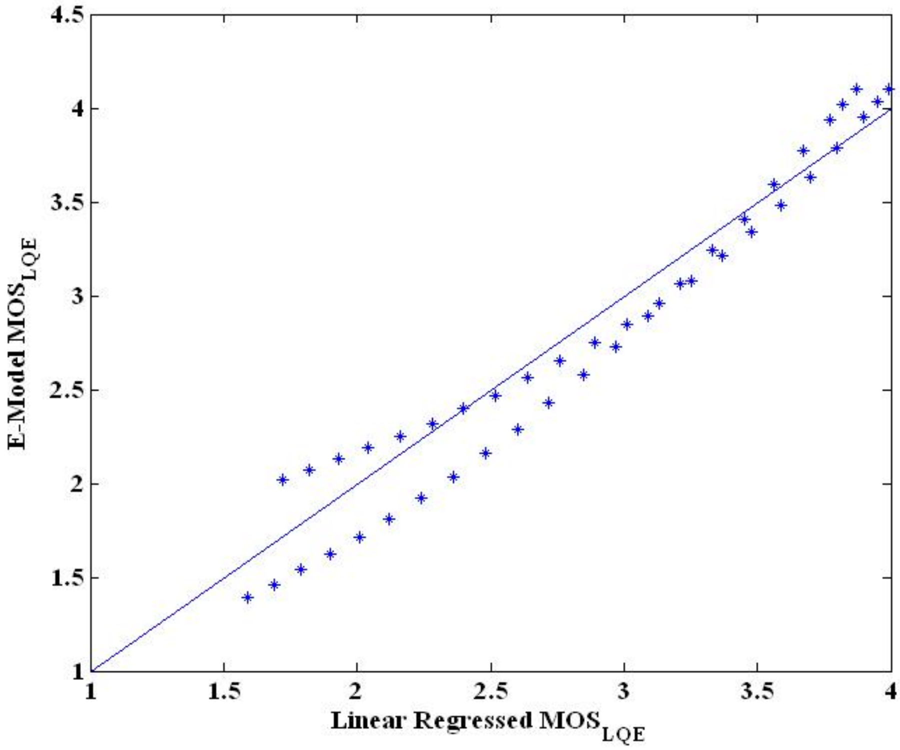

Figure 6.9: Scatter Diagram of quality prediction using Linear Regression

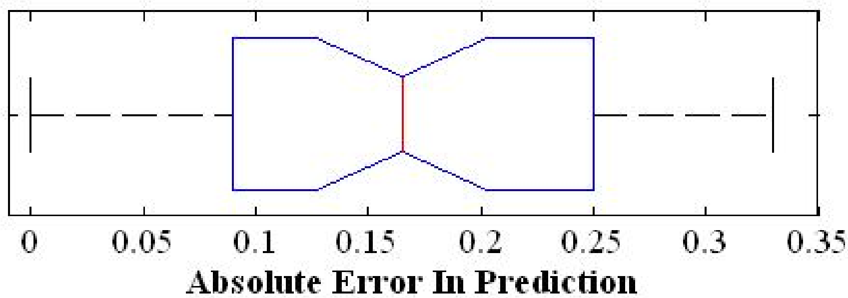

Figure 6.10: Box Plot of the error in Linearly Regressed prediction

0.2955 MOS below or equal to 0.10 MOS, 0.6591 MOS below or equal to 0.20 MOS and 0.8864 MOS below or equal to 0.3 MOS.

The results above indicate that linear regression is able to model speech quality in term of MOS_{LQE} with a certain level of accuracy and this provided the motivation to derive non linear models as discussed in the next section.

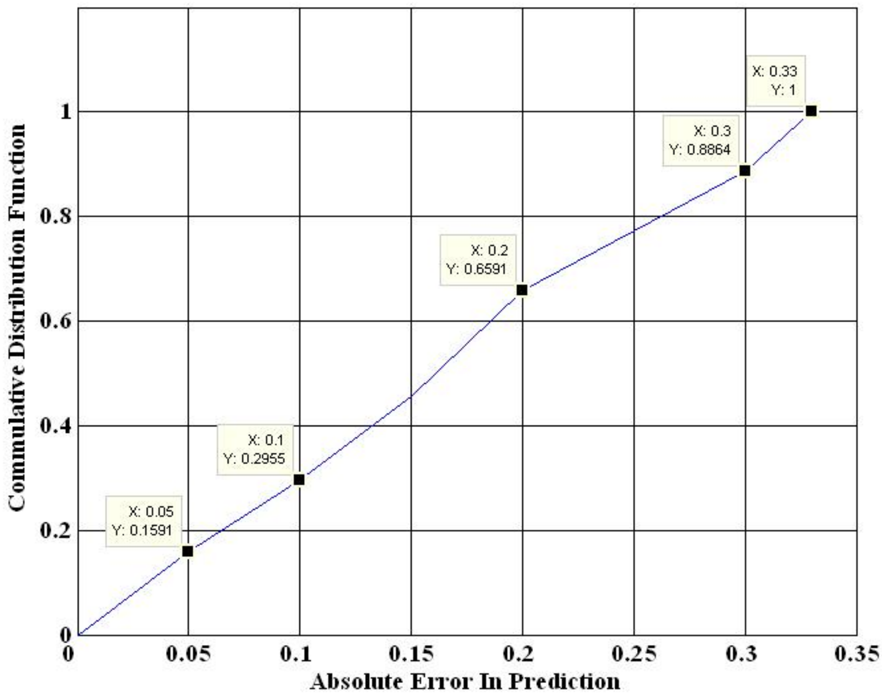

Figure 6.11: Cumulative Distributed Function (CDF) of quality prediction error

6.3 Non Linear Regression

Using SPSS version 12 [163] , and the data for *Ie-eff* shown in Figure 6.1 and listed in Tables A.1-A.5 in Appendix A, a multivariate non linear model is derived to relate *Ie-eff* with both *Ppl* and *BurstR*. SPSS was configured to go as far as a cubic polynomial of degree 3 according the following general equation:

$$\begin{aligned} \textit{Ie-eff} &= a_1 * BurstR + a_2 * BurstR^2 + a_3 * BurstR^3 \\ &\quad + b_1 * Ppl + b_2 * Ppl^2 + b_3 * Ppl^3 + c \end{aligned} \tag{6.4}$$

The equation that resulted from the non linear regression analysis is:

$$\begin{aligned} \textit{Ie-eff} &= 25.0885 * BurstR - 6.6627 * BurstR^2 + 0.5910 * BurstR^3 \\ &\quad + 4.01085 * Ppl - 0.0858 * Ppl^2 + 0.0011 * Ppl^3 - 14.6495 \end{aligned} \tag{6.5}$$

Equation (6.5) relates *Ie-eff* to both *Ppl* and *BurstR*. If you compare this to the original *Ie-eff*'s equation

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{6.6}$$

It can be noticed that the speech coder dependent parameters (Ie and Bpl) have disappeared. These parameters resulted from subjective tests. As such *Ie-eff* as it is defined in equation (6.5) does not depend on the time-consuming subjective tests to normalise its equations. This was the objective of deriving such model, to bypass these subjective tests.

Using equation (6.5) a new 3-dimensional graph can be drawn to relate *Ie-eff* against Ppl and $BurstR$. This new graph is shown in Figure 6.12 and all the *Ie-eff* values shown in Figure 6.12 are listed in Tables A.11-A.15 in Appendix A.

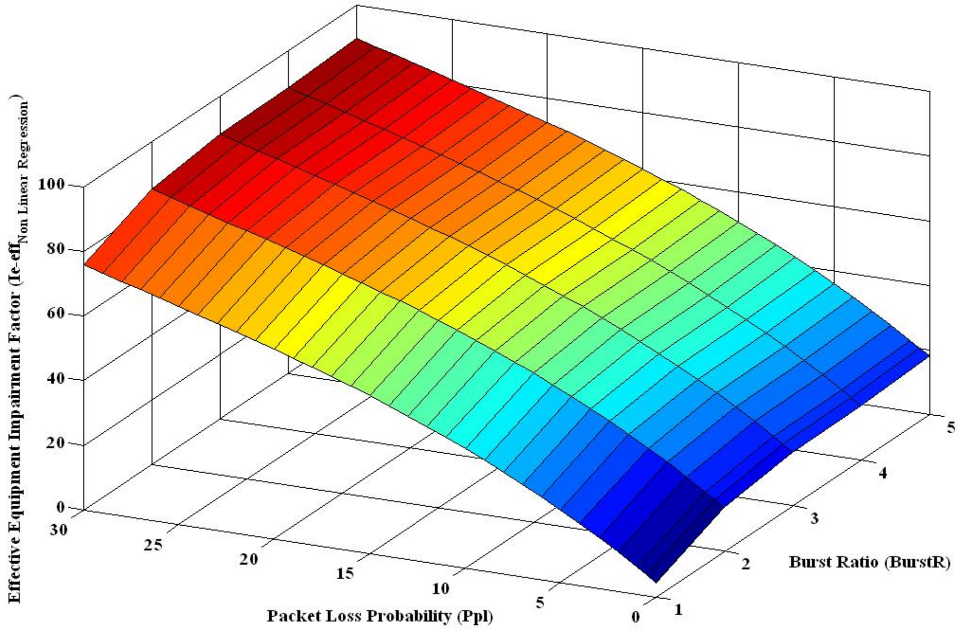

Figure 6.12: *Ie-eff* (Non Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

Figure 6.13 compares the *Ie-eff* from the experiments as shown in Figure 6.1 with the *Ie-eff* from the non linear regression model in Figure 6.12. The figure was rotated from its original angle to offer the best possible view for comparison. In general the fit was good from the visual aspect although some differences can appear specially in both ends of the surface.

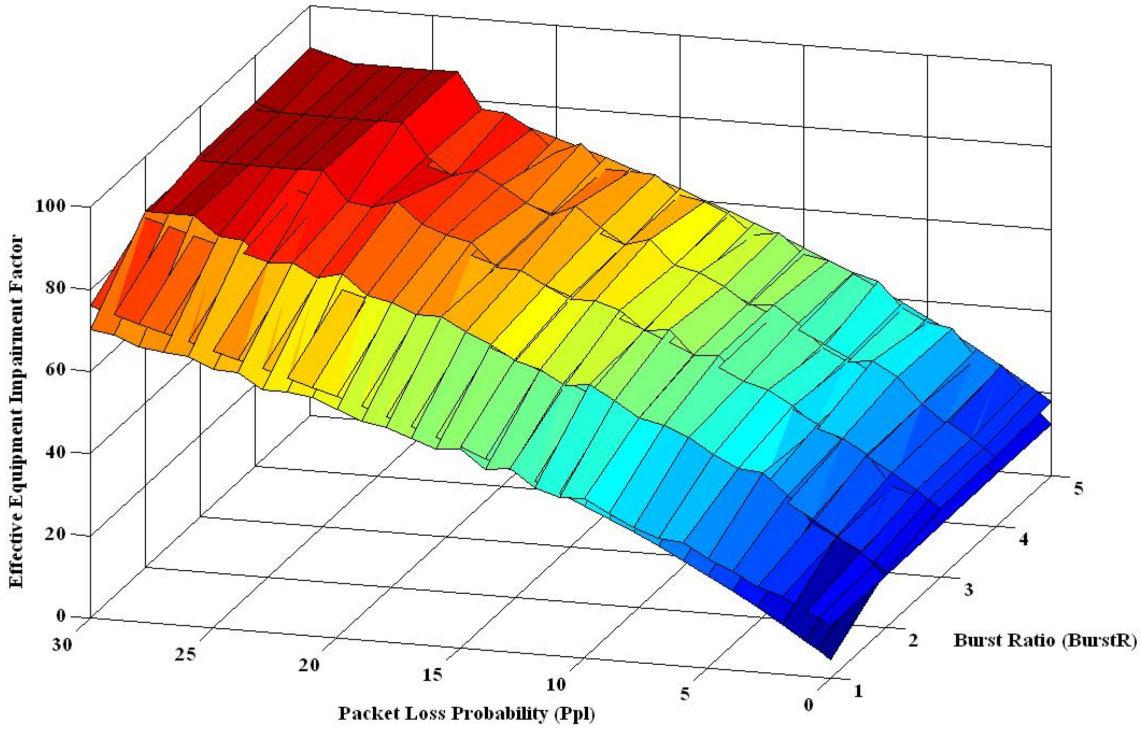

Figure 6.13: *Ie-eff* (Experimental and Non Linear Regressed) vs. Packet Loss Probability and Burst Ratio

From the figure, some similarities and correlations are noticed between the two models. Both graphs have their minimum point at Ppl and $BurstR$ combination of 0 and 1, respectively. Both have their maximum value at Ppl and $BurstR$ values of 30 and 5, respectively, also both of them have the same slope.

If the fit of the non linear regression is compared with the fit of the linear regression analysis, it can be noticed that the non linear regression model offers more accurate model for *Ie-eff* than its linear counterpart.

The multiple correlation coefficient (R) between the observed and the model-predicted values for the dependent variable (*Ie-eff*) has the value of 0.992 which indicates a strong relation and a good fit. The R^2, the coefficient of determination has the value of 0.984 which indicates that 98.4% of the time the variation in the independent variable is explained by the model.

From the non linear relation derived above, now a reverse to what was done in the previous chapter can be calculated. i.e. derive R-Rating Factor values from the

Ie-eff using equation (5.12). Based on this equation, Figure 6.14 is derived and the data is listed in Tables A.11-A.15 in Appendix A.

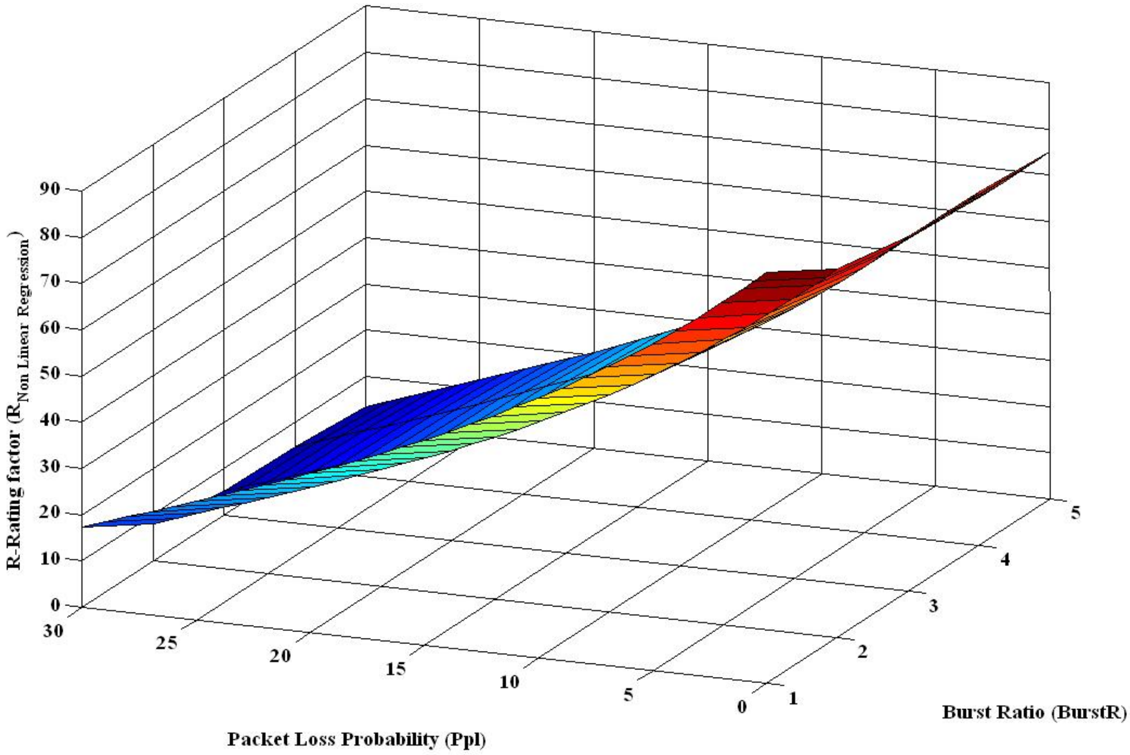

Figure 6.14: R-Rating Factor vs. (Non Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

Similarly, MOS_{LQE} can be derived from the R-Rating Factor using equation (3.9) to produce Figure 6.15. Again the data is listed in Tables A.11-A.15 in Appendix A.

Using MOS_{LQE} values produced based on the non linear regression, the effect of the non linear regression model on the predictive accuracy of the speech quality can be measured by comparing the performance of the non linear regression model with the performance of the original E-model.

MOS_{LQE} values can also be used to derive MOS_{LQO} values using equation (5.11) which is the reverse equation for the correction formula developed in chapter 5 to correct the deviation between quality prediction between the E-model and PESQ. Utilising this derivation, Figure 6.16 is produced to show the values of MOS_{LQO} with Ppl and $BurstR$. MOS_{LQO} data is listed in Tables A.11-A.15 in Appendix A. MOS_{LQO} values in this figure can be compared against PESQ-derived MOS_{LQO}

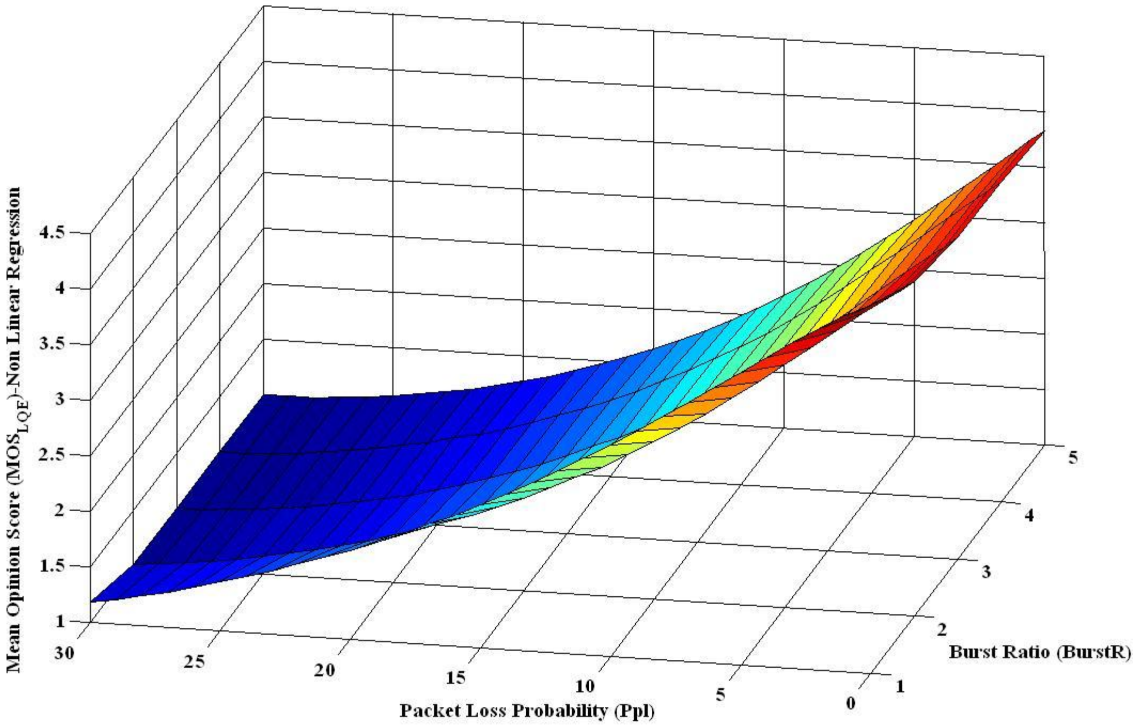

Figure 6.15: MOS_{LQE} (Non Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

values obtained empirically.

In the same way $PESQ$ scores can be derived from the MOS_{LQO} values using equation (3.3) to produce Figure 6.17 which relates $PESQ$ to Ppl and $BurstR$. Also the data are listed in Tables A.11-A.15 in Appendix A. These $PESQ$ scores can be used for comparison with $PESQ$ scores obtained empirically.

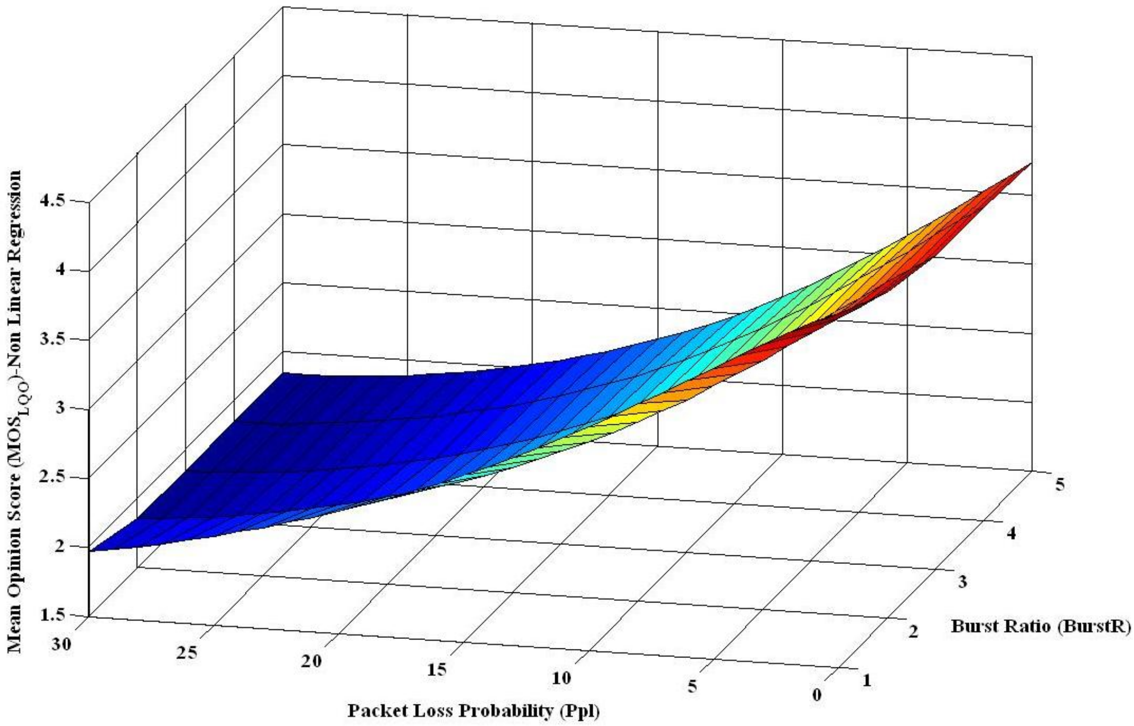

Figure 6.16: MOS_{LQO} (Non Linear Regressed) vs. Packet Loss Probability and Burst Ratio

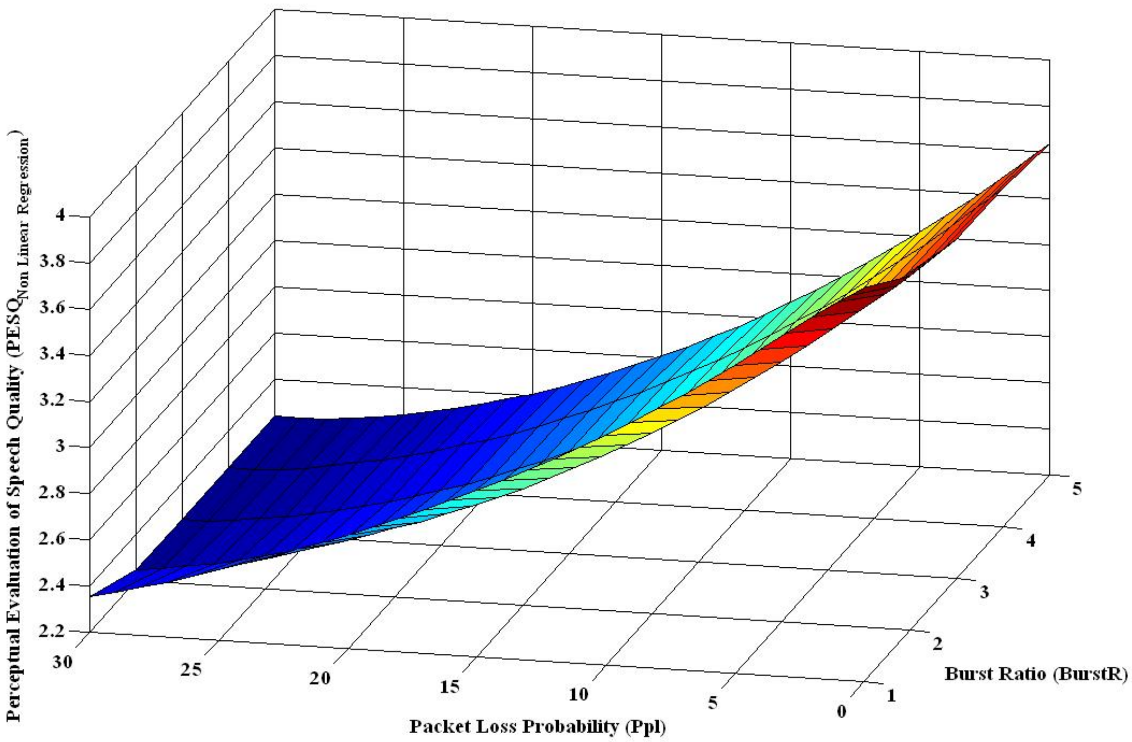

Figure 6.17: $PESQ$ (Non Linear Regressed) vs. Packet Loss Probability and Burst Ratio

Comparison Between Quality Prediction in Non Linear Regression Analysis and E-model:

The derived MOS_{LQE} values in Figure 6.15 represent the model prediction (Non linear regression of the quality) values. These values should be compared against the E-model's predicted values to determine the successfulness of the non linear regression model in extending the E-model to new network conditions and new speech coders.

To study the effectiveness of the non linear regression in modelling *Ie-eff* and ultimately predicting the speech quality in terms of MOS score, a comparison is performed between model-predicted MOS_{LQE} (as shown in Figure 6.15 and listed in Tables A.11-A.15 in Appendix A) and E-model predicted MOS_{LQE} (as shown in Figure 4.2 and listed in Table 4.3 in chapter 4).

The comparison will be constrained to the MOS_{LQE} values corresponding to *Ppl* range 0 to 20 and $BurstR$ range 1 to 2 as defined by the E-model and listed in Table 3.1. For comparison between the non linear regression values and the E-model predicted values outside the above ranges, the E-model range need to be extended outside these ranges which is something can only be done if further subjective tests are performed so that the E-model covers wider range.

Tables 6.3-6.4 list the MOS_{LQE} values for both the E-model and the non linear regression model in the above ranges.

By applying equation (6.3) to calculate the correlation coefficient between MOS_{LQE} values in the non linear model and the E-model (both list in Tables 6.3-6.4), the resultant correlation coefficients value is found to be 0.9882 which indicates strong positive correlation between the non Linearly regressed MOS_{LQE} values and the E-model MOS_{LQE} values. This correlation is higher than the correlation found between the linear regression model and the E-model which was 0.9762.

The average absolute difference is 0.1100 MOS which indicates an improvement over the estimation of the linear model while the maximum absolute difference is 0.3000 MOS_{LQE} and the standard deviation is 0.0976.

Ppl	**E-model** MOS_{LQE}	**Non Linear Regression** MOS_{LQE}
0	4.10	4.31
0.5	4.03	4.25
1	3.95	4.20
2	3.79	4.07
3	3.63	3.93
4	3.48	3.78
5	3.34	3.63
6	3.21	3.48
7	3.08	3.33
8	2.96	3.19
9	2.85	3.04
10	2.75	2.90
11	2.65	2.77
12	2.56	2.64
13	2.47	2.52
14	2.40	2.40
15	2.32	2.29
16	2.25	2.18
17	2.19	2.08
18	2.13	1.98
19	2.07	1.89
20	2.02	1.80

Table 6.3: E-model MOS_{LQE} and Non Linear Regression MOS_{LQE} for different possible values of Ppl for speech coder G.729. $BurstR = 1$

Ppl	E-model MOS_{LQE}	Non Linear Regression MOS_{LQE}
0	4.10	4.01
0.5	4.02	3.93
1	3.94	3.85
2	3.77	3.69
3	3.59	3.52
4	3.41	3.35
5	3.24	3.18
6	3.06	3.02
7	2.89	2.86
8	2.73	2.70
9	2.58	2.56
10	2.43	2.42
11	2.29	2.29
12	2.16	2.16
13	2.03	2.05
14	1.92	1.94
15	1.81	1.83
16	1.71	1.74
17	1.62	1.65
18	1.54	1.57
19	1.46	1.49
20	1.39	1.43

Table 6.4: E-model MOS_{LQE} and Non Linear Regression MOS_{LQE} for different possible values of Ppl for speech coder G.729. $BurstR = 2$

It is worth noting that if the correction formula discussed in section 5.3 is not used and the MOS_{LQE} values retrieved from the PESQ experiment are used directly to calculate the R-Rating Factor, the absolute difference was 0.7700 MOS_{LQE} which indicates that the correction formula helped in improving the accuracy of the prediction considerably and decreased the maximum difference to 0.3000.

To visually compare the MOS_{LQE} values from the E-model with the MOS_{LQE} values from the non linear regression model, both graphs (in the specified range) are shown in Figure 6.18. From Figure 6.18 it is clear that the two relations for the MOS (E-model based and non linear regression) both have similar characteristics but the E-model seems to be more pessimistic at low loss values.

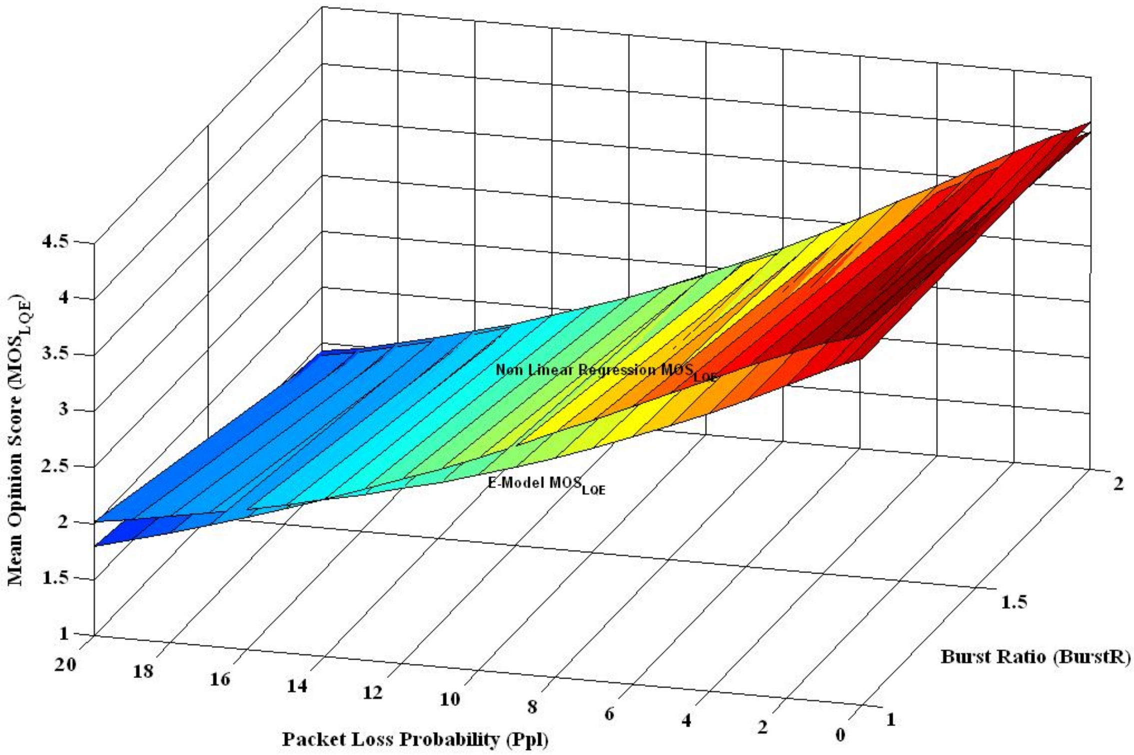

Figure 6.18: MOS_{LQE} (E-model and Non Linearly Regressed) vs. Packet Loss Probability and Burst Ratio

In Figure 6.19 a scatter diagram visualises the correlation between the E-model-based prediction and the non linear regression prediction. As the correlation is high, most of the points are concentrated around the perfect fit line.

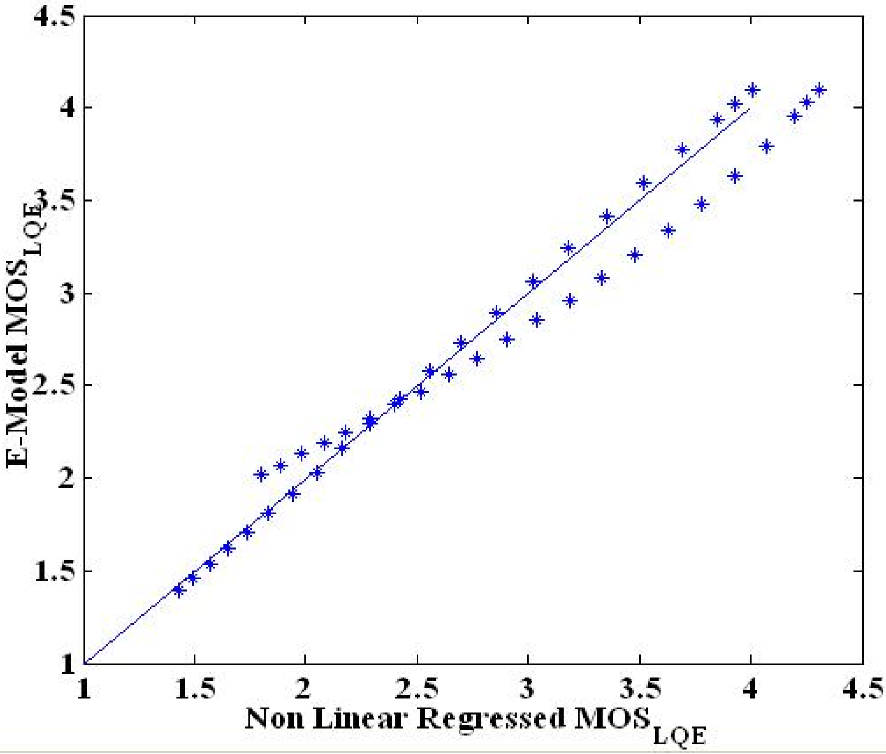

Figure 6.19: Scatter Diagram of quality prediction using Non Linear Regression

Figure 6.20 shows the box plot of difference in quality prediction between the E-model and the non linear regression model. the purpose of this figure is to visualise the distribution range of the differences and in which range they are concentrated. From the figure it appears that the predicted error values are clustered in the lower range i.e. toward zero. The first quartile (first 25% of the data) lie in the range 0 to 0.03 MOS, and the first two quartiles (50% of the data) are in the range 0 to 0.075 (median value) MOS which is less than the value for the first quartile in the linear model. This indicates better overall approximation for the non linear regression in comparison with the linear model for large percentage of the points. The third quartile lies between 0.075 and 0.2 while the last quartile lies between 0.2 and 0.3 MOS. This figure was plotted using the same scale used to plot the box plot for the linear regression model to simplify comparison.

Figure 6.21 shows the Cumulative Distribution Function (CDF) of the difference in quality prediction between the E-model and the non linear regression model. The upper bound for the error was 0.3 MOS, with 40.91% below or equal to 0.05 MOS, 61.36% below or equal 0.10 MOS, 0.75% below 0.20 MOS and 88.64% below 0.25

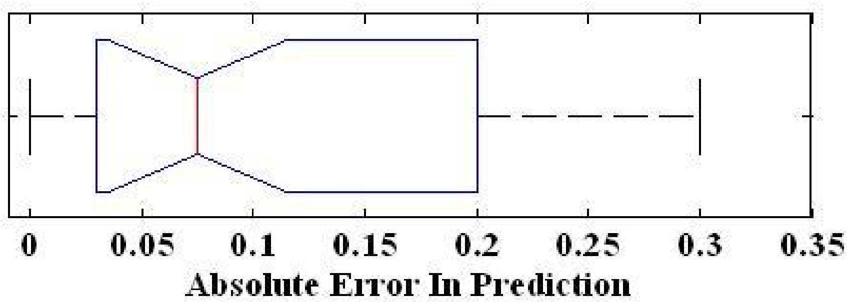

Figure 6.20: Box Plot of the error in Non Linearly Regressed prediction

MOS. These numbers indicate a better approximation than the linear regression models as higher percentages of points are below certain error levels in comparison with the same error levels in linear regression analysis.

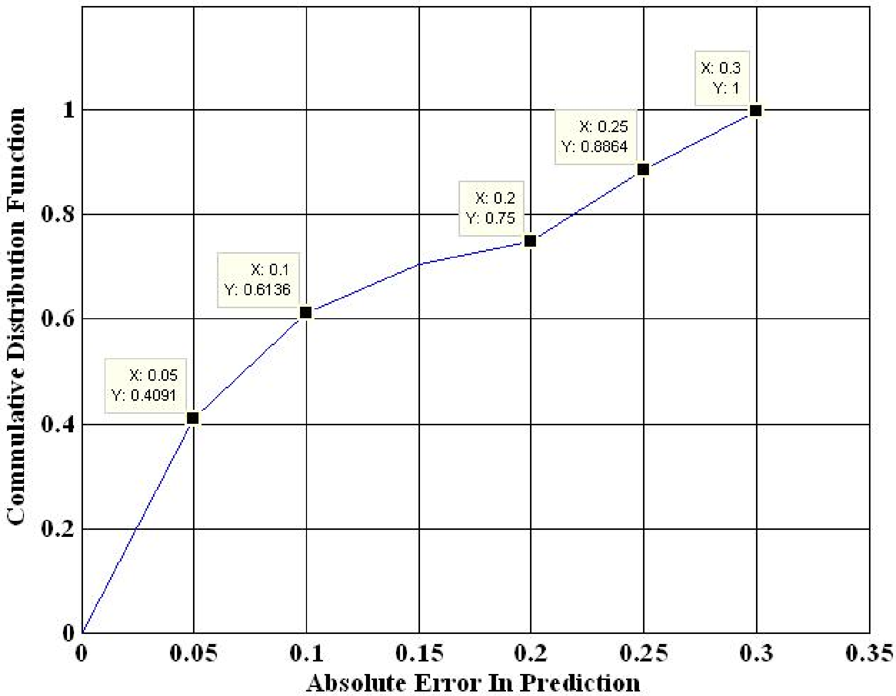

Figure 6.21: Cumulative Distributed Function (CDF) of quality prediction error

The results above indicates that non linear regression is able to model speech quality in terms of MOS_{LQE} with better level of accuracy than linear regression models. The next section discuss the performance of Artificial Neural Network (ANN) models in estimating the speech quality.

6.4 Artificial Neural Network

In the previous two sections, linear and non linear regression analysis techniques were used to derive a relation between *Ie-eff* and both of *Ppl* and *BurstR*. The performance of both techniques was tested using several statistical approaches. As finding such a potential relation is a function approximation problem, any technique that is able to approximate a function accurately from set of values can be used for this task. As the task here is to find a relation between *Ie-eff* and both of *Ppl* and *BurstR* and the focus is not on the function approximation techniques themselves, the performance of Artificial Neural Networks (ANNs) will be tested in this section as ANNs are usually used for either pattern classification or function approximation problems. The performance of ANN will also be tested against the performance of linear and no-linear regression [4, 136].

ANNs are adjusted, or trained so that a particular input leads to a specific target output. Training is performed using input vectors and the corresponding target vectors based on a comparison of the network output and the target, until the network output matches the target. During training, weights and biases of the ANN are modified until it can approximate a function, associate input vectors with a specific output vectors, or classify input vectors in an appropriate way as defined by network designer.

Multi-layer Feedforward ANNs, trained using the Backpropagation algorithm, are used to train a network in order to approximate a function. Multiple-layer networks can be powerful and can approximate functions that can not be approximated using single layer networks because single layer networks are limited in power. For instance, two-Layer networks with biases, a sigmoid transfer function in the first layer and a linear transfer function in the output layer can be used as general function approximator networks as they are capable of approximating any function with a finite number of discontinuities arbitrarily well, given sufficient neurons in the hidden layer. Using a nonlinear transfer function such as the sigmoid function allows the network to learn nonlinear and linear relationships between input and output vectors. The sigmoid functions squash the output into a limited range, between 0 and 1 or between -1 and +1, therefore to be able to produce values outside the range -1 to +1, it is necessary to use a function that is capable of mapping to a wider range such as a linear function. With many nodes in the hidden layer, the power of the network increases, as the number of available weights to adjust increases, given the the network the capability to approximate complex functions and relations [4, 59].

Several algorithms can be used to train multi-layer backpropagation networks. For instance, Levenberg-Marquardt (LM) algorithm performs well in terms of convergence speed for function approximation problems using moderate-sized feedforward ANNs (up to several hundred weights) where the approximation must be very accurate. Other algorithms such Resilient Backpropagation are faster and more accurate in tasks such as pattern recognition. LM also has a very efficient MATLAB implementation, since the solution of the matrix equation is a built-in function, so its importance becomes more pronounced in a MATLAB setting [4, 120].

The above characteristics fit the problem in hand, finding a function approximation model to characterise the relation between *Ppl* and *BurstR* with *Ie-eff*. For this problem a two layer neural network with sigmoid transfer function in the first layer and linear transfer function in the output layer will be used and the network will be trained using LM algorithm.

In the previous two sections an attempt was made to derive a linear and non linear regression models and in this section the performance of an ANN architecture will be tested. In this network the number of input units are restricted to two corresponding to *Ppl* and *BurstR*, the number of output unit is one corresponding to *Ie-eff*. The number of neurons in the hidden layer are free parameters determined empirically. Figure 6.22 shows an example feedforward neural network with two inputs, one output, and 4 neurons in the hidden layer, sigmoid function is used in the first layer and linear transfer function is used in the output layer.

The first step in the experiment is to divide the *Ie-eff* data shown in Figure 6.1 and listed in Tables A.1-A.5 in Appendix A into training, validation and test subsets to improve generalisation accuracy and avoid overfitting the trained network into the training data. For the 160 input vectors available, 100 vectors will be used for training, 30 for validation and 30 for testing.

Among the 160 vectors, there are five possible burst ratios and 32 possible packet loss probabilities belong to each burst ratio. The training, validation and test sets were selected as equally spaced points throughout the original data to avoid bias in

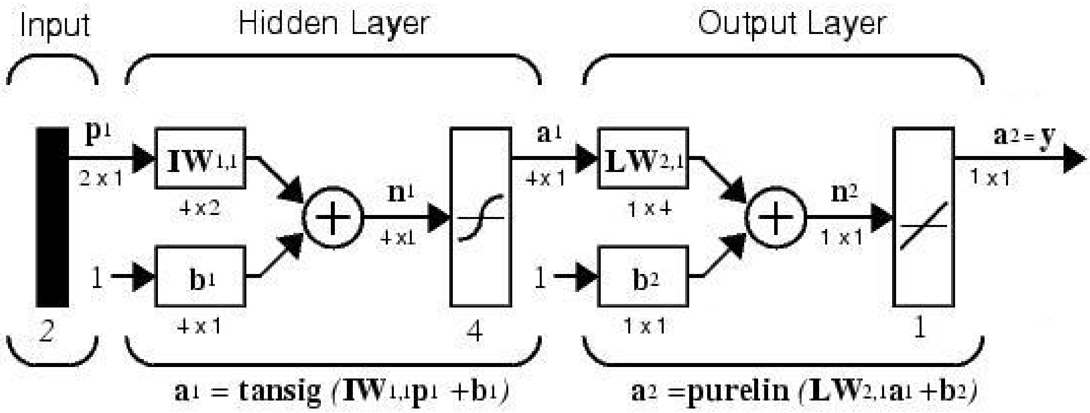

Figure 6.22: Multi-layer Feedforward Neural Network

the training set. Table 8.1 illustrates the division of data into training, validation and testing sets for any burst ratio. The same pattern is repeated for other burst ratios.

Different number of neurons in the hidden layer were attempted ranging from one neuron to 40 neurons. With one neuron the total number of weights and biases in the network equals 5, knowing with 40 neurons the network will have the capability to fully remember the training set, this was done to study the effect of the number of neurons on the performance of the test set. For each setting the experiment was repeated for 30 different trials, where different random initial weights are used in each trial. This counts to 1200 experiments in total (40x30).

The performances of all the experiments in terms of training set and test set are listed in Appendix B in Tables B.1-B.20. During the experiment the network was allowed to be trained for up to 10000 epochs, although in all cases training stopped before reaching this number due to the error in the validation set exceeding the error in the training set.

The best network in terms of performance of the test set was found to be a network with 5 neurons in the hidden layer. As the purpose is to find how good ANNs perform in modelling *Ie-eff*, the best retrieved network will be used for subsequent derivations in this section. This is similar to the approach with linear regression and non linear regression where the best approximation function was used.

Ppl	**Assigned data set**	**Ppl**	**Assigned data set**
0	Training	15	Training
0.5	Training	16	Training
1	Training	17	Training
2	Validation	18	Validation
3	Testing	19	Testing
4	Training	20	Training
5	Training	21	Training
6	Training	22	Training
7	Validation	23	Validation
8	Testing	24	Testing
9	Training	25	Training
10	Training	26	Training
11	Training	27	Training
12	Training	28	Training
13	Validation	29	Validation
14	Testing	30	Testing

Table 6.5: Division of data set into training, validation, and testing

The input to this network are Ppl and $BurstR$, if you compare this to the original $Ie\text{-}eff$'s equation

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{6.7}$$

It can be noticed that the speech coder dependent parameters (Ie and Bpl) have been absorbed in the form of weights and biases in the ANN model. These parameters were resulted from subjective tests. As such $Ie\text{-}eff$ as derived from the ANN model does not depend on the time-consuming subjective tests to normalise its equations which is the goal of this derivation.

When different combinations of Ppl and $BurstR$ are fed into the best network obtained from the experiment, which has 5 neurons in the hidden layer, a new 3-dimensional graph can be drawn to relate $Ie\text{-}eff$ against Ppl and $BurstR$. This new graph is shown in Figure 6.23 and all the *Ie-eff* values shown in Figure 6.23 are listed in Tables B.21-B.25 in Appendix B.

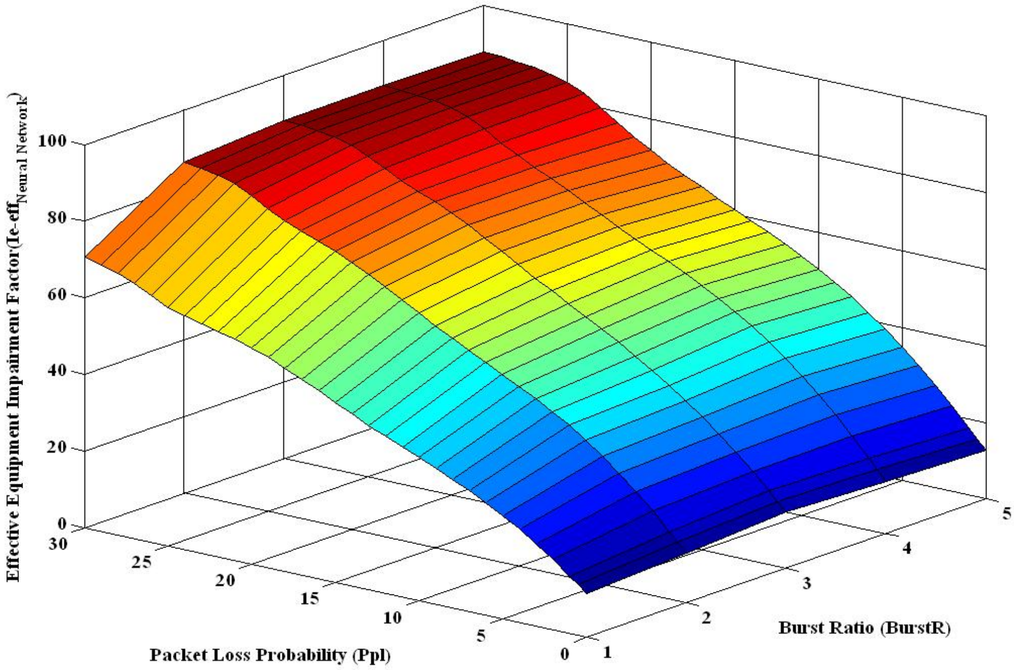

Figure 6.23: *Ie-eff* (Neural Network derived) vs. Packet Loss Probability and Burst Ratio

Figure 6.24 visually compares *Ie-eff* from the experiments as shown in Figure 6.1 with *Ie-eff* from the ANN derivation shown in Figure 6.23 by drawing both surfaces in the figure. The fit was good to the degree it is hard to distinguish between the experimental values and the ANN predicted values. It is better than both linear and non linear approximations.

The multiple correlation coefficient (R) between the observed and the ANN-predicted values for the dependent variable (*Ie-eff*) has the value of 0.998 which indicates strong positive correlation and a good fit. The R^2, the coefficient of determination has the value of 0.996 which indicates that 99.6% of the time the variation in the independent variable is explained by the model. The correlation value is higher than that retrieved by the linear regression model (97.6%) and non linear regression model (99.2%).

Using *Ie-eff* values derived using the ANN model, R-Rating Factor can be de-

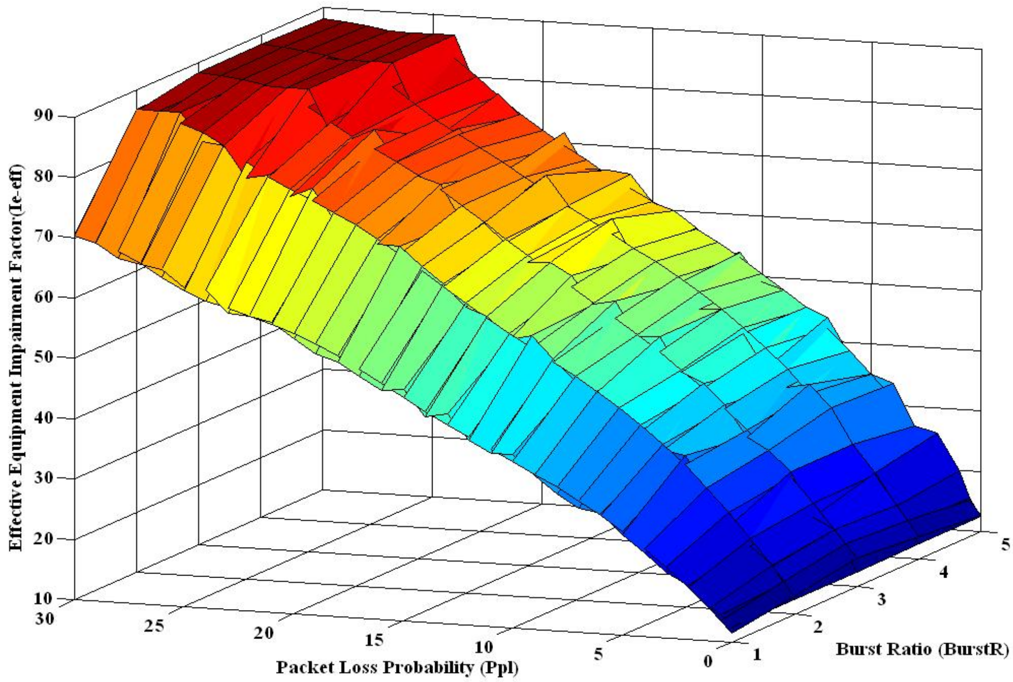

Figure 6.24: *Ie-eff* (Experimental and Neural Network predicted) vs. Packet Loss Probability and Burst Ratio

rived using equation (5.12). Based on this equation, Figure 6.25 is produced and the full data is listed in Tables B.21-B.25 in Appendix B.

Similarly, MOS_{LQE} can be derived from R-Rating Factor using equation (3.9) to produce Figure 6.26. Again the data is listed in tables B.21-B.25 in Appendix B.

MOS_{LQE} values can be used to test the performance of the ANN in comparison with the performance of the original E-model in measuring the speech quality. Also, MOS_{LQE} values can be used to derive MOS_{LQO} values using equation (5.11) which is the reverse equation for the correction formula that was developed in chapter 5 to correct the deviation between quality prediction between the E-model and PESQ. Utilising this derivation, Figure 6.27 is produced to show the values of MOS_{LQO} with *Ppl* and $BurstR$. MOS_{LQO} data is listed in Tables B.21-B.25 in Appendix B.

In the same way $PESQ$ scores can be derived from the MOS_{LQO} values using equation (3.3) to produce Figure 6.28 which relates $PESQ$ to *Ppl* and $BurstR$. These scores could be used for comparison with $PESQ$ scores obtained empirically. The data are listed in Tables B.21-B.25 in Appendix B.

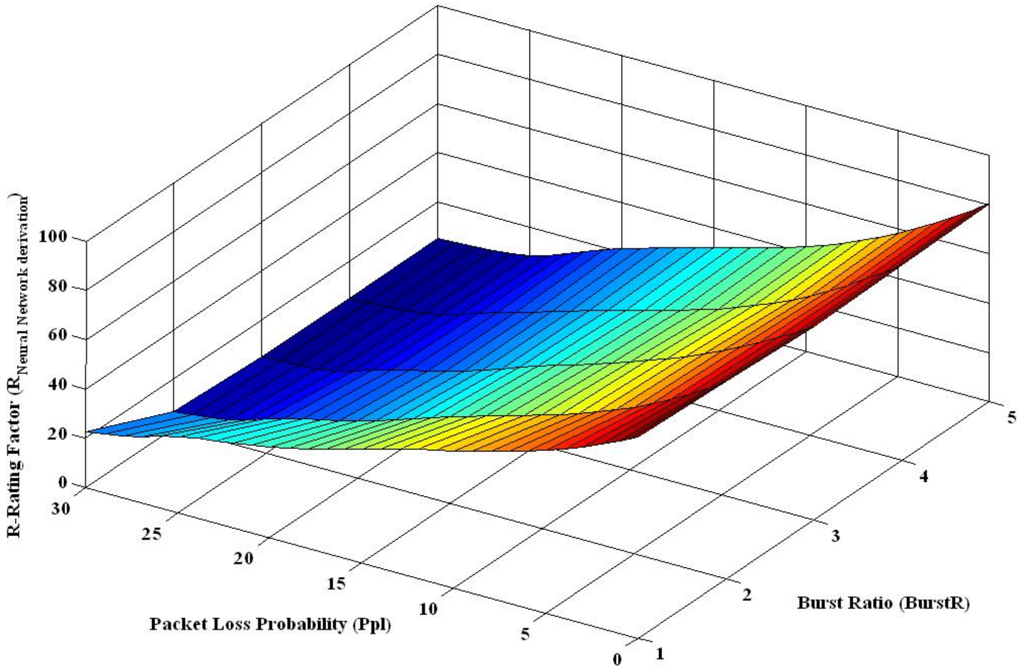

Figure 6.25: R-Rating Factor (Neural Network derived) vs. Packet Loss Probability and Burst Ratio

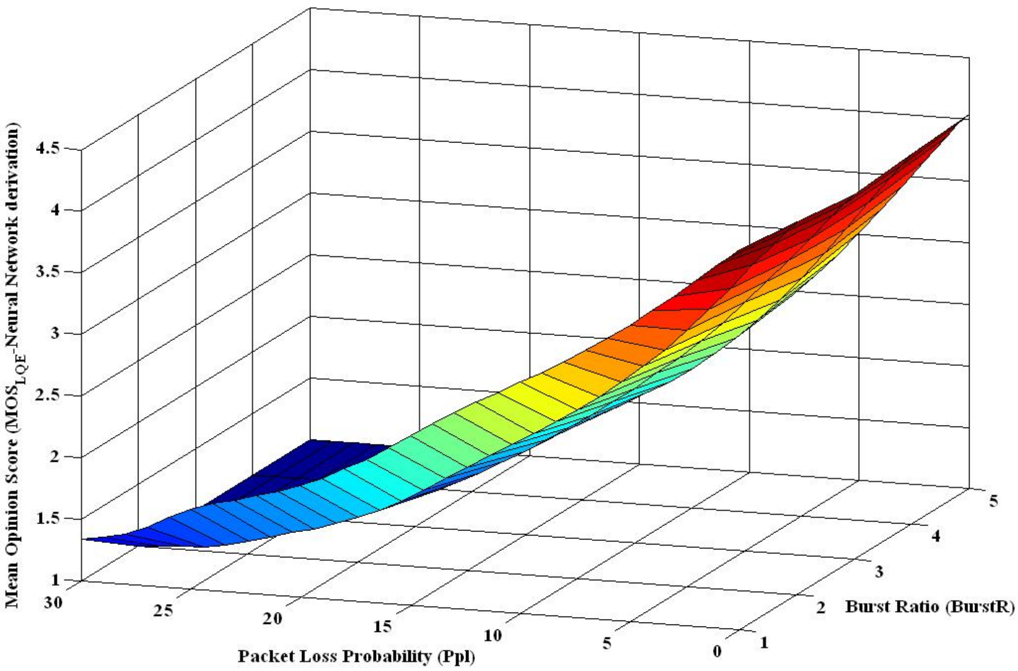

Figure 6.26: MOS_{LQE} (Neural Network derived) vs. Packet Loss Probability and Burst Ratio

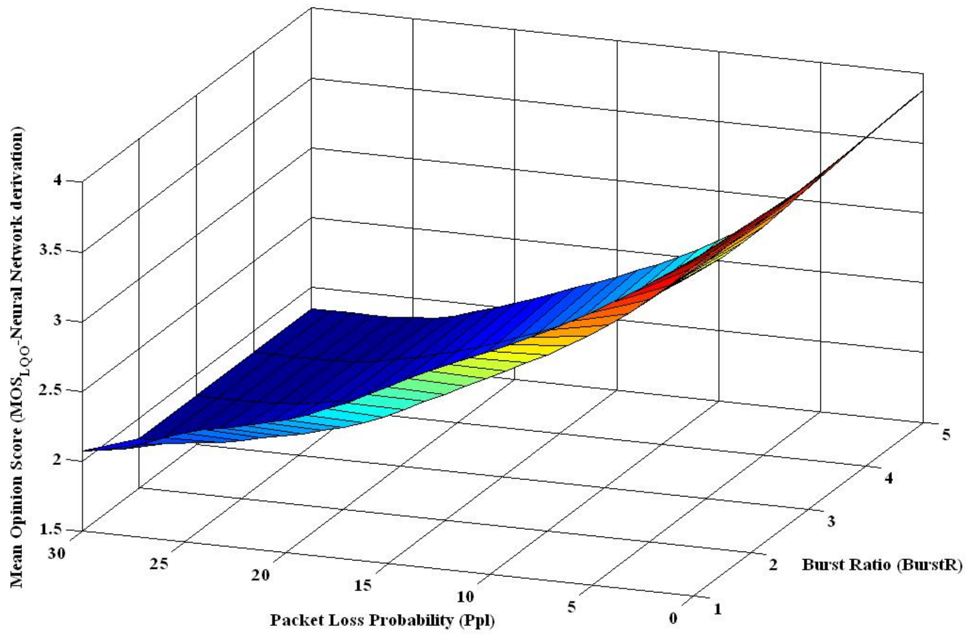

Figure 6.27: MOS_{LQO} (Neural Network derived) vs. Packet Loss Probability and Burst Ratio

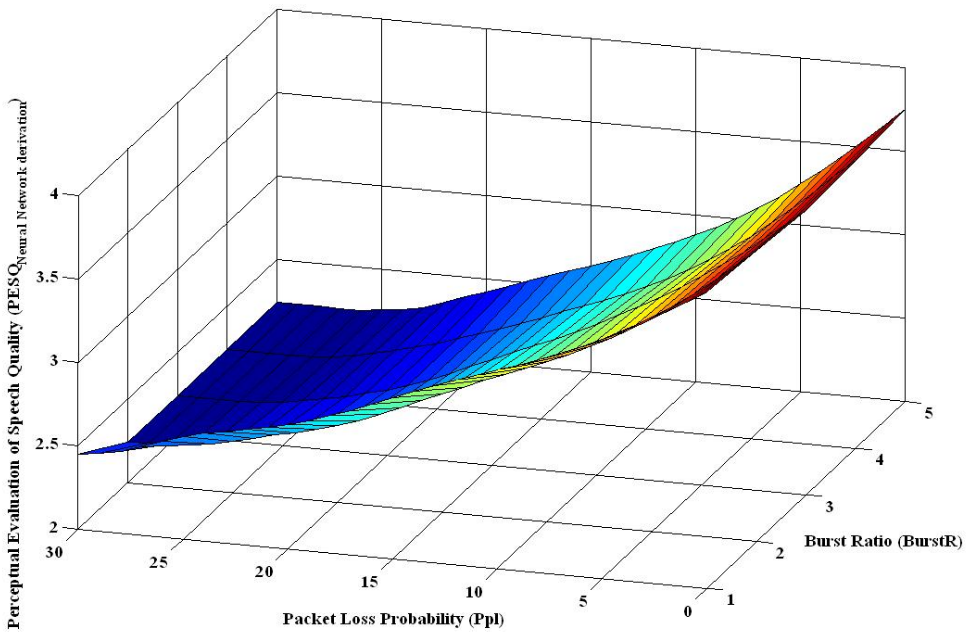

Figure 6.28: $PESQ$ (Neural Network derived) vs. Packet Loss Probability and Burst Ratio

Comparison Between Quality Prediction in Neural Network Model and E-model:

The derived MOS_{LQE} values in Figure 6.26 represent the model prediction (ANN prediction of the quality) values. These values should be compared against the E-model's predicted values to determine the successfulness of the ANN model in extending the E-model to new network conditions and new speech coders.

To study the effectiveness of the ANN model in modelling *Ie-eff* and ultimately predicting the speech quality in terms of MOS score, a comparison is made between ANN-predicted MOS_{LQE} (as shown in Figure 6.26 and listed in Tables B.21-B.25 in Appendix B) and E-model predicted MOS_{LQE} (as shown in Figure 4.2 and listed in Table 4.3 in chapter 4). The comparison is constrained to the MOS_{LQE} values corresponding to Ppl in the range 0 to 20 and $BurstR$ in the range 1 to 2 as defined by the E-model and listed in Table 3.1. Tables 6.6-6.7 lists the MOS_{LQE} values for both the E-model and the ANN in the above ranges.

Using equation (6.3) on the MOS_{LQE} values listed in Tables 6.6-6.7, the resultant correlation coefficient is found to be 0.9942 which indicates strong positive correlation between the ANN MOS_{LQE} values and the E-model MOS_{LQE} values. This is higher than the correlation found in case of the linear regression (0.9762) and higher than the correlation found in case of non linear regression (0.9882).

The average absolute difference is 0.0716 MOS which indicates an improvement over the estimation of the linear and non linear models while the maximum absolute difference is 0.1600 MOS_{LQE} and the standard deviation is 0.0486.

Visual comparison between the MOS_{LQE} values from the E-model with the MOS_{LQE} values from the ANN model (in the specified range) is shown in Figure 6.29.

It appears from the figure that the two relations for the MOS (E-model and ANN) both have very similar characteristics with very small differences which are hardly noticeable from the figure.

Figure 6.30 shows a scatter diagram between the E-model-based prediction and

Ppl	E-model MOS_{LQE}	ANN MOS_{LQE}
0	4.10	4.09
0.5	4.03	4.02
1	3.95	3.95
2	3.79	3.81
3	3.63	3.67
4	3.48	3.54
5	3.34	3.41
6	3.21	3.29
7	3.08	3.18
8	2.96	3.07
9	2.85	2.97
10	2.75	2.88
11	2.65	2.78
12	2.56	2.69
13	2.47	2.59
14	2.40	2.49
15	2.32	2.38
16	2.25	2.27
17	2.19	2.16
18	2.13	2.07
19	2.07	1.98
20	2.02	1.91

Table 6.6: E-model MOS_{LQE} and ANN MOS_{LQE} for different possible values of Ppl for speech coder G.729. $BurstR = 1$

Ppl	**E-model** MOS_{LQE}	**ANN** MOS_{LQE}
0	4.10	4.03
0.5	4.02	3.94
1	3.94	3.84
2	3.77	3.65
3	3.59	3.45
4	3.41	3.26
5	3.24	3.08
6	3.06	2.91
7	2.89	2.76
8	2.73	2.62
9	2.58	2.49
10	2.43	2.37
11	2.29	2.26
12	2.16	2.15
13	2.03	2.05
14	1.92	1.94
15	1.81	1.83
16	1.71	1.73
17	1.62	1.62
18	1.54	1.52
19	1.46	1.43
20	1.39	1.36

Table 6.7: E-model MOS_{LQE} and ANN MOS_{LQE} for different possible values of Ppl for speech coder G.729. $BurstR = 2$

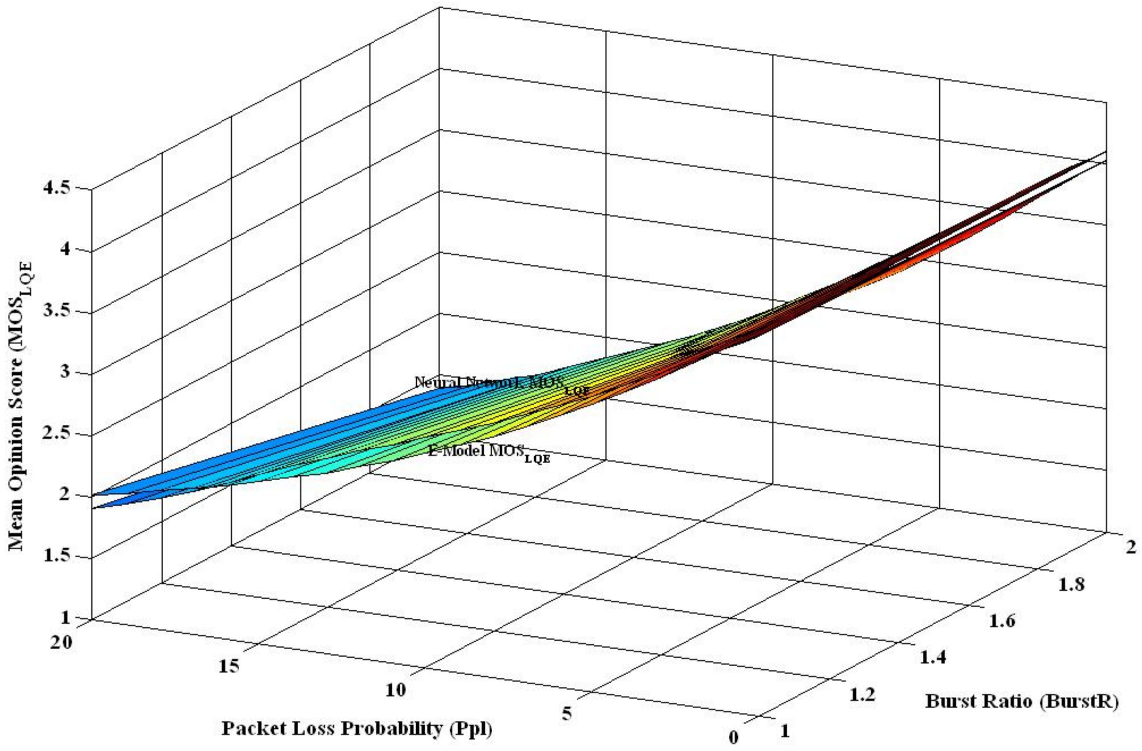

Figure 6.29: MOS_{LQE} (E-model and Neural Network) vs. Packet Loss Probability and Burst Ratio

the ANN prediction to visualise the correlation between the two predictions. Most of the points are concentrated near the perfect fit line due to the very high correlation.

Figure 6.31 shows the box plot of difference in quality prediction between the E-model and the ANN model. From the figure it appears that the values of error in prediction are clustered in the lower range i.e. toward zero. The first quartile (first 25% of the data) lies in the range 0 to 0.02 MOS, and the first two quartiles 50% are in the range 0 to 0.07 (median value) MOS which is less than the value for the first quartile in the linear and non linear models, the third quartile lies between 0.07 and 0.15 which is a noticeable reduction from the non linear model while the last quartile is 0.15 and 0.16 MOS which is again a noticeable reduction from the non linear model which is in its turn less than the linear model. This figure is plotted using the same scale used to plot the box plot for both the linear and non linear regression models to simplify comparison.

Figure 6.32 shows the Cumulative Distribution Function (CDF) of the difference

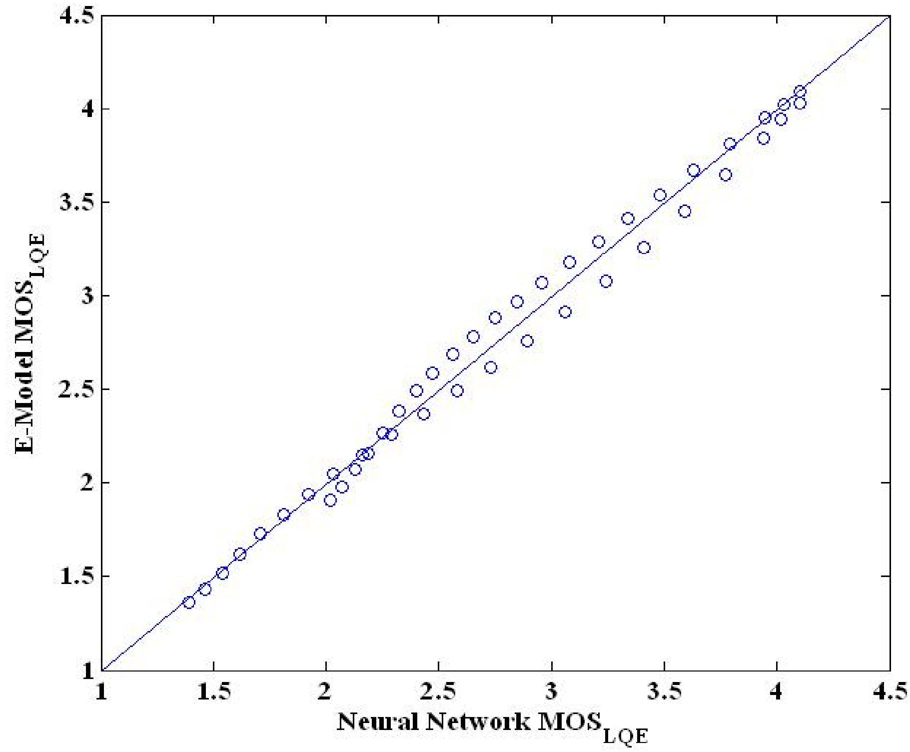

Figure 6.30: Scatter Diagram of quality prediction

in quality prediction between the E-model and the ANN model. The upper bound for the error was 0.16 MOS, with 38.64% below or equal 0.05 MOS, 68.18% below or equal 0.10 MOS, 97.73% below 0.15 MOS and 100.% below 0.16 MOS. These numbers indicate better approximation than both the linear and non linear regression models due to the existence of higher percentage of points below a certain threshold in comparison with lower percentage of values under the same error level.

Next section compares the performance of linear regression, non linear regression and ANN models in estimating the quality.

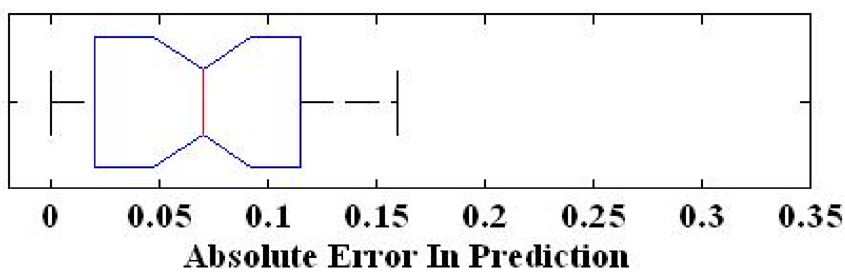

Figure 6.31: Box Plot of the error in Neural Network prediction

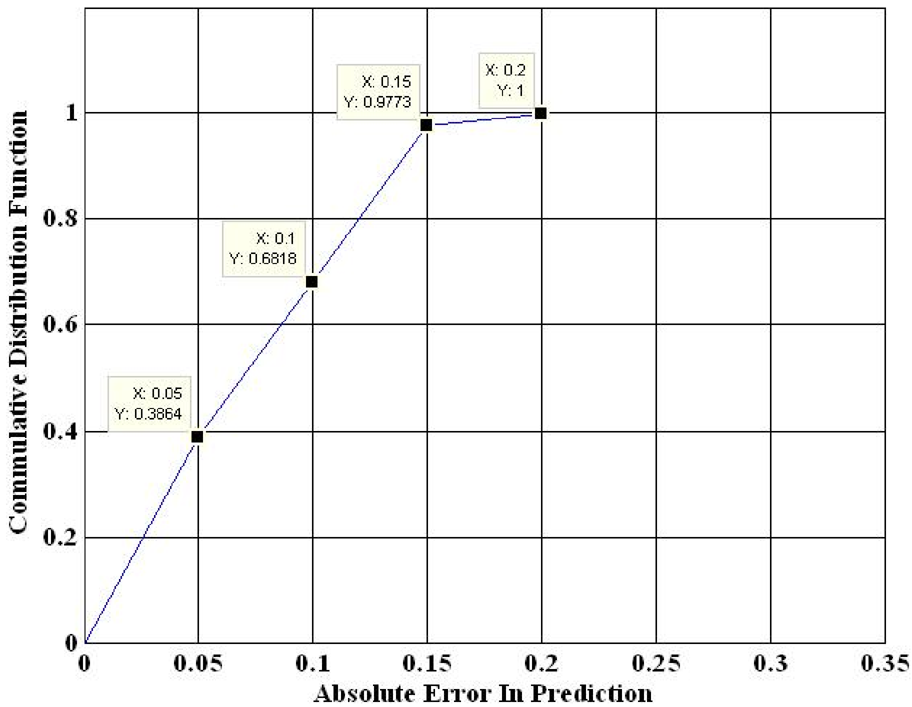

Figure 6.32: Cumulative Distributed Function (CDF) of quality prediction error

6.5 Evaluation of the Model

In the previous sections Linear regression models, non linear regression and ANN models are compared with the E-model in terms of their capabilities of predicting the speech quality. Table 6.8 summarises different comparison aspects between the 3 models. In the table each model is assigned a column and each comparison criteria is assigned a row.

From different aspects of comparisons it seems that ANN model offers the best approximation among the 3 models considered for this problem although the non linear regression is very close in many aspects. The most prominent criteria for the ANN model is in terms of maximum difference estimation where it differs significantly (almost half) of both linear and non linear regression models.

Although the ANN derivation does take more time and preparation (training and testing) than the other models, but the estimation accuracy justifies its use in predicting the quality. The recommendation therefore is to use an ANN model to predict quality degradation due to *Ie-eff*. Next section compares the performance of the ANN with previous attempts to extend the E-model utlising PESQ.

Comparison Criteria	**Model**		
	Linear Regression	**Non Linear Regression**	**ANN**
Correlation Factor (*Ie-eff*)	0.976	0.992	0.998
Variance Explained (*Ie-eff*)	95.3%	98.4%	99.6%
Correlation Factor (MOS_{LQE})	0.9762	0.9882	0.9942
Maximum Difference (MOS_{LQE})	0.3300	0.3000	0.1600
Average Difference (MOS_{LQE})	0.1705	0.1100	0.0716
Standard Difference (MOS_{LQE})	0.0968	0.9760	0.0486
1st Quartile Difference	0-0.09	0-0.03	0-0.02
2nd Quartile Difference	0.09-0.165	0.03-0.075	0.02-0.07
3rd Quartile Difference	0.165-0.25	0.075-0.2	0.07-0.15
% of less than 0.05 Difference	15.91%	40.91%	38.64%
% of less than 0.1 Difference	29.55%	61.36%	68.18%
% of less than 0.2 Difference	65.91%	75.00%	97.73%

Table 6.8: Comparison between Linear Regression, Non Linear Regression and Artificial Neural Network techniques

6.6 Comaprison with previous work

Previous efforts have been going on to extend the E-model based on the *PESQ* intrusive-based speech quality prediction methodology [30, 31, 165, 168, 167, 169].

As previous attempts to extend the the E-model was based on a previous version of the E-model, 2000 [78] prior to the latest version, 2005 [84], one of the major differences between the work presented in this chapter and the previous work is packet loss paramteres.

As the impact of packet loss in the older versions of the E-model (prior to the current version, 2005) was characterised by the Ie factor, specific impairment factor values for codec operating under random packet loss have been previously tabulated to be packet-loss dependent. As such packet loss was treated as random packet loss without taking the effect of burstiness into account and it was characterised by Equipment Impairment (Ie) factor instead of Effective Equipment Impairment Factor (*Ie-eff*) as the case of the current version.

The corresponding (Ie) values for different speech coders operating under specific percentages of packet loss are listed in ITU-T Recommendation G.113, 1999 [76]. For the used speech coder G.729 [71], these values are reproduced in Table 6.9. In contrast in the current version of the E-model, 2005 [84] *Bpl* is defined as codec-specific value and Ie is replaced by *Ie-eff*.

Ppl	**Equipment Impairment Ie**
0	11
0.5	13
1	15
2	19
3	23
4	26
8	36
16	49

Table 6.9: Values for equipment impairment factor Ie under conditions of random packet loss for speech coder G.729 [76]

Through Internet statistics several studies [14, 107] have shown that that packet loss can be bursty and appropriately bursty loss was introduced in the latest version of the E-model [84].

Based on the above and due to the different parameters used in different studies, any comparison between the results obtained in this thesis and the results obtained in previous studies should be read with care. This section compares the results obtained using ANN with previous efforts in terms of clossness to the E-model. Since different versions of the E-model are used in this study and other studies, the results of previous studies will be tested against the old E-model, 2000 [78] and our results will be compared against the new E-model, 2005 [84] to see how close each extension to the corresponding version of the E-model.

In the work presented by Ding and Goubran (DG) [30, 31] packet loss for the used speech coder was modelled by the following equation

$$Ie = Ie_{opt} + C1.ln(1 + C2.Ppl) \tag{6.8}$$

where

Ie_{opt}	Ie when packet loss is zero from ITU-T G.113
Ppl	Packet loss Probability
$C1, C2$	Curve fitting parameters

For the used speech coder G.729, $Ie_{opt} = 11$, $C1 = 25.21$ and $C2 = 0.150$ [30]. It should be noted that this work depends on the published Ie values for the speech coder, as such it may extend to new packet loss values but it does not extend to new speech coders as the work in this thesis is able to do.

Using equation (6.8) the values for Ie, R-Rating Factor and MOS are calculated and listed in Table 6.10. Using Ie values from the old E-model, 2000 [78], R-Rating Factor and MOS scores are calculated and listed in Table 6.11.

On the other hand, the ANN model derived in section 6.4 is used to calculate the corresponding values for *Ie-eff* (to replace Ie in the new E-model), R-Rating Factor, and MOS as listed in Table 6.12 for BurstR=1 and BurstR=2, while the corresponding values from the new E-model, 2005 [84] are listed in Table 6.13.

Ppl	Ie	R**-Rating Factor**	**MOS**
0	11.0000	82.2000	4.1000
0.5	12.8232	80.3768	4.0400
1.0	14.5234	78.6766	3.9700
2.0	17.6142	75.5858	3.8500
3.0	20.3671	72.8329	3.7300
4.0	22.8488	70.3512	3.6100
8.0	30.8770	62.3230	3.2200
16.0	41.8514	51.3486	2.6500

Table 6.10: Ie, R-Rating Factor and MOS derived according to Ding and Goubran study

Ppl	Ie	R**-Rating Factor**	**MOS**
0	11.0000	82.2000	4.1000
0.5	13.0000	80.2000	4.0300
1.0	15.0000	78.2000	3.9500
2.0	19.0000	74.2000	3.7900
3.0	23.0000	70.2000	3.6100
4.0	26.0000	67.2000	3.4600
8.0	36.0000	57.2000	2.9500
16.0	49.0000	44.2000	2.2700

Table 6.11: Ie, R-Rating Factor and MOS derived according to the E-model, 2000 [78]

It should be noticed that with the new E-model two sets of values are retrieved, corresponding to BurstR=1 and BurstR=2 and this is reflected in the tables.

BurstR=1				**BurstR=2**			
Ppl	Ie	**R**	**MOS**	**Ppl**	Ie	**R**	**MOS**
0	11.4580	81.7420	4.0900	0	12.9502	80.2498	4.0300
0.5	13.3260	79.8740	4.0200	0.5	15.3495	77.8505	3.9400
1.0	15.1259	78.0741	3.9500	1.0	17.6811	75.5189	3.8400
2.0	18.5207	74.6793	3.8100	2.0	22.1263	71.0737	3.6500
3.0	21.6452	71.5548	3.6700	3.0	26.2624	66.9376	3.4500
4.0	24.5088	68.6912	3.5400	4.0	30.0780	63.1220	3.2600
8.0	33.7440	59.4560	3.0700	8.0	42.3221	50.8779	2.6200
16.0	49.0968	44.1032	2.2700	16.0	60.4717	32.7283	1.7300

Table 6.12: Ie, R-Rating Factor and MOS derived according to ANN

BurstR=1				**BurstR=2**			
Ppl	Ie	**R**	**MOS**	**Ppl**	Ie	**R**	**MOS**
0	11.0000	82.2000	4.1000	0	11.0000	82.2000	4.1000
0.5	13.1540	80.1000	4.0300	0.5	13.1820	80.0000	4.0200
1.0	15.2000	78.0000	3.9500	1.0	15.3080	77.9000	3.9400
2.0	19.0000	74.2000	3.7900	2.0	19.4000	73.8000	3.7700
3.0	22.4550	70.8000	3.6300	3.0	23.2930	69.9000	3.5900
4.0	25.6090	67.6000	3.4800	4.0	27.0000	66.2000	3.4100
8.0	35.8890	57.3000	2.9600	8.0	40.2170	53.0000	2.7300
16.0	49.4000	43.8000	2.2500	16.0	60.7780	32.4000	1.7100

Table 6.13: Ie, R-Rating Factor and MOS derived according to the E-model, 2005 [84]

By calculating the correlation in predicting the E-model output for both DG study and this study in terms of MOS, it was found to be equal to 0.998 in both cases which indicates strong positive correlation with the E-model. This correlation was calculated by comparing MOS values calculated from DG study with MOS values from the old E-model, 2000 [78]. The correlation of our study is calculated by comparing MOS values from ANN with MOS values from the new E-model, 2005 [84].

Although the calculated correlation is high and indicates strong positive relation, but when it comes to the difference in prediction, the maximum difference between DG study and the old E-model is 0.3800 MOS with standard deviation of 0.1367 MOS. The maximum difference between our study and the new E-model is 0.1500 with 0.0506 standard deviation. It appears from these figures that the study presented in this thesis offers closer approximation to the new E-model than what DG study offers to the old E-model. This limitation is in favor of this thesis in addition to the restriction that their study is able to extend to new loss conditions without being able to extend to new speech coders as the case of our study.

Figure 6.33 presents scatter diagram between DG results and the old E-model results while Figure 6.34 presents scatter diagram between the study in this chapter and the results obtained from the new E-model. It appears from the figures that DG results are more far from the perfect fit line than the ANN results.

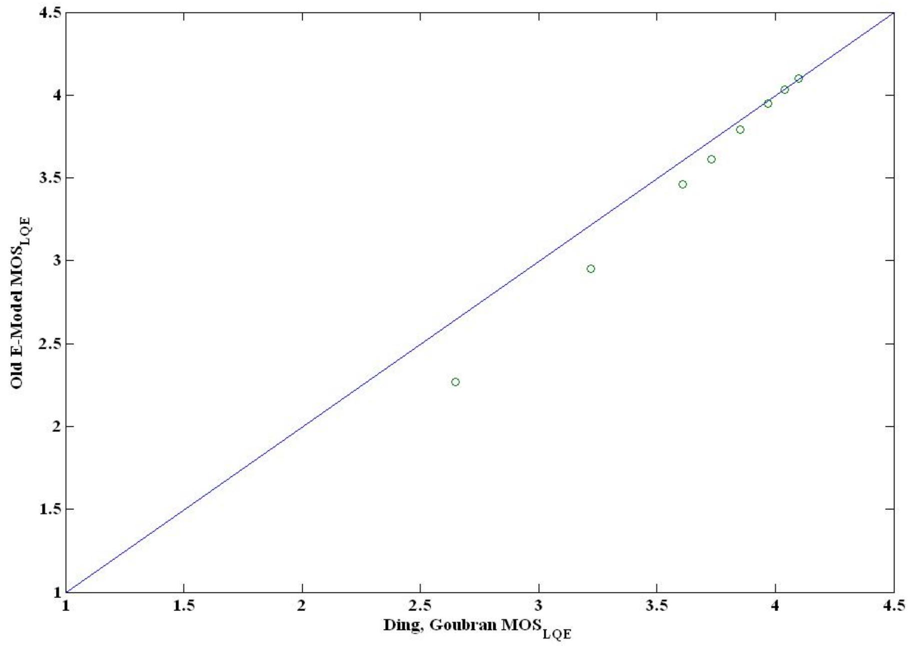

Figure 6.33: Scatter Diagram of quality prediction using Ding and Goubran Study

Box plot of the error in prediction between DG results and the old E-model is shown in Figure 6.35 and similar box plot between the error between the ANN pre-

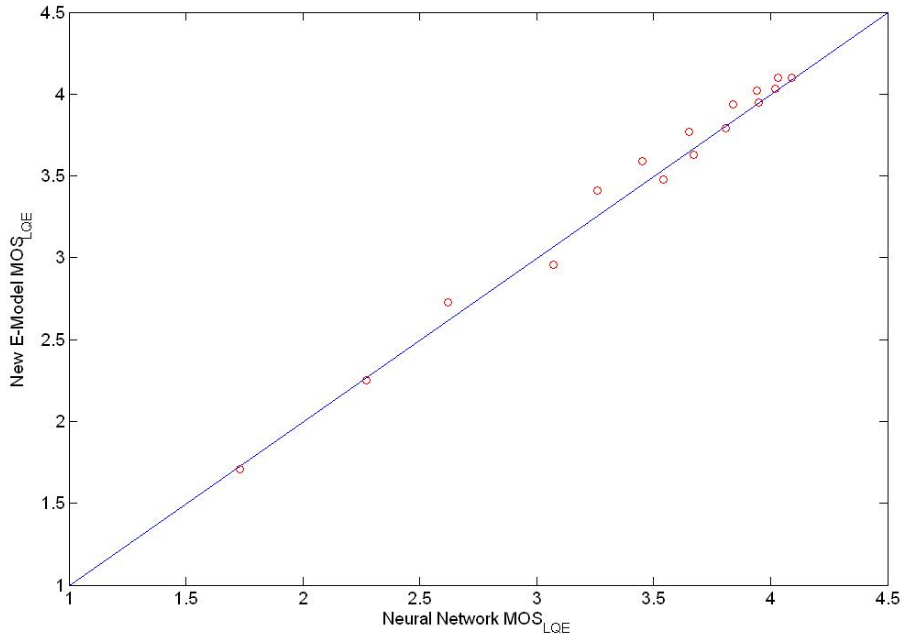

Figure 6.34: Scatter Diagram of quality prediction using ANN Prediction

diction and the new E-model prediction is shown in Figure 6.36.

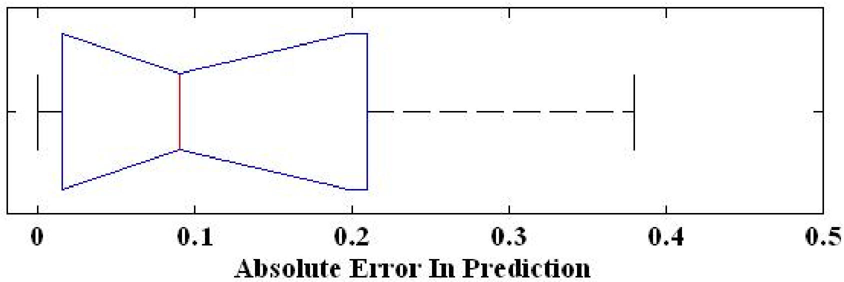

Figure 6.35: Box Plot of the error in Ding and Goubran Study

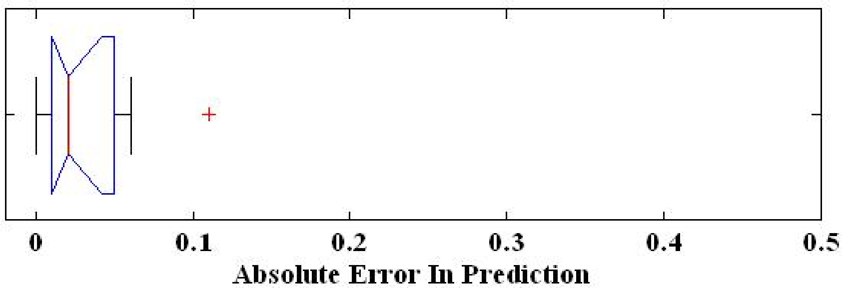

Figure 6.36: Box Plot of the error in ANN prediction

It can be noticed easily from the two figures that the error in prediction is concentrated in lower ranges with the ANN solution in comparison with the DG solution which indicates more accurate prediction of the quality in case of the ANN solution.

Another way of showing the difference is illustrated in Figure 6.37 where both DG prediction of the quality and the old E-model prediction are illustrated. Figure 6.38 shows the the ANN and the new E-model prediction of the quality.

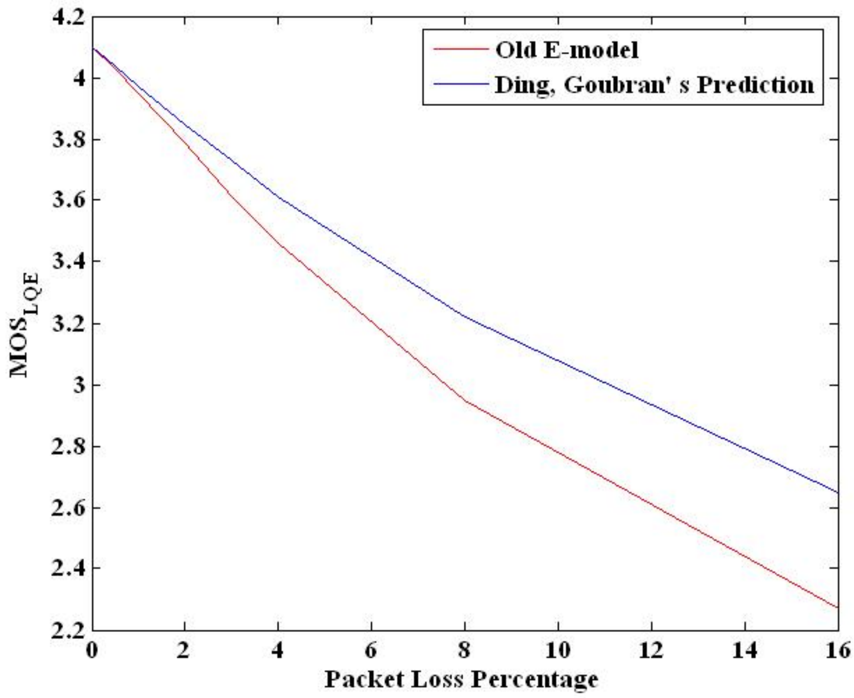

Figure 6.37: Difference between Ding and Goubran Prediction and the old E-model

It is noticed from the figures that the error in prediction using the ANN model in Figure 6.38 is much less than the error in prediction in Figure 6.37 using DG study. This can be seen from the very close values between the two graphs in Figure as compared with the previous case in Figure 6.37 where there is some distance between the two graphs indicating the gap between DG prediction in comparison with the old E-model.

From the above discussion, it can be concluded that the ANN estimation of the quality is much closer to the new E-model than the previous work done by DG to the old E-model.

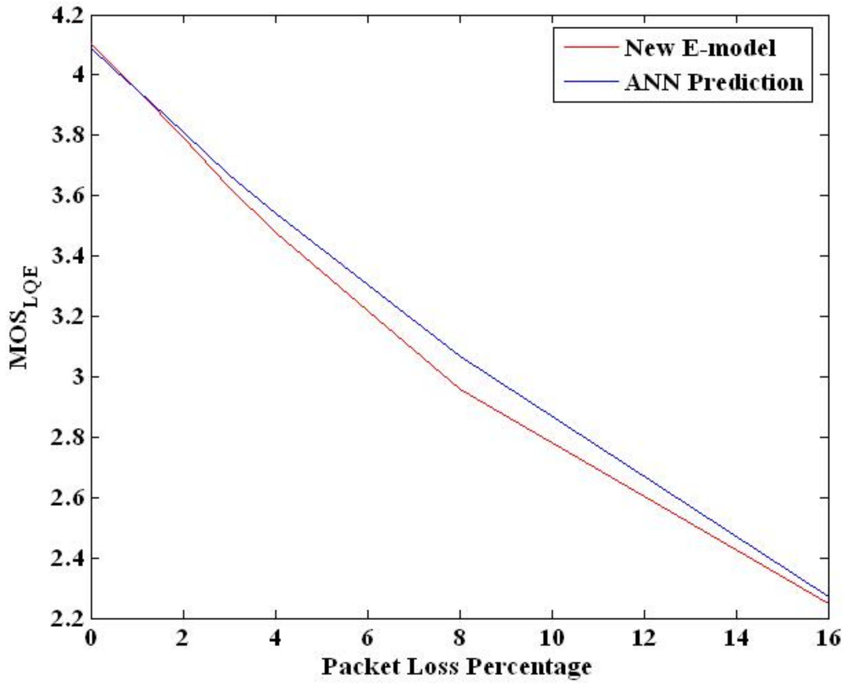

Figure 6.38: Difference between ANN Prediction and the new E-model

In addition to the difference in the estimation accuracy, the work done by DG does not deal with the problem of extension to new speech coders and is able to extend to new packet loss conditions. This is in addition to the fact that their extension deals with an older version of the E-model, without propoer handling of burstiness in packet loss as the case of the linear regression, non-linear regression and ANN extensions examined in this thesis.

Similar to the work of Ding and Goubran, Sun and Ifeachor (SI) in a series of publications attempted to extend the E-model using PESQ [165, 168, 167, 169]. The work of SI avoided the need for the optimal Ie value (Ie_{opt}) as it was needed in Ding and Goubran's study. However, they still used the old E-model, 2000 [78] which does not consider the effect of burstiness on the speech quality.

The effect of Ie was modelled by the following equation:

$$Ie = a.ln(1 + b.Ppl) + c \tag{6.9}$$

where

Ppl	Packet loss Probability
a, b, c	Curve fitting parameters

For the used speech coder G.729, SI reported the values of a, b and c to be 21.14, 0.1273 and 22.45, respectively [165, 168, 167, 169].

As the value of Ie_{opt} for the used encoder is not used in the derived equation, the work of SI becomes applicable to new speech coders as well as to new networking conditions. However, still their reliance on an older version of the E-model, 2000 [78], hinders its use, as the old E-model does not handle burstiness in packet loss which makes their results unrealistic in real network as several studies have shown burstiness in Internet packet loss statistics [14, 107].

Utilising equation (6.9) to derive the values for Ie, R-Rating Factor and MOS as listed in Table 6.14 while the corresponding Ie values from the old E-model, 2000 [78] are listed earlier in Table 6.11.

Ppl	Ie	R**-Rating Factor**	**MOS**
0.0	22.4500	70.7500	3.6300
0.5	23.7545	69.4455	3.5700
1.0	24.9831	68.2169	3.5100
2.0	27.2449	65.9551	3.4000
3.0	29.2879	63.9121	3.3000
4.0	31.1508	62.0492	3.2100
8.0	37.2967	55.9033	2.8900
16.0	45.9324	47.2676	2.4300

Table 6.14: Ie, R-Rating Factor and MOS derived according to Sun and Ifeachor study

The correlation in predicting the E-model's output between the SI study and the old E-model in terms of MOS was found to be equivelant to 0.999 which indicates very strong positive correlation with the E-model. This correlation was calculated by comparing MOS values calculated from SI study with MOS values from the old E-model. The correlation between ANN model and the new E-model was 0.998 which is still strong correlation.

Although the calculated correlation in SI study is higher than the correlation of the ANN with the new E-model. By comparing the difference in prediction, the maximum difference between SI study and the old E-model is 0.4700 MOS with standard deviation of 0.15 MOS while the maximum difference between our study and the new E-model is 0.1500 with 0.0506 standard deviation (Tables 6.12 and 6.13). The big difference in quality estimation between the ANN model and SI study shows the benefit of the the study presented in this thesis by offering an extension with much closer approximation to the new E-model than what SI study offers to the old E-model. This deficits comes due to the reason that SI in their series of studies did not compare their results after deriving the formula with the original E-model to check its validity. This is in addition to the fact that they used old E-model which does not handle burstiness in packet loss.

Figure 6.39 presents scatter diagram between SI results and the old E-model results while the scatter diagram between the study in this chapter and the results obtained using the new E-model was presented earlier in Figure 6.34. By comparing the two figures it appears how the ANN prediction is much closer and concentrated surrounding the perfect fit line than the case of SI scatter diagram.

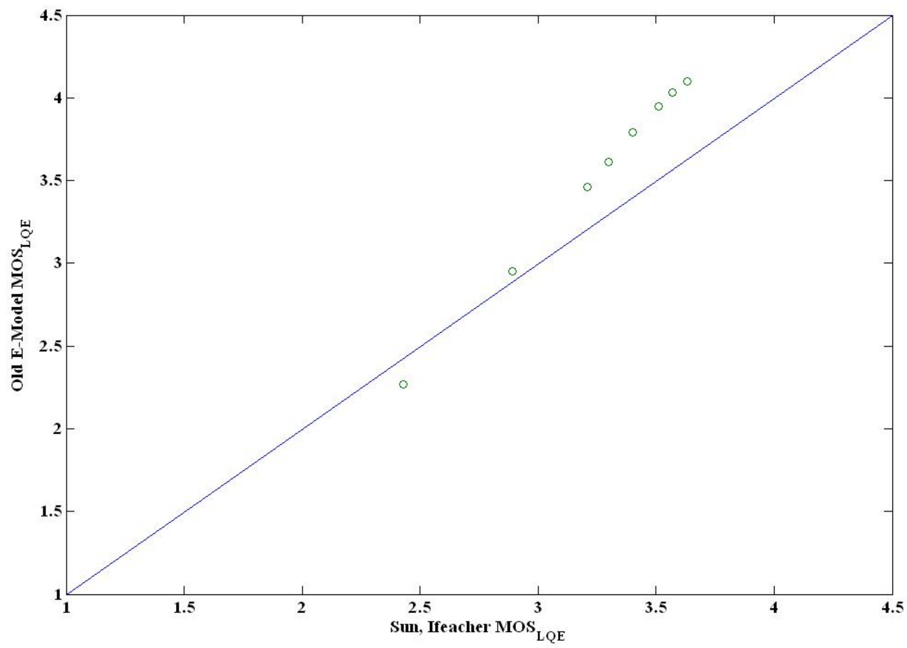

Figure 6.39: Scatter Diagram of quality prediction using Sun and Ifeachor Study

Box plot of the error in prediction between SI results and the old E-model is shown in Figure 6.40 and similar box plot between the error between the ANN prediction and the new E-model prediction was shown earlier in Figure 6.36.

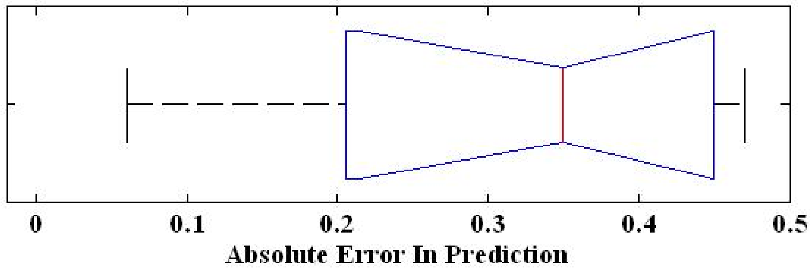

Figure 6.40: Box Plot of the error in Sun and Ifeachor Study

By quick comparison, the difference in the estimation accuracy between the two models seems clear as the ANN model offers far more accurate estimation of the quality with much less difference from the new E-model than what SI study offers in comparison with the old E-model.

The difference is also illustrated in Figure 6.41 where SI prediction of the quality and the old E-model prediction are both illustrated. The big difference in quality estimation can be noticed almost everywhere although there was an exact match when the packet loss was 10 but the difference in quality estimation is big elsewhere.

Figure 6.38 shown earlier illsutrates how the ANN and the new E-model prediction of the quality are close to each other. It is noticed from the figures that the error in prediction is much less in this case than the error in Figure 6.41.

From the above comparisons, it is clear that although SI extension to the E-model based on PESQ was able to avoid the subjectivity in estimating the quality but it suffers from several defícts. Most notably its estimation accuracy is far from being accurate in comparison with the old E-model it was supposed to extend. Also it does not take burstiness in packet loss as a factor which hinders its applicability in relistic environment.

In contrast the proposed model in this chapter such as the most accurate ANN model was able to extend the new E-model to avoid the subjectivity which makes its applicable to new network conditions and to new speech coders. At the same time its extension accuracy was better than previous efforts done by Ding and Goubran

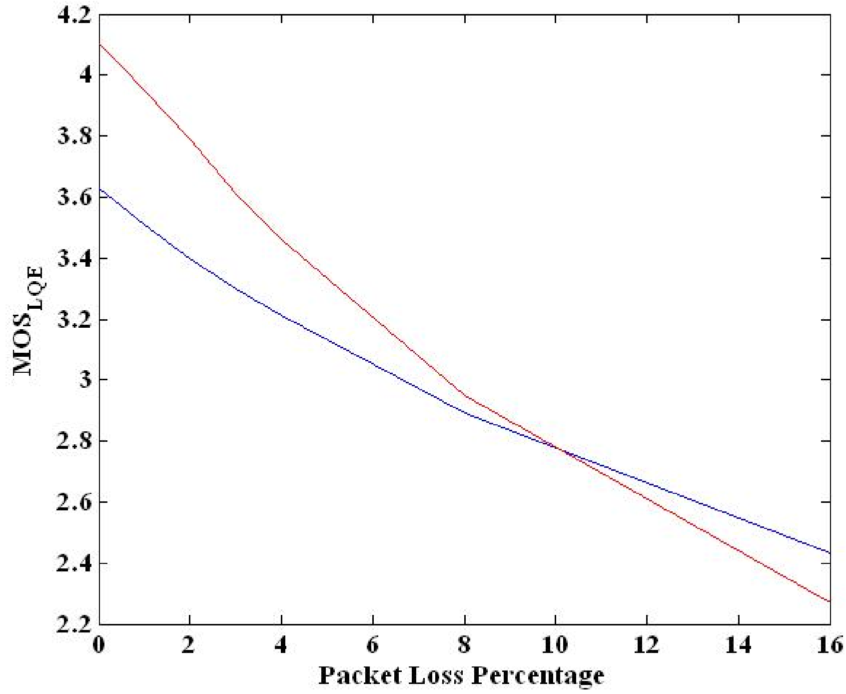

Figure 6.41: Difference between Sun and Ifeachor Prediction and the old E-model

[30, 31] and Sun and Ifeachor [165, 168, 167, 169].

This section compares the dervied ANN model with previous effors aimed at extending the E-model. Next section describes how can the ANN model be used in conjunction with the E-model to predict the conversational speech quality non-intrusively.

6.7 Use of the New Model in Combination with the E-model

As the ANN model provides the best prediction accuracy, it usage is assumed in this section for predicting the quality although the discussion in this section can be applied equally to both linear regression and non linear regression models.

The derived ANN model can be integrated with the E-model in monitoring live traffic non-intrusively and predicting conversational speech quality as depicted in Figure 6.42.

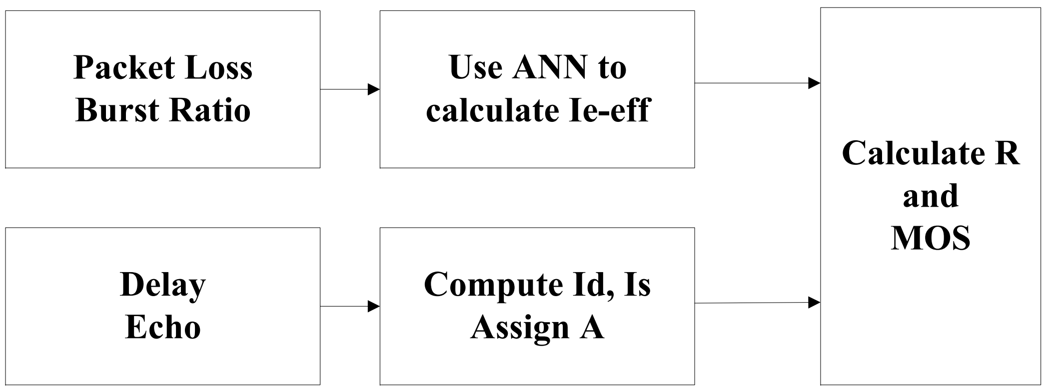

Figure 6.42: Conceptual diagram of E-model integration with the ANN to monitor live traffic

IP packets can be captured at some appropriate point in the IP network (possibly at an ingress gateway) and the information about packet loss, packet loss distribution, delay, and speech coder are extracted from Real-time Transport Protocol (RTP) header as explained in section 2.2.3.

By identifying packet loss characteristics of the received stream (Ppl and $BurstR$), then the ANN model is used and the information about packet loss are fed into that model to calculate the corresponding *Ie-eff* value. Similarly the information about the delay and other parameters in the E-model are used to calculate Id, and Is to use them in calculating the R-Rating Factor in equation (5.1). The value of the advantage factor is also added according to the characteristics of the system in-hand (wired, wireless, etc).

The *Ie-eff* value computed from the ANN model and the results from the computation of Id, Is are combined with A to calculate an overall R-Rating Factor. The R-Rating Factor can then be mapped into an MOS score to give an estimation for the overall conversational quality.

6.8 Summary

In this chapter three methods for predicting *Ie-eff* from *Ppl* and *BurstR* are derived, their accuracy in comparison with the E-model is evaluated. It has been found that the ANN method is more accurate than both linear regression and non linear regression models in estimating *Ie-eff* and then ultimately estimating the quality.

The proposed model avoids the hard to conduct, time-consuming, expensive, and lack of repeatability subjective tests required to estimate the E-model parameters and as such makes the E-model readily extensible to new network conditions and to new speech coders. Using the proposed technique, if a new speech coder emerges, it is readily applicable to the E-model as soon as the required objective tests are performed to derive a relation between *Ie-eff* and *Ppl* and *BurstR*. Using this relation R-Rating Factor and MOS values can be computed.

The proposed model has wide applicability in estimating the speech quality for voice applications over IP networks which makes it significantly important for this widely interesting, growing application in the continuously changing world of communication.

Chapter 7

Improving the E-model Based on Packet Loss Classification

7.1 Introduction

As discussed in the previous chapters, the E-model does not require the injection of the reference signal to estimate the speech quality due to its non-intrusive nature, consequently the E-model is suitable for monitoring live traffic. However, the content of the signal is not taken into consideration in the estimation of the quality and packet loss is taken as a whole without considering whether that loss happened in voiced or unvoiced parts of the signal.

On the other hand Perceptual Evaluation of Speech Quality (PESQ) as a full-reference, intrusive method for measuring the quality, provides a more accurate measurement for speech quality as it requires the original or reference speech signal as input and produces measurement of the listening MOS by comparing the post-transmitted signal with the original one. However, such method is inapplicable in monitoring live traffic because it is difficult or impossible to obtain actual speech samples as the reference signal is not available at the receiver.

In the previous two chapters the accuracy of the E-model was not questioned as the aim was to extend the E-model to avoid the need for the expensive, time-consuming subjective tests. As a very interesting finding, the results obtained using PESQ do not match the results obtained using the E-model. This difference is reproduced here again in Figure 7.1 for easy reference.

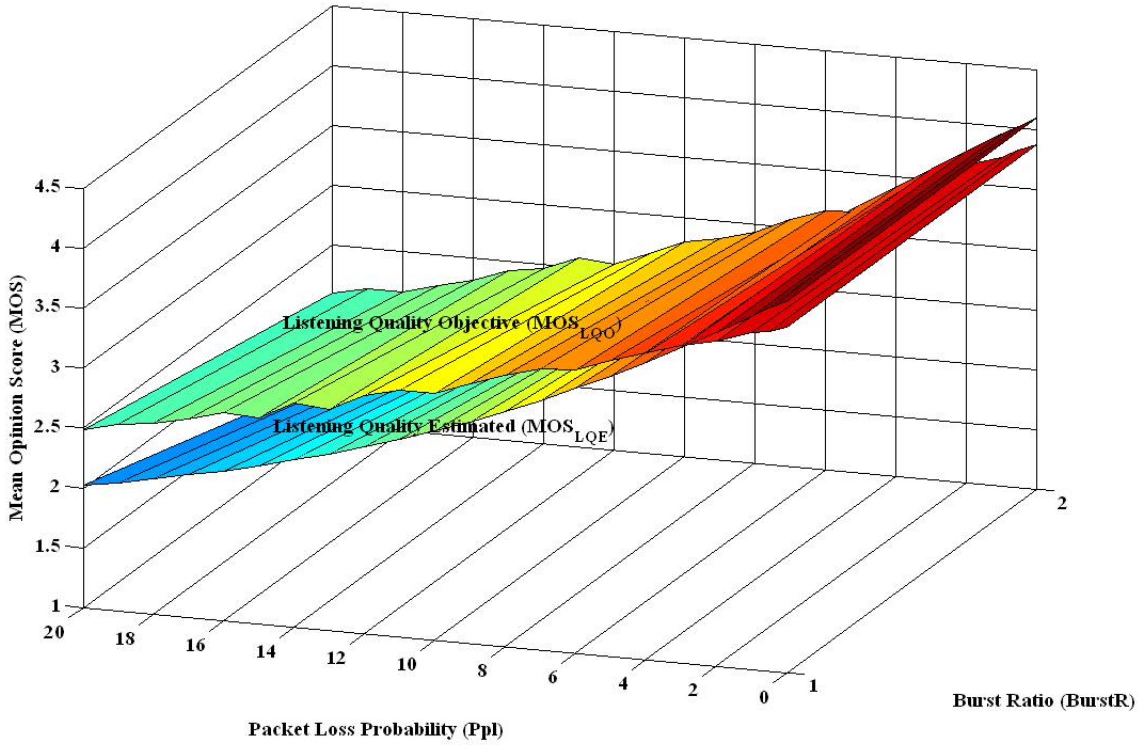

Figure 7.1: MOS_{LQO} and MOS_{LQE} vs. Packet Loss Probability and Burst Ratio

To alleviate the deviation in quality estimation and to allow PESQ to be used in extending the E-model, a correction formula (Equation (5.10)) was proposed. As a result of the previous two chapters and based on the usage of the correction formula, it was possible to extend the E-model to new network conditions based on the results obtained using PESQ.

Based on the finding that the results of the two models do not ideally match and some deviation is found between the two models, it was found useful and necessary to study this deviation and try to find a solution to reduce its effect.

One such possibility is to modify the E-model so that it can consider the content of the lost frames and whether they represent voiced parts of the signal or unvoiced parts. This would improve the E-model and could bring its estimation closer to the PESQ measurement of the quality as PESQ is more accurate due to its intrusive nature as previous studies [165, 170] have shown that packet loss during voiced parts of the signal has more perceptual effect on the quality than packet loss during unvoiced parts of the signal.

If packet loss can be broken into 2 classes one of them to represent packet loss during voiced parts of the signal and the other one to represent unvoiced parts of the signal, the accuracy of the E-model could be improved by bringing its estimation closer to the measurement value obtained using PESQ. This possibility is the subject of this chapter.

Section 7.2 explains how speech frames can be classified as voiced or unvoiced through what is called Voice Activity Detection (VAD). Section 7.3 describes rules used to determine if a missing packet lies in a Voiced or Unvoiced region based on the surrounding received packets. Section 7.4 illustrates how a received stream can be used to calculate packet loss characteristics in case of packet loss is not classified as Voiced or Unvoiced or in case it is classified. Section 7.5 describes how the accuracy of the E-model can be improved by taking packet contents into consideration. In section 7.6 an empirical study performed to support the finding in this chapter is described. When packet loss is classified into Voiced and Unvoiced, two weighting factors are introduced to represent the effect of voiced and unvoiced loss. Since finding the best values for these two factors is an optimisation problem, section 7.7 describes a Genetic Algorithm's approach to derive these model parameters. Section 7.8 evaluates the proposed method and section 7.9 summarises the chapter.

7.2 Voice Activity Detection

Several algorithms have been proposed for Voice Activity Detection (VAD) for the purpose of saving bandwidth during silence periods through discontinuous transmission. The simpler ones take account of the zero-crossings and the amplitude of the signal to detect the lower frequencies and higher amplitudes of speech segments when compared to noisy frames. Other algorithms are based on spectral characteristics of the signal such as the amount of energy contained at a certain frame. This chapter will use VAD algorithms for a slightly different purpose, to classify speech frames and aid in improving quality estimation accuracy.

ITU-T Recommendation G.729 way of classifying frames into Voiced or Unvoiced (V/UV) frames will be used which is based on the 10^{th} order LSF coefficients, zero crossing rate, and energy of the frame [71, 104, 161].

Figure 7.2 shows a sample speech signal from ITU-T Recommendation P.50, Appendix I [73] and labelling for the signal with Voiced and Unvoiced.

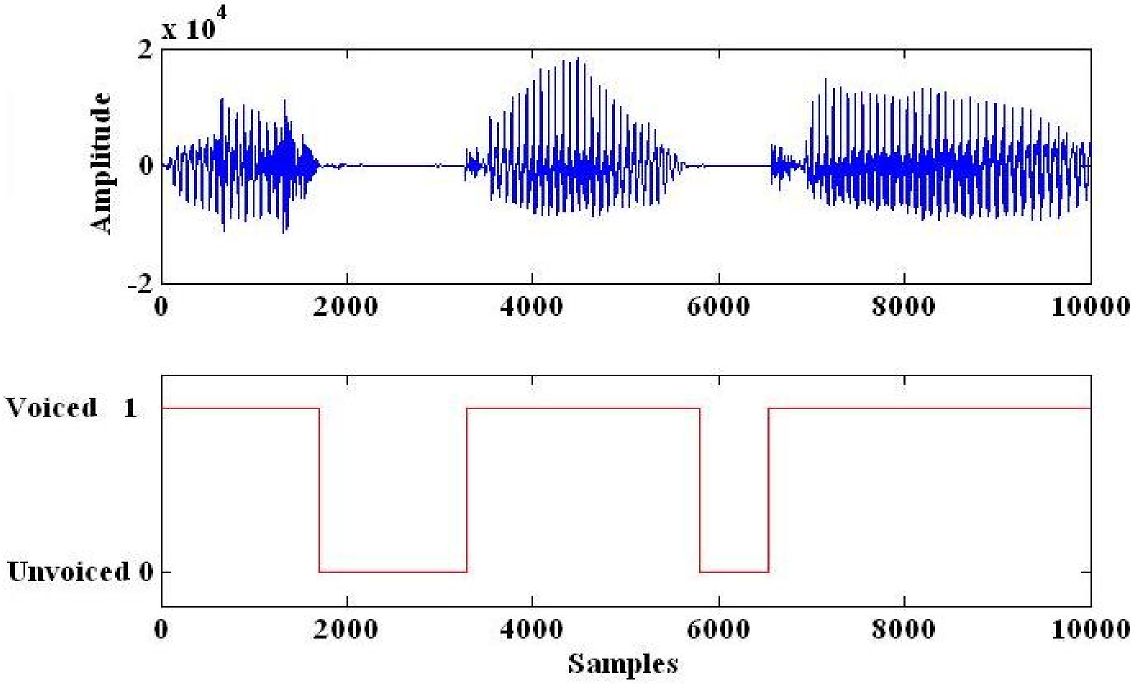

Figure 7.2: Speech Signal and Voiced/Unvoiced Label

The above classification uses the signal in hand to classify each frame into Voiced/Unvoiced. During classification the contents of the frame are used to test its energy and number of zero crossings. But as the purpose is to test the effect of a missing frame on the speech quality, it is necessary to determine whether a missing frame rather than an existing frame is Voiced or Unvoiced, this can be determined from the surrounding received packets as explained in the next section.

7.3 Voiced/Unvoiced Classification for Missing Packets

Based on the fact that the shape of the vocal tract and its mode of excitation changes relatively slowly, therefore speech signal can be considered to be quasi-stationary over a short period of time which allow it to show high degree of predictability. This fact will be exploited to allow predicting the state (Voiced/Unvoiced) of missing speech frames based on the surrounding packets.

Few rules are formulated to decide whether a missing packet is Voiced or Unvoiced based on two factors:

- The state of the previous and/or following packets.
- Whether packet loss is in isolation or in bursts.

This is explained in Table 7.1 where

X	Indicates packet loss position
*	Indicates packet loss position for 2 or more frames
V	Indicates received Voiced frame
U	Indicates received Unvoiced frame
$\underline{\text{V}}$	Indicates lost frame classified as Voiced
$\underline{\text{U}}$	Indicates lost frame classified as Unvoiced

The received speech stream is processed to determine how different loss segments match a pattern from the list in the table.

An example of a stream processed according to the rules in Table 7.1 is illustrated in Figure 7.3. Where the first line represent the received stream where packet loss is indicated by the zeros. The second line represents labelling of the received speech frames into Voiced or Unvoiced based on ITU-T Recommendation G.729 and the missing frames are indicated by X. The last line illustrates how different lost packets are labelled with either Voiced or Unvoiced based on the surrounding packets and according to the rules in Table 7.1.

Received Stream	0	1	1	0	1	1	0	0	1	1	0	0	1	1	1	0	1
Labelling for Received Packets	X	V	V	X	U	U	X	X	V	V	X	X	V	V	U	X	U
Labelling for missing packets	$\underline{\text{V}}$	V	V	$\underline{\text{V}}$	U	U	$\underline{\text{U}}$	X	V	V	$\underline{\text{V}}$	$\underline{\text{V}}$	V	V	U	$\underline{\text{U}}$	U

Figure 7.3: Stream processing for Voiced/ Unvoiced Processing

This is followed by, and based on the classification of lost frames, breaking the overall packet loss into voiced packet loss and unvoiced packet loss and also some

Loss Pattern	**Loss pattern After V/UV Decision**	**Explanation**
XU	UU	Classified as Unvoiced as the next frame
XV	VV	Classified as Voiced as the next frame
UX	UU	Classified as Unvoiced as the previous frame
VX	VV	Classified as Voiced as the previous frame
UXU	UUU	Classified as Unvoiced as the surrounding frames
VXV	VVV	Classified as Voiced as the surrounding frames
UXV	UUV	Classified as Unvoiced as the previous frame
VXU	VVU	Classified as Voiced as the previous frame
UXXU	UUUU	Classified as Unvoiced as the surrounding frames
VXXV	VVVV	Classified as Voiced as the surrounding frames
UXXV	UUXV	Classify the first frame as Unvoiced as the previous frame. Do Not classify the other frame
VXXU	VVXU	Classify the first frame as Voiced as the previous frame. Do Not classify the other frame
VX*V	VV*V	Classify the first frame as Voiced as the previous frame. Do Not classify the other frames
UX*U	UU*U	Classify the first frame as Unvoiced as the previous frame. Do Not classify the other frames
VX*U	VV*U	Classify the first frame as Voiced as the previous frame. Do Not classify the other frames
UX*V	UU*V	Classify the first frame as Unvoiced as the previous frame. Do Not classify the other frames

Table 7.1: Voiced/Unvoiced Classification

packet loss is unclassified neither as Voiced nor as Unvoiced. Then different packet loss categories are then combined together to have an overall estimation of the packet loss effect on the quality. This estimation will hopefully bring the E-model's estimation of the speech quality closer to the measurement provided by PESQ.

By experimenting with speech coded using G.729 speech coder [71] and using speech packet loss simulation according to 2-state Gilbert model as described in chapter 4, Packet loss Probability (Ppl) in the range 0 to 20 and Burst Ratio ($BurstR$) in the range 1 to 2 will be used. For each combination of Ppl and $BurstR$, 30 iterations will be performed.

With the above settings, the accuracy of the above classification rules in estimating the status of a missing packet into either Voiced or Unvoiced is tested. It is found that using the above rules, the status is accurately predicted in about 87.35% of the cases which is a good indication that the classification rules can be used for the derivations in the subsequent sections.

7.4 An Illustrative Example of how packet loss characteristics can be calculated for a received stream

This section illustrates how packet loss pattern in a received stream can be used to calculate Voiced, Unvoiced and Unclassified packet loss characteristics of a stream. Section 7.4.1 describes how packet loss characteristics can be calculated for a received stream when no classification is applied. Section 7.4.2 describes how these characteristics can be calculated for a received stream when classification of the missing packets is applied.

7.4.1 Packet Loss for a stream without packet classification

In case no packet classification is applied to the received stream, the calculation is limited to the calculation of the overall Ppl and $BurstR$ for the loss locations in the received stream. These calculation can be described as follows:

Calculation of Ppl

Ppl can be calculated as a ratio of how many packet has been lost divided by the total number of packets. i.e.

$$Ppl = \frac{\text{Number of Lost Packets}}{\text{Total Number of Packets}} \tag{7.1}$$

Calculation of $BurstR$

$BurstR$ is used to count for burstiness in the received stream. As explained earlier, $BurstR$ can be calculated as a ratio between average length of observed bursts in an arrival sequence over average length of bursts expected for the network under "random" loss. i.e.

$$BurstR = \frac{\text{Average length of observed bursts in an arrival sequence}}{\text{Average length of bursts expected for the network under "random" loss}} \tag{7.2}$$

Where average length of burst in an arrival sequence can be calculated by counting the number of lost packets and divide them how they are distributed

$$BurstR_{ArrivalSequence} = \frac{\text{Number of Lost packets}}{\text{Number of loss fragments}} \tag{7.3}$$

As an example, consider the example shown in Figure 7.3 where there are 7 lost packets distributed over 5 bursts which result in observed Burst length of 1.4 in this stream.

Whereas average length of burst expected for network under random packet loss can be calculated using the following equation [125]:

$$BurstR_{Random} = \frac{1}{1 - Ppl} \tag{7.4}$$

This equations gives the expected length of packet loss under random packet loss conditions. After calculating $BurstR_{ArrivalSequence}$ and $BurstR_{Random}$, $BurstR$ is calculated by dividing $BurstR_{ArrivalSequence}$ over $BurstR_{Random}$.

7.4.2 Packet Loss for a stream with packet classification

In case packet classification is applied to the received stream, packet classification is required prior to any calculations. Packet classification is performed based on the rules provided in Table 7.1.

After performing the classification, packet loss in a stream is divided into one of 3 classes: Voiced, Unvoiced, or Unclassified, after that Ppl and $BurstR$ is calculated for each of these 3 classes.

Calculation of Ppl

Ppl is calculated as a ratio of how many frames has been lost divided by the total number of frames. However, when packet classification is applied first, it yields 3 Ppl values corresponding to Voiced, Unvoiced, and Unclassified losses, these values are called Ppl_{Voiced}, $Ppl_{Unoiced}$, and $Ppl_{Unclassified}$ respectively.

$$Ppl_{Voiced} = \frac{\text{Number of Voiced-Classified Lost Frames}}{\text{Total Number of Frames}} \tag{7.5}$$

$$Ppl_{Unvoiced} = \frac{\text{Number of Unvoiced-Classified Lost Frames}}{\text{Total Number of Frames}} \tag{7.6}$$

$$Ppl_{Unclassified} = \frac{\text{Number of Unclassified Lost Frames}}{\text{Total Number of Frames}} \tag{7.7}$$

Calculation of $BurstR$

As before $BurstR$ is calculated as a ratio between average length of observed bursts in an arrival sequence over average length of bursts expected for the network under “random” loss. However, as packet classification is applied first, this yields 3 $BurstR$ values corresponding to Voiced, Unvoiced, and Unclassified losses, these values are called $BurstR_{Voiced}$, $BurstR_{Unoiced}$, and $BurstR_{Unclassified}$ respectively.

Calculation of each class considers only packet loss within that class. For example for Voiced losses, average length of burst in an arrival sequence can be calculated by counting the number of lost packets which are classified as Voiced loss and divide them on how they are distributed. Whereas average length of burst expected for network under random packet loss can be calculated using equation (7.4) by replacing Ppl by Ppl_{Voiced}. Similar calculations are applied to calculate $BurstR_{Unvoiced}$ and

$BurstR_{Unclassified}$. The corresponding equation for $BurstR_{Voiced}$, $BurstR_{Unvoiced}$, and $BurstR_{Unclassified}$ are:

$$BurstR_{Voiced} = \frac{\text{Average observed burst length in Voiced-Classified Lost Frames within arrival sequence}}{\text{Average burst length expected for networks with “random” loss equals } Ppl_{Voiced}} \tag{7.8}$$

$$BurstR_{Unvoiced} = \frac{\text{Average observed burst length in Unvoiced-Classified Lost Frames within arrival sequence}}{\text{Average burst length expected for networks with “random” loss equals } Ppl_{Unvoiced}} \tag{7.9}$$

$$BurstR_{Unclassified} = \frac{\text{Average observed burst length in Unclassified Lost Frames within arrival sequence}}{\text{Average burst length expected for networks with “random” loss equals } Ppl_{Unclassified}} \tag{7.10}$$

7.5 Modified E-model based on Voiced/Unvoiced Classification for Lost Packets

Recall from chapter 3 that the effect of packet loss is characterised by the E-model using packet loss dependent Effective Equipment Impairment Factor (*Ie-eff*) which can be calculated according to the following formula [84]:

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{7.11}$$

Ppl refers here to the overall packet loss percentage in the stream. The hope here is to replace this value by breaking it up according to the lost packets and whether they are Voiced or Unvoiced in a formula that looks like this formula:

$$Ppl = Ppl_{Voiced} + Ppl_{Unvoiced} \tag{7.12}$$

where

Ppl_{Voiced}	Packet loss during Voiced parts of the speech
$Ppl_{Unvoiced}$	Packet loss during Unvoiced parts of the speech

But as the overall packet loss equals the packet loss during Voiced segments + packet loss during Unvoiced segments, the above equation will not alter the overall value for *Ie-eff* as computed using equation (7.11). To put forward a new formula that has the power to modify equation (7.11) and bring the output of the E-model closer to the output of PESQ, the basic equation (7.12) is modified into a new formula:

$$Ppl = \alpha_V.Ppl_{Voiced} + \alpha_U.Ppl_{Unvoiced} \tag{7.13}$$

where

α_V Voicing Factor
α_U Unvoicing Factor

If the values of α_V and α_U are both set to 1, the overall *Ppl* will not change and the E-model estimation will remain the same, so the goal is to find an ideal values for α_V and α_U.

Also as illustrated in the previous sections, is the fact that some frames can not classified into either Voiced or Unvoiced due to their existence in the middle of loss burst as explained in section 7.3. To take packet loss during unclassified frames into account, a new *Ppl* term for unclassified losses is introduced, it is called $Ppl_{UnClassified}$. Based on this, equation (7.13) develops into a new equation to consider these frames

$$Ppl = \alpha_V.Ppl_{Voiced} + \alpha_U.Ppl_{Unvoiced} + 1.Ppl_{UnClassified} \tag{7.14}$$

Based on the fact that packet loss during voiced parts of the speech has more perceptual effect than packet loss during unvoiced parts of the signal, the value of α_V is expected to be more than 1 and the value of α_U to decrease to less than 1. This is yet to be confirmed by experimental analysis as explained in the coming section. Also, the coefficient of $Ppl_{UnClassified}$ which is given the value of 1 will not be allowed to change, as no extra or less weight is desired to unclassified packets.

In the same way $BurstR$ is broken into $BurstR_{Voiced}$, $BurstR_{Unvoiced}$, and $BurstR_{Unclassified}$ where the overall $BurstR$ can be calculated according to the following equation:

$$BurstR = \alpha_V.BurstR_{Voiced} + \alpha_U.BurstR_{Unvoiced} + 1.BurstR_{UnClassified} \tag{7.15}$$

Based on equations (7.14) and (7.15), *Ie-eff* can be calculated using the following equation

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{\alpha_V.Ppl_{Voiced} + \alpha_U.Ppl_{Unvoiced} + 1.Ppl_{UnClassified}}{\frac{\alpha_V.Ppl_{Voiced}+\alpha_U.Ppl_{Unvoiced}+1.Ppl_{UnClassified}}{\alpha_V.BurstR_{Voiced}+\alpha_U.BurstR_{Unvoiced}+1.BurstR_{UnClassified}} + Bpl} \tag{7.16}$$

The alteration of the E-model to cope with speech classification based on PESQ is illustrated in Figure 7.4.

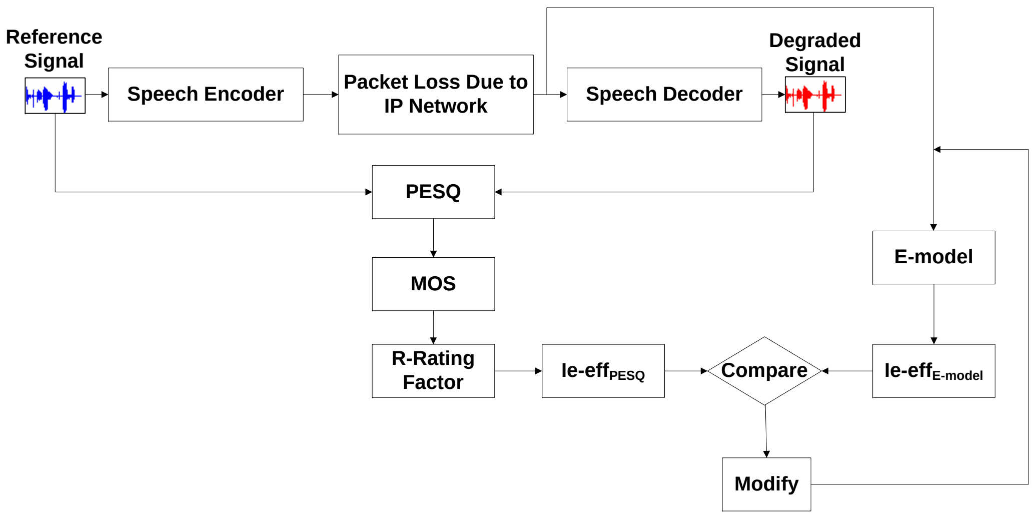

Figure 7.4: Increasing E-model's Accuracy Based on PESQ

PESQ in Figure 7.4, is used as a base measurement as it provides a more accurate measurement of speech quality due to its intrusive nature. Based on PESQ measurement of speech quality, MOS value is derived and R-Rating Factor is calculated. From R-Rating Factor, *Ie-eff* is calculated and this is called Ie-eff$_{PESQ}$ as it is derived from PESQ measurements. At the same time E-model calculates *Ie-eff* utilising equation (7.16) which is called Ie-eff$_{E-model}$ to distinguish it from Ie-eff$_{PESQ}$.

By comparing Ie-eff$_{PESQ}$ and Ie-eff$_{E-model}$, the E-model can be modified to allow its estimation to come closer to PESQ measurement. This can be achieved by modifying the values of α_V and α_U.

One question remains to answer is how the best values for α_V and α_U can be determined. Next section explains the empirical study performed to retrieve the data needed to determine the values of α_V and α_U. In section 7.7 a method to work out the ideal values for α_V and α_U is described.

7.6 Empirical Study for Voicing Factor Calculation

In order to have a wide variety of data available to determine the best values for α_V and α_U, packet loss is simulated using 2-state Gilbert model as explained in section 4.3. Packet loss simulation uses *Ppl* in the range 0 to 20 and $BurstR$ in the range 1 to 2. For each combination of *Ppl* and $BurstR$, the experiment is repeated for 30 times making up a total of 1320 runs.

During each run, packet loss is simulated using 2-state Gilbert model constructed based on the values of *Ppl* and $BurstR$ to retrieve a degraded signal. The degraded signal is compared against the original signal to calculate $PESQ$ score. This $PESQ$ score is then used to calculate MOS_{LQO} (equation (3.2)) and R-Rating Factor is calculated (equation (3.10)). Finally *Ie-eff* is derived from R-Rating Factor (equation (3.8)), which is called Ie-eff$_{PESQ}$ because it is based on PESQ calculations.

Then the degraded signal is used for further calculations. The Non-missing packets are classified into Voiced or Unvoiced according to the method explained in section 7.2. Then these packets are used to classify the missing packets according to the rules defined in Table 7.1 in section 7.3.

As described in section 7.4.2, the missing packets are used to calculate packet loss statistics during Voiced, Unvoiced, and Unclassified packets. From the above there are 1320 vectors corresponding to the above settings, for each vector, statistics about Ie-eff$_{PESQ}$, *Ppl* and $BurstR$ for different classes of packets are calculated. From this data set the best values of α_V and α_U that makes E-model estimation of the quality as close as possible to the PESQ measurement will be derived.

7.7 Genetic Algorithms Approach for Calculating Voicing Coefficients

Finding the the best values for α_V and α_U in equation 7.16 is an optimisation problem. The optimisation criteria is based on minimising the difference between the E-model's estimation of the quality and the PESQ measurement of the quality over the whole available data set to make the Ie-eff$_{E-model}$ closer to Ie-eff$_{PESQ}$.

As Artificial Neural Networks (ANNs) are normally used for either pattern classification or function approximation problems, they will not be considered as a possible solution for this optimisation problem.

Several techniques can be used for solving optimisation problems such as successive approximation, exhaustive search, and Genetic Algorithm (GA). Successive approximation can be used for problems with search space of one slope and without local minima, which is not guaranteed for our problem.

Exhaustive search can be used for limited search space problems by exploring all possible values of α_V and α_U and testing their performance to find the best solution. As the search space for α_V and α_U is not limited, searching for the best solution using exhaustive search is a difficult proposition.

For this problem a Genetic Algorithm (GA) approach will be used to find the best combination of α_V and α_U due to the ability of GAs of solving optimisation problems based on natural selection [145].

In section 7.7.1 different parameters used in the GA experiment are discussed and their choice is justified. Section 7.7.2 describes the GA experiments performed to find optimal values for the voicing parameters α_V and α_U.

7.7.1 Setting of Genetic Algorithms Parameters

Each chromosome consists of two variables corresponding to the parameters α_V and α_U. In the GA experiments a population size of 100 will be used which determines how many possibilities will be examined in each generation. By setting the population size to 100, each generation will contain 100 pair of values corresponding to the two parameters need to be optimised.

Although with a chromosome having just two variables, the population size could be selected to be a much larger value than 100, but the selected fitness function (described in next section) is computationally expensive. As the fitness function is used by each chromosome in each generation, the complexity of the fitness function should also be taken into consideration, therefore the population size is selected to be 100.

The population input range will be set corresponding to the coefficients to [0 10]. By setting the upper limit for the range to 10, this allows wider search space which allows higher diversity for the parameters and avoids local minima problem. However, GA with enough generations has the power to set the parameter to a much smaller values if it found that the optimal values for the parameters are much less than 10. It even can set the coefficients value to zero to indicate the effect of the related parameter is negligible.

For the production of the next generation, the best 2 chromosomes of the current generation are taken to the next generation (elitism). After experimenting with many values for crossover, it is chosen to produce 80% of the children (other than children produced with elitism) using crossover, while the rest of the children are produced using mutation.

For the number of generations produced, maximum number of generations will be set to Infinite (Inf) so that new generations continue to be produced and the generation process is stopped when no improvement occurs to the optimisation function for 50 successive generations.

After the genetic algorithm terminates and the final value using the final generation is produced, a hybrid function is used which takes the final point of the genetic algorithm as its initial point and returns a more accurate result by fine tuning in the region surrounding that point.

7.7.2 Genetic Algorithms Experiments

To find the optimal values for the parameters α_V and α_U, GA needs a fitness function to determine how successful a solution is in bringing the E-model's estimation of the quality closer to the PESQ measurement.

The fitness function used in the experiment is listed in Figure 7.5. In this fitness function, the difference between Ie-eff$_{E-model}$ as computed by equation (7.16) and Ie-eff$_{PESQ}$ as computed empirically over the whole data set is used as a fitness criteria in the GA experiments. i.e. the chromosome that produces the minimum overall difference from Ie-eff$_{PESQ}$ over the whole data set is chosen to be the fittest in its generation.

The setting for the GA parameters set as described in the previous section, i.e population size of 100, 2 elite chromosomes, crossover of 80% of the children other than the elite children, while the rest of the children are produced using mutation.

After running several experiment, it is noticed that no improvement occur after 100 generations as the values of α_V and α_U stabilise to [2.364 0.00238], respectively as shown in Figures 7.6 and 7.7. Figure 7.6 shows the how the best fit function stabilise over generations while 7.7 shows the best individual.

7.8 Evaluation of the Proposed Packet Loss Classification Model

To determine the best pair of values α_V and α_U, the benchmark criteria is to make the E-model estimation closer to the accurate PESQ measurement. The fitness criteria used to determined the best pair of values was based on the sum of differences in estimation between the new model and the PESQ measurements over the whole data set (1320 vectors).

The differences between Ie-eff$_{E-model}$ and Ie-eff$_{PESQ}$ over the 1320 vectors is 30305 unit, when the new model is applied the difference between Ie-eff$_{PESQ}$ and Ie-eff$_{NewModel}$ is reduced to 9956 which is less than 33% of the original deviation in the estimation. In terms of the worst case for the difference, it was 47.98 between PESQ and the E-model while it is reduced into 21.84 in case of the new model.

```
function FitnessValue=minimise_Ieeff(Alphas)
 %The fitness function takes the GA chromosome as a vector
 %and return the fitness of this chromosome


 Diff=0;%Variable to hold the sum of all difference
 Ie=11;%Equipment Impairment for G.729
 Bpl=19;%Packet Loss Robustness for G.729

 %Loop through all vectors in the data set
 for i=1:1320
   Begin

    %Calculate the overall packet loss probability by
    %combining Voiced loss multiplied by its factor with
    %Unvoiced Loss Multiplied by its factor with
    %Unclassified Loss

    OverlAllPpl=Alphas(1)*Dataset(i).PplVoiced
               +Alphas(2)*Dataset(i).PplUnvoiced
               +1.Dataset(i).PplUnclassified


    %Calculate the overall burst ratio by
    %combining Voiced Burst ratio multiplied by its factor with
    %Unvoiced burst ratio multiplied by its factor with
    %Unclassified burst

    OverlAllBurstR=Alphas(1)*Dataset(i).BurstRVoiced
                  +Alphas(2)*Dataset(i).BurstRUnvoiced
                  +1.Dataset(i).BurstRUnclassified

    %Calculate the new Equipment impairment factor
    newIeeff=Ie+(95-Ie)*OverlAllPpl / (OverlAllPpl/OverlAllBurstR + Bpl)

    %Calculate the difference for this record between
    %the empirical Ie-eff and newly calculated Ie-eff
    %and add it to the overall differences

    Diff=Diff + Abs(Dataset(i).Ieeff_PESQ - newIeeff)

   End

 %Return the fitness value for this pair of Alphas values
 FitnessValue=Diff
```

Figure 7.5: Pseudo code for the GA Fitness function

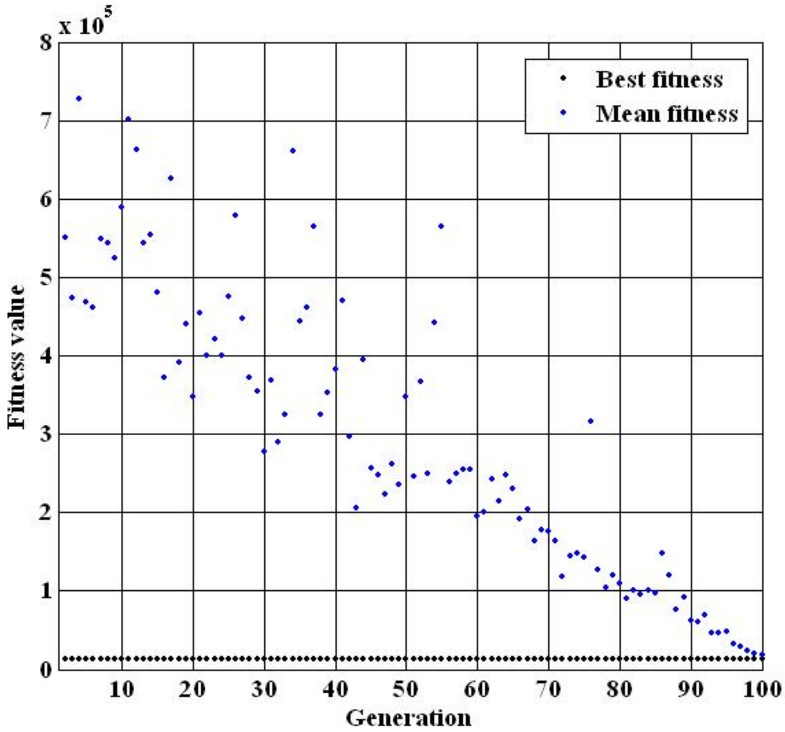

Figure 7.6: Best Function Value Vs. Generation

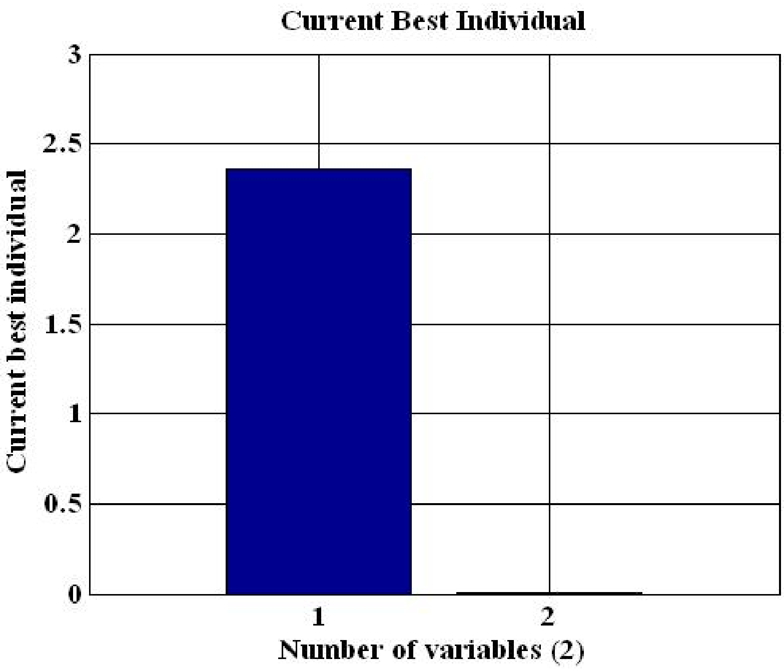

Figure 7.7: Vector Entries of the Individual with the Best Fitness Function

The mean difference in estimation between the E-model's calculated Ie-eff MOS and empirical Ie-eff based on PESQ measurement was 22.96 while in case of the new model, the mean difference is reduced to 7.54 which indicates more accurate estimation of the quality in case of the new model. Also the correlation factor between PESQ values and the new model values is 0.9056 while using the original E-model, the correlation was 0.8358.

To visualise the correlation with quality prediction using PESQ, the scatter diagram for Ie-eff$_{E-model}$ with Ie-eff$_{PESQ}$ is shown in Figure 7.8 while the scatter diagram for the new model and Ie-eff$_{PESQ}$ is shown in Figure 7.9.

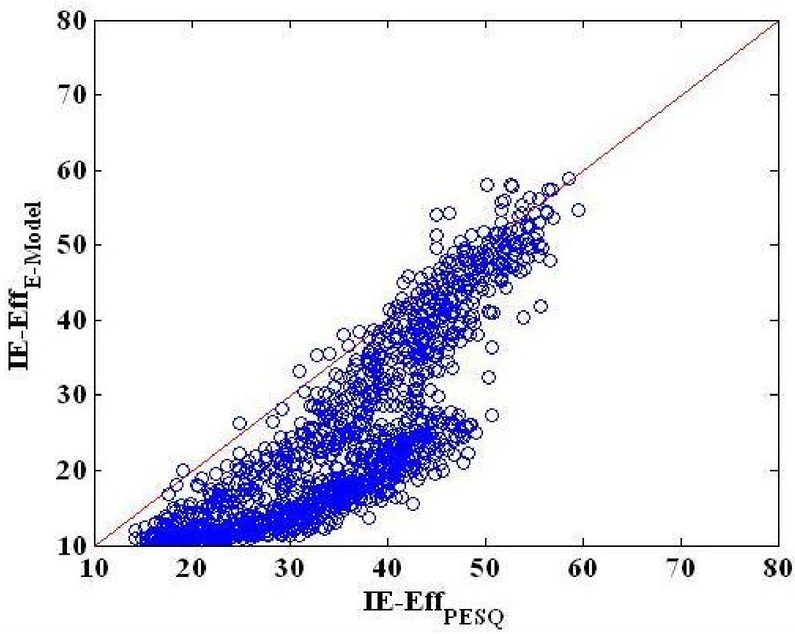

Figure 7.8: Scatter Diagram between Ie-eff$_{E-model}$ and Ie-eff$_{PESQ}$

By comparing the two Figures it is noted that in the first figure when the E-model is compared against PESQ, most of the points are located at one end of the perfect linear model which implicates the existence of deviation between the two models. In the second Figure when the new model is compared against PESQ most of the points are around the perfect linear model which implicates the deviation between the two models is less clear.

In Figure 7.10 a box plot is plotted to demonstrate the distribution of difference between the E-model's estimation and the $PESQ$ measurement in terms of Ie-eff. Figure 7.11 shows the same relation but this time between the new model's estima-

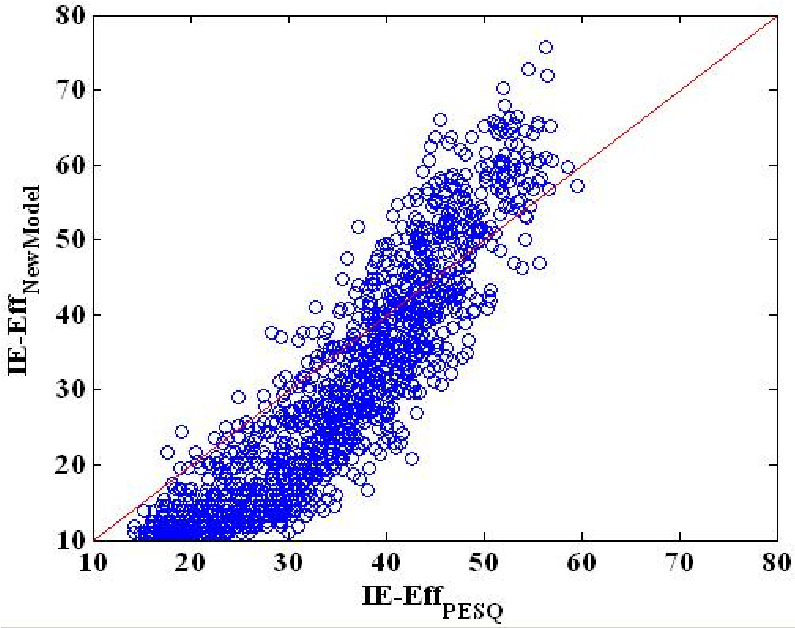

Figure 7.9: Scatter Diagram between Ie-eff from the new model and Ie-eff$_{PESQ}$

tion and the $PESQ$ measurement.

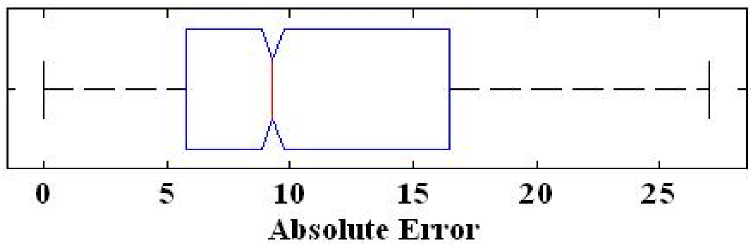

Figure 7.10: Box plot of the deviation between the E-model's estimation of the quality and $PESQ$ measurement in terms of Ie-eff

By comparing the two Figures it is clear that the majority of points in the second figure are closer to zero (less deviation) with a median of 6.89 in comparison with the first figure with deviation of 9.31 in case of the E-model.

From the above measurements and Figures it is clear that the new model improves the estimation accuracy of the E-model by classifying the packet loss into Voiced and Unvoiced contents. However, to make the comparison more meaningful MOS scores from the new model in comparison with the MOS score retrieved from PESQ should be compared against the quality estimation in the E-model with the

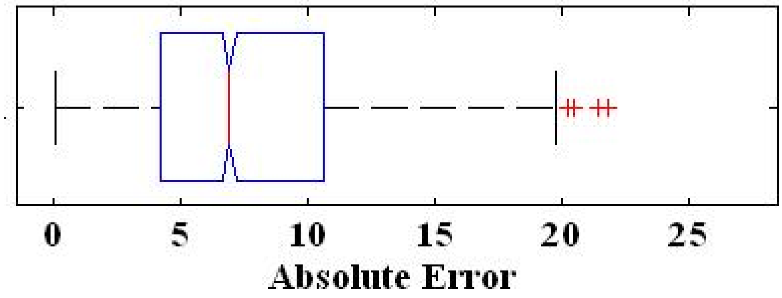

Figure 7.11: Box plot of the deviation between the new model estimation of the quality and $PESQ$ measurement in terms of Ie-eff

MOS scores from PESQ to measure the difference in deviation between the two model. Ie-eff from PESQ, the E-model and the new model are used to calculate R-Rating Factor and then MOS scores. These MOS scores are compared next.

The differences between E-model's MOS and PESQ-derived MOS over the 1320 vectors is 667.0130 MOS units, when the new model is applied the difference is reduced to 469.5634. In terms of the worst case for the difference, it is 1.37 between $PESQ$ and the E-model while it reduces into 1.10 in case of the new Model.

The mean difference in estimation between the E-model and $PESQ$ is 0.5053 MOS while in case of the new model the mean difference reduces to 0.3557 which indicates more accurate estimation of the quality in case of the new model. Also the correlation factor between PESQ-derived MOS scores and the new model MOS scores is 0.9027 while the E-model's correlation was 0.8348.

The scatter diagram for E-model's MOS with the PESQ-derived MOS is shown in Figure 7.12 while the scatter diagram for the new model with the PESQ-derived MOS is shown in Figure 7.13.

Comparison of the two Figures reveals that in the first figure when the E-model is compared against $PESQ$ most of the points are located at one end of the perfect linear model which implicates the existence of deviation between the two models. In the second figure when the new model is compared against $PESQ$ the points are much closer to the line and good percentages are distributed in both sides of the line which implicates less deviation between the two models than the first case.

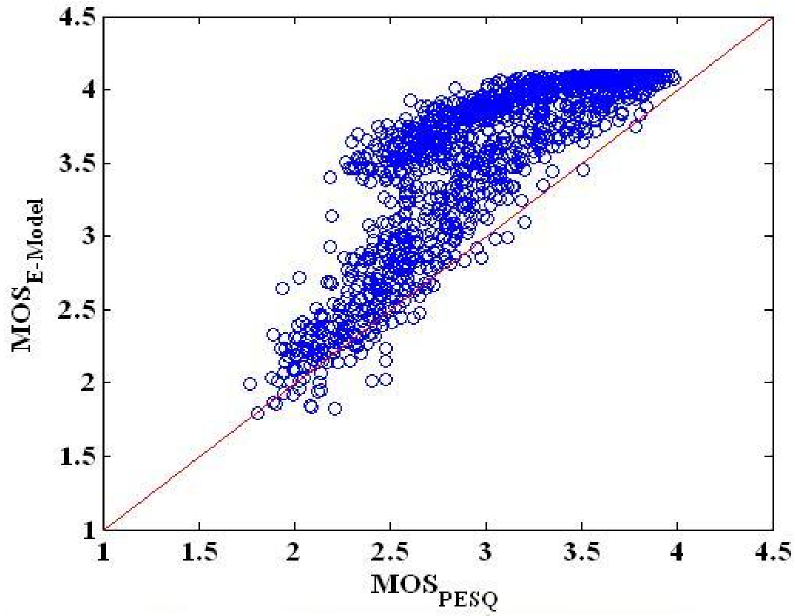

Figure 7.12: Scatter Diagram between the E-model MOS and PESQ-derived MOS

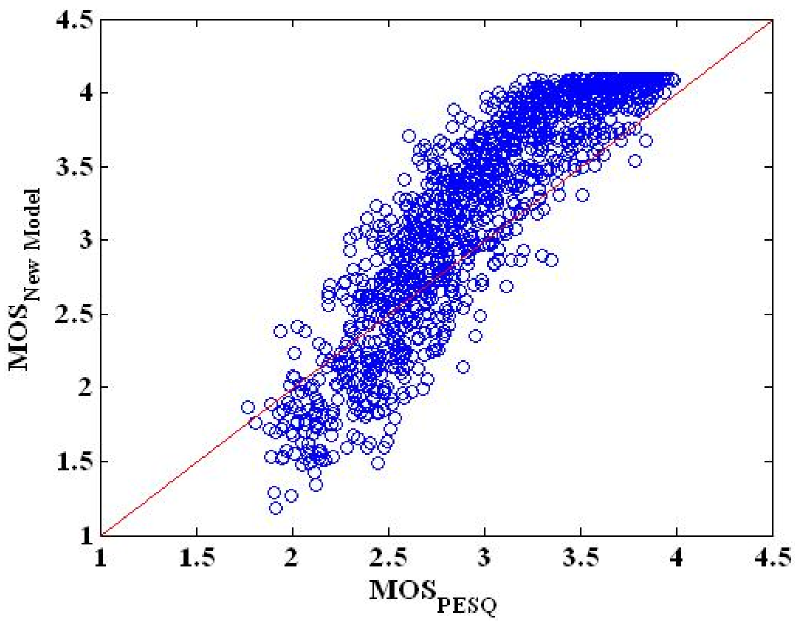

Figure 7.13: Scatter Diagram between MOS from the new model and PESQ-derived MOS

In Figure 7.14 a box plot is plotted to demonstrate the distribution of differences between the E-model's estimation and the $PESQ$ measurement of MOS. While Figure 7.15 shows the same relation but this time between the new model's estimation and the $PESQ$ measurement.

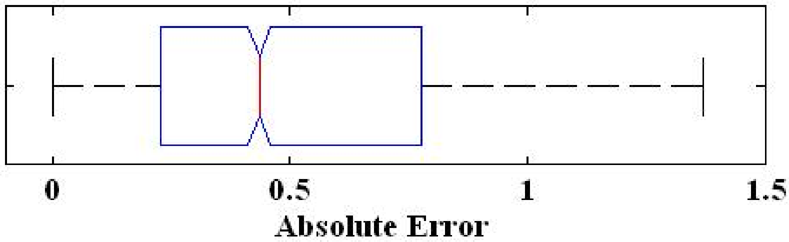

Figure 7.14: Box plot of the deviation between the E-model's estimation of the quality and $PESQ$ measurement in terms of MOS

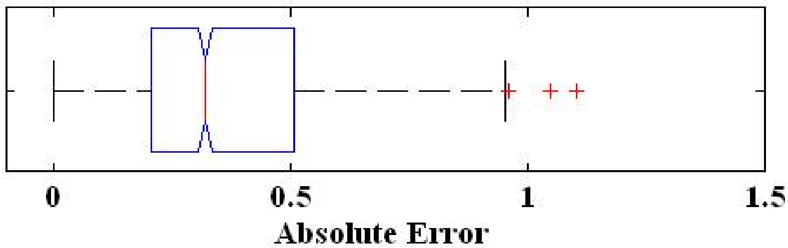

Figure 7.15: Box plot of the deviation between the new model estimation of the quality and $PESQ$ measurement in terms of MOS

By comparing the two figures it is very clear that the majority of points in the second figure are closer to zero (less deviation) than the first figure with 75% of values less than 0.51 MOS while the 75% mark in the first figure is for values less than or equal 0.78. The median in the second graph is 0.32 while it is 0.44 in the first figure.

From the above, there is a clear indication that the new model's estimation of the quality in terms of MOS is more accurate than the E-model's estimation when both models are compared against the PESQ measurement of the quality.

7.9 Summary

In this chapter a method for improving the accuracy of the E-model in comparison with PESQ was proposed. The proposed method uses packets surrounding the lost packet to predict whether the content of lost packets are Voiced or Unvoiced. Based on Voicing and Unvoicing prediction and using empirical results obtained about quality measurement using PESQ, E-model's estimation of the quality is modified in such a way its estimation comes closer to the PESQ measurement which is more accurate due to its intrusive nature.

The new model proved to gives more accurate in estimating the quality than the E-model which is illustrated through several measurements and comparisons between the E-model and the new model from one side and PESQ measurement from the other side. The new model's results showed less deviation from PESQ results than their E-model's counterpart.

Chapter 8

Combined Subjective-Test Free, Voicing Classification Extension to the E-model

8.1 Introduction

In chapters 5 and 6 an attempt was made to extend the E-model using Perceptual Evaluation of Speech Quality (PESQ). The purpose was to avoid the need for the time-consuming, expensive subjective tests which are necessary to calibrate the parameters of the E-model. The main equation to calibrate was the one that is used to calculate packet loss Effective Equipment Impairment Factor (*Ie-eff*):

$$Ie\text{-}eff = Ie + (95 - Ie).\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{8.1}$$

In the above equation both Ie and Bpl depend on subjective tests and they depend on the speech coder under test. As a result of the above extension, the new model avoids the use of these two parameters, this new model could be a linear or non linear regression model or an Artificial Neural Networks (ANN) model.

In case of either linear or non linear regression model, a new equation that does not depend on these two parameters was developed where the new equation is defined in terms of Ppl and $Burst$. Both Ppl and $Burst$ depend on packet loss characteristics of the system and how this packet loss affect the received speech signal.

In case an ANN model is used, Ie and Bpl variables are being absorbed in form of weights and biases of the network where the input to the ANN model are Ppl and $Burst$. By comparing ANN model with both linear and non linear regression, ANN seemed to offer more accurate measurement of the quality than the regression models.

During the experiments and the derivations that led to the extended E-model, the accuracy of the E-model was not questioned as the assumption was made that the calculations run by the E-model are accurate and the goal was to avoid the subjective tests. However, an observation was made that there is a deviation between quality estimation calculated according to the E-model (MOS_{LQE}) and quality measurements according to PESQ (MOS_{LQO}). Based on this observation, a correction formula was proposed to correct this deviation in quality estimation.

In chapter 7 this observation was further investigated and an attempt was made to improve the E-model accuracy by bringing its estimation closer to the measurement provided by the more accurate, intrusive-based PESQ technique. This attempt was based on classification of lost packets into either Voiced or Unvoiced loss according to the classification of the surrounding non-missing packets.

The proposed model improves the accuracy achieved by the E-model by reducing the deviation in quality estimation between the E-model estimation and $PESQ$ measurement.

In the new model, the equation for calculating *Ie-eff* was modified to accommodate packet loss classification:

$$Ie\text{-}eff = Ie + (95 - Ie) . \frac{\alpha_V . Ppl_{Voiced} + \alpha_U . Ppl_{Unvoiced} + 1 . Ppl_{UnClassified}}{\frac{\alpha_V . Ppl_{Voiced} + \alpha_U . Ppl_{Unvoiced} + 1 . Ppl_{UnClassified}}{\alpha_V . BurstR_{Voiced} + \alpha_U . BurstR_{Unvoiced} + 1 . BurstR_{UnClassified}} + Bpl} \quad (8.2)$$

The new classified losses are integrated with the E-model using the same equation used to calculate *Ie-eff*. This puts some restriction on the power of the classification extension for the E-model as the same form of equation is used with or without classification.

If the above ideas can be combined together to produce a non-intrusive extension for the E-model that is as accurate as $PESQ$ and at the same time does not depend on subjective tests to calibrate its parameters, this new model will have a wide applicability in estimating the speech quality for real-time applications.

8.2 The Proposed Technique

In this section the proposed technique is described. The setup for the system is depicted in Figure 8.1. In the system setup PESQ is used as a base criteria for comparison to avoid the need for subjective tests required to retrieve E-model's parameters as performed earlier in chapters 5 and 6. Also, by comparing the E-model's estimation of the quality with PESQ's measurement of the quality, the accuracy of the E-model is improved by bringing its estimation closer to the accurate PESQ measurement as performed in chapter 7.

With this system setup, the performance of the E-model is improved based on the PESQ and at the same time the subjectivity in calibrating its parameters is avoided, as such this modified model satisfies the requirements of quality estimation of voice traffic in IP networks.

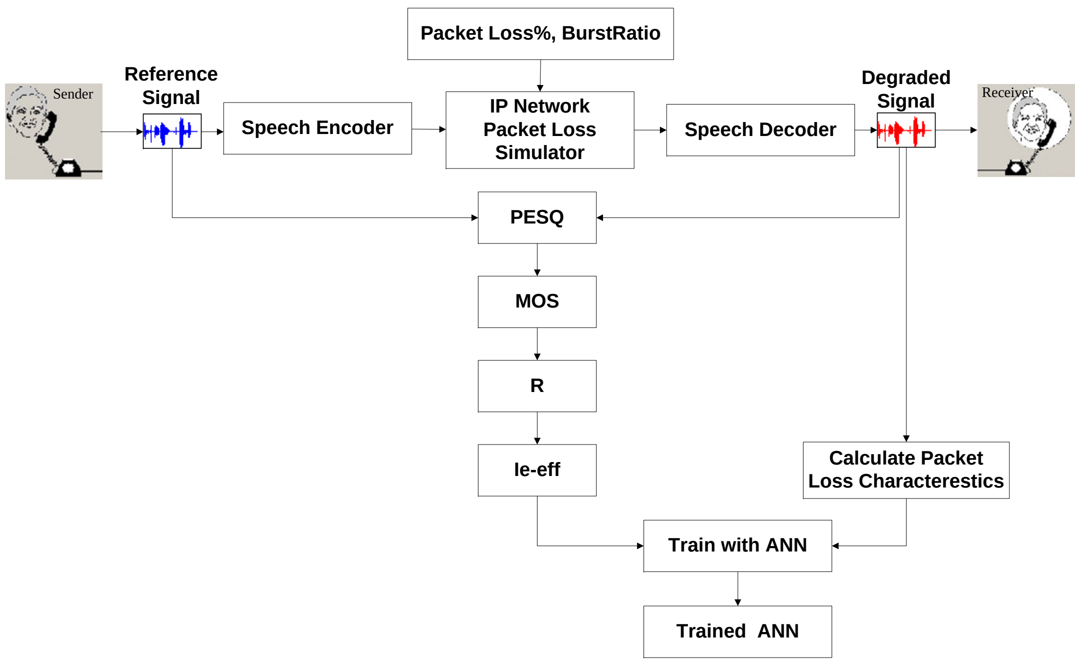

Figure 8.1: System setup for the E-model extension based on PESQ with voice classification

In the system setup shown above, the reference speech signal is first encoded and then packet loss is simulated. The received stream is decoded to retrieve the degraded speech signal and quality is measured by comparing the reference speech signal with the degraded speech signal using PESQ. This measured PESQ value is then mapped into MOS score which in turn is used to calculate R-Rating Factor and then *Ie-eff*. The calculated *Ie-eff* is considered an accurate measurement as it is calculated using the accurate PESQ algorithm.

At the same time, the degraded signal is analysed to calculate packet loss statistics for Voiced and Unvoiced parts of the signal similar to the classification performed in the previous chapter.

By feeding packet loss statistics as input information and *Ie-eff* as target information, an ANN model can be trained to find a relation between packet loss statistics and *Ie-eff*.

The choice of ANNs over linear or non linear regression models to find a relation between packet loss statistics and *Ie-eff* is coming from the fact that ANN performed the best in modelling *Ie-eff* with *Ppl* and *Burst* as explained in chapter 6. Also by choosing linear regression, we estimate the underlying relation to be linear which could be not. In case of non linear regression, we need to determine the form and degree of the polynomial while the relation could be modelled by a non-polynomial function.

As packet loss statistics (Voiced and Unvoiced are used as input information) and *Ie-eff* from PESQ is used as output information, this scheme gives more accurate estimation of the speech quality than the original E-model. Additionally, as the subjective-dependent parameters, namely Ie and *Bpl* are not used as input parameters, this scheme also does not depend on subjective tests to calibrate its parameters.

Based on the above, the combined scheme offers a solution for monitoring live traffic non-intrusively accurately and without the need for subjective tests to calibrate the parameters. As such this model has wide applicability in estimating the speech quality for real-time applications. Figure 8.2 shows how the new system can be used to monitor conversational speech quality non-intrusively.

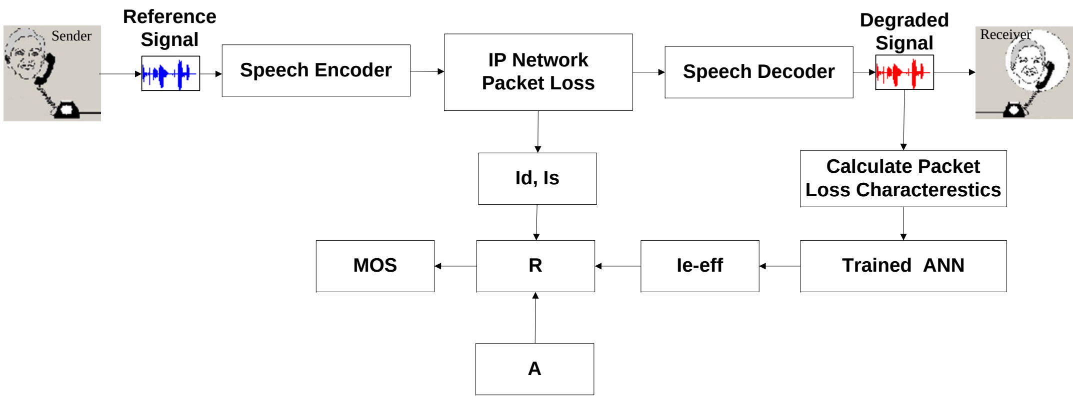

Figure 8.2: Application of the new system in monitoring live systems non-intrusively

In Figure 8.2 the degraded speech signal is analysed to extract packet loss statistics, then these statistics are fed into the trained ANN to estimate *Ie-eff*. This *Ie-eff* is then combined with Id, and Is and the Advantage factor is added to calculate R-Rating Factor. This R-Rating Factor is then mapped into a conversational speech estimation in terms of MOS.

8.3 Performance of ANN in Estimating *Ie-eff*

Packet loss is simulated using 2-state Gilbert model which uses *Ppl* in the range 0 to 20 and $BurstR$ in the range 1 to 2. For each combination of *Ppl* and $BurstR$, the experiment is repeated for 30 times making up a total of 1320 runs.

During each run, packet loss is simulated using 2-state Gilbert model constructed based on *Ppl* and $BurstR$ to retrieve a degraded signal. The degraded signal is compared against the original signal to calculate $PESQ$ score. This $PESQ$ score is then used to calculate MOS score (equation (3.2)), R-Rating Factor (equation (3.10)), and *Ie-eff* (equation (3.8)).

The degraded signal is also used for further calculations. The non-missing packets are classified into Voiced or Unvoiced. Then these packets are used to classify the missing packets. Statistics about the missing packets are then calculated.

This yields 1320 vectors, each vector contains statistics about *Ppl* and $BurstR$

for Voiced, Unvoiced, and Unclassified packets as well as estimation of the quality according to PESQ, mapped MOS value, R-Rating Factor, and *Ie-eff*.

The data is divided into training, validation and test subsets to improve generalisation accuracy and avoid overfitting the trained network into the training data. For the 1320 input vector 792 vectors were used for training, 264 for validation and 264 for testing. This corresponds to 0.6, 0.2, and 0.2 of the available data respectively where training, validation and test sets are picked as equally spaced points throughout the original data to avoid bias in the training set. Table 8.1 illustrates the division of data into training, validation and testing sets for each combination of Ppl and $BurstR$.

Iteration	**Assigned data set**	**Iteration**	**Assigned data set**
1	Training	16	Training
2	Training	17	Training
3	Training	18	Training
4	Validation	19	Validation
5	Testing	20	Testing
6	Training	21	Training
7	Training	22	Training
8	Training	23	Training
9	Validation	24	Validation
10	Testing	25	Testing
11	Training	26	Training
12	Training	27	Training
13	Training	28	Training
14	Validation	29	Validation
15	Testing	30	Testing

Table 8.1: Division of data set into training, validation, and testing for each combination of Ppl and Burst

For approximation of *Ie-eff*, a two-layer neural network with sigmoid transfer function in the first layer and linear transfer function in the output layer is used and the network is trained using LM algorithm.

Different number of neurons in the hidden layer are attempted ranging from 1 neuron to 100 neurons. With one neuron, the total number of weights and biases in the network equals 5, with 100 neuron the network will have the capability to fully remember the training set (792). For each setting the experiment is repeated for 30 different trials, where different random initial weights are used in each trial. This counts to 3000 experiments in total (100x30).

The performances of all the experiments in terms of training set and test set are listed in Appendix C in Tables C.1-C.25. During the experiment each network was allowed to train as far as 10000 epochs, although in all cases training stopped before reaching this number due to the error in the validation set exceeding the error in the training set.

The best network in terms of performance of the test set was found to be a network with 3 neurons in the hidden layer, this network has 25 of weights and biases in total. This network will be used for subsequent derivations in this section.

8.4 Results of Applying ANN in Quality Estimation

Using the best network retrieved from the last section, *Ie-eff* from the ANN can be compared with *Ie-eff* obtained experimentally over the whole data set. Performing such comparison results in multiple correlation coefficient (R) of value 0.9559 which indicates strong positive correlation and a good fit. The R^2, the coefficient of determination has the value of 0.9137 which indicates that 91.37% of the time the variation in the independent variable is explained by the model. From the *Ie-eff*, R-Rating Factor can be calculated which can be mapped into MOS score.

When the derived MOS values from the ANN model are compared with the MOS values obtained from the empirical $PESQ$ score to determine the accuracy of the ANN model, the resultant correlation coefficients value is found to be 0.9566 which indicates a strong positive correlation.

A scatter diagram between the ANN prediction and the empirical PESQ-derived MOS score is shown in Figure 8.3 to visualise the correlation between the corresponding values. Most of the points are concentrated near the perfect fit line due to the very high correlation.

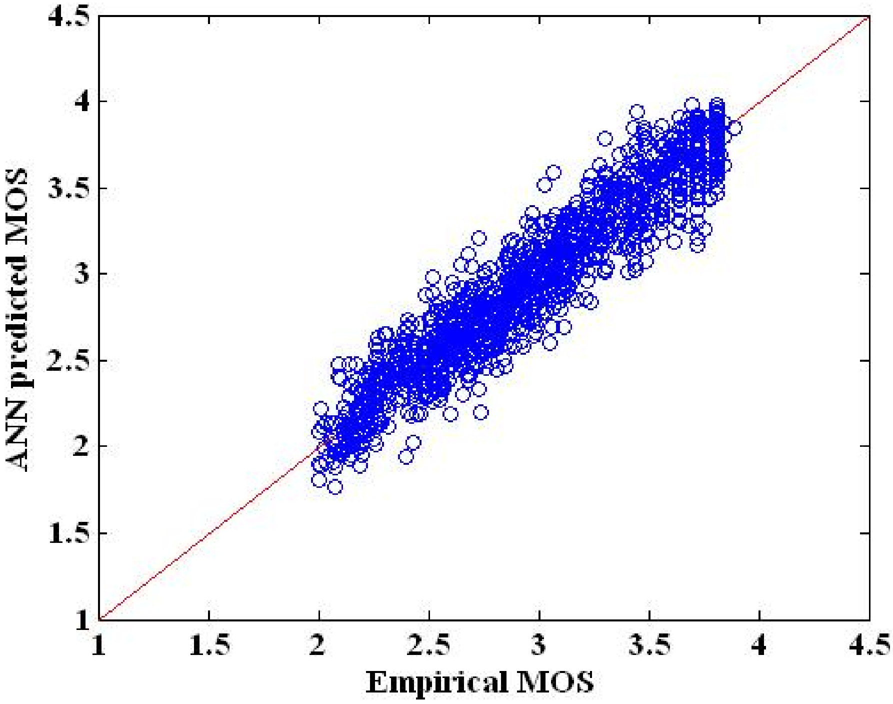

Figure 8.3: Scatter Diagram of quality prediction

Figure 8.4 shows the box plot of difference in quality prediction between the empirical values and the ANN model. From the figure it appears that the values of prediction error are clustered in the lower range with the first two quartiles are below 0.1 (median value) MOS. More than 75% of the data are below 0.2 MOS. There are few outliers (out of 1320) with high prediction error.

8.5 Summary

In this chapter, previous efforts in the previous chapters are combined to predict the speech quality accurately, non-intrusively in live networks without the need for

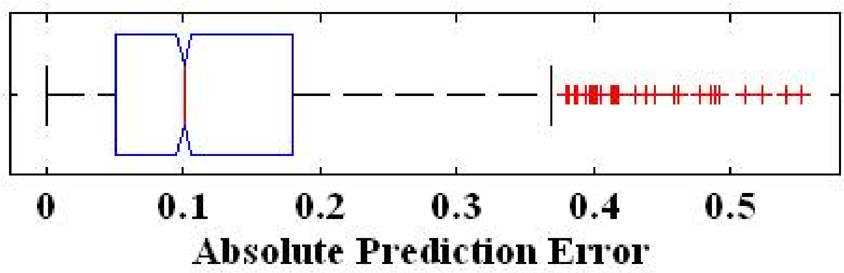

Figure 8.4: Box Plot of the error in Neural Network prediction

the subjective tests.

The proposed model avoids the hard to conduct, time-consuming, expensive, and lack repeatability subjective tests required to estimate the E-model parameters by using PESQ to find a model (ANN in this case) that does not use the subjective test related parameters Ie and Bpl. Also, the new model offers more accurate measurement to the speech quality depending on the more accurate intrusive-based PESQ method.

The proposed model has wide applicability in estimating the speech quality for voice applications over IP networks which increases its significance as it provides better features than the E-model for VoIP traffic by being more accurate in estimating the quality and able to avoid the subjectivity in estimating model's parameters.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Transmission of Voice over IP networks (VoIP) is one of the most active research areas in the world of communication due to the great expectations in terms of cost reduction and service innovations. However, as IP networks was designed to carry data rather than voice, several challenges arise when an attempt is made to transmit voice in addition to data traffic over such networks. These challenges occur due to the best effort nature of these network and due to the characteristics of real-time voice traffic.

For VoIP to be commercially viable and competitive to the traditional telephony solutions, these challenges should be tackled and their solutions should be evaluated. Therefore, it is important to measure the speech quality provided by such services for legal, commercial, and technical reasons because regardless of how cheap the service is, if the quality of such services is poor, it is not expected to attract users who are used to the high quality provided by traditional telephony.

Several solutions have been proposed to measure the speech quality. These solutions can be classified into: subjective methods, Intrusive-based objective methods such as Perceptual Evaluation of Speech Quality (PESQ) method, and non-intrusive objective methods. It is the non-intrusive based objective methods that are most suitable for monitoring live traffic in a productive network. The most famous method for monitoring the speech quality non-intrusively is the ITU-T E-model.

Several defects in the E-model were identified, including:

1 It depends on subjects tests to calibrate its parameters, as such it is applicable to limited network conditions and limited number of speech coders corresponding to those speech coder where subjective tests were implemented.

2 It does not look into the content of the received, degraded speech signal and estimates the speech quality purely depending on the networking conditions and statistics about the received speech signal. Based on this, the E-model provides a less accurate estimation of the received quality than the intrusive based PESQ method.

This research was stimulated by the lack of an appropriate method for measuring the speech quality accurately, non-intrusively, and without the need for the time-consuming, expensive, and hard-to-conduct subjective tests.

In this study an attempt was made to answer few questions related to speech quality measurement in VoIP applications such as:

- Is it possible to extend the E-model's applicability range depending on the intrusive-based, objective method of PESQ without the need to conduct the time-consuming, expensive, and hard to conduct subjective tests?
- What is the relation between quality estimation using the E-model and quality measurement using PESQ?
- Is it possible to make estimation of speech quality non-intrusively using the E-model as accurate as quality measurement using the intrusive PESQ?
- Is it possible to extend the E-model without the need for the subjective tests and at the same time improves its accuracy to make it closer to the quality measurement using PESQ?
- What are the opportunities for applying the E-model or the modified E-model in Service Oriented Computing (SOC) to measure the quality of the services provided?

To answer the above research questions, a series of experiments were conducted and the results were evaluated in comparison with the original methods. Several statistics about the achieved results were calculated and the goodness of these methods was also tested.

The results obtained indicates that the E-model can be extended using PESQ so that the need for the subjective tests is avoided. This extension was implemented using 3 methods: linear regression, non linear regression and Artificial Neural Networks (ANN). The result of these three approach were evaluated by comparing them to the original E-model and the ANN was the most successful one as it provided the most accurate measurement method in comparison with the E-model with the minimum degree of deviation in quality estimation and highest degree of correlation.

Using the above method, the E-model could be extended to new network conditions and to new speech coders without the need for the subjective tests, so the applicability of the E-model is extended in a continuously changing world of communication.

During the extension of the E-model, the relation between quality estimation using the E-model (MOS_{LQE}) and quality measurement using PESQ (MOS_{LQO}) was investigated and as a deviation is noticed between the outcome of the two methods, a correction formula between the two output was proposed.

To alleviate the deviation in quality estimation and make the quality estimation of speech using the E-model as accurate as possible in comparison with quality measurement using the accurate, intrusive-based method of PESQ, a new method was proposed to improve the E-model's estimation based on classification of missing voice packets into either Voiced or Unvoiced based on the surrounding non-missing packets.

When the output of the new method is compared against the original E-model, it was shown that the modified E-model provides more accurate measurement of speech quality than the original E-model without voicing classification. The proposed method showed higher degree of correlation and less deviation with the PESQ than the E-model.

Finally the E-model's extension to avoid the subjective tests using PESQ was combined with E-model's accuracy improvement by taking packet loss classification into account. This combination results in a modified E-model that is capable of providing accurate measurement of speech quality, non-intrusively and applicable to new network conditions and new speech coders without the need for the time-consuming, expensive, hard to conduct subjective tests.

The original or the modified E-model could also be applied to SOC applications by providing a specialised service for measuring service quality of other services. This proposal is explained next as a possible research direction.

9.2 Future Work

Finding of the thesis have lent themselves to further explore the following issues:

- Although the extension of the E-model using PESQ was tested using a number of possible ways of comparisons, but these results were not evaluated against the subjective tests which are the most accurate measurement of speech quality. Subjective tests provides measurement of quality as received by the user measured by real users and their perception reflects the most important and accurate measurement.

- The deviation in quality estimation was based on quality is measurement by PESQ and how the quality is estimated by the E-model, subjective tests should be conducted to confirm this result.

- Language dependency was not considered as a possible factor in this study which is something could be investigated.

- The proposed method extend the non-intrusive method of estimating the speech quality using intrusive methods, but what about the applicability of the new methods for estimating the video quality? Is the proposed method equally applicable to video applications as well as voice applications?

- The E-model or the modified E-model can be applied in service quality measurement in SOC applications. Among other services provided many services within SOC community are concerned about provisioning of services for multimedia applications from a service provider to set of clients as shown in Figure 9.1. This could be further investigated to see how QoS can be offered as a unique and specialised self-contained service.

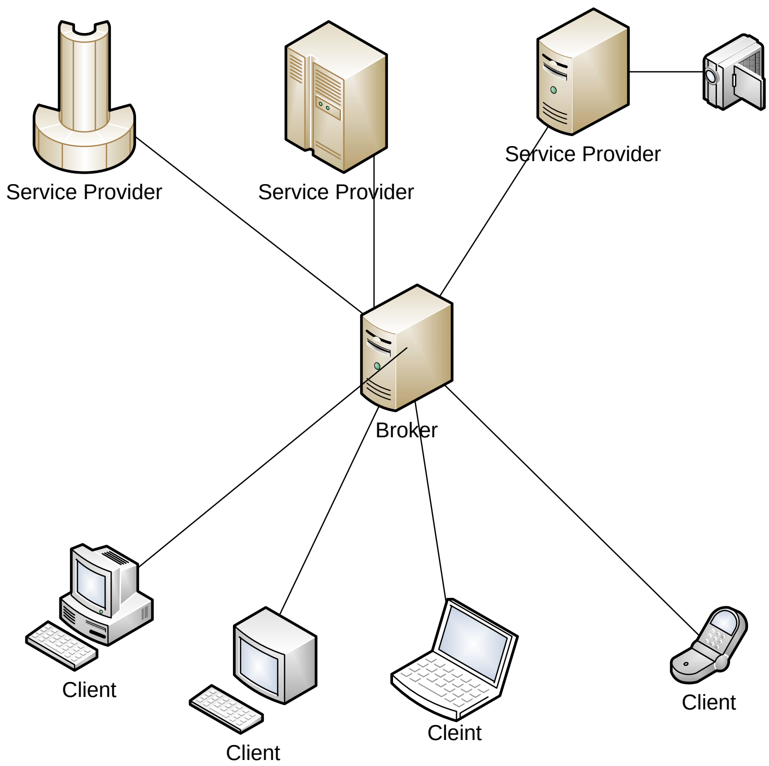

Figure 9.1: Service Scenario

The research directions within SOC have concentrated mainly on areas such as compositionality, service description languages, orchestration, models, etc. However a fundamental research question to SOC is how these services provide functionalities within an acceptable level of quality. Developers or designers within SOC often relegate these issues to the service providers themselves. This responsibility could be lift by identifiable self-contained service that can measure QoS of a given service(s) within a given domain. The QoS issues can be packaged into a separate and unique service that could be used, composed, interact with other services [3].

The QoS specialised service could interact with other services such as a media playback service responsible for playing the media contents received by the service provider to the client and then report the received quality to the service provider to calculate the required payment by the client as shown in Figure 9.2.

In this service provisioning architecture, the service quality is automatically and continuously monitored and client's profile is update based on the received quality [3].

The above discussion about the applicability of the E-model in measuring the quality is readily applicable using either the E-model or the modified E-model, but the integration of the quality measurement service with other services is still to be investigated and this is very much embedded within SOC area.

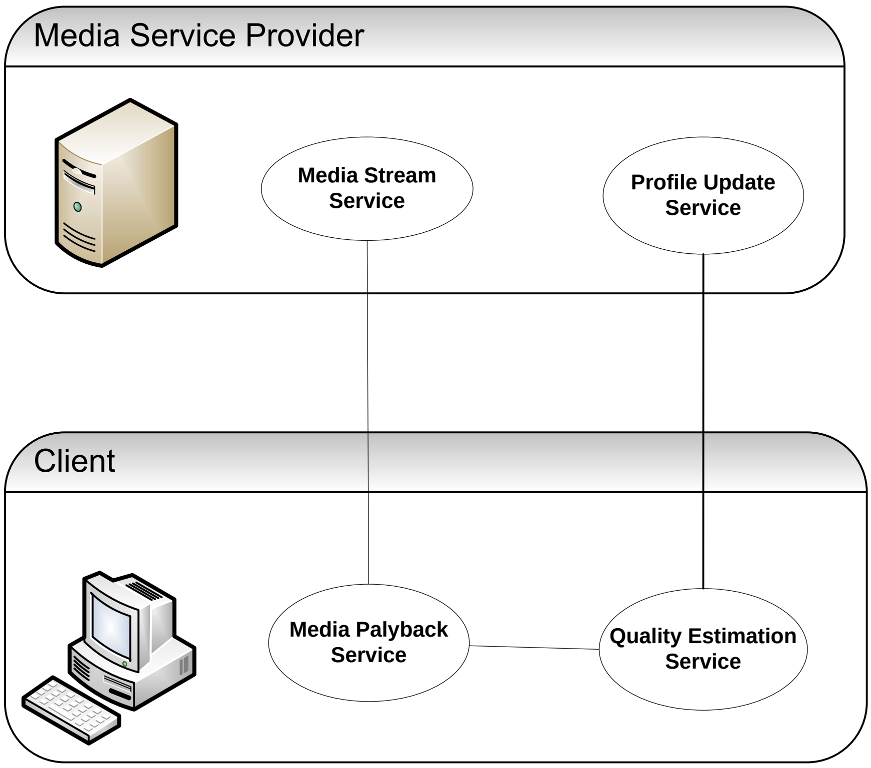

Figure 9.2: Service Provision with Quality Measurement Service

Appendix A

Extending The E-model Based on PESQ-Detailed Results

In this appendix the detailed results of derivation of the *Ie-eff* from PESQ scores as discussed in chapter 5 are listed.

Tables A.1-A.5 show the full results of PESQ scores, MOS_{LQO} Scores, MOS_{LQE} Scores, R-Rating Factor values and *Ie-eff* values for each combination of *Ppl* and *BurstR* as calculated experimentally for loss in speech file B_eng_f1 (see section 4.2). The data is splitted over 5 tables where each table is for a specific Burst Ratio (*BurstR*) from 1 to 5.

Each PESQ score was calculated as an average over 30 iterations to have an accuracy ± 0.01 MOS with 95% confidence. The MOS score was calculated from the PESQ score using equation 3.2. The R-Rating Factor was calculated using equation 3.10. Finally *Ie-eff* was calculated using equation 5.13.

Tables A.6-A.10 show the full results of *Ie-eff*, R-Rating Factor, MOS_{LQE} Scores, MOS_{LQO} Scores, and PESQ scores values and for each combination of *Ppl* and *BurstR* as calculated from the linear regression model derived in chapter 6. The equation for the linear regression model is:

$$\textit{Ie-eff} = 3.080 * BurstR + 2.331 * Ppl + 10.886 \tag{A.1}$$

Each table is for a specific Burst Ratio. The R-Rating Factor was calculated using equation 5.12. The MOS score was calculated from R-Rating factor using equation 3.9. Finally the PESQ score was calculated from the MOS score using equation 3.3.

Tables A.11-A.15 show the full results of *Ie-eff*, *R*-Rating Factor, MOS_{LQE} Scores, MOS_{LQO} Scores, and PESQ scores values and for each combination of *Ppl* and *BurstR* as calculated from the non linear regression model derived in chapter 6. The equation for the non linear regression model is:

$$\begin{aligned} \textit{Ie-eff} &= 25.0885 * BurstR - 6.6627 * BurstR^2 + 0.5910 * BurstR^3 \\ &\quad +4.01085 * Ppl - 0.0858 * Ppl^2 + 0.0011 * Ppl^3 - 14.6495 \qquad \text{(A.2)} \end{aligned}$$

Each table is for a specific Burst Ratio. The *R*-Rating Factor was calculated using equation 5.12. The MOS score was calculated from the MOS score using equation 3.9. Finally the PESQ score was calculated from the MOS score using equation 3.3.

Ppl	**PESQ Score**	MOS_{LQO}	MOS_{LQE}	R**-Rating factor**	***Ie-eff***
0	3.7535	3.8826	4.0496	80.6834	12.5166
0.5	3.7151	3.8359	3.9795	78.8425	14.3575
1	3.6945	3.8102	3.9410	77.8703	15.3297
2	3.6169	3.7111	3.7925	74.3192	18.8808
3	3.5748	3.6556	3.7093	72.4400	20.7600
4	3.5065	3.5630	3.5704	69.4350	23.7650
5	3.4431	3.4748	3.4381	66.6892	26.5108
6	3.3647	3.3628	3.2702	63.3204	29.8796
7	3.3522	3.3447	3.2431	62.7861	30.4139
8	3.3015	3.2707	3.1321	60.6216	32.5784
9	3.2338	3.1708	2.9824	57.7427	35.4573
10	3.1578	3.0577	2.8128	54.5141	38.6859
11	3.1604	3.0615	2.8184	54.6217	38.5783
12	3.1145	2.9929	2.7156	52.6717	40.5283
13	3.0165	2.8467	2.4965	48.5017	44.6983
14	3.0323	2.8702	2.5317	49.1740	44.0260
15	2.9323	2.7224	2.3101	44.9012	48.2988
16	2.9406	2.7346	2.3283	45.2570	47.9430
17	2.8899	2.6606	2.2174	43.0815	50.1185
18	2.8435	2.5936	2.1169	41.0741	52.1259
19	2.8173	2.5562	2.0609	39.9375	53.2625
20	2.7653	2.4829	1.9510	37.6616	55.5384
21	2.7107	2.4076	1.8381	35.2461	57.9539
22	2.6927	2.3832	1.8015	34.4418	58.7582
23	2.6869	2.3753	1.7897	34.1812	59.0188
24	2.6056	2.2678	1.6285	30.4688	62.7312
25	2.6028	2.2642	1.6232	30.3413	62.8587
26	2.5445	2.1900	1.5119	27.5765	65.6235
27	2.5324	2.1750	1.4893	26.9901	66.2099
28	2.5165	2.1553	1.4599	26.2106	66.9894
29	2.4717	2.1011	1.3786	23.9495	69.2505
30	2.4524	2.0783	1.3444	22.9419	70.2581

Table A.1: $PESQ$, MOS_{LQO}, MOS_{LQE}, R and *Ie-eff* with different *Ppl* values, $BurstR = 1$

Ppl	PESQ Score	MOS_{LQO}	MOS_{LQE}	R-Rating factor	*Ie-eff*
0	3.7535	3.8826	4.0496	80.6834	12.5166
0.5	3.7005	3.8178	3.9524	78.1546	15.0454
1	3.6391	3.7400	3.8357	75.3247	17.8753
2	3.5724	3.6524	3.7044	72.3322	20.8678
3	3.5032	3.5584	3.5636	69.2912	23.9088
4	3.3605	3.3566	3.2610	63.1390	30.0610
5	3.2406	3.1809	2.9975	58.0302	35.1698
6	3.2284	3.1628	2.9703	57.5113	35.6887
7	3.1605	3.0616	2.8187	54.6269	38.5731
8	3.0731	2.9311	2.6230	50.9132	42.2868
9	3.0169	2.8474	2.4975	48.5205	44.6795
10	2.9809	2.7940	2.4174	46.9825	46.2175
11	2.8979	2.6721	2.2347	43.4231	49.7769
12	2.8143	2.5520	2.0546	39.8075	53.3925
13	2.8344	2.5806	2.0974	40.6798	52.5202
14	2.7535	2.4665	1.9265	37.1431	56.0569
15	2.7220	2.4231	1.8613	35.7499	57.4501
16	2.6415	2.3148	1.6990	32.1262	61.0738
17	2.5847	2.2409	1.5882	29.4927	63.7073
18	2.5194	2.1589	1.4652	26.3527	66.8473
19	2.5167	2.1556	1.4602	26.2193	66.9807
20	2.4683	2.0971	1.3725	23.7731	69.4269
21	2.4441	2.0685	1.3297	22.4973	70.7027
22	2.3607	1.9737	1.1876	17.6775	75.5225
23	2.3874	2.0034	1.2321	19.3123	73.8877
24	2.3356	1.9463	1.1465	16.0200	77.1800
25	2.3461	1.9577	1.1636	16.7292	76.4708
26	2.2799	1.8874	1.0582	11.5771	81.6229
27	2.2758	1.8833	1.0520	11.1811	82.0189
28	2.1747	1.7836	1.0000	6.5153	86.6847
29	2.1810	1.7895	1.0000	6.5153	86.6847
30	2.1366	1.7482	1.0000	6.5153	86.6847

Table A.2: $PESQ$, MOS_{LQO}, MOS_{LQE}, R and *Ie-eff* with different *Ppl* values, $BurstR = 2$

Ppl	**PESQ Score**	MOS_{LQO}	MOS_{LQE}	R**-Rating factor**	***Ie-eff***
0	3.7535	3.8826	4.0496	80.6834	12.5166
0.5	3.6812	3.7935	3.9160	77.2518	15.9482
1	3.6079	3.6994	3.7749	73.9160	19.2840
2	3.5682	3.6468	3.6961	72.1486	21.0514
3	3.4003	3.4139	3.3469	64.8454	28.3546
4	3.3221	3.3008	3.1774	61.4999	31.7001
5	3.2773	3.2351	3.0787	59.5895	33.6105
6	3.1565	3.0557	2.8099	54.4587	38.7413
7	3.0790	2.9399	2.6363	51.1651	42.0349
8	3.0409	2.8831	2.5510	49.5429	43.6571
9	2.9060	2.6840	2.2526	43.7747	49.4253
10	2.9127	2.6938	2.2672	44.0615	49.1385
11	2.7824	2.5069	1.9869	38.4125	54.7875
12	2.7999	2.5316	2.0240	39.1794	54.0206
13	2.6995	2.3924	1.8154	34.7474	58.4526
14	2.6547	2.3322	1.7251	32.7268	60.4732
15	2.6609	2.3405	1.7375	33.0081	60.1919
16	2.5949	2.2540	1.6078	29.9707	63.2293
17	2.5289	2.1707	1.4829	26.8219	66.3781
18	2.5261	2.1672	1.4777	26.6833	66.5167
19	2.4154	2.0352	1.2798	20.9241	72.2759
20	2.3959	2.0130	1.2465	19.8107	73.3893
21	2.3603	1.9733	1.1870	17.6541	75.5459
22	2.2908	1.8987	1.0751	12.5740	80.6260
23	2.3118	1.9208	1.1083	14.2922	78.9078
24	2.3011	1.9096	1.0914	13.4496	79.7504
25	2.2278	1.8348	1.0000	6.5153	86.6847
26	2.1910	1.7990	1.0000	6.5153	86.6847
27	2.1181	1.7315	1.0000	6.5153	86.6847
28	2.1555	1.7656	1.0000	6.5153	86.6847
29	2.1168	1.7304	1.0000	6.5153	86.6847
30	2.1031	1.7182	1.0000	6.5153	86.6847

Table A.3: $PESQ$, MOS_{LQO}, MOS_{LQE}, R and *Ie-eff* with different *Ppl* values, $BurstR = 3$

Ppl	PESQ Score	MOS_{LQO}	MOS_{LQE}	R-Rating factor	*Ie-eff*
0	3.7535	3.8826	4.0496	80.6834	12.5166
0.5	3.6879	3.8020	3.9288	77.5658	15.6342
1	3.6238	3.7202	3.8061	74.6329	18.5671
2	3.5032	3.5584	3.5636	69.2909	23.9091
3	3.3986	3.4115	3.3433	64.7732	28.4268
4	3.2614	3.2117	3.0437	58.9164	34.2836
5	3.1786	3.0887	2.8593	55.3973	37.8027
6	3.1940	3.1116	2.8936	56.0485	37.1515
7	3.1295	3.0153	2.7493	53.3097	39.8903
8	2.9974	2.8183	2.4539	47.6854	45.5146
9	2.9066	2.6848	2.2537	43.7973	49.4027
10	2.9128	2.6939	2.2674	44.0658	49.1342
11	2.8213	2.5619	2.0695	40.1126	53.0874
12	2.7568	2.4711	1.9334	37.2898	55.9102
13	2.7281	2.4315	1.8739	36.0218	57.1782
14	2.5977	2.2576	1.6132	30.1020	63.0980
15	2.6397	2.3123	1.6953	32.0410	61.1590
16	2.5587	2.2079	1.5387	28.2599	64.9401
17	2.4637	2.0916	1.3644	23.5354	69.6646
18	2.5177	2.1569	1.4622	26.2724	66.9276
19	2.4658	2.0941	1.3680	23.6426	69.5574
20	2.3977	2.0151	1.2496	19.9191	73.2809
21	2.3453	1.9569	1.1624	16.6800	76.5200
22	2.3738	1.9883	1.2094	18.4959	74.7041
23	2.3126	1.9217	1.1095	14.3522	78.8478
24	2.2389	1.8459	1.0000	6.5153	86.6847
25	2.1944	1.8023	1.0000	6.5153	86.6847
26	2.1824	1.7909	1.0000	6.5153	86.6847
27	2.1250	1.7378	1.0000	6.5153	86.6847
28	2.1384	1.7499	1.0000	6.5153	86.6847
29	2.0307	1.6566	1.0000	6.5153	86.6847
30	2.0706	1.6900	1.0000	6.5153	86.6847

Table A.4: $PESQ$, MOS_{LQO}, MOS_{LQE}, R and *Ie-eff* with different Ppl values, $BurstR = 4$

Ppl	PESQ Score	MOS_{LQO}	MOS_{LQE}	R-Rating factor	*Ie-eff*
0	3.7535	3.7535	4.0496	80.6834	12.5166
0.5	3.6923	3.6923	3.9369	77.7675	15.4325
1	3.5948	3.5948	3.7489	73.3279	19.8721
2	3.4569	3.4569	3.4671	67.2830	25.9170
3	3.4334	3.4334	3.4174	66.2674	26.9326
4	3.2735	3.2735	3.0705	59.4309	33.7691
5	3.2453	3.2453	3.0078	58.2284	34.9716
6	3.1753	3.1753	2.8518	55.2540	37.9460
7	3.0360	3.0360	2.5400	49.3328	43.8672
8	2.9989	2.9989	2.4574	47.7524	45.4476
9	2.9380	2.9380	2.3225	45.1447	48.0553
10	2.9139	2.9139	2.2698	44.1125	49.0875
11	2.8041	2.8041	2.0329	39.3620	53.8380
12	2.8256	2.8256	2.0787	40.2991	52.9009
13	2.7239	2.7239	1.8651	35.8320	57.3680
14	2.6672	2.6672	1.7502	33.2958	59.9042
15	2.6428	2.6428	1.7015	32.1845	61.0155
16	2.5458	2.5458	1.5144	27.6424	65.5576
17	2.5378	2.5378	1.4995	27.2553	65.9447
18	2.5383	2.5383	1.5003	27.2768	65.9232
19	2.4246	2.4246	1.2957	21.4366	71.7634
20	2.4753	2.4753	1.3850	24.1334	69.0666
21	2.3841	2.3841	1.2266	19.1164	74.0836
22	2.3315	2.3315	1.1398	15.7313	77.4687
23	2.3232	2.3232	1.1265	15.1405	78.0595
24	2.2314	2.2314	1.0000	6.5153	86.6847
25	2.1987	2.1987	1.0000	6.5153	86.6847
26	2.1186	2.1186	1.0000	6.5153	86.6847
27	2.1331	2.1331	1.0000	6.5153	86.6847
28	2.1133	2.1133	1.0000	6.5153	86.6847
29	2.0863	2.0863	1.0000	6.5153	86.6847
30	2.0356	2.0356	1.0000	6.5153	86.6847

Table A.5: $PESQ$, MOS_{LQO}, MOS_{LQE}, R and *Ie-eff* with different Ppl values, $BurstR = 5$

Ppl	*Ie-eff*	R-Rating factor	MOS_{LQE}	MOS_{LQO}	PESQ Score
0	13.9660	79.2340	3.9900	3.8429	3.7209
0.5	15.1315	78.0685	3.9500	3.8163	3.6993
1	16.2970	76.9030	3.9000	3.7829	3.6727
2	18.6280	74.5720	3.8000	3.7162	3.6208
3	20.9590	72.2410	3.7000	3.6495	3.5702
4	23.2900	69.9100	3.5900	3.5761	3.5160
5	25.6210	67.5790	3.4800	3.5028	3.4631
6	27.9520	65.2480	3.3700	3.4294	3.4111
7	30.2830	62.9170	3.2500	3.3494	3.3554
8	32.6140	60.5860	3.1300	3.2693	3.3006
9	34.9450	58.2550	3.0100	3.1893	3.2463
10	37.2760	55.9240	2.8900	3.1092	3.1924
11	39.6070	53.5930	2.7600	3.0225	3.1343
12	41.9380	51.2620	2.6400	2.9425	3.0807
13	44.2690	48.9310	2.5200	2.8624	3.0271
14	46.6000	46.6000	2.4000	2.7824	2.9731
15	48.9310	44.2690	2.2800	2.7024	2.9186
16	51.2620	41.9380	2.1600	2.6223	2.8635
17	53.5930	39.6070	2.0400	2.5423	2.8075
18	55.9240	37.2760	1.9300	2.4689	2.7552
19	58.2550	34.9450	1.8200	2.3955	2.7018
20	60.5860	32.6140	1.7200	2.3288	2.6521
21	62.9170	30.2830	1.6200	2.2621	2.6012
22	65.2480	27.9520	1.5300	2.2021	2.5541
23	67.5790	25.6210	1.4400	2.1421	2.5056
24	69.9100	23.2900	1.3600	2.0887	2.4613
25	72.2410	20.9590	1.2800	2.0354	2.4155
26	74.5720	18.6280	1.2100	1.9887	2.3742
27	76.9030	16.2970	1.1500	1.9487	2.3378
28	79.2340	13.9660	1.1000	1.9153	2.3066
29	81.5650	11.6350	1.0600	1.8886	2.2810
30	83.8960	9.3040	1.0300	1.8686	2.2615

Table A.6: Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 1$

Ppl	*Ie-eff*	R-Rating factor	MOS_{LQE}	MOS_{LQO}	PESQ Score
0	17.0460	76.1540	3.8700	3.7629	3.6570
0.5	18.2115	74.9885	3.8200	3.7295	3.6311
1	19.3770	73.8230	3.7700	3.6962	3.6055
2	21.7080	71.4920	3.6700	3.6295	3.5553
3	24.0390	69.1610	3.5600	3.5561	3.5015
4	26.3700	66.8300	3.4500	3.4828	3.4488
5	28.7010	64.4990	3.3300	3.4027	3.3925
6	31.0320	62.1680	3.2100	3.3227	3.3371
7	33.3630	59.8370	3.0900	3.2426	3.2824
8	35.6940	57.5060	2.9700	3.1626	3.2283
9	38.0250	55.1750	2.8500	3.0826	3.1745
10	40.3560	52.8440	2.7200	2.9958	3.1165
11	42.6870	50.5130	2.6000	2.9158	3.0629
12	45.0180	48.1820	2.4800	2.8358	3.0091
13	47.3490	45.8510	2.3600	2.7557	2.9550
14	49.6800	43.5200	2.2400	2.6757	2.9003
15	52.0110	41.1890	2.1200	2.5956	2.8449
16	54.3420	38.8580	2.0100	2.5223	2.7933
17	56.6730	36.5270	1.9000	2.4489	2.7408
18	59.0040	34.1960	1.7900	2.3755	2.6870
19	61.3350	31.8650	1.6900	2.3088	2.6370
20	63.6660	29.5340	1.5900	2.2421	2.5856
21	65.9970	27.2030	1.5000	2.1821	2.5381
22	68.3280	24.8720	1.4100	2.1221	2.4892
23	70.6590	22.5410	1.3300	2.0687	2.4443
24	72.9900	20.2100	1.2600	2.0220	2.4038
25	75.3210	17.8790	1.1900	1.9753	2.3622
26	77.6520	15.5480	1.1400	1.9420	2.3316
27	79.9830	13.2170	1.0900	1.9086	2.3002
28	82.3140	10.8860	1.0500	1.8820	2.2746
29	84.6450	8.5550	1.0200	1.8619	2.2549
30	86.9760	6.2240	1.0000	1.8486	2.2417

Table A.7: Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 2$

Ppl	***Ie-eff***	R**-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	20.1260	73.0740	3.7400	3.6762	3.5903
0.5	21.2915	71.9085	3.6900	3.6428	3.5652
1	22.4570	70.7430	3.6300	3.6028	3.5356
2	24.7880	68.4120	3.5200	3.5294	3.4822
3	27.1190	66.0810	3.4100	3.4561	3.4299
4	29.4500	63.7500	3.2900	3.3760	3.3739
5	31.7810	61.4190	3.1700	3.2960	3.3188
6	34.1120	59.0880	3.0500	3.2159	3.2643
7	36.4430	56.7570	2.9300	3.1359	3.2103
8	38.7740	54.4260	2.8100	3.0559	3.1566
9	41.1050	52.0950	2.6900	2.9758	3.1031
10	43.4360	49.7640	2.5600	2.8891	3.0450
11	45.7670	47.4330	2.4400	2.8091	2.9911
12	48.0980	45.1020	2.3200	2.7290	2.9368
13	50.4290	42.7710	2.2000	2.6490	2.8819
14	52.7600	40.4400	2.0900	2.5756	2.8309
15	55.0910	38.1090	1.9700	2.4956	2.7743
16	57.4220	35.7780	1.8600	2.4222	2.7214
17	59.7530	33.4470	1.7600	2.3555	2.6721
18	62.0840	31.1160	1.6600	2.2888	2.6217
19	64.4150	28.7850	1.5600	2.2221	2.5699
20	66.7460	26.4540	1.4700	2.1621	2.5220
21	69.0770	24.1230	1.3800	2.1021	2.4725
22	71.4080	21.7920	1.3100	2.0554	2.4329
23	73.7390	19.4610	1.2400	2.0087	2.3921
24	76.0700	17.1300	1.1700	1.9620	2.3500
25	78.4010	14.7990	1.1200	1.9286	2.3191
26	80.7320	12.4680	1.0700	1.8953	2.2875
27	83.0630	10.1370	1.0400	1.8753	2.2681
28	85.3940	7.8060	1.0100	1.8553	2.2483
29	87.7250	5.4750	1.0000	1.8486	2.2417
30	90.0560	3.1440	1.0000	1.8486	2.2417

Table A.8: Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 3$

Ppl	***Ie-eff***	***R*-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	23.2060	69.9940	3.6000	3.5828	3.5209
0.5	24.3715	68.8285	3.5400	3.5428	3.4918
1	25.5370	67.6630	3.4900	3.5094	3.4678
2	27.8680	65.3320	3.3700	3.4294	3.4111
3	30.1990	63.0010	3.2500	3.3494	3.3554
4	32.5300	60.6700	3.1300	3.2693	3.3006
5	34.8610	58.3390	3.0100	3.1893	3.2463
6	37.1920	56.0080	2.8900	3.1092	3.1924
7	39.5230	53.6770	2.7700	3.0292	3.1388
8	41.8540	51.3460	2.6500	2.9491	3.0852
9	44.1850	49.0150	2.5200	2.8624	3.0271
10	46.5160	46.6840	2.4000	2.7824	2.9731
11	48.8470	44.3530	2.2800	2.7024	2.9186
12	51.1780	42.0220	2.1600	2.6223	2.8635
13	53.5090	39.6910	2.0500	2.5490	2.8122
14	55.8400	37.3600	1.9400	2.4756	2.7600
15	58.1710	35.0290	1.8300	2.4022	2.7067
16	60.5020	32.6980	1.7200	2.3288	2.6521
17	62.8330	30.3670	1.6200	2.2621	2.6012
18	65.1640	28.0360	1.5300	2.2021	2.5541
19	67.4950	25.7050	1.4400	2.1421	2.5056
20	69.8260	23.3740	1.3600	2.0887	2.4613
21	72.1570	21.0430	1.2800	2.0354	2.4155
22	74.4880	18.7120	1.2200	1.9953	2.3802
23	76.8190	16.3810	1.1600	1.9553	2.3439
24	79.1500	14.0500	1.1000	1.9153	2.3066
25	81.4810	11.7190	1.0600	1.8886	2.2810
26	83.8120	9.3880	1.0300	1.8686	2.2615
27	86.1430	7.0570	1.0000	1.8486	2.2417
28	88.4740	4.7260	1.0000	1.8486	2.2417
29	90.8050	2.3950	1.0000	1.8486	2.2417
30	93.1360	0.0640	1.0000	1.8486	2.2417

Table A.9: Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 4$

Ppl	***Ie-eff***	***R*-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	26.2860	66.9140	3.4500	3.4828	3.4488
0.5	27.4515	65.7485	3.3900	3.4427	3.4205
1	28.6170	64.5830	3.3300	3.4027	3.3925
2	30.9480	62.2520	3.2200	3.3293	3.3416
3	33.2790	59.9210	3.1000	3.2493	3.2869
4	35.6100	57.5900	2.9700	3.1626	3.2283
5	37.9410	55.2590	2.8500	3.0826	3.1745
6	40.2720	52.9280	2.7300	3.0025	3.1209
7	42.6030	50.5970	2.6100	2.9225	3.0673
8	44.9340	48.2660	2.4800	2.8358	3.0091
9	47.2650	45.9350	2.3600	2.7557	2.9550
10	49.5960	43.6040	2.2400	2.6757	2.9003
11	51.9270	41.2730	2.1300	2.6023	2.8496
12	54.2580	38.9420	2.0100	2.5223	2.7933
13	56.5890	36.6110	1.9000	2.4489	2.7408
14	58.9200	34.2800	1.7900	2.3755	2.6870
15	61.2510	31.9490	1.6900	2.3088	2.6370
16	63.5820	29.6180	1.5900	2.2421	2.5856
17	65.9130	27.2870	1.5000	2.1821	2.5381
18	68.2440	24.9560	1.4100	2.1221	2.4892
19	70.5750	22.6250	1.3300	2.0687	2.4443
20	72.9060	20.2940	1.2600	2.0220	2.4038
21	75.2370	17.9630	1.2000	1.9820	2.3682
22	77.5680	15.6320	1.1400	1.9420	2.3316
23	79.8990	13.3010	1.0900	1.9086	2.3002
24	82.2300	10.9700	1.0500	1.8820	2.2746
25	84.5610	8.6390	1.0200	1.8619	2.2549
26	86.8920	6.3080	1.0000	1.8486	2.2417
27	89.2230	3.9770	1.0000	1.8486	2.2417
28	91.5540	1.6460	1.0000	1.8486	2.2417
29	93.1800	0.0200	1.0000	1.8486	2.2417
30	93.2000	0.0000	1.0000	1.8486	2.2417

Table A.10: Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 5$

Ppl	***Ie-eff***	R**-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	4.3673	88.8327	4.3100	4.0564	3.9059
0.5	6.3514	86.8486	4.2500	4.0164	3.8692
1	8.2934	84.9066	4.2000	3.9830	3.8395
2	12.0544	81.1456	4.0700	3.8963	3.7649
3	15.6564	77.5436	3.9300	3.8029	3.6886
4	19.1058	74.0942	3.7800	3.7029	3.6106
5	22.4090	70.7910	3.6300	3.6028	3.5356
6	25.5722	67.6278	3.4800	3.5028	3.4631
7	28.6018	64.5982	3.3300	3.4027	3.3925
8	31.5040	61.6960	3.1900	3.3093	3.3279
9	34.2853	58.9147	3.0400	3.2093	3.2598
10	36.9519	56.2481	2.9000	3.1159	3.1969
11	39.5102	53.6898	2.7700	3.0292	3.1388
12	41.9664	51.2336	2.6400	2.9425	3.0807
13	44.3268	48.8732	2.5200	2.8624	3.0271
14	46.5979	46.6021	2.4000	2.7824	2.9731
15	48.7859	44.4141	2.2900	2.7090	2.9232
16	50.8971	42.3029	2.1800	2.6357	2.8727
17	52.9379	40.2621	2.0800	2.5690	2.8263
18	54.9145	38.2855	1.9800	2.5023	2.7791
19	56.8333	36.3667	1.8900	2.4422	2.7359
20	58.7007	34.4993	1.8000	2.3822	2.6920
21	60.5228	32.6772	1.7200	2.3288	2.6521
22	62.3061	30.8939	1.6500	2.2821	2.6166
23	64.0568	29.1432	1.5700	2.2288	2.5752
24	65.7813	27.4187	1.5100	2.1888	2.5435
25	67.4859	25.7141	1.4400	2.1421	2.5056
26	69.1769	24.0231	1.3800	2.1021	2.4725
27	70.8607	22.3393	1.3200	2.0620	2.4386
28	72.5434	20.6566	1.2700	2.0287	2.4097
29	74.2316	18.9684	1.2200	1.9953	2.3802
30	75.9314	17.2686	1.18	1.9687	2.3561

Table A.11: Non Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 1$

Ppl	*Ie-eff*	R-Rating factor	MOS_{LQE}	MOS_{LQO}	PESQ Score
0	13.6049	79.5951	4.0100	3.8563	3.7318
0.5	15.5890	77.6110	3.9300	3.8029	3.6886
1	17.5310	75.6690	3.8500	3.7496	3.6466
2	21.2919	71.9081	3.6900	3.6428	3.5652
3	24.8940	68.3060	3.5200	3.5294	3.4822
4	28.3434	64.8566	3.3500	3.4161	3.4018
5	31.6465	61.5535	3.1800	3.3027	3.3233
6	34.8098	58.3902	3.0200	3.1959	3.2508
7	37.8393	55.3607	2.8600	3.0892	3.1790
8	40.7416	52.4584	2.7000	2.9825	3.1075
9	43.5229	49.6771	2.5600	2.8891	3.0450
10	46.1895	47.0105	2.4200	2.7957	2.9821
11	48.7477	44.4523	2.2900	2.7090	2.9232
12	51.2040	41.9960	2.1600	2.6223	2.8635
13	53.5644	39.6356	2.0500	2.5490	2.8122
14	55.8355	37.3645	1.9400	2.4756	2.7600
15	58.0235	35.1765	1.8300	2.4022	2.7067
16	60.1347	33.0653	1.7400	2.3422	2.6622
17	62.1755	31.0245	1.6500	2.2821	2.6166
18	64.1521	29.0479	1.5700	2.2288	2.5752
19	66.0709	27.1291	1.4900	2.1754	2.5327
20	67.9383	25.2617	1.4300	2.1354	2.5002
21	69.7604	23.4396	1.3600	2.0887	2.4613
22	71.5437	21.6563	1.3000	2.0487	2.4271
23	73.2944	19.9056	1.2500	2.0153	2.3980
24	75.0189	18.1811	1.2000	1.9820	2.3682
25	76.7235	16.4765	1.1600	1.9553	2.3439
26	78.4145	14.7855	1.1200	1.9286	2.3191
27	80.0982	13.1018	1.0800	1.9020	2.2939
28	81.7810	11.4190	1.0600	1.8886	2.2810
29	83.4692	9.7308	1.0300	1.8686	2.2615
30	85.1690	8.0310	1.0100	1.8553	2.2483

Table A.12: Non Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 2$

Ppl	***Ie-eff***	**R-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	16.6095	76.5905	3.8900	3.7762	3.6675
0.5	18.5936	74.6064	3.8000	3.7162	3.6208
1	20.5356	72.6644	3.7200	3.6628	3.5802
2	24.2966	68.9034	3.5500	3.5495	3.4966
3	27.8986	65.3014	3.3700	3.4294	3.4111
4	31.3480	61.8520	3.2000	3.3160	3.3325
5	34.6512	58.5488	3.0200	3.1959	3.2508
6	37.8144	55.3856	2.8600	3.0892	3.1790
7	40.8440	52.3560	2.7000	2.9825	3.1075
8	43.7462	49.4538	2.5500	2.8825	3.0405
9	46.5275	46.6725	2.4000	2.7824	2.9731
10	49.1941	44.0059	2.2600	2.6890	2.9095
11	51.7524	41.4476	2.1400	2.6090	2.8542
12	54.2086	38.9914	2.0100	2.5223	2.7933
13	56.5690	36.6310	1.9000	2.4489	2.7408
14	58.8401	34.3599	1.8000	2.3822	2.6920
15	61.0281	32.1719	1.7000	2.3155	2.6421
16	63.1393	30.0607	1.6100	2.2555	2.5960
17	65.1801	28.0199	1.5300	2.2021	2.5541
18	67.1567	26.0433	1.4500	2.1488	2.5111
19	69.0756	24.1244	1.3800	2.1021	2.4725
20	70.9429	22.2571	1.3200	2.0620	2.4386
21	72.7650	20.4350	1.2600	2.0220	2.4038
22	74.5483	18.6517	1.2100	1.9887	2.3742
23	76.2990	16.9010	1.1700	1.9620	2.3500
24	78.0235	15.1765	1.1300	1.9353	2.3254
25	79.7281	13.4719	1.0900	1.9086	2.3002
26	81.4191	11.7809	1.0600	1.8886	2.2810
27	83.1029	10.0971	1.0400	1.8753	2.2681
28	84.7856	8.4144	1.0200	1.8619	2.2549
29	86.4738	6.7262	1.0000	1.8486	2.2417
30	88.1736	5.0264	1.0000	1.8486	2.2417

Table A.13: Non Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 3$

Ppl	***Ie-eff***	***R*-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	16.9274	76.2726	3.8800	3.7696	3.6622
0.5	18.9115	74.2885	3.7900	3.7095	3.6157
1	20.8535	72.3465	3.7100	3.6562	3.5752
2	24.6144	68.5856	3.5300	3.5361	3.4870
3	28.2164	64.9836	3.3500	3.4161	3.4018
4	31.6659	61.5341	3.1800	3.3027	3.3233
5	34.9690	58.2310	3.0100	3.1893	3.2463
6	38.1322	55.0678	2.8400	3.0759	3.1700
7	41.1618	52.0382	2.6800	2.9692	3.0986
8	44.0641	49.1359	2.5300	2.8691	3.0315
9	46.8454	46.3546	2.3800	2.7691	2.9640
10	49.5120	43.6880	2.2500	2.6824	2.9049
11	52.0702	41.1298	2.1200	2.5956	2.8449
12	54.5264	38.6736	2.0000	2.5156	2.7886
13	56.8869	36.3131	1.8900	2.4422	2.7359
14	59.1580	34.0420	1.7800	2.3689	2.6821
15	61.3460	31.8540	1.6900	2.3088	2.6370
16	63.4572	29.7428	1.6000	2.2488	2.5908
17	65.4980	27.7020	1.5200	2.1954	2.5488
18	67.4746	25.7254	1.4400	2.1421	2.5056
19	69.3934	23.8066	1.3700	2.0954	2.4669
20	71.2607	21.9393	1.3100	2.0554	2.4329
21	73.0829	20.1171	1.2600	2.0220	2.4038
22	74.8661	18.3339	1.2000	1.9820	2.3682
23	76.6169	16.5831	1.1600	1.9553	2.3439
24	78.3414	14.8586	1.1200	1.9286	2.3191
25	80.0460	13.1540	1.0900	1.9086	2.3002
26	81.7370	11.4630	1.0600	1.8886	2.2810
27	83.4207	9.7793	1.0300	1.8686	2.2615
28	85.1035	8.0965	1.0100	1.8553	2.2483
29	86.7916	6.4084	1.0000	1.8486	2.2417
30	88.4914	4.7086	1.0000	1.8486	2.2417

Table A.14: Non Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 4$

Ppl	***Ie-eff***	R**-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	18.1047	75.0953	3.8300	3.7362	3.6362
0.5	20.0888	73.1112	3.7400	3.6762	3.5903
1	22.0308	71.1692	3.6500	3.6162	3.5454
2	25.7917	67.4083	3.4700	3.4961	3.4583
3	29.3937	63.8063	3.2900	3.3760	3.3739
4	32.8432	60.3568	3.1200	3.2626	3.2960
5	36.1463	57.0537	2.9500	3.1493	3.2193
6	39.3095	53.8905	2.7800	3.0359	3.1432
7	42.3391	50.8609	2.6200	2.9291	3.0718
8	45.2414	47.9586	2.4700	2.8291	3.0046
9	48.0227	45.1773	2.3200	2.7290	2.9368
10	50.6893	42.5107	2.1900	2.6423	2.8773
11	53.2475	39.9525	2.0600	2.5556	2.8169
12	55.7037	37.4963	1.9400	2.4756	2.7600
13	58.0642	35.1358	1.8300	2.4022	2.7067
14	60.3353	32.8647	1.7300	2.3355	2.6572
15	62.5233	30.6767	1.6400	2.2755	2.6115
16	64.6345	28.5655	1.5500	2.2155	2.5647
17	66.6753	26.5247	1.4700	2.1621	2.5220
18	68.6519	24.5481	1.4000	2.1154	2.4836
19	70.5707	22.6293	1.3300	2.0687	2.4443
20	72.4380	20.7620	1.2700	2.0287	2.4097
21	74.2602	18.9398	1.2200	1.9953	2.3802
22	76.0435	17.1565	1.1700	1.9620	2.3500
23	77.7942	15.4058	1.1300	1.9353	2.3254
24	79.5187	13.6813	1.1000	1.9153	2.3066
25	81.2233	11.9767	1.0600	1.8886	2.2810
26	82.9143	10.2857	1.0400	1.8753	2.2681
27	84.5980	8.6020	1.0200	1.8619	2.2549
28	86.2808	6.9192	1.0000	1.8486	2.2417
29	87.9689	5.2311	1.0000	1.8486	2.2417
30	89.6688	3.5312	1.0000	1.8486	2.2417

Table A.15: Non Linear Regression for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 5$

Appendix B

Extending The E-model Based on PESQ-Detailed Results ANN

In this appendix the detailed results of performance of the Artificial Neural Network constructed to predict *Ie-eff* from *Ppl* and *Burst* as discussed in chapter 6 are listed.

Tables B.1-B.20 show the full results of training set performance and test set performance for the tested networks ranging from simple networks with 1 neuron in the hidden layer to large networks with 40 neurons in the hidden layer.

Tables B.21-B.25 show the full results of *Ie-eff*, *R*-Rating Factor, MOS Scores, and PESQ scores values and for each combination of *Ppl* and *BurstR* as calculated from the neural network model derived in chapter 6. Each table is for a specific Burst Ratio. The *R*-Rating Factor was calculated from *Ie-eff* using equation 5.12. The MOS score was calculated from *R*-Rating Factor using equation 3.9. Finally the PESQ score was calculated from the MOS score using equation 3.3.

1 Hidden Neurons		2 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.04330	0.04290	0.00670	0.01110
0.04280	0.04350	0.03800	0.04940
0.04360	0.04260	0.00670	0.01110
0.04330	0.04280	0.00670	0.01120
0.04270	0.04380	0.01040	0.01110
0.04340	0.04270	0.04140	0.04330
0.04330	0.04280	0.02700	0.03160
0.04310	0.04300	0.01040	0.01110
0.04300	0.04320	0.02810	0.03140
0.04340	0.04280	0.02730	0.03070
0.04300	0.04320	0.02730	0.03070
0.04270	0.04370	0.00670	0.01110
0.04340	0.04270	0.01040	0.01110
0.04270	0.04380	0.03880	0.03780
0.04340	0.04270	0.00690	0.01080
0.04320	0.04300	0.01060	0.01120
0.04300	0.04330	0.01040	0.01110
0.04340	0.04270	0.01040	0.01110
0.04270	0.04370	0.03810	0.03780
0.04350	0.04260	0.01040	0.01110
0.04270	0.04380	0.00670	0.01110
0.04270	0.04370	0.01050	0.01110
0.04320	0.04300	0.00670	0.01110
0.04290	0.04340	0.02670	0.03000
0.04330	0.04280	0.01040	0.01110
0.04300	0.04330	0.01040	0.01110
0.04280	0.04370	0.01050	0.01120
0.04270	0.04370	0.01040	0.01110
0.04340	0.04270	0.02660	0.02940
0.04330	0.04280	0.02670	0.03050

Table B.1: Training and Testing MSE for 1 and 2 hidden neurons

3 Hidden Neurons		4 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00570	0.01720	0.00640	0.01950
0.00640	0.01210	0.00480	0.00970
0.00610	0.01040	0.00390	0.01440
0.00910	0.00850	0.00440	0.00900
0.00920	0.01050	0.00370	0.01100
0.00960	0.01040	0.00370	0.01090
0.00450	0.01210	0.00420	0.01580
0.00560	0.01590	0.00600	0.01460
0.00700	0.00850	0.00430	0.01560
0.00470	0.01020	0.00390	0.01120
0.00550	0.00870	0.00470	0.01040
0.00460	0.01100	0.00610	0.00550
0.00480	0.00950	0.00540	0.01340
0.00450	0.01200	0.00380	0.01110
0.00450	0.01250	0.00370	0.00740
0.00940	0.01060	0.00430	0.01340
0.00490	0.01680	0.00560	0.00960
0.00900	0.01060	0.00440	0.00820
0.00700	0.00850	0.00350	0.00850
0.00470	0.01030	0.00370	0.01100
0.00450	0.01230	0.00430	0.01280
0.00490	0.01680	0.00410	0.01130
0.01020	0.01710	0.00410	0.00680
0.00550	0.00870	0.00430	0.01640
0.00580	0.01680	0.00550	0.00900
0.00700	0.00850	0.00430	0.01620
0.00480	0.01700	0.00360	0.00970
0.00590	0.01060	0.00430	0.01640
0.00650	0.00920	0.00380	0.01110
0.00590	0.01720	0.00510	0.00950

Table B.2: Training and Testing MSE for 3 and 4 hidden neurons

5 Hidden Neurons		6 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00340	0.01200	0.00310	0.00510
0.00450	0.01640	0.00380	0.01290
0.00370	0.00960	0.00320	0.00730
0.00370	0.00970	0.00280	0.00880
0.00390	0.01240	0.00340	0.00860
0.00350	0.00730	0.00380	0.00920
0.00310	0.00650	0.00260	0.00840
0.00320	0.00740	0.00290	0.00700
0.00380	0.01060	0.00340	0.00940
0.00330	0.01090	0.00280	0.00540
0.00340	0.00840	0.00340	0.01160
0.00330	0.00750	0.00380	0.01160
0.00340	0.01160	0.00480	0.00930
0.00360	0.00850	0.00290	0.00510
0.00390	0.01210	0.00350	0.01270
0.00440	0.00860	0.00270	0.00580
0.00380	0.01220	0.00310	0.01040
0.00330	0.00450	0.00330	0.01140
0.00590	0.00900	0.00290	0.00650
0.00430	0.00730	0.00360	0.00750
0.00310	0.01340	0.00310	0.00720
0.00370	0.01100	0.00300	0.00610
0.00320	0.00780	0.00310	0.00630
0.00420	0.01060	0.00300	0.00970
0.00430	0.01480	0.00330	0.01060
0.00330	0.00750	0.00370	0.00710
0.00310	0.00550	0.00340	0.01200
0.00330	0.00700	0.00300	0.00640
0.00340	0.01000	0.00290	0.00750
0.00430	0.01480	0.00300	0.00540

Table B.3: Training and Testing MSE for 5 and 6 hidden neurons

7 Hidden Neurons		8 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00320	0.01120	0.00280	0.00800
0.00310	0.00830	0.00220	0.00930
0.00360	0.01400	0.00280	0.00750
0.00300	0.00820	0.00260	0.00960
0.00290	0.01000	0.00340	0.01380
0.00310	0.00720	0.00320	0.00800
0.00380	0.00810	0.00250	0.01460
0.00300	0.00620	0.00280	0.00970
0.00230	0.01190	0.00380	0.00660
0.00360	0.01350	0.00280	0.01160
0.00280	0.00700	0.00210	0.00970
0.00270	0.00840	0.00330	0.00980
0.00370	0.00950	0.00300	0.01050
0.00320	0.00830	0.00260	0.01230
0.00330	0.01190	0.00290	0.00870
0.00230	0.01710	0.00350	0.00940
0.00300	0.01220	0.00360	0.01080
0.00300	0.00780	0.00300	0.00750
0.00290	0.00910	0.00290	0.00990
0.00350	0.01390	0.00240	0.00600
0.00330	0.01130	0.00260	0.00620
0.00330	0.01320	0.00290	0.00750
0.00250	0.00950	0.00290	0.01090
0.00270	0.00560	0.00250	0.01320
0.00290	0.00630	0.00290	0.00670
0.00340	0.01240	0.00280	0.00570
0.00330	0.00720	0.00280	0.00590
0.00280	0.00550	0.00230	0.00790
0.00310	0.00920	0.00270	0.00690
0.00310	0.00800	0.00290	0.00640

Table B.4: Training and Testing MSE for 7 and 8 hidden neurons

9 Hidden Neurons		10 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00230	0.00760	0.00290	0.00770
0.00280	0.00580	0.00300	0.00890
0.00270	0.00620	0.00200	0.01300
0.00320	0.01260	0.00210	0.00840
0.00280	0.00980	0.00270	0.00890
0.00210	0.00700	0.00260	0.01500
0.00270	0.00950	0.00260	0.01230
0.00240	0.00670	0.00230	0.01240
0.00270	0.00550	0.00250	0.01970
0.00230	0.00740	0.00220	0.00710
0.00290	0.00720	0.00250	0.01270
0.00270	0.00660	0.00290	0.00630
0.00230	0.00670	0.00370	0.00720
0.00340	0.01100	0.00230	0.01060
0.00200	0.01420	0.00210	0.01000
0.00220	0.00930	0.00230	0.00970
0.00230	0.00660	0.00200	0.01810
0.00280	0.00570	0.00220	0.01200
0.00210	0.00970	0.00250	0.01160
0.00260	0.00610	0.00200	0.00940
0.00260	0.00650	0.00260	0.00730
0.00340	0.01030	0.00200	0.01060
0.00310	0.00670	0.00210	0.00760
0.00290	0.00600	0.00230	0.00760
0.00280	0.00730	0.00330	0.00710
0.00230	0.00730	0.00230	0.00660
0.00300	0.01250	0.00200	0.00800
0.00310	0.00800	0.00280	0.00840
0.00270	0.01260	0.00280	0.00660
0.00310	0.01140	0.00250	0.00710

Table B.5: Training and Testing MSE for 9 and 10 hidden neurons

11 Hidden Neurons		12 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00240	0.01160	0.00200	0.00800
0.00260	0.00860	0.00230	0.01000
0.00230	0.00910	0.00200	0.00770
0.00180	0.01010	0.00190	0.00940
0.00190	0.00830	0.00200	0.00500
0.00270	0.01050	0.00200	0.01020
0.00180	0.01510	0.00200	0.00920
0.00170	0.00870	0.00230	0.00870
0.00210	0.00970	0.00180	0.00920
0.00230	0.00670	0.00210	0.01030
0.00210	0.00950	0.00180	0.01090
0.00310	0.00810	0.00240	0.00990
0.00210	0.00550	0.00230	0.00800
0.00200	0.01590	0.00260	0.01310
0.00220	0.00740	0.00210	0.00920
0.00230	0.00580	0.00180	0.00740
0.00270	0.01000	0.00190	0.00900
0.00210	0.01490	0.00220	0.00770
0.00310	0.01410	0.00210	0.00700
0.00250	0.00860	0.00350	0.00880
0.00210	0.00860	0.00200	0.01570
0.00220	0.01130	0.00200	0.00940
0.00270	0.01050	0.00190	0.01070
0.00310	0.01310	0.00250	0.00650
0.00290	0.00710	0.00190	0.00660
0.00250	0.00950	0.00220	0.00810
0.00220	0.00780	0.00180	0.01080
0.00210	0.00960	0.00170	0.01040
0.00230	0.00950	0.00210	0.00740
0.00240	0.01490	0.00260	0.00670

Table B.6: Training and Testing MSE for 11 and 12 hidden neurons

13 Hidden Neurons		14 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00210	0.00860	0.00290	0.01130
0.00220	0.01080	0.00240	0.00740
0.00230	0.00860	0.00190	0.00760
0.00200	0.00880	0.00210	0.00810
0.00200	0.00480	0.00170	0.01040
0.00230	0.00760	0.00200	0.01180
0.00170	0.01000	0.00160	0.00990
0.00200	0.01000	0.00270	0.00920
0.00280	0.00840	0.00170	0.00900
0.00170	0.00790	0.00180	0.01600
0.00200	0.01840	0.00210	0.01200
0.00270	0.00720	0.00200	0.00650
0.00170	0.00760	0.00160	0.02350
0.00180	0.00930	0.00200	0.01310
0.00180	0.00850	0.00160	0.01630
0.00180	0.00890	0.00270	0.01040
0.00240	0.00720	0.00220	0.01120
0.00190	0.01170	0.00170	0.01230
0.00200	0.01510	0.00160	0.01230
0.00210	0.00860	0.00210	0.00850
0.00170	0.00990	0.00230	0.01460
0.00220	0.01380	0.00170	0.01190
0.00190	0.01180	0.00190	0.01010
0.00230	0.01020	0.00220	0.00780
0.00200	0.01100	0.00160	0.01130
0.00230	0.00750	0.00170	0.01200
0.00190	0.01090	0.00200	0.01160
0.00160	0.01020	0.00230	0.01430
0.00220	0.00960	0.00230	0.00930
0.00210	0.01110	0.00190	0.01820

Table B.7: Training and Testing MSE for 13 and 14 hidden neurons

15 Hidden Neurons		16 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00180	0.00890	0.00150	0.01190
0.00190	0.01110	0.00180	0.01090
0.00150	0.00980	0.00210	0.01190
0.00140	0.00700	0.00170	0.01110
0.00200	0.01110	0.00170	0.00660
0.00150	0.01360	0.00180	0.01240
0.00240	0.00660	0.00170	0.01220
0.00190	0.01270	0.00210	0.01310
0.00170	0.01320	0.00160	0.01200
0.00210	0.01220	0.00240	0.01420
0.00210	0.01120	0.00220	0.01090
0.00180	0.01550	0.00180	0.02620
0.00160	0.01230	0.00190	0.00890
0.00220	0.00800	0.00200	0.01490
0.00200	0.01210	0.00190	0.00890
0.00190	0.01090	0.00180	0.00940
0.00150	0.01140	0.00170	0.01060
0.00180	0.01090	0.00160	0.01510
0.00200	0.01690	0.00220	0.01070
0.00180	0.00870	0.00240	0.00860
0.00200	0.01060	0.00230	0.00940
0.00150	0.01000	0.00170	0.01370
0.00180	0.00800	0.00160	0.01770
0.00190	0.01360	0.00150	0.01050
0.00220	0.01130	0.00190	0.01090
0.00170	0.01420	0.00190	0.01070
0.00240	0.01050	0.00230	0.00850
0.00180	0.01160	0.00210	0.01040
0.00160	0.01550	0.00170	0.00960
0.00190	0.00730	0.00220	0.00830

Table B.8: Training and Testing MSE for 15 and 16 hidden neurons

17 Hidden Neurons			18 Hidden Neurons	
Training MSE	**Testing MSE**		**Training MSE**	**Testing MSE**
0.00140	0.01350		0.00190	0.01520
0.00170	0.01460		0.00230	0.01640
0.00170	0.01430		0.00230	0.01260
0.00150	0.01120		0.00150	0.01330
0.00180	0.01230		0.00120	0.01490
0.00180	0.01400		0.00140	0.01030
0.00230	0.00910		0.00210	0.01260
0.00180	0.01190		0.00160	0.00840
0.00220	0.01210		0.00220	0.01150
0.00190	0.01120		0.00160	0.00960
0.00180	0.01000		0.00130	0.01480
0.00180	0.01190		0.00180	0.00860
0.00180	0.00910		0.00170	0.01290
0.00160	0.01140		0.00190	0.01490
0.00160	0.02470		0.00160	0.01090
0.00170	0.00940		0.00180	0.01370
0.00200	0.01660		0.00190	0.00700
0.00210	0.01230		0.00170	0.01160
0.00220	0.00790		0.00180	0.00840
0.00170	0.01430		0.00200	0.01090
0.00180	0.01130		0.00180	0.01100
0.00140	0.01230		0.00150	0.01450
0.00170	0.01130		0.00160	0.01220
0.00150	0.01260		0.00140	0.01190
0.00150	0.01850		0.00210	0.01610
0.00160	0.00840		0.00160	0.01380
0.00130	0.01540		0.00140	0.01170
0.00180	0.01030		0.00210	0.01210
0.00180	0.01440		0.00160	0.01190
0.00170	0.00940		0.00170	0.01910

Table B.9: Training and Testing MSE for 17 and 18 hidden neurons

19 Hidden Neurons		20 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00130	0.02860	0.00110	0.01240
0.00160	0.01480	0.00170	0.01530
0.00190	0.00950	0.00150	0.01150
0.00190	0.01170	0.00170	0.01070
0.00170	0.00890	0.00150	0.01260
0.00150	0.01300	0.00190	0.00940
0.00180	0.00790	0.00150	0.01170
0.00180	0.01220	0.00150	0.00930
0.00170	0.01150	0.00130	0.00980
0.00160	0.01190	0.00150	0.01500
0.00190	0.01860	0.00170	0.01810
0.00180	0.00880	0.00140	0.01820
0.00130	0.02010	0.00190	0.00900
0.00170	0.00730	0.00160	0.01350
0.00230	0.01310	0.00150	0.01360
0.00150	0.01870	0.00210	0.01120
0.00160	0.01340	0.00210	0.01240
0.00160	0.01380	0.00180	0.01340
0.00190	0.01370	0.00190	0.00990
0.00190	0.00800	0.00200	0.01260
0.00140	0.01040	0.00150	0.01220
0.00150	0.01040	0.00190	0.01260
0.00150	0.00910	0.00140	0.01820
0.00190	0.00950	0.00160	0.01420
0.00150	0.01190	0.00180	0.01420
0.00160	0.01280	0.00170	0.01000
0.00180	0.00900	0.00160	0.01180
0.00200	0.01200	0.00110	0.02130
0.00170	0.01180	0.00150	0.01480
0.00150	0.02030	0.00140	0.01040

Table B.10: Training and Testing MSE for 19 and 20 hidden neurons

21 Hidden Neurons		22 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00170	0.01420	0.00140	0.01640
0.00130	0.01130	0.00140	0.01400
0.00140	0.01560	0.00130	0.01000
0.00150	0.00970	0.00170	0.01030
0.00150	0.00950	0.00140	0.01480
0.00170	0.01020	0.00140	0.01420
0.00160	0.02410	0.00170	0.00970
0.00140	0.01360	0.00150	0.01310
0.00130	0.01200	0.00150	0.01520
0.00120	0.01080	0.00150	0.01700
0.00140	0.01330	0.00140	0.01770
0.00130	0.01100	0.00150	0.00960
0.00140	0.01570	0.00170	0.01200
0.00170	0.01290	0.00170	0.01140
0.00130	0.02120	0.00180	0.01210
0.00140	0.01120	0.00110	0.01420
0.00200	0.00740	0.00220	0.01720
0.00140	0.01260	0.00130	0.02110
0.00120	0.02110	0.00150	0.02510
0.00130	0.01530	0.00130	0.01330
0.00150	0.01190	0.00150	0.01150
0.00120	0.01050	0.00200	0.01430
0.00170	0.01240	0.00120	0.01210
0.00200	0.01160	0.00150	0.01100
0.00120	0.01980	0.00130	0.01170
0.00150	0.01060	0.00140	0.01620
0.00130	0.01050	0.00120	0.01190
0.00130	0.01240	0.00170	0.01260
0.00140	0.01470	0.00140	0.01630
0.00170	0.01310	0.00180	0.00820

Table B.11: Training and Testing MSE for 21 and 22 hidden neurons

23 Hidden Neurons		24 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00100	0.01520	0.00110	0.02520
0.00190	0.01450	0.00140	0.02720
0.00170	0.01190	0.00130	0.01080
0.00200	0.01340	0.00140	0.01240
0.00090	0.03020	0.00140	0.01730
0.00200	0.00720	0.00120	0.01630
0.00170	0.02300	0.00160	0.00940
0.00170	0.01310	0.00190	0.00970
0.00140	0.00980	0.00120	0.01310
0.00130	0.01210	0.00150	0.01240
0.00150	0.01450	0.00100	0.01180
0.00160	0.01060	0.00110	0.01880
0.00140	0.01710	0.00120	0.02580
0.00200	0.01550	0.00150	0.01750
0.00140	0.01270	0.00120	0.02220
0.00150	0.01100	0.00140	0.01340
0.00130	0.01170	0.00130	0.01660
0.00130	0.01460	0.00160	0.01190
0.00140	0.01100	0.00100	0.01510
0.00180	0.01570	0.00150	0.01590
0.00160	0.01140	0.00120	0.01370
0.00160	0.01800	0.00190	0.01680
0.00140	0.01120	0.00150	0.01420
0.00100	0.01700	0.00110	0.01630
0.00190	0.01410	0.00120	0.01460
0.00120	0.01030	0.00140	0.00990
0.00130	0.01340	0.00140	0.01270
0.00210	0.01060	0.00150	0.01110
0.00160	0.01390	0.00150	0.00990
0.00130	0.01980	0.00160	0.01020

Table B.12: Training and Testing MSE for 23 and 24 hidden neurons

25 Hidden Neurons		26 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00120	0.01190	0.00150	0.00940
0.00130	0.01780	0.00100	0.01420
0.00120	0.01420	0.00130	0.01250
0.00140	0.02250	0.00140	0.01260
0.00120	0.01030	0.00090	0.02450
0.00130	0.01360	0.00140	0.01570
0.00110	0.01500	0.00160	0.01240
0.00140	0.01010	0.00180	0.01220
0.00150	0.00870	0.00110	0.01610
0.00140	0.02080	0.00180	0.01750
0.00140	0.01260	0.00150	0.01100
0.00110	0.01740	0.00180	0.01400
0.00130	0.01320	0.00130	0.01040
0.00160	0.00980	0.00170	0.00930
0.00170	0.01480	0.00100	0.01410
0.00120	0.02730	0.00140	0.01660
0.00120	0.01390	0.00140	0.02720
0.00230	0.00980	0.00110	0.01330
0.00160	0.00970	0.00240	0.02430
0.00120	0.01610	0.00140	0.02630
0.00130	0.01450	0.00140	0.01130
0.00250	0.01150	0.00200	0.01290
0.00160	0.02020	0.00150	0.01610
0.00180	0.01400	0.00120	0.02430
0.00120	0.01310	0.00120	0.01430
0.00180	0.01540	0.00100	0.01600
0.00110	0.02650	0.00130	0.01350
0.00100	0.01130	0.00140	0.02350
0.00110	0.01180	0.00180	0.01560
0.00180	0.00940	0.00120	0.01150

Table B.13: Training and Testing MSE for 25 and 26 hidden neurons

27 Hidden Neurons			28 Hidden Neurons	
Training MSE	**Testing MSE**		**Training MSE**	**Testing MSE**
0.00120	0.01340		0.00120	0.01940
0.00110	0.01420		0.00110	0.01700
0.00110	0.01640		0.00100	0.01800
0.00110	0.01780		0.00200	0.01300
0.00130	0.01350		0.00120	0.01540
0.00070	0.01900		0.00140	0.01200
0.00130	0.01790		0.00160	0.00880
0.00110	0.01100		0.00120	0.01420
0.00070	0.04880		0.00120	0.02150
0.00130	0.02050		0.00120	0.01240
0.00180	0.01170		0.00150	0.01230
0.00140	0.01400		0.00140	0.01090
0.00140	0.01040		0.00140	0.01950
0.00090	0.01540		0.00140	0.01170
0.00150	0.01810		0.00110	0.01540
0.00100	0.02120		0.00150	0.01110
0.00130	0.01390		0.00140	0.01840
0.00140	0.01540		0.00090	0.02200
0.00140	0.01180		0.00150	0.01930
0.00120	0.01660		0.00120	0.01500
0.00100	0.01500		0.00140	0.01600
0.00120	0.01610		0.00130	0.02480
0.00130	0.01420		0.00160	0.01300
0.00140	0.01290		0.00100	0.02120
0.00120	0.01490		0.00120	0.01500
0.00140	0.01320		0.00120	0.01180
0.00110	0.01280		0.00150	0.01570
0.00120	0.01240		0.00160	0.01700
0.00120	0.01260		0.00120	0.01820
0.00110	0.02350		0.00150	0.01500

Table B.14: Training and Testing MSE for 27 and 28 hidden neurons

29 Hidden Neurons		30 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00130	0.01120	0.00180	0.01780
0.00170	0.01020	0.00150	0.01430
0.00140	0.01310	0.00120	0.01230
0.00130	0.02150	0.00130	0.01050
0.00120	0.01410	0.00120	0.01400
0.00100	0.01250	0.00100	0.01760
0.00130	0.02060	0.00130	0.01650
0.00120	0.01100	0.00100	0.01410
0.00110	0.01840	0.00150	0.01700
0.00110	0.01180	0.00110	0.01720
0.00110	0.02080	0.00150	0.01290
0.00120	0.01390	0.00130	0.01210
0.00140	0.01820	0.00250	0.01570
0.00140	0.02520	0.00150	0.01320
0.00130	0.00990	0.00110	0.01430
0.00160	0.01740	0.00090	0.01610
0.00110	0.01140	0.00130	0.01290
0.00180	0.02750	0.00180	0.01640
0.00160	0.01160	0.00130	0.01240
0.00120	0.01310	0.00120	0.01220
0.00120	0.01370	0.00130	0.01500
0.00120	0.01450	0.00170	0.01600
0.00120	0.01500	0.00190	0.01200
0.00090	0.03150	0.00180	0.01250
0.00140	0.01010	0.00130	0.01280
0.00140	0.00950	0.00110	0.01330
0.00150	0.00930	0.00130	0.01750
0.00140	0.01120	0.00140	0.01280
0.00130	0.01750	0.00110	0.01670
0.00140	0.01170	0.00070	0.01700

Table B.15: Training and Testing MSE for 29 and 30 hidden neurons

31 Hidden Neurons		32 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00150	0.01780	0.00120	0.01660
0.00180	0.01430	0.00130	0.01830
0.00120	0.01230	0.00120	0.01190
0.00110	0.01050	0.00120	0.02270
0.00100	0.01400	0.00120	0.00890
0.00080	0.01760	0.00180	0.01370
0.00120	0.01650	0.00140	0.01320
0.00110	0.01410	0.00130	0.01440
0.00100	0.01700	0.00130	0.01420
0.00110	0.01720	0.00110	0.01480
0.00110	0.01290	0.00130	0.01120
0.00090	0.01210	0.00120	0.01350
0.00160	0.01570	0.00100	0.01660
0.00120	0.01320	0.00160	0.01560
0.00130	0.01430	0.00110	0.01850
0.00100	0.01610	0.00150	0.01810
0.00100	0.01290	0.00140	0.01040
0.00110	0.01640	0.00160	0.01120
0.00130	0.01240	0.00120	0.01290
0.00100	0.01220	0.00080	0.01990
0.00100	0.01500	0.00110	0.01480
0.00110	0.01600	0.00130	0.01290
0.00140	0.01200	0.00080	0.02780
0.00160	0.01250	0.00110	0.01170
0.00110	0.01280	0.00150	0.01370
0.00110	0.01330	0.00110	0.01170
0.00110	0.01750	0.00160	0.01070
0.00140	0.01280	0.00150	0.02360
0.00090	0.01670	0.00110	0.01660
0.00140	0.01700	0.00090	0.02150

Table B.16: Training and Testing MSE for 31 and 32 hidden neurons

33 Hidden Neurons			34 Hidden Neurons	
Training MSE	**Testing MSE**		**Training MSE**	**Testing MSE**
0.00140	0.02330		0.00120	0.01670
0.00170	0.01210		0.00140	0.01980
0.00120	0.02350		0.00120	0.02630
0.00140	0.01020		0.00120	0.01380
0.00100	0.01450		0.00080	0.01920
0.00150	0.01380		0.00110	0.01370
0.00120	0.01070		0.00100	0.01440
0.00090	0.01820		0.00070	0.03580
0.00140	0.01580		0.00130	0.01450
0.00100	0.01440		0.00080	0.01640
0.00140	0.01950		0.00170	0.01660
0.00090	0.01480		0.00130	0.01160
0.00150	0.01260		0.00090	0.01440
0.00120	0.01150		0.00120	0.00930
0.00110	0.01420		0.00150	0.01000
0.00110	0.01940		0.00130	0.01620
0.00120	0.01630		0.00220	0.01220
0.00110	0.01340		0.00090	0.02240
0.00110	0.01190		0.00140	0.01510
0.00070	0.04100		0.00120	0.02610
0.00140	0.01430		0.00130	0.01450
0.00100	0.01560		0.00140	0.01470
0.00130	0.01610		0.00100	0.01620
0.00090	0.04940		0.00100	0.03980
0.00090	0.01520		0.00120	0.01220
0.00100	0.02720		0.00170	0.01540
0.00130	0.01910		0.00100	0.01790
0.00140	0.01150		0.00110	0.02370
0.00110	0.02660		0.00090	0.02700
0.00090	0.01280		0.00140	0.01180

Table B.17: Training and Testing MSE for 33 and 34 hidden neurons

35 Hidden Neurons		36 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00120	0.01090	0.00120	0.01850
0.00110	0.02770	0.00100	0.01870
0.00090	0.02100	0.00090	0.01860
0.00120	0.01450	0.00090	0.01360
0.00080	0.01970	0.00070	0.01730
0.00090	0.01540	0.00120	0.01670
0.00090	0.01660	0.00060	0.02170
0.00080	0.01440	0.00100	0.01640
0.00110	0.01840	0.00110	0.01480
0.00150	0.01200	0.00130	0.01710
0.00090	0.01060	0.00070	0.01900
0.00140	0.02230	0.00120	0.01380
0.00160	0.01460	0.00100	0.02130
0.00110	0.02400	0.00110	0.02350
0.00140	0.01700	0.00080	0.03310
0.00070	0.02180	0.00090	0.01680
0.00130	0.01040	0.00060	0.01730
0.00070	0.01390	0.00090	0.01380
0.00110	0.02070	0.00120	0.01370
0.00110	0.01960	0.00100	0.02400
0.00120	0.01380	0.00120	0.01580
0.00090	0.02480	0.00070	0.01610
0.00120	0.01610	0.00140	0.02140
0.00120	0.01520	0.00130	0.02240
0.00130	0.01600	0.00070	0.03680
0.00090	0.01660	0.00120	0.01540
0.00130	0.01560	0.00110	0.01150
0.00090	0.02340	0.00110	0.01340
0.00150	0.01560	0.00130	0.01270
0.00120	0.01270	0.00090	0.01520

Table B.18: Training and Testing MSE for 35 and 36 hidden neurons

37 Hidden Neurons		38 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00110	0.01330	0.00110	0.01810
0.00120	0.01850	0.00080	0.02050
0.00070	0.01460	0.00090	0.02330
0.00110	0.02320	0.00120	0.01310
0.00110	0.01890	0.00080	0.01850
0.00100	0.01250	0.00100	0.01560
0.00120	0.02010	0.00080	0.02220
0.00100	0.01870	0.00110	0.01710
0.00100	0.01320	0.00100	0.01500
0.00130	0.01030	0.00080	0.01880
0.00110	0.01920	0.00110	0.01210
0.00110	0.01500	0.00100	0.01300
0.00100	0.01270	0.00120	0.02400
0.00070	0.01550	0.00100	0.01680
0.00120	0.01410	0.00150	0.02080
0.00070	0.02470	0.00070	0.04400
0.00080	0.03040	0.00110	0.01560
0.00100	0.01650	0.00130	0.01720
0.00080	0.01690	0.00130	0.02080
0.00060	0.02100	0.00070	0.02440
0.00070	0.02660	0.00170	0.01600
0.00100	0.02260	0.00080	0.01360
0.00090	0.02200	0.00100	0.01360
0.00070	0.01730	0.00060	0.02770
0.00090	0.01330	0.00110	0.02620
0.00090	0.01430	0.00100	0.01580
0.00100	0.01430	0.00120	0.02040
0.00070	0.03010	0.00050	0.03070
0.00120	0.01860	0.00090	0.01490
0.00090	0.01200	0.00110	0.01160

Table B.19: Training and Testing MSE for 37 and 38 hidden neurons

39 Hidden Neurons		40 Hidden Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**
0.00110	0.02270	0.00100	0.01410
0.00100	0.02800	0.00060	0.05390
0.00090	0.01260	0.00110	0.01380
0.00110	0.03380	0.00070	0.02130
0.00170	0.01570	0.00120	0.01460
0.00130	0.01520	0.00100	0.01780
0.00080	0.04720	0.00110	0.01330
0.00130	0.01430	0.00040	0.02680
0.00120	0.01830	0.00130	0.01060
0.00140	0.01840	0.00090	0.01790
0.00120	0.02970	0.00100	0.02200
0.00110	0.01780	0.00120	0.01490
0.00120	0.01910	0.00090	0.01480
0.00090	0.01870	0.00130	0.01490
0.00110	0.01320	0.00100	0.03110
0.00060	0.02460	0.00110	0.01220
0.00100	0.01900	0.00170	0.03030
0.00120	0.01530	0.00110	0.01460
0.00090	0.01590	0.00120	0.02580
0.00120	0.00970	0.00110	0.01420
0.00090	0.01660	0.00100	0.02810
0.00110	0.01600	0.00110	0.01280
0.00070	0.01810	0.00080	0.02020
0.00120	0.01290	0.00110	0.01290
0.00130	0.01390	0.00110	0.01440
0.00070	0.01680	0.00140	0.01460
0.00130	0.02010	0.00080	0.02130
0.00090	0.01550	0.00090	0.01630
0.00080	0.01950	0.00080	0.01880
0.00110	0.01480	0.00110	0.01310

Table B.20: Training and Testing MSE for 39 and 40 hidden neurons

Ppl	***Ie-eff***	***R*-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	11.4580	81.7420	4.0900	3.9096	3.9375
0.5	13.3260	79.8740	4.0200	3.8629	3.8725
1	15.1259	78.0741	3.9500	3.8163	3.8106
2	18.5207	74.6793	3.8100	3.7229	3.6943
3	21.6452	71.5548	3.6700	3.6295	3.5856
4	24.5088	68.6912	3.5400	3.5428	3.4898
5	27.1267	66.0733	3.4100	3.4561	3.3975
6	29.5198	63.6802	3.2900	3.3760	3.3147
7	31.7148	61.4852	3.1800	3.3027	3.2400
8	33.7440	59.4560	3.0700	3.2293	3.1661
9	35.6461	57.5539	2.9700	3.1626	3.0992
10	37.4664	55.7336	2.8800	3.1026	3.0388
11	39.2552	53.9448	2.7800	3.0359	2.9714
12	41.0651	52.1349	2.6900	2.9758	2.9101
13	42.9427	50.2573	2.5900	2.9091	2.8410
14	44.9164	48.2836	2.4900	2.8424	2.7703
15	46.9826	46.2174	2.3800	2.7691	2.6903
16	49.0968	44.1032	2.2700	2.6957	2.6073
17	51.1796	42.0204	2.1600	2.6223	2.5203
18	53.1396	40.0604	2.0700	2.5623	2.4454
19	54.9025	38.2975	1.9800	2.5023	2.3664
20	56.4313	36.7687	1.9100	2.4556	2.3015
21	57.7312	35.4688	1.8500	2.4156	2.2431
22	58.8450	34.3550	1.8000	2.3822	2.1920
23	59.8570	33.3430	1.7500	2.3489	2.1385
24	60.9223	32.2777	1.7100	2.3222	2.0937
25	62.3110	30.8890	1.6500	2.2821	2.0226
26	64.2678	28.9322	1.5700	2.2288	1.9191
27	66.4785	26.7215	1.4800	2.1688	1.7870
28	68.2583	24.9417	1.4100	2.1221	1.6686
29	69.5050	23.6950	1.3700	2.0954	1.5926
30	70.4916	22.7084	1.3400	2.0754	1.5307

Table B.21: Results of Neural Network for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 1$

Ppl	*Ie-eff*	R-Rating factor	MOS_{LQE}	MOS_{LQO}	PESQ Score
0	12.9502	80.2498	4.0300	3.8696	3.8816
0.5	15.3495	77.8505	3.9400	3.8096	3.8020
1	17.6811	75.5189	3.8400	3.7429	3.7185
2	22.1263	71.0737	3.6500	3.6162	3.5706
3	26.2624	66.9376	3.4500	3.4828	3.4256
4	30.0780	63.1220	3.2600	3.3560	3.2942
5	33.5740	59.6260	3.0800	3.2360	3.1728
6	36.7629	56.4371	2.9100	3.1226	3.0590
7	39.6680	53.5320	2.7600	3.0225	2.9579
8	42.3221	50.8779	2.6200	2.9291	2.8619
9	44.7665	48.4335	2.4900	2.8424	2.7703
10	47.0498	46.1502	2.3700	2.7624	2.6829
11	49.2280	43.9720	2.2600	2.6890	2.5995
12	51.3616	41.8384	2.1500	2.6157	2.5121
13	53.5118	39.6882	2.0500	2.5490	2.4282
14	55.7316	37.4684	1.9400	2.4756	2.3297
15	58.0528	35.1472	1.8300	2.4022	2.2229
16	60.4717	32.7283	1.7300	2.3355	2.1163
17	62.9420	30.2580	1.6200	2.2621	1.9851
18	65.3825	27.8175	1.5200	2.1954	1.8481
19	67.7022	25.4978	1.4300	2.1354	1.7041
20	69.8299	23.3701	1.3600	2.0887	1.5725
21	71.7373	21.4627	1.3000	2.0487	1.4399
22	73.4542	19.7458	1.2400	2.0087	1.2804
23	75.0940	18.1060	1.2000	1.9820	1.1519
24	76.8951	16.3049	1.1500	1.9487	0.9518
25	79.1278	14.0722	1.1000	1.9153	0.6740
26	81.5895	11.6105	1.0600	1.8886	0.3298
27	83.5751	9.6249	1.0300	1.8686	-0.1282
28	84.8663	8.3337	1.0200	1.8619	-0.3905
29	85.7136	7.4864	1.0100	1.8553	-0.5000
30	86.3357	6.8643	1.0000	1.8486	-0.5000

Table B.22: Results of Neural Network for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 2$

Ppl	***Ie-eff***	R**-Rating factor**	MOS_{LQE}	MOS_{LQO}	**PESQ Score**
0	14.7854	78.4146	3.9600	3.8229	3.8193
0.5	17.0399	76.1601	3.8700	3.7629	3.7431
1	19.2409	73.9591	3.7800	3.7029	3.6704
2	23.4744	69.7256	3.5800	3.5695	3.5189
3	27.4750	65.7250	3.3900	3.4427	3.3836
4	31.2392	61.9608	3.2000	3.3160	3.2535
5	34.7699	58.4301	3.0200	3.1959	3.1326
6	38.0760	55.1240	2.8400	3.0759	3.0120
7	41.1715	52.0285	2.6800	2.9692	2.9033
8	44.0745	49.1255	2.5300	2.8691	2.7988
9	46.8077	46.3923	2.3900	2.7757	2.6977
10	49.3981	43.8019	2.2500	2.6824	2.5918
11	51.8780	41.3220	2.1300	2.6023	2.4957
12	54.2846	38.9154	2.0100	2.5223	2.3932
13	56.6596	36.5404	1.9000	2.4489	2.2920
14	59.0450	34.1550	1.7900	2.3755	2.1815
15	61.4749	31.7251	1.6800	2.3022	2.0588
16	63.9629	29.2371	1.5800	2.2355	1.9327
17	66.4888	26.7112	1.4800	2.1688	1.7870
18	68.9939	24.2061	1.3900	2.1087	1.6315
19	71.3928	21.8072	1.3100	2.0554	1.4636
20	73.6031	19.5969	1.2400	2.0087	1.2804
21	75.5856	17.6144	1.1900	1.9753	1.1160
22	77.3921	15.8079	1.1400	1.9420	0.9042
23	79.2181	13.9819	1.1000	1.9153	0.6740
24	81.3529	11.8471	1.0600	1.8886	0.3298
25	83.7212	9.4788	1.0300	1.8686	-0.1282
26	85.6569	7.5431	1.0100	1.8553	-0.5000
27	86.8481	6.3519	1.0000	1.8486	-0.5000
28	87.5313	5.6687	1.0000	1.8486	-0.5000
29	87.9591	5.2409	1.0000	1.8486	-0.5000
30	88.2597	4.9403	1.0000	1.8486	-0.5000

Table B.23: Results of Neural Network for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 3$

Ppl	*Ie-eff*	R-Rating factor	MOS_{LQE}	MOS_{LQO}	PESQ Score
0	13.7279	79.4721	4.0000	3.8496	3.8546
0.5	16.2808	76.9192	3.9000	3.7829	3.7681
1	18.7807	74.4193	3.8000	3.7162	3.6863
2	23.6027	69.5973	3.5800	3.5695	3.5189
3	28.1617	65.0383	3.3600	3.4227	3.3628
4	32.4345	60.7655	3.1400	3.2760	3.2131
5	36.4074	56.7926	2.9300	3.1359	3.0724
6	40.0758	53.1242	2.7400	3.0092	2.9443
7	43.4442	49.7558	2.5600	2.8891	2.8200
8	46.5252	46.6748	2.4000	2.7824	2.7051
9	49.3396	43.8604	2.2600	2.6890	2.5995
10	51.9156	41.2844	2.1300	2.6023	2.4957
11	54.2889	38.9111	2.0100	2.5223	2.3932
12	56.5026	36.6974	1.9100	2.4556	2.3015
13	58.6069	34.5931	1.8100	2.3889	2.2024
14	60.6565	32.5435	1.7200	2.3288	2.1051
15	62.7056	30.4944	1.6300	2.2688	1.9977
16	64.7986	28.4014	1.5400	2.2088	1.8771
17	66.9566	26.2434	1.4600	2.1554	1.7548
18	69.1652	24.0348	1.3800	2.1021	1.6123
19	71.3746	21.8254	1.3100	2.0554	1.4636
20	73.5228	19.6772	1.2400	2.0087	1.2804
21	75.5908	17.6092	1.1900	1.9753	1.1160
22	77.6831	15.5169	1.1300	1.9353	0.8532
23	80.0376	13.1624	1.0900	1.9086	0.6026
24	82.6479	10.5521	1.0400	1.8753	0.0606
25	84.8857	8.3143	1.0200	1.8619	-0.3905
26	86.3300	6.8700	1.0000	1.8486	-0.5000
27	87.1771	6.0229	1.0000	1.8486	-0.5000
28	87.7066	5.4934	1.0000	1.8486	-0.5000
29	88.0723	5.1277	1.0000	1.8486	-0.5000
30	88.3445	4.8555	1.0000	1.8486	-0.5000

Table B.24: Results of Neural Network for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 4$

Ppl	*Ie-eff*	R-Rating factor	MOS_{LQE}	MOS_{LQO}	PESQ Score
0	12.7272	80.4728	4.0400	3.8763	3.8908
0.5	15.2226	77.9774	3.9500	3.8163	3.8106
1	17.6680	75.5320	3.8400	3.7429	3.7185
2	22.3926	70.8074	3.6300	3.6028	3.5557
3	26.8721	66.3279	3.4200	3.4627	3.4045
4	31.0856	62.1144	3.2100	3.3227	3.2603
5	35.0197	58.1803	3.0100	3.1893	3.1259
6	38.6689	54.5311	2.8100	3.0559	2.9917
7	42.0351	51.1649	2.6400	2.9425	2.8757
8	45.1270	48.0730	2.4700	2.8291	2.7560
9	47.9601	45.2399	2.3300	2.7357	2.6530
10	50.5562	42.6438	2.2000	2.6490	2.5524
11	52.9435	40.2565	2.0800	2.5690	2.4539
12	55.1572	38.0428	1.9700	2.4956	2.3573
13	57.2391	35.9609	1.8700	2.4289	2.2629
14	59.2377	33.9623	1.7800	2.3689	2.1709
15	61.2059	31.9941	1.6900	2.3088	2.0706
16	63.1956	30.0044	1.6100	2.2555	1.9722
17	65.2489	27.9511	1.5300	2.2021	1.8627
18	67.3868	25.8132	1.4500	2.1488	1.7382
19	69.6062	23.5938	1.3700	2.0954	1.5926
20	71.9018	21.2982	1.2900	2.0420	1.4155
21	74.3321	18.8679	1.2200	1.9953	1.2189
22	77.0682	16.1318	1.1500	1.9487	0.9518
23	80.1212	13.0788	1.0800	1.9020	0.5230
24	82.8856	10.3144	1.0400	1.8753	0.0606
25	84.8118	8.3882	1.0200	1.8619	-0.3905
26	86.0232	7.1768	1.0000	1.8486	-0.5000
27	86.8176	6.3824	1.0000	1.8486	-0.5000
28	87.3781	5.8219	1.0000	1.8486	-0.5000
29	87.7954	5.4046	1.0000	1.8486	-0.5000
30	88.1168	5.0832	1.0000	1.8486	-0.5000

Table B.25: Results of Neural Network for *Ie-eff*, R, MOS_{LQE}, MOS_{LQO}, $PESQ$. $BurstR = 5$

Appendix C

Combined Subjective-Test Free, Voice Classification Extension for the E-model Based on PESQ-Detailed Results ANN

In this appendix the detailed results of performance of the Artificial Neural Network (ANN) constructed to predict *Ie-eff* from *Ppl* and *Burst* for Voiced, Unvoiced and Unclassified speech losses as discussed in chapter 8 are listed.

Tables C.1-C.25 show the full results of training set performance and test set performance for the tested networks ranging from simple networks with 1 neuron in the hidden layer to large networks with 100 neurons in the hidden layer.

1 Neurons		2 Neurons		3 Neurons		4 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.12640	0.12550	0.10810	0.10260	0.09890	0.08980	0.09640	0.08590
0.12640	0.12540	0.10380	0.09470	0.09570	0.08560	0.08420	0.10570
0.12640	0.12550	0.10430	0.09550	0.09890	0.08980	0.09550	0.08130
0.12640	0.12540	0.10380	0.09480	0.10600	0.10680	0.09240	0.09530
0.12640	0.12540	0.10390	0.09480	0.09420	0.08830	0.09180	0.09130
0.12640	0.12550	0.10550	0.09480	0.09140	0.08720	0.09120	0.11610
0.12640	0.12540	0.10860	0.10220	0.10440	0.09560	0.09240	0.08200
0.12640	0.12540	0.10810	0.10260	0.09900	0.08780	0.09770	0.09600
0.12640	0.12540	0.10370	0.09470	0.09560	0.08570	0.09100	0.09960
0.12640	0.12540	0.11410	0.11110	0.10520	0.09350	0.08530	0.07620
0.12640	0.12540	0.10940	0.10080	0.09560	0.08560	0.08720	0.08040
0.12640	0.12540	0.10410	0.09510	0.09740	0.09280	0.08330	0.08250
0.12640	0.12540	0.10500	0.09770	0.09620	0.08720	0.08270	0.07710
0.12640	0.12540	0.10550	0.09480	0.09440	0.08670	0.09170	0.08680
0.12640	0.12540	0.10490	0.09760	0.10010	0.09280	0.09090	0.07880
0.12640	0.12540	0.10410	0.09510	0.08680	0.07470	0.08950	0.07870
0.12640	0.12540	0.10420	0.09490	0.09430	0.08700	0.08490	0.08330
0.12640	0.12540	0.10430	0.09540	0.09560	0.08570	0.10010	0.10060
0.12640	0.12540	0.10370	0.09460	0.09990	0.10070	0.09870	0.09310
0.12640	0.12550	0.10370	0.09450	0.08690	0.07410	0.08630	0.07770
0.12640	0.12540	0.10540	0.09470	0.09420	0.08760	0.09410	0.09080
0.12640	0.12540	0.10380	0.09460	0.10180	0.09300	0.08370	0.07450
0.12640	0.12540	0.08610	0.08570	0.08630	0.07790	0.08980	0.08330
0.12640	0.12540	0.10990	0.09680	0.09580	0.08630	0.08710	0.08450
0.12640	0.12540	0.10370	0.09440	0.08480	0.07580	0.08450	0.07640
0.12640	0.12540	0.10390	0.09490	0.10130	0.09310	0.09230	0.08510
0.12640	0.12540	0.10360	0.09440	0.08500	0.07580	0.08380	0.07860
0.12640	0.12540	0.10440	0.09570	0.09940	0.09730	0.09210	0.09990
0.12640	0.12550	0.10400	0.09500	0.09570	0.08610	0.08210	0.07880
0.12640	0.12540	0.12740	0.12570	0.08480	0.07580	0.09300	0.08310

Table C.1: Training and Testing MSE for 1 to 4 hidden neurons

5 Neurons		6 Neurons		7 Neurons		8 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.09160	0.10050	0.07910	0.07830	0.08250	0.10070	0.07800	0.11100
0.08980	0.13330	0.08890	0.08430	0.08960	0.09440	0.08220	0.09140
0.09190	0.09740	0.09230	0.09960	0.08860	0.08750	0.08450	0.09740
0.08510	0.08740	0.08780	0.08580	0.07960	0.09380	0.08660	0.07680
0.09240	0.08280	0.07980	0.08640	0.08010	0.08210	0.07970	0.11880
0.09280	0.09690	0.08170	0.08630	0.08210	0.08860	0.07930	0.09310
0.09320	0.08660	0.08590	0.08370	0.08890	0.09360	0.08520	0.10360
0.08920	0.08350	0.09000	0.08440	0.08630	0.09090	0.08780	0.08460
0.08420	0.08560	0.09750	0.09060	0.08810	0.08370	0.07580	0.09190
0.09800	0.08460	0.07960	0.07870	0.09370	0.09360	0.07910	0.08220
0.09250	0.09280	0.09260	0.09360	0.09440	0.08440	0.09200	0.08340
0.09260	0.08980	0.08030	0.07470	0.09710	0.09780	0.09190	0.08380
0.08750	0.08130	0.08840	0.08760	0.08090	0.09760	0.08950	0.09680
0.08480	0.07850	0.08370	0.08020	0.07780	0.08140	0.08690	0.08860
0.08880	0.08190	0.08810	0.08450	0.09030	0.08200	0.07860	0.08500
0.08540	0.08700	0.09440	0.08740	0.08840	0.09580	0.08380	0.09540
0.09060	0.08450	0.08770	0.08530	0.08610	0.08650	0.08740	0.10940
0.09350	0.09410	0.08740	0.09110	0.08440	0.08650	0.08410	0.08600
0.09420	0.08150	0.09010	0.08960	0.08810	0.08420	0.09030	0.08090
0.10000	0.08850	0.08670	0.08800	0.08820	0.08910	0.08720	0.09370
0.08250	0.11410	0.08050	0.07720	0.08280	0.08510	0.07910	0.08960
0.08340	0.07490	0.08060	0.07480	0.09340	0.08410	0.08350	0.09100
0.09100	0.08410	0.08600	0.08720	0.09050	0.09430	0.08360	0.08550
0.09670	0.09580	0.08380	0.08240	0.07760	0.08880	0.08170	0.09530
0.09220	0.09970	0.08730	0.07940	0.08070	0.07960	0.07720	0.08480
0.08350	0.08570	0.07970	0.07790	0.08550	0.08160	0.08200	0.08740
0.08400	0.08690	0.09740	0.08870	0.08740	0.08940	0.07820	0.08190
0.09430	0.09200	0.09250	0.08580	0.08760	0.09090	0.08640	0.08770
0.08930	0.08140	0.08530	0.08220	0.08310	0.08030	0.08830	0.08440
0.08880	0.08830	0.09030	0.09500	0.08120	0.07850	0.09170	0.09970

Table C.2: Training and Testing MSE for 5 to 8 hidden neurons

9 Neurons		10 Neurons		11 Neurons		12 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.08510	0.08280	0.07600	0.08810	0.08360	0.08210	0.08060	0.11170
0.08270	0.07860	0.07790	0.09980	0.07910	0.09480	0.08150	0.09950
0.08780	0.08700	0.07870	0.08850	0.08460	0.08510	0.07770	0.10020
0.08300	0.08980	0.07870	0.08220	0.08020	0.08920	0.08220	0.08680
0.08690	0.09560	0.07620	0.08750	0.07160	0.10190	0.07910	0.10850
0.08320	0.08780	0.08850	0.09310	0.07810	0.09630	0.07920	0.09780
0.07520	0.13560	0.08430	0.09970	0.07650	0.09080	0.07870	0.09550
0.08170	0.09060	0.07710	0.09160	0.07860	0.09250	0.08360	0.16120
0.08040	0.11070	0.07990	0.14230	0.07820	0.10020	0.07560	0.08930
0.07960	0.09500	0.08900	0.08740	0.08650	0.09820	0.07240	0.11460
0.08050	0.09800	0.08720	0.09340	0.07860	0.08260	0.08540	0.10740
0.08450	0.08900	0.08190	0.08210	0.08710	0.08630	0.08220	0.09020
0.07860	0.11270	0.08400	0.09050	0.08640	0.08990	0.08420	0.11160
0.08360	0.10280	0.07610	0.09760	0.08400	0.09280	0.08350	0.08650
0.07710	0.08550	0.07600	0.09240	0.07640	0.09530	0.08540	0.10950
0.07750	0.09960	0.08250	0.09080	0.07590	0.11280	0.08180	0.09060
0.07720	0.10300	0.08140	0.08780	0.07510	0.09590	0.08100	0.11000
0.07970	0.08520	0.08290	0.09000	0.08280	0.09770	0.07970	0.11570
0.08320	0.09430	0.07470	0.11310	0.08260	0.09940	0.08360	0.12310
0.08970	0.09200	0.07680	0.08730	0.08700	0.12330	0.07500	0.09580
0.07610	0.09950	0.08240	0.11590	0.07380	0.10700	0.07350	0.10360
0.07580	0.09430	0.08460	0.08770	0.07600	0.08980	0.07600	0.10010
0.08810	0.08810	0.07740	0.08910	0.08810	0.09000	0.08590	0.08430
0.08190	0.09470	0.08890	0.09300	0.07600	0.11040	0.08050	0.08470
0.07740	0.09580	0.08410	0.14690	0.08580	0.11290	0.08320	0.09500
0.08370	0.09780	0.07760	0.15410	0.07660	0.11420	0.07790	0.09660
0.08820	0.08430	0.07920	0.08610	0.08140	0.09190	0.08030	0.09000
0.07910	0.09620	0.08150	0.09990	0.08200	0.10180	0.07760	0.09410
0.08440	0.09610	0.07900	0.10030	0.07530	0.09910	0.07870	0.14530
0.07990	0.08280	0.08430	0.08540	0.08460	0.11090	0.07940	0.10150

Table C.3: Training and Testing MSE for 9 to 12 hidden neurons

13 Neurons		14 Neurons		15 Neurons		16 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.07780	0.08670	0.08170	0.10250	0.08130	0.09710	0.07740	0.09600
0.08160	0.09300	0.08130	0.09410	0.07410	0.09590	0.08810	0.09070
0.07590	0.10490	0.07770	0.09650	0.07170	0.17560	0.07420	0.12190
0.07670	0.09920	0.08000	0.08270	0.07420	0.10510	0.08470	0.09180
0.08500	0.09160	0.08110	0.09460	0.08260	0.10230	0.07540	0.09770
0.08580	0.08800	0.07410	0.09790	0.07720	0.09210	0.08510	0.09280
0.06950	0.11150	0.07470	0.11820	0.07560	0.09600	0.06800	0.10840
0.07880	0.09630	0.08920	0.08860	0.07980	0.09520	0.08030	0.17430
0.08260	0.08740	0.08370	0.09610	0.07470	0.09210	0.07650	0.10290
0.07510	0.08970	0.08330	0.09190	0.08410	0.09180	0.07580	0.09220
0.08120	0.09090	0.07780	0.13270	0.07910	0.09640	0.07030	0.11680
0.07010	0.13550	0.07470	0.10980	0.08750	0.09270	0.07610	0.11910
0.08500	0.10440	0.07730	0.10680	0.08180	0.08480	0.08270	0.10890
0.07950	0.08960	0.07880	0.10870	0.07420	0.08830	0.08090	0.10210
0.07860	0.11140	0.07490	0.11350	0.07970	0.09530	0.07560	0.13840
0.08580	0.10090	0.08600	0.09120	0.07290	0.10920	0.07280	0.11410
0.08510	0.09060	0.07570	0.10830	0.08200	0.09210	0.07310	0.11650
0.07540	0.09380	0.08510	0.10960	0.07240	0.12210	0.07980	0.09970
0.07610	0.10550	0.08010	0.08870	0.07200	0.11830	0.07240	0.10190
0.07500	0.09460	0.07790	0.13570	0.08530	0.10010	0.07540	0.12200
0.07870	0.10370	0.07440	0.09490	0.08020	0.10650	0.07420	0.12090
0.08580	0.08120	0.07820	0.08970	0.08030	0.10730	0.07690	0.10180
0.07180	0.14480	0.08290	0.11480	0.07740	0.10220	0.08000	0.09250
0.07290	0.10660	0.07560	0.12860	0.07720	0.08950	0.08310	0.09350
0.07700	0.10540	0.08440	0.10270	0.07180	0.09810	0.07260	0.10800
0.07350	0.10190	0.08100	0.10020	0.07570	0.11730	0.07560	0.09080
0.07480	0.08550	0.07390	0.11780	0.07200	0.11000	0.07500	0.12340
0.08270	0.09300	0.07230	0.09930	0.08200	0.08970	0.07390	0.10770
0.07730	0.12360	0.07520	0.08760	0.07730	0.10350	0.07700	0.08820
0.07800	0.09420	0.07270	0.12380	0.07460	0.11380	0.07510	0.10600

Table C.4: Training and Testing MSE for 13 to 16 hidden neurons

17 Neurons		18 Neurons		19 Neurons		20 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.07920	0.11900	0.07230	0.09630	0.07560	0.09540	0.07890	0.10360
0.07140	0.13020	0.07260	0.12330	0.07280	0.11340	0.06870	0.12110
0.07170	0.10110	0.08110	0.09250	0.07840	0.10080	0.07620	0.10750
0.08250	0.10040	0.07080	0.12120	0.07460	0.14290	0.07730	0.10170
0.07400	0.10290	0.07620	0.10070	0.06960	0.09830	0.07690	0.09780
0.07740	0.12650	0.07700	0.09350	0.07210	0.10950	0.06760	0.13060
0.07800	0.08780	0.06970	0.14930	0.07950	0.10320	0.06780	0.13740
0.07320	0.11720	0.07980	0.10380	0.07690	0.09810	0.08440	0.08940
0.07710	0.09520	0.07700	0.10350	0.07300	0.11920	0.06830	0.12770
0.07590	0.10770	0.07780	0.11570	0.07600	0.10460	0.07720	0.10260
0.07350	0.10040	0.07780	0.09920	0.08040	0.08990	0.07620	0.09660
0.07760	0.09640	0.07040	0.11980	0.07950	0.09940	0.07400	0.13530
0.07730	0.08930	0.06980	0.10320	0.07710	0.11160	0.07720	0.09550
0.06980	0.13160	0.07790	0.09590	0.07280	0.10690	0.07430	0.14440
0.07320	0.11420	0.07460	0.10560	0.07890	0.10490	0.07100	0.14690
0.07640	0.10690	0.06650	0.15010	0.07640	0.11290	0.07570	0.11160
0.06850	0.11240	0.07610	0.13050	0.07270	0.12300	0.07390	0.11760
0.07440	0.12310	0.07080	0.12270	0.07280	0.11530	0.07850	0.10540
0.08250	0.09610	0.07060	0.13730	0.10470	0.12900	0.08080	0.09170
0.07990	0.12210	0.07500	0.09980	0.07760	0.11660	0.07510	0.10470
0.07190	0.10100	0.08160	0.10950	0.08090	0.10670	0.07390	0.12250
0.07120	0.10450	0.07380	0.10640	0.06810	0.12190	0.06750	0.12020
0.07860	0.09140	0.07560	0.10860	0.07800	0.09860	0.07820	0.09780
0.07430	0.13600	0.08290	0.09430	0.07510	0.10340	0.07020	0.11250
0.07710	0.09060	0.07380	0.15410	0.07610	0.09870	0.07510	0.10050
0.08100	0.08690	0.07250	0.10500	0.07190	0.10600	0.07760	0.10140
0.07280	0.11040	0.07360	0.13260	0.08090	0.11400	0.06950	0.13380
0.07420	0.09920	0.07610	0.10110	0.07640	0.10500	0.07630	0.10020
0.07660	0.09560	0.07120	0.10520	0.07020	0.11260	0.07080	0.12870
0.07580	0.11120	0.07910	0.09630	0.08280	0.09240	0.08490	0.08800

Table C.5: Training and Testing MSE for 17 to 20 hidden neurons

21 Neurons		22 Neurons		23 Neurons		24 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.07240	0.16840	0.06830	0.10570	0.06930	0.10880	0.07350	0.10000
0.07220	0.10650	0.06790	0.11270	0.07210	0.11160	0.07220	0.11920
0.07820	0.10160	0.07310	0.12740	0.07800	0.09790	0.06600	0.16900
0.07200	0.10500	0.07290	0.11350	0.06820	0.15010	0.07630	0.11000
0.07740	0.10780	0.06680	0.15350	0.07240	0.13350	0.07660	0.12920
0.07380	0.11030	0.07030	0.10750	0.07220	0.10710	0.06930	0.13730
0.06860	0.13280	0.07590	0.10550	0.07720	0.09870	0.06360	0.12380
0.06950	0.15630	0.07400	0.11680	0.07660	0.10190	0.07650	0.12470
0.07050	0.12980	0.06730	0.11540	0.07540	0.10830	0.07370	0.11210
0.06660	0.14620	0.07780	0.10580	0.06810	0.11600	0.07510	0.12970
0.07320	0.11910	0.07260	0.11380	0.06710	0.19710	0.07210	0.10680
0.07270	0.10170	0.07330	0.11510	0.06690	0.12970	0.07540	0.10300
0.07060	0.12440	0.07150	0.11030	0.07170	0.10070	0.07000	0.10730
0.08320	0.09030	0.06790	0.11720	0.07250	0.11900	0.06370	0.13130
0.07510	0.11090	0.06500	0.17690	0.07800	0.10710	0.07640	0.10430
0.07460	0.11450	0.07100	0.13180	0.07550	0.10830	0.06500	0.13990
0.07230	0.10970	0.07880	0.09720	0.07470	0.11790	0.07200	0.10540
0.07800	0.11740	0.07000	0.12210	0.06630	0.15740	0.07930	0.10210
0.08220	0.10660	0.07860	0.09610	0.07390	0.12720	0.07130	0.10590
0.06940	0.12430	0.07100	0.11780	0.07060	0.18080	0.06960	0.12490
0.07320	0.09490	0.07590	0.09960	0.07000	0.10150	0.06820	0.10610
0.06940	0.12160	0.07230	0.09340	0.07570	0.09130	0.07490	0.10550
0.07440	0.09970	0.07690	0.09960	0.06850	0.12800	0.07730	0.15020
0.06560	0.13060	0.07870	0.10610	0.06700	0.12220	0.07450	0.13290
0.07760	0.10240	0.07360	0.12830	0.06930	0.13880	0.07870	0.10030
0.06950	0.11610	0.06690	0.10820	0.07480	0.10180	0.06800	0.11590
0.08760	0.11310	0.06910	0.13310	0.07200	0.11250	0.07410	0.11050
0.07420	0.10660	0.07190	0.10190	0.07950	0.09610	0.07380	0.09950
0.08190	0.10850	0.07440	0.11750	0.07680	0.10420	0.06430	0.17810
0.07200	0.10110	0.07630	0.11360	0.07260	0.11900	0.07920	0.10580

Table C.6: Training and Testing MSE for 21 to 24 hidden neurons

25 Neurons		26 Neurons		27 Neurons		28 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.07800	0.10370	0.06880	0.19220	0.0669	0.12420	0.07270	0.10330
0.07670	0.10930	0.07130	0.12470	0.0702	0.14320	0.06570	0.17720
0.06900	0.14270	0.06350	0.11520	0.076	0.11130	0.07070	0.13370
0.07240	0.13850	0.07700	0.12320	0.0736	0.12020	0.07300	0.12620
0.07030	0.12310	0.07420	0.11310	0.0669	0.13510	0.07110	0.11770
0.07640	0.10000	0.07780	0.09570	0.07	0.12770	0.06980	0.14950
0.07410	0.12480	0.07190	0.11830	0.0708	0.18540	0.06370	0.12790
0.07100	0.16460	0.06650	0.14910	0.075	0.10570	0.07090	0.11530
0.07080	0.10830	0.08220	0.10250	0.0795	0.09390	0.07070	0.17870
0.07480	0.10130	0.07620	0.10380	0.0759	0.14240	0.07250	0.10760
0.06970	0.12000	0.07590	0.10070	0.0791	0.09960	0.07640	0.10400
0.07290	0.09680	0.07670	0.10080	0.0674	0.11810	0.06530	0.13770
0.07030	0.12310	0.06950	0.11820	0.0716	0.11380	0.07280	0.10590
0.07110	0.12680	0.06680	0.12980	0.0731	0.11190	0.06510	0.13000
0.07720	0.10060	0.06620	0.12690	0.0742	0.14900	0.06950	0.12030
0.07590	0.10920	0.07340	0.09690	0.0703	0.11210	0.07570	0.17760
0.07340	0.11330	0.07140	0.11950	0.0664	0.13120	0.06930	0.16750
0.07190	0.11370	0.07620	0.10090	0.0758	0.11170	0.06870	0.12550
0.07820	0.09570	0.07180	0.10120	0.0776	0.11600	0.07030	0.10630
0.06800	0.12100	0.07310	0.16660	0.0697	0.10430	0.06860	0.17600
0.07110	0.14890	0.07830	0.15310	0.0708	0.10260	0.07040	0.12350
0.06650	0.13050	0.07880	0.08490	0.0692	0.12250	0.06220	0.18090
0.07540	0.10940	0.07140	0.16660	0.0699	0.13520	0.06890	0.13210
0.07400	0.12860	0.06520	0.12210	0.0704	0.09940	0.07890	0.09170
0.07140	0.09800	0.07030	0.14550	0.0633	0.16290	0.07140	0.12150
0.07190	0.12890	0.07260	0.11650	0.0699	0.12640	0.06930	0.11830
0.07650	0.09750	0.07490	0.10670	0.0708	0.10000	0.07590	0.12980
0.07330	0.09760	0.07220	0.12020	0.0743	0.12510	0.07060	0.13040
0.07390	0.11510	0.07260	0.12310	0.0749	0.15620	0.07020	0.10580
0.07270	0.11080	0.06830	0.10370	0.0706	0.11360	0.08050	0.10040

Table C.7: Training and Testing MSE for 25 to 28 hidden neurons

29 Neurons		30 Neurons		31 Neurons		32 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.07660	0.1179	0.06910	0.11040	0.07720	0.10960	0.07290	0.11460
0.07380	0.1106	0.07370	0.10920	0.07540	0.09480	0.07080	0.14370
0.06970	0.1119	0.07360	0.12180	0.07460	0.11320	0.07310	0.15150
0.07460	0.1113	0.07000	0.10310	0.07490	0.12750	0.07360	0.14800
0.07200	0.154	0.06990	0.11730	0.07250	0.10830	0.06460	0.13200
0.06010	0.1553	0.06460	0.19630	0.06990	0.13560	0.06840	0.11700
0.07380	0.1191	0.06520	0.12140	0.06730	0.13110	0.06760	0.14490
0.06470	0.2009	0.07320	0.13050	0.06650	0.11000	0.06690	0.12850
0.06880	0.1248	0.06830	0.10230	0.06700	0.10660	0.06490	0.11970
0.06980	0.1107	0.06490	0.16970	0.06310	0.13460	0.06510	0.13150
0.07240	0.0981	0.07050	0.16440	0.08080	0.09330	0.06540	0.13950
0.06550	0.1438	0.06810	0.13920	0.06740	0.13530	0.07040	0.11540
0.07060	0.1448	0.06550	0.12960	0.07280	0.10690	0.07150	0.12160
0.06880	0.1145	0.06790	0.12170	0.06730	0.11020	0.06970	0.11300
0.07450	0.1034	0.07010	0.12870	0.06640	0.13470	0.06660	0.14300
0.07010	0.1384	0.06830	0.12920	0.07600	0.11220	0.07410	0.12190
0.06960	0.1203	0.07440	0.12210	0.06710	0.13000	0.06790	0.11560
0.07870	0.1072	0.07720	0.12800	0.06160	0.16800	0.06770	0.14770
0.06450	0.1245	0.06790	0.13850	0.06420	0.15570	0.06660	0.11450
0.07460	0.0989	0.07310	0.13120	0.07000	0.12410	0.07100	0.11200
0.07730	0.0944	0.07650	0.10170	0.06900	0.11080	0.07560	0.11490
0.06220	0.2187	0.07160	0.12560	0.07440	0.09600	0.06610	0.14790
0.06990	0.1352	0.06850	0.11980	0.06180	0.20410	0.06500	0.13840
0.06890	0.1296	0.06730	0.13200	0.06770	0.12550	0.07020	0.11400
0.07430	0.0891	0.06930	0.13560	0.06700	0.12470	0.06780	0.13130
0.06790	0.1185	0.07060	0.11810	0.07560	0.12300	0.06890	0.12720
0.07660	0.1216	0.06860	0.13580	0.06780	0.16980	0.07350	0.13090
0.07770	0.1039	0.06870	0.11590	0.07480	0.12810	0.07290	0.10730
0.06600	0.1474	0.06600	0.14810	0.07220	0.10180	0.06880	0.14160
0.06790	0.1211	0.07540	0.09480	0.06750	0.12110	0.07420	0.09440

Table C.8: Training and Testing MSE for 29 to 32 hidden neurons

33 Neurons		34 Neurons		35 Neurons		36 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06640	0.14560	0.07950	0.10110	0.06570	0.15860	0.06880	0.11820
0.07560	0.15090	0.06620	0.16800	0.06470	0.14230	0.07000	0.11350
0.06560	0.15410	0.07260	0.11190	0.06820	0.12330	0.06850	0.14490
0.06400	0.15720	0.06920	0.12150	0.06600	0.13290	0.05780	0.16350
0.07120	0.12190	0.06450	0.17980	0.06850	0.12360	0.06830	0.11860
0.07220	0.11330	0.06870	0.11290	0.07020	0.11430	0.07720	0.10010
0.06380	0.14980	0.07530	0.12340	0.07850	0.11620	0.06410	0.12970
0.07330	0.11900	0.07200	0.12750	0.06690	0.13000	0.07130	0.11270
0.06950	0.12140	0.06750	0.13170	0.07160	0.12460	0.06920	0.13580
0.06690	0.11210	0.06460	0.15670	0.07170	0.15680	0.06650	0.16810
0.07010	0.11340	0.06150	0.18930	0.06720	0.11830	0.06750	0.12360
0.06370	0.17350	0.06320	0.18150	0.06100	0.16540	0.07530	0.10850
0.06680	0.13710	0.06480	0.14340	0.06720	0.14470	0.07130	0.14770
0.07340	0.12120	0.06950	0.12540	0.08450	0.09900	0.06540	0.12430
0.06620	0.16310	0.06840	0.16860	0.07070	0.14440	0.07220	0.11270
0.06450	0.12270	0.06050	0.16780	0.07270	0.10960	0.06570	0.12350
0.06780	0.12480	0.06660	0.12080	0.06830	0.11720	0.06560	0.13290
0.06920	0.11160	0.06370	0.18120	0.06710	0.15030	0.07410	0.11920
0.06340	0.14720	0.06880	0.11790	0.07280	0.12430	0.06310	0.14310
0.06840	0.14080	0.07290	0.09940	0.06850	0.15230	0.07020	0.17510
0.07460	0.11870	0.06480	0.13880	0.06570	0.12750	0.06220	0.15020
0.07210	0.15230	0.06980	0.13810	0.06430	0.11530	0.06300	0.15310
0.06200	0.12330	0.06710	0.11500	0.07010	0.13480	0.06680	0.13680
0.07390	0.12310	0.07640	0.10310	0.06500	0.13020	0.07550	0.11110
0.06970	0.12960	0.07130	0.11610	0.06620	0.11620	0.07150	0.10790
0.07200	0.12440	0.06580	0.13850	0.07230	0.09470	0.06840	0.10950
0.06610	0.12790	0.07300	0.12470	0.06880	0.14330	0.06480	0.12500
0.07530	0.10220	0.06740	0.15410	0.07070	0.13540	0.06950	0.15720
0.06850	0.11230	0.07180	0.10810	0.07560	0.09610	0.06540	0.13650
0.06600	0.12850	0.07430	0.12490	0.06630	0.11130	0.06460	0.15770

Table C.9: Training and Testing MSE for 33 to 36 hidden neurons

37 Neurons		38 Neurons		39 Neurons		40 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06880	0.10440	0.06420	0.15150	0.07720	0.10150	0.06430	0.13400
0.07040	0.12580	0.06910	0.14130	0.06760	0.14650	0.06810	0.12450
0.07000	0.12960	0.07460	0.11200	0.06460	0.14400	0.06870	0.09970
0.06560	0.12020	0.07160	0.15050	0.06560	0.11190	0.06710	0.11450
0.06870	0.13360	0.06760	0.16180	0.06820	0.16740	0.06770	0.13470
0.06810	0.12010	0.06720	0.14470	0.06480	0.12910	0.07110	0.12280
0.06950	0.18860	0.06780	0.14490	0.06650	0.13250	0.06490	0.18450
0.07270	0.11000	0.06620	0.12120	0.06930	0.10750	0.05870	0.15920
0.06790	0.15100	0.06970	0.13500	0.06280	0.14040	0.06780	0.12270
0.07330	0.13590	0.06460	0.13730	0.07040	0.12090	0.06660	0.11930
0.07150	0.12080	0.06700	0.12940	0.06390	0.12110	0.05830	0.16190
0.06800	0.12020	0.06880	0.16330	0.07580	0.10420	0.06470	0.12690
0.06990	0.14460	0.07120	0.17270	0.06900	0.12960	0.06670	0.12530
0.06450	0.12330	0.06160	0.14270	0.06620	0.11610	0.07160	0.13940
0.06580	0.13790	0.06720	0.13730	0.06430	0.12770	0.06340	0.19690
0.06680	0.13510	0.06850	0.12710	0.06860	0.11830	0.06190	0.25890
0.06890	0.11390	0.06960	0.11480	0.06670	0.13670	0.06540	0.15160
0.06680	0.12420	0.06260	0.12730	0.06600	0.11960	0.06800	0.12320
0.07070	0.15150	0.07220	0.11820	0.06380	0.13730	0.06630	0.14650
0.06150	0.13410	0.06270	0.13250	0.06400	0.11660	0.07110	0.11920
0.06580	0.14210	0.06580	0.13560	0.06770	0.11430	0.06870	0.11010
0.07260	0.11660	0.06160	0.15630	0.05800	0.16870	0.06380	0.14790
0.06410	0.13820	0.06730	0.13130	0.06860	0.11360	0.06510	0.12590
0.06130	0.13670	0.07060	0.15140	0.06840	0.13360	0.06480	0.11290
0.06820	0.15320	0.07250	0.11060	0.06610	0.20260	0.06530	0.13000
0.06810	0.11720	0.06920	0.14550	0.06130	0.13950	0.07170	0.10820
0.06980	0.17040	0.06670	0.17220	0.06940	0.13300	0.07290	0.11050
0.07250	0.20190	0.06290	0.18640	0.07050	0.13800	0.06810	0.11870
0.07100	0.13780	0.06830	0.14050	0.06430	0.18640	0.06790	0.12800
0.06920	0.12110	0.06590	0.13910	0.06300	0.14310	0.07350	0.10720

Table C.10: Training and Testing MSE for 37 to 40 hidden neurons

41 Neurons		42 Neurons		43 Neurons		44 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05890	0.14540	0.06330	0.14610	0.05970	0.16480	0.05800	0.18920
0.05860	0.19340	0.06670	0.13020	0.06580	0.13150	0.06800	0.12530
0.06940	0.10800	0.06540	0.13830	0.06300	0.15450	0.06410	0.19550
0.07110	0.09690	0.06340	0.16690	0.06890	0.12070	0.06440	0.14260
0.06540	0.13990	0.06270	0.15610	0.06140	0.18850	0.06030	0.15710
0.06020	0.15110	0.06520	0.25290	0.06730	0.13140	0.06480	0.12730
0.05870	0.14340	0.05990	0.13510	0.06890	0.11610	0.06480	0.11570
0.06300	0.14380	0.06450	0.12470	0.06200	0.16360	0.06170	0.15090
0.06010	0.19290	0.06580	0.14040	0.06360	0.13450	0.05960	0.17820
0.06720	0.13830	0.06700	0.12840	0.07120	0.10800	0.06360	0.14820
0.07040	0.13660	0.06600	0.15210	0.06350	0.13680	0.07060	0.11450
0.06310	0.18640	0.06890	0.17030	0.06250	0.13690	0.05490	0.17220
0.06510	0.13520	0.06480	0.11210	0.06800	0.12870	0.06240	0.15030
0.06230	0.18480	0.06310	0.13290	0.07260	0.11030	0.06830	0.13220
0.06610	0.15050	0.06910	0.16950	0.06530	0.16340	0.06890	0.12860
0.06110	0.14150	0.07140	0.15880	0.06010	0.13370	0.06670	0.11560
0.06150	0.22090	0.06590	0.19160	0.06180	0.16430	0.06880	0.13170
0.06350	0.13630	0.06680	0.14990	0.05530	0.14820	0.06070	0.15850
0.06460	0.15900	0.06670	0.16300	0.07350	0.14270	0.06540	0.12570
0.06390	0.12310	0.05840	0.22390	0.06530	0.13900	0.06980	0.11410
0.06250	0.11890	0.06090	0.13900	0.06780	0.18720	0.07080	0.10010
0.06390	0.12400	0.06870	0.12280	0.07040	0.11790	0.06400	0.14230
0.06980	0.11120	0.07160	0.12160	0.06510	0.14430	0.07100	0.11580
0.06890	0.13020	0.06590	0.14570	0.07530	0.13200	0.06360	0.13690
0.06900	0.17480	0.07410	0.15220	0.07230	0.09840	0.06440	0.13520
0.06420	0.13240	0.06740	0.14090	0.06570	0.14980	0.06910	0.15460
0.07100	0.11960	0.06380	0.16510	0.07120	0.11140	0.06380	0.18500
0.06800	0.13110	0.06660	0.12620	0.06490	0.23360	0.06820	0.12080
0.06560	0.11660	0.06620	0.15970	0.05790	0.17610	0.06260	0.19550
0.06830	0.13970	0.06670	0.10430	0.06240	0.14390	0.05770	0.16000

Table C.11: Training and Testing MSE for 41 to 44 hidden neurons

45 Neurons		46 Neurons		47 Neurons		48 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06530	0.13910	0.05750	0.18820	0.06470	0.12130	0.06820	0.12370
0.06510	0.16430	0.06210	0.17370	0.05930	0.20160	0.06380	0.19340
0.06760	0.11920	0.06270	0.14560	0.05940	0.19890	0.06510	0.15490
0.07240	0.13220	0.06480	0.17220	0.06840	0.11500	0.06330	0.17650
0.06810	0.15240	0.07300	0.13100	0.06890	0.12960	0.07100	0.11590
0.06260	0.15070	0.06660	0.13260	0.06520	0.13490	0.07250	0.11780
0.06440	0.17090	0.06360	0.13120	0.06870	0.13020	0.06170	0.17070
0.05860	0.14820	0.06360	0.16000	0.06010	0.16480	0.06560	0.23620
0.06000	0.21160	0.06550	0.17200	0.06650	0.11600	0.06250	0.13730
0.06930	0.12160	0.06340	0.14940	0.06150	0.17290	0.06630	0.18250
0.07360	0.12500	0.06420	0.14500	0.06680	0.10600	0.06550	0.13640
0.06410	0.17150	0.06020	0.18550	0.06520	0.13570	0.05900	0.17290
0.06710	0.11130	0.06290	0.13910	0.06890	0.14740	0.05980	0.15200
0.06330	0.15770	0.06680	0.14320	0.05780	0.16010	0.06800	0.14700
0.06920	0.11330	0.07330	0.10360	0.06770	0.20220	0.06380	0.12970
0.07290	0.12770	0.05450	0.17290	0.06880	0.11670	0.05720	0.17430
0.07320	0.10260	0.06500	0.15410	0.07160	0.13130	0.05770	0.14000
0.05620	0.18430	0.07490	0.11600	0.06710	0.15610	0.05970	0.21350
0.06210	0.24190	0.06660	0.12910	0.06370	0.14390	0.05990	0.16120
0.06230	0.17450	0.06150	0.15920	0.07450	0.11580	0.06380	0.13600
0.06710	0.12210	0.05980	0.15360	0.06450	0.12940	0.06220	0.16520
0.07140	0.11650	0.06210	0.15780	0.06250	0.17110	0.06410	0.15380
0.06110	0.15020	0.07440	0.11930	0.06410	0.14040	0.06930	0.12890
0.06330	0.15230	0.06820	0.14580	0.07290	0.12600	0.06700	0.12870
0.05880	0.15090	0.05970	0.14500	0.05880	0.17200	0.05810	0.17480
0.06340	0.12150	0.06290	0.18380	0.06490	0.13770	0.06160	0.19800
0.06440	0.12290	0.06120	0.28420	0.07210	0.11810	0.06020	0.14650
0.06500	0.14200	0.06810	0.11510	0.06410	0.13110	0.06390	0.16560
0.06870	0.13160	0.06360	0.17190	0.06820	0.13700	0.06560	0.20150
0.06500	0.15680	0.06760	0.12820	0.06470	0.13160	0.06590	0.12690

Table C.12: Training and Testing MSE for 45 to 48 hidden neurons

49 Neurons		50 Neurons		51 Neurons		52 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06230	0.15270	0.06340	0.15070	0.06920	0.13410	0.06190	0.14170
0.05710	0.14650	0.06650	0.13950	0.05830	0.17520	0.05500	0.21690
0.06930	0.11480	0.05950	0.15190	0.06970	0.13840	0.06180	0.13380
0.06150	0.12890	0.05840	0.14650	0.06570	0.15300	0.06140	0.21770
0.05610	0.18450	0.06120	0.14940	0.06200	0.14750	0.06790	0.12280
0.07760	0.10610	0.05950	0.14750	0.06180	0.14790	0.07180	0.13210
0.06250	0.14080	0.06420	0.13870	0.05820	0.13640	0.06150	0.15130
0.06780	0.12640	0.05270	0.15680	0.06410	0.14730	0.06620	0.13430
0.06290	0.17520	0.06860	0.12960	0.06740	0.11670	0.06430	0.14040
0.05810	0.15840	0.06110	0.17270	0.06100	0.18410	0.06100	0.17810
0.06550	0.14770	0.06670	0.21770	0.06120	0.18180	0.06090	0.14490
0.06920	0.11720	0.06940	0.14360	0.06090	0.19000	0.07270	0.09820
0.06700	0.13030	0.06880	0.13320	0.06820	0.14950	0.05490	0.19500
0.07060	0.17070	0.07290	0.13140	0.06030	0.18130	0.05950	0.12680
0.06050	0.19040	0.06540	0.12640	0.06420	0.12760	0.06360	0.13830
0.05690	0.17930	0.06490	0.17650	0.06120	0.15550	0.05680	0.30000
0.06360	0.13130	0.05680	0.20180	0.06430	0.12390	0.05640	0.14000
0.06550	0.16050	0.07030	0.12240	0.07180	0.11250	0.05350	0.18460
0.05950	0.17960	0.06740	0.13050	0.06990	0.12360	0.05450	0.17630
0.06380	0.13350	0.05960	0.13830	0.06040	0.16260	0.06680	0.11810
0.06230	0.22940	0.06560	0.15490	0.05580	0.15940	0.06380	0.14450
0.06350	0.13670	0.05540	0.17860	0.06050	0.19220	0.06400	0.11800
0.06590	0.13000	0.06090	0.13420	0.05700	0.22260	0.06500	0.12260
0.06300	0.14250	0.07290	0.14530	0.05750	0.12080	0.06160	0.15800
0.07430	0.12860	0.06250	0.17020	0.05610	0.14700	0.05930	0.21140
0.05950	0.15760	0.06520	0.15040	0.06060	0.16010	0.06330	0.16090
0.07110	0.10730	0.06620	0.19720	0.06180	0.16700	0.06480	0.12530
0.06880	0.13930	0.06000	0.15900	0.06080	0.11150	0.06510	0.13750
0.06190	0.15820	0.05900	0.15610	0.06120	0.13910	0.06510	0.18520
0.06270	0.15760	0.06080	0.18950	0.05770	0.18690	0.06290	0.13960

Table C.13: Training and Testing MSE for 49 to 52 hidden neurons

53 Neurons		54 Neurons		55 Neurons		56 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06600	0.14900	0.06870	0.13300	0.06730	0.11310	0.06880	0.10860
0.06510	0.14380	0.07040	0.10810	0.06280	0.15350	0.06570	0.14450
0.07090	0.12670	0.06070	0.19350	0.05800	0.19990	0.05760	0.14900
0.06440	0.11670	0.05850	0.13880	0.06440	0.20330	0.06170	0.14650
0.06230	0.13500	0.06840	0.15000	0.06110	0.13450	0.06240	0.13750
0.06670	0.16680	0.06280	0.13870	0.06610	0.13530	0.06620	0.14180
0.06360	0.18340	0.06130	0.13410	0.06030	0.20400	0.06740	0.14400
0.05750	0.17690	0.06660	0.13900	0.06090	0.14780	0.05890	0.17310
0.05830	0.22620	0.06160	0.14580	0.06690	0.12660	0.05540	0.15430
0.05480	0.17860	0.06830	0.15840	0.06070	0.19150	0.05630	0.13540
0.06730	0.18340	0.05620	0.20140	0.06460	0.18170	0.06390	0.15130
0.05740	0.18550	0.06980	0.11850	0.06010	0.17750	0.05590	0.18380
0.06660	0.12670	0.06070	0.14870	0.06880	0.11500	0.07110	0.14260
0.07230	0.10140	0.06160	0.16800	0.06620	0.14710	0.06180	0.22020
0.05450	0.18740	0.06140	0.12650	0.06370	0.15450	0.05610	0.14650
0.05610	0.14230	0.05950	0.14510	0.06290	0.16510	0.06270	0.11630
0.06560	0.12670	0.06580	0.13730	0.06380	0.12770	0.05910	0.15980
0.06540	0.12200	0.05030	0.21050	0.06870	0.12800	0.06980	0.10510
0.05750	0.18670	0.06630	0.12810	0.06340	0.15330	0.06380	0.13800
0.06610	0.14290	0.05810	0.16210	0.06610	0.12420	0.06080	0.19890
0.05870	0.17580	0.05720	0.18840	0.05860	0.14590	0.06080	0.17560
0.06910	0.13660	0.06160	0.18650	0.05850	0.21590	0.05890	0.17760
0.06110	0.15220	0.05710	0.18670	0.06860	0.21520	0.06050	0.18310
0.06120	0.14260	0.05190	0.24950	0.06230	0.13350	0.06070	0.17690
0.06470	0.16870	0.06290	0.18160	0.05580	0.16000	0.05960	0.20290
0.06110	0.18010	0.06350	0.15450	0.06600	0.16560	0.06580	0.12430
0.06270	0.12860	0.06760	0.15340	0.07350	0.15080	0.06620	0.13760
0.06970	0.12610	0.07790	0.12010	0.06700	0.12770	0.05800	0.13160
0.06380	0.19820	0.06630	0.13070	0.06270	0.17430	0.06070	0.14610
0.06530	0.14550	0.06940	0.11660	0.07840	0.16070	0.06040	0.18180

Table C.14: Training and Testing MSE for 53 to 56 hidden neurons

57 Neurons		58 Neurons		59 Neurons		60 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05700	0.23020	0.06620	0.12350	0.06130	0.14920	0.06490	0.12600
0.06770	0.11850	0.06500	0.12260	0.06310	0.15630	0.06870	0.12320
0.05300	0.19120	0.06710	0.11470	0.06570	0.12370	0.05850	0.20040
0.05800	0.19550	0.05880	0.16250	0.05940	0.14950	0.05900	0.20510
0.06730	0.16180	0.06770	0.14940	0.05800	0.16510	0.06060	0.15610
0.06960	0.27060	0.05200	0.15760	0.05360	0.24050	0.05960	0.16830
0.07180	0.15200	0.06450	0.14230	0.06270	0.13370	0.07020	0.18400
0.05330	0.24020	0.06650	0.15210	0.06730	0.18420	0.06330	0.27400
0.06640	0.12860	0.05950	0.16810	0.06780	0.12870	0.05590	0.15180
0.06340	0.12640	0.05550	0.15870	0.06240	0.15360	0.06010	0.18340
0.06290	0.13970	0.06130	0.13300	0.06210	0.19000	0.06690	0.11600
0.05930	0.17760	0.05630	0.18930	0.05620	0.20080	0.06950	0.12190
0.06380	0.22060	0.06170	0.13070	0.07040	0.16510	0.06750	0.17340
0.06020	0.14810	0.06430	0.16130	0.06720	0.12810	0.06150	0.13210
0.06340	0.13530	0.06340	0.15190	0.06000	0.14640	0.06740	0.11060
0.06540	0.14480	0.06350	0.16770	0.06420	0.13330	0.06250	0.17820
0.06130	0.25170	0.06580	0.14330	0.07240	0.14740	0.05730	0.23110
0.06380	0.25020	0.05970	0.27100	0.05630	0.16040	0.06880	0.13750
0.05810	0.19620	0.06430	0.19490	0.06370	0.14140	0.05880	0.16930
0.06490	0.14420	0.06540	0.13410	0.06040	0.16960	0.06870	0.17250
0.06290	0.18190	0.06480	0.13630	0.06030	0.12340	0.06730	0.20490
0.06950	0.13060	0.06410	0.12220	0.05920	0.15130	0.06250	0.13310
0.06490	0.13670	0.05860	0.26940	0.05800	0.15180	0.06720	0.13630
0.05350	0.18360	0.05950	0.15540	0.06310	0.16950	0.06880	0.14830
0.05670	0.24520	0.05910	0.12990	0.05910	0.21750	0.06850	0.12480
0.05710	0.13470	0.06490	0.13780	0.06420	0.13540	0.06180	0.22110
0.07320	0.12810	0.06210	0.14830	0.06650	0.15260	0.07110	0.24750
0.06120	0.13150	0.05230	0.18730	0.06950	0.12690	0.05850	0.17030
0.07090	0.13980	0.06320	0.16960	0.05960	0.16020	0.06310	0.13360
0.06150	0.19840	0.05890	0.17250	0.05760	0.18210	0.05070	0.22420

Table C.15: Training and Testing MSE for 57 to 60 hidden neurons

61 Neurons		62 Neurons		63 Neurons		64 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05860	0.21040	0.05880	0.26340	0.06480	0.14150	0.06240	0.15070
0.05790	0.22370	0.05410	0.16590	0.05720	0.18860	0.06150	0.22010
0.05590	0.21190	0.06080	0.15580	0.05950	0.15510	0.06470	0.16030
0.05680	0.19200	0.06260	0.13550	0.06200	0.14070	0.06060	0.13100
0.06070	0.14200	0.05970	0.13970	0.06610	0.12410	0.06490	0.15720
0.07510	0.18910	0.05920	0.16870	0.06790	0.12580	0.05720	0.17390
0.05480	0.23060	0.05970	0.24340	0.06490	0.14600	0.06310	0.17700
0.06780	0.14380	0.06350	0.12020	0.05800	0.18950	0.06300	0.19070
0.06170	0.15140	0.07300	0.12110	0.05710	0.16200	0.05720	0.26550
0.06310	0.14080	0.06310	0.14680	0.06420	0.14650	0.05920	0.15400
0.06380	0.18290	0.05300	0.19070	0.05850	0.17570	0.05460	0.16720
0.06180	0.13480	0.06370	0.12010	0.06550	0.13810	0.05750	0.16580
0.05900	0.18560	0.06330	0.16610	0.05940	0.13490	0.06350	0.12520
0.06180	0.24460	0.06730	0.12770	0.06260	0.13040	0.05870	0.15310
0.06790	0.11710	0.06870	0.12770	0.05850	0.13650	0.05600	0.16460
0.06650	0.19600	0.05950	0.15180	0.06460	0.17010	0.06580	0.12260
0.05800	0.14360	0.06290	0.17360	0.05480	0.14620	0.06090	0.17040
0.06100	0.17150	0.05570	0.16810	0.06850	0.13510	0.05780	0.15250
0.06020	0.18530	0.05620	0.24490	0.05780	0.12920	0.06020	0.17460
0.06710	0.19550	0.06620	0.13260	0.06910	0.13130	0.05990	0.17760
0.05840	0.15040	0.06310	0.12960	0.05450	0.18590	0.05480	0.17220
0.06550	0.19360	0.06200	0.12870	0.05540	0.17050	0.05380	0.16790
0.05980	0.18140	0.06820	0.18100	0.06160	0.12730	0.06190	0.15980
0.06400	0.16550	0.06580	0.18750	0.06180	0.19570	0.05410	0.21350
0.06160	0.14250	0.05680	0.20810	0.06240	0.15580	0.05820	0.21330
0.06160	0.15250	0.06390	0.15450	0.06290	0.12490	0.05700	0.17480
0.05780	0.14210	0.06090	0.15810	0.06320	0.21650	0.06530	0.16760
0.06570	0.18060	0.06060	0.14060	0.05980	0.14360	0.06670	0.11940
0.06940	0.12090	0.06290	0.16650	0.06510	0.15000	0.06270	0.12870
0.05870	0.15240	0.06400	0.12930	0.06780	0.13860	0.05210	0.19850

Table C.16: Training and Testing MSE for 61 to 64 hidden neurons

65 Neurons		66 Neurons		67 Neurons		68 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06050	0.12620	0.06090	0.18370	0.05950	0.20240	0.06370	0.13030
0.05410	0.16790	0.06480	0.12850	0.05670	0.15970	0.06100	0.18740
0.06710	0.12160	0.05300	0.21860	0.05690	0.17180	0.05410	0.20920
0.06240	0.15120	0.06360	0.21420	0.06750	0.13880	0.05830	0.16940
0.07400	0.12040	0.05550	0.18140	0.05830	0.18860	0.05970	0.19480
0.06170	0.17270	0.04890	0.21390	0.05450	0.15380	0.06380	0.16220
0.06180	0.12580	0.06210	0.14400	0.06550	0.15850	0.05630	0.18710
0.05570	0.17660	0.06080	0.14750	0.06890	0.20150	0.05230	0.21530
0.05660	0.23730	0.05530	0.18350	0.05770	0.17630	0.05880	0.22130
0.05660	0.17480	0.05780	0.15560	0.06020	0.18380	0.06910	0.15340
0.05270	0.22280	0.05580	0.21870	0.05990	0.17740	0.06050	0.15800
0.06720	0.13040	0.06080	0.14970	0.05880	0.18150	0.07070	0.17950
0.06360	0.13050	0.05540	0.15930	0.05840	0.15420	0.05730	0.17660
0.05890	0.15080	0.06480	0.12580	0.05140	0.19130	0.06060	0.13050
0.06040	0.15350	0.05890	0.17390	0.06370	0.14430	0.05640	0.16150
0.06700	0.15750	0.05370	0.22430	0.05930	0.15430	0.05830	0.16440
0.05630	0.26460	0.06000	0.13870	0.06050	0.18840	0.06380	0.17710
0.06630	0.12120	0.05680	0.18930	0.07230	0.19460	0.05780	0.16550
0.06400	0.17610	0.05050	0.15760	0.06040	0.17040	0.05760	0.19120
0.06080	0.14520	0.06370	0.19180	0.06110	0.15140	0.06000	0.17130
0.06340	0.14390	0.05900	0.14830	0.05820	0.32460	0.06540	0.15660
0.05980	0.22140	0.06640	0.17530	0.05280	0.21890	0.06460	0.12050
0.06780	0.25090	0.06010	0.16910	0.06090	0.15240	0.05870	0.14850
0.06130	0.16660	0.06280	0.13880	0.06280	0.19570	0.06040	0.17440
0.05900	0.15030	0.06290	0.15270	0.06600	0.24020	0.05960	0.19740
0.05990	0.19210	0.05810	0.19070	0.06120	0.15780	0.06290	0.13870
0.07100	0.11890	0.06150	0.13040	0.06100	0.14570	0.06120	0.17010
0.05960	0.16370	0.06700	0.14020	0.05670	0.27540	0.06040	0.16730
0.06020	0.16270	0.05970	0.14070	0.06160	0.15680	0.06250	0.12760
0.05330	0.20530	0.06100	0.15030	0.05770	0.15210	0.05860	0.15300

Table C.17: Training and Testing MSE for 65 to 68 hidden neurons

69 Neurons		70 Neurons		71 Neurons		72 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05760	0.14660	0.06670	0.15870	0.06300	0.20000	0.05610	0.19580
0.05980	0.17480	0.05880	0.18410	0.05180	0.19390	0.05880	0.17710
0.06500	0.12960	0.05700	0.19280	0.05850	0.19340	0.05930	0.20990
0.06880	0.15590	0.05660	0.16830	0.05670	0.19540	0.06080	0.22790
0.05790	0.13700	0.05860	0.18310	0.05890	0.16160	0.05830	0.13880
0.06570	0.12070	0.05510	0.19940	0.06110	0.15540	0.05960	0.22200
0.06630	0.14740	0.06070	0.14330	0.05970	0.27830	0.06300	0.19580
0.07110	0.20620	0.06400	0.16850	0.05760	0.16420	0.05670	0.14520
0.05710	0.39190	0.05940	0.14540	0.06160	0.15820	0.05590	0.23580
0.05940	0.14860	0.06240	0.21590	0.05920	0.14890	0.06720	0.14670
0.05010	0.21110	0.05720	0.19600	0.05470	0.21430	0.05990	0.15810
0.05290	0.15220	0.06710	0.13130	0.06630	0.13610	0.06180	0.12180
0.05990	0.15610	0.05490	0.28850	0.05800	0.18770	0.05370	0.20460
0.05420	0.16240	0.05750	0.16480	0.06290	0.10230	0.05830	0.20830
0.06170	0.14340	0.05520	0.18710	0.05460	0.15970	0.05790	0.17290
0.05710	0.22880	0.06470	0.15140	0.05870	0.16640	0.06350	0.29560
0.06270	0.15300	0.05590	0.17710	0.06580	0.15270	0.07060	0.11770
0.05940	0.15690	0.06180	0.14110	0.05580	0.14580	0.06410	0.15640
0.05920	0.15420	0.06120	0.14620	0.06350	0.14860	0.06110	0.12610
0.05200	0.19510	0.06170	0.12960	0.05880	0.14780	0.05940	0.14610
0.05490	0.28510	0.05270	0.19050	0.06130	0.17710	0.05310	0.28150
0.05510	0.20650	0.05800	0.16200	0.05410	0.19740	0.06190	0.16520
0.05760	0.15400	0.05270	0.18400	0.05820	0.14510	0.05770	0.15110
0.05600	0.18520	0.06280	0.18100	0.05260	0.15730	0.05660	0.14400
0.06140	0.15050	0.05860	0.16210	0.06510	0.14310	0.06330	0.13630
0.05110	0.21350	0.05660	0.22380	0.06190	0.14990	0.05730	0.18300
0.06470	0.19820	0.05780	0.17300	0.06490	0.12870	0.05660	0.18850
0.05520	0.15280	0.06350	0.17890	0.06330	0.19790	0.06340	0.19800
0.06070	0.15290	0.05930	0.16340	0.06010	0.15360	0.06590	0.14210
0.05810	0.21300	0.06140	0.16250	0.06060	0.14480	0.06420	0.12430

Table C.18: Training and Testing MSE for 69 to 72 hidden neurons

73 Neurons		74 Neurons		75 Neurons		76 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05540	0.24620	0.07050	0.11910	0.05420	0.17720	0.05860	0.24070
0.06100	0.19210	0.05600	0.20870	0.06410	0.12540	0.05640	0.19380
0.05370	0.17550	0.05320	0.14620	0.06890	0.15610	0.05360	0.17890
0.06100	0.18180	0.05840	0.18150	0.06240	0.22060	0.05470	0.19520
0.05870	0.13120	0.05630	0.19310	0.05710	0.16970	0.05400	0.16320
0.05390	0.16130	0.05620	0.17300	0.05890	0.14640	0.04980	0.18700
0.05710	0.26480	0.06020	0.24590	0.06250	0.13350	0.06330	0.18380
0.05780	0.15810	0.05910	0.15630	0.05500	0.15430	0.06780	0.14240
0.05920	0.13680	0.05570	0.14430	0.05790	0.15700	0.05230	0.22770
0.06740	0.11670	0.05240	0.19850	0.05470	0.15510	0.05130	0.23240
0.06030	0.15700	0.06340	0.24820	0.05630	0.15520	0.05790	0.18230
0.06330	0.12920	0.05930	0.22510	0.05820	0.13780	0.05450	0.17170
0.06040	0.13250	0.05310	0.21090	0.05850	0.15040	0.05170	0.14990
0.06150	0.13460	0.05760	0.30450	0.05560	0.18450	0.05680	0.16540
0.05650	0.17740	0.05580	0.19880	0.05450	0.18710	0.06480	0.13530
0.05170	0.19280	0.05350	0.17120	0.05240	0.22340	0.06230	0.14730
0.05430	0.14480	0.05140	0.17740	0.06050	0.14350	0.06850	0.12610
0.06080	0.22630	0.05990	0.16760	0.05410	0.15130	0.05880	0.14100
0.06260	0.15370	0.05710	0.15270	0.05450	0.18460	0.05990	0.15220
0.06360	0.15310	0.05500	0.31090	0.05910	0.13200	0.05110	0.27760
0.05280	0.17790	0.05540	0.19290	0.05970	0.26730	0.05880	0.18760
0.05740	0.26950	0.05890	0.15480	0.05850	0.14750	0.05370	0.27330
0.06440	0.12240	0.06420	0.20110	0.05150	0.27550	0.05820	0.26390
0.06470	0.19810	0.05350	0.17950	0.05870	0.17350	0.06440	0.12810
0.05260	0.28890	0.05820	0.14910	0.06020	0.18400	0.05390	0.25780
0.06300	0.22200	0.06780	0.15690	0.05070	0.22320	0.06070	0.21100
0.05340	0.16890	0.05600	0.15320	0.05380	0.22810	0.05510	0.17920
0.05950	0.14090	0.05990	0.17250	0.05250	0.21850	0.05880	0.15850
0.05490	0.14260	0.05490	0.15890	0.05880	0.16200	0.05880	0.17520
0.05380	0.26660	0.06060	0.15180	0.06020	0.17570	0.06110	0.18690

Table C.19: Training and Testing MSE for 73 to 76 hidden neurons

77 Neurons		78 Neurons		79 Neurons		80 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05740	0.18500	0.05590	0.14090	0.05350	0.17910	0.05060	0.20420
0.05250	0.19040	0.05670	0.18840	0.05920	0.14420	0.05680	0.15290
0.04910	0.24240	0.05300	0.18440	0.05610	0.30130	0.06080	0.15650
0.06020	0.16610	0.06310	0.14740	0.05750	0.14000	0.06050	0.15880
0.05730	0.20490	0.05980	0.16030	0.05830	0.26640	0.05700	0.16640
0.06040	0.12680	0.06270	0.18880	0.05870	0.14400	0.05850	0.32810
0.06030	0.19690	0.05570	0.16720	0.05830	0.14040	0.05510	0.20960
0.06280	0.12830	0.06470	0.15920	0.05360	0.20830	0.05560	0.16020
0.04950	0.19260	0.05960	0.15320	0.06150	0.16910	0.06440	0.12400
0.06150	0.13520	0.05410	0.19570	0.05200	0.22360	0.05670	0.18120
0.05910	0.29240	0.05600	0.15970	0.05030	0.21150	0.06440	0.13480
0.05620	0.18510	0.05660	0.21850	0.05560	0.14920	0.06090	0.15110
0.05370	0.19240	0.05910	0.15370	0.06070	0.14160	0.05670	0.15890
0.05250	0.18660	0.05870	0.13670	0.05380	0.18490	0.05890	0.18020
0.06060	0.16820	0.05630	0.20160	0.05540	0.21850	0.05720	0.16530
0.06110	0.13000	0.05840	0.17560	0.06190	0.16360	0.05340	0.15730
0.06720	0.11640	0.05890	0.15760	0.05850	0.15850	0.05890	0.17650
0.05830	0.17250	0.05640	0.20700	0.06020	0.25950	0.05570	0.12360
0.06280	0.13930	0.06000	0.15670	0.06030	0.17380	0.05530	0.18530
0.05260	0.18630	0.05490	0.24150	0.05430	0.27980	0.04630	0.20780
0.06240	0.19940	0.06040	0.14680	0.05280	0.19620	0.05540	0.20130
0.05800	0.18440	0.05930	0.13020	0.05860	0.25360	0.05930	0.16840
0.06380	0.12390	0.06500	0.16690	0.05610	0.14900	0.05580	0.23400
0.06130	0.17860	0.06220	0.19950	0.05450	0.16680	0.05790	0.17260
0.05350	0.17190	0.05140	0.15630	0.06020	0.15540	0.04920	0.20940
0.07060	0.13380	0.05320	0.16170	0.05590	0.16230	0.06160	0.13340
0.05520	0.17390	0.04790	0.17100	0.05390	0.26020	0.06510	0.13600
0.06210	0.14290	0.05520	0.15590	0.05280	0.22700	0.06200	0.27340
0.06050	0.18820	0.05880	0.17480	0.05760	0.24530	0.06200	0.16120
0.05560	0.15550	0.05980	0.16710	0.05810	0.15620	0.06310	0.18880

Table C.20: Training and Testing MSE for 77 to 80 hidden neurons

81 Neurons		82 Neurons		83 Neurons		84 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05340	0.23030	0.05140	0.21010	0.04880	0.18720	0.05880	0.15090
0.06120	0.13980	0.06330	0.12240	0.04960	0.18350	0.06030	0.11630
0.06170	0.18590	0.05350	0.16520	0.06240	0.18680	0.05990	0.18170
0.05760	0.24740	0.06920	0.17020	0.05340	0.18950	0.05340	0.20200
0.06300	0.15610	0.05520	0.23700	0.05380	0.18790	0.06030	0.16260
0.05910	0.23470	0.07140	0.12490	0.05960	0.14790	0.05410	0.22750
0.05920	0.15270	0.05500	0.17520	0.05320	0.17830	0.05640	0.16880
0.05720	0.16170	0.05970	0.15690	0.05190	0.17080	0.05640	0.18370
0.06190	0.14090	0.05610	0.33460	0.05120	0.20990	0.05990	0.13570
0.05660	0.14920	0.05260	0.18880	0.05700	0.13130	0.05180	0.20100
0.05870	0.20770	0.05240	0.14830	0.06340	0.16820	0.05070	0.19040
0.06500	0.14100	0.06230	0.16080	0.05570	0.26850	0.05170	0.23780
0.05920	0.30650	0.05550	0.14800	0.05070	0.17330	0.05660	0.23270
0.06260	0.18000	0.05710	0.20420	0.05190	0.19020	0.05210	0.18380
0.05660	0.16080	0.05740	0.20050	0.05480	0.14930	0.05430	0.19400
0.05080	0.22000	0.05600	0.17700	0.05570	0.14830	0.05840	0.19600
0.05930	0.15150	0.05920	0.17300	0.06020	0.14020	0.04930	0.29620
0.05860	0.17240	0.06440	0.13130	0.05800	0.23460	0.06380	0.15580
0.05710	0.17680	0.05980	0.15720	0.05410	0.15180	0.05400	0.20630
0.05840	0.15940	0.04990	0.16550	0.04610	0.19640	0.05470	0.15000
0.05560	0.30120	0.05290	0.18490	0.05870	0.13430	0.05860	0.20840
0.06210	0.14880	0.05330	0.18750	0.04730	0.27050	0.05940	0.13920
0.05700	0.15730	0.05420	0.17890	0.05520	0.15430	0.05540	0.15570
0.05600	0.21490	0.05700	0.21550	0.06520	0.13810	0.05360	0.19260
0.06100	0.14880	0.06010	0.14820	0.05860	0.19730	0.06080	0.15910
0.05700	0.20710	0.05800	0.18900	0.06020	0.13760	0.05710	0.16100
0.06400	0.12550	0.05440	0.18990	0.06000	0.15420	0.06190	0.15660
0.06020	0.20460	0.05490	0.16540	0.05790	0.13580	0.06090	0.14660
0.06140	0.15370	0.05350	0.16960	0.05930	0.13780	0.05130	0.19070
0.06170	0.14560	0.06210	0.13910	0.06170	0.13080	0.05800	0.15150

Table C.21: Training and Testing MSE for 81 to 84 hidden neurons

85 Neurons		86 Neurons		87 Neurons		88 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.04820	0.17310	0.05380	0.17330	0.06610	0.19330	0.05830	0.16480
0.05890	0.14410	0.05500	0.16890	0.05210	0.15810	0.05110	0.16950
0.05310	0.20620	0.05690	0.16250	0.05820	0.17550	0.05140	0.23670
0.05490	0.15620	0.05790	0.16600	0.05530	0.17880	0.06070	0.17230
0.05480	0.15780	0.06100	0.19320	0.05370	0.18550	0.05580	0.21380
0.05680	0.20140	0.06050	0.22140	0.05680	0.16000	0.05500	0.17000
0.05270	0.22580	0.06090	0.24850	0.06270	0.22040	0.05720	0.15510
0.05400	0.14620	0.05650	0.29840	0.05840	0.21610	0.06260	0.22100
0.06230	0.14350	0.05890	0.22610	0.05520	0.22620	0.05080	0.23880
0.05070	0.18380	0.05740	0.13610	0.05080	0.24440	0.05150	0.17690
0.05620	0.17570	0.05330	0.15630	0.05060	0.20650	0.06480	0.14400
0.05790	0.19670	0.05870	0.14540	0.05500	0.18970	0.05430	0.19440
0.05530	0.18380	0.05260	0.23140	0.05820	0.17960	0.05900	0.17150
0.05360	0.17960	0.05560	0.15730	0.05650	0.16850	0.05250	0.17190
0.05320	0.16790	0.04930	0.19650	0.05510	0.14920	0.05480	0.19060
0.05770	0.13420	0.05780	0.13800	0.05540	0.24430	0.06570	0.17070
0.05290	0.22390	0.06060	0.18580	0.05540	0.16510	0.05530	0.18860
0.04970	0.17080	0.06230	0.14250	0.05280	0.18840	0.05110	0.17740
0.05330	0.25560	0.06170	0.17600	0.05780	0.17060	0.05860	0.24760
0.05660	0.17390	0.05920	0.18830	0.05500	0.15930	0.05250	0.21860
0.05170	0.19940	0.05380	0.21920	0.05580	0.20610	0.05820	0.19100
0.06000	0.15800	0.05460	0.31520	0.06510	0.13480	0.05490	0.23200
0.05520	0.17060	0.05730	0.22890	0.05180	0.16700	0.05470	0.35150
0.06230	0.13570	0.05080	0.35260	0.06210	0.13580	0.05410	0.17710
0.05070	0.17280	0.05980	0.18400	0.05240	0.24630	0.05650	0.20740
0.06040	0.13990	0.04990	0.17740	0.05070	0.21210	0.04660	0.24850
0.05900	0.18580	0.05300	0.22810	0.05470	0.15360	0.05250	0.17440
0.05970	0.13100	0.06310	0.16140	0.05140	0.17040	0.05160	0.16840
0.05530	0.14820	0.05410	0.21960	0.05330	0.21090	0.05590	0.22750
0.05630	0.18630	0.05140	0.20370	0.05470	0.15360	0.05350	0.33890

Table C.22: Training and Testing MSE for 85 to 88 hidden neurons

89 Neurons		90 Neurons		91 Neurons		92 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05440	0.15930	0.05060	0.24890	0.05290	0.19780	0.05610	0.15110
0.05990	0.11920	0.05270	0.18830	0.05540	0.17350	0.04970	0.22940
0.05710	0.21250	0.05500	0.16630	0.04900	0.25430	0.05400	0.16120
0.06370	0.27400	0.05630	0.16130	0.05480	0.26890	0.05610	0.13480
0.04690	0.28140	0.05630	0.16410	0.06110	0.20720	0.05400	0.18240
0.06130	0.14950	0.05290	0.20740	0.06480	0.18980	0.06020	0.14380
0.05370	0.28160	0.05690	0.17110	0.05180	0.20800	0.05260	0.15440
0.06510	0.16360	0.05350	0.16740	0.05800	0.24710	0.05290	0.17080
0.05230	0.16320	0.05610	0.15420	0.05150	0.19890	0.05560	0.16780
0.05430	0.22050	0.05850	0.22340	0.05240	0.21550	0.05130	0.15940
0.05180	0.23330	0.04750	0.20100	0.05320	0.18310	0.05930	0.13120
0.05660	0.16540	0.05440	0.20980	0.06150	0.16350	0.04800	0.21850
0.05930	0.17750	0.05340	0.17830	0.05750	0.15300	0.05670	0.22950
0.06440	0.14450	0.05540	0.25810	0.06240	0.18320	0.04790	0.28390
0.05980	0.18060	0.05590	0.16910	0.05780	0.18020	0.04690	0.21300
0.05440	0.16620	0.05520	0.14600	0.05960	0.12960	0.06510	0.13350
0.05940	0.12320	0.05620	0.14180	0.05370	0.27310	0.05630	0.12300
0.05300	0.40250	0.05670	0.17380	0.05870	0.12870	0.04980	0.25570
0.06100	0.12190	0.05700	0.14890	0.04470	0.26520	0.04940	0.19830
0.06290	0.13290	0.05500	0.13840	0.05420	0.36550	0.05730	0.15630
0.05970	0.16980	0.05100	0.18980	0.05320	0.21270	0.05050	0.21010
0.05270	0.24470	0.05610	0.26470	0.05890	0.24840	0.05930	0.22630
0.05770	0.17820	0.05270	0.19280	0.06190	0.13090	0.05830	0.16830
0.06470	0.23370	0.05570	0.17740	0.05860	0.21290	0.05240	0.40400
0.05280	0.20650	0.06000	0.14010	0.04800	0.22730	0.06130	0.18190
0.06070	0.13100	0.05660	0.12630	0.06720	0.11780	0.04950	0.20820
0.05700	0.19160	0.06230	0.20610	0.05570	0.20280	0.05490	0.20770
0.05470	0.18930	0.05670	0.23830	0.05750	0.17050	0.06110	0.12320
0.05320	0.18200	0.05920	0.16530	0.05290	0.15440	0.05240	0.18680
0.05320	0.15430	0.05690	0.22210	0.05580	0.16120	0.05380	0.15820

Table C.23: Training and Testing MSE for 89 to 92 hidden neurons

93 Neurons		94 Neurons		95 Neurons		96 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.05800	0.16830	0.06240	0.17800	0.05420	0.16230	0.05780	0.29310
0.05940	0.13620	0.05220	0.20980	0.05850	0.15580	0.06360	0.20060
0.05160	0.24900	0.05190	0.24250	0.04710	0.20840	0.05280	0.21230
0.05580	0.13370	0.05610	0.19020	0.05490	0.15280	0.06020	0.16320
0.05080	0.17800	0.05460	0.16620	0.05440	0.16070	0.05360	0.18080
0.05170	0.17820	0.05630	0.15140	0.06300	0.13850	0.05410	0.17430
0.05340	0.16210	0.06510	0.24460	0.05770	0.19110	0.05280	0.18900
0.04830	0.18480	0.05430	0.23640	0.05360	0.16810	0.06560	0.15170
0.05950	0.16360	0.05610	0.17800	0.05400	0.22260	0.05980	0.21150
0.05900	0.13710	0.05930	0.15010	0.06580	0.19340	0.05260	0.21120
0.05290	0.18430	0.05990	0.18480	0.05130	0.17690	0.05360	0.14360
0.05390	0.16550	0.05940	0.14420	0.05170	0.16590	0.05980	0.13460
0.04970	0.20280	0.05730	0.15570	0.05090	0.21500	0.05080	0.22100
0.04860	0.22910	0.05830	0.15280	0.05650	0.23900	0.05750	0.24130
0.05280	0.20660	0.05150	0.20380	0.05060	0.22310	0.05450	0.15110
0.05790	0.16010	0.04750	0.21870	0.05840	0.18000	0.06130	0.15910
0.05080	0.20770	0.07200	0.21370	0.05170	0.22150	0.05640	0.16890
0.05600	0.16930	0.05110	0.20100	0.04820	0.22110	0.05500	0.21570
0.04760	0.16100	0.04790	0.17610	0.05850	0.19880	0.05970	0.18560
0.05680	0.16850	0.05970	0.14720	0.06210	0.12530	0.05820	0.15660
0.05520	0.33080	0.04550	0.23720	0.05700	0.17510	0.04950	0.25740
0.05060	0.22610	0.05350	0.16390	0.04760	0.29570	0.05070	0.19350
0.06230	0.20720	0.05780	0.14080	0.05110	0.31160	0.05650	0.19850
0.05790	0.27320	0.05650	0.30660	0.04910	0.22120	0.05360	0.16880
0.06490	0.20120	0.05100	0.27150	0.05280	0.15580	0.05960	0.14920
0.06010	0.24530	0.05510	0.18470	0.05230	0.16770	0.04320	0.29470
0.05980	0.22740	0.05490	0.21600	0.06050	0.17100	0.05390	0.17410
0.05740	0.22150	0.05810	0.20810	0.05960	0.19500	0.05580	0.18270
0.05960	0.16480	0.05520	0.19080	0.05730	0.19300	0.05090	0.36000
0.06010	0.16090	0.05410	0.20140	0.05430	0.21470	0.05570	0.19250

Table C.24: Training and Testing MSE for 93 to 96 hidden neurons

97 Neurons		98 Neurons		99 Neurons		100 Neurons	
Training MSE	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**	**Training MSE**	**Testing MSE**
0.06010	0.21880	0.06010	0.19060	0.05320	0.18540	0.05970	0.22170
0.05480	0.17520	0.04460	0.34530	0.05300	0.18500	0.05540	0.26820
0.05680	0.17280	0.05020	0.25460	0.05410	0.19590	0.05580	0.22880
0.05330	0.24560	0.05760	0.17860	0.05110	0.35140	0.05050	0.18870
0.04470	0.23350	0.06980	0.14790	0.05770	0.21350	0.05560	0.23480
0.04730	0.19390	0.05230	0.22340	0.05180	0.20140	0.06270	0.14600
0.04810	0.21910	0.04940	0.17210	0.05230	0.16840	0.05760	0.22150
0.05520	0.17800	0.05480	0.23310	0.04670	0.20460	0.05310	0.17910
0.06470	0.13700	0.05350	0.20370	0.04680	0.20500	0.04880	0.22570
0.05500	0.19670	0.04770	0.27060	0.04830	0.20090	0.05250	0.19810
0.05080	0.16780	0.06240	0.14170	0.04970	0.21630	0.05060	0.21310
0.05850	0.18580	0.05470	0.23170	0.06220	0.23530	0.05420	0.32330
0.05240	0.19010	0.04910	0.20470	0.05540	0.21140	0.05230	0.19980
0.06560	0.18590	0.06000	0.16980	0.05900	0.16180	0.05420	0.22700
0.05690	0.17910	0.06030	0.18970	0.06020	0.22330	0.05560	0.20330
0.05170	0.22290	0.05870	0.20240	0.05510	0.16730	0.05730	0.14430
0.05930	0.16300	0.05890	0.24060	0.05950	0.16020	0.05250	0.21770
0.05430	0.18380	0.04700	0.22520	0.05250	0.20580	0.05460	0.16880
0.05650	0.24110	0.05870	0.14520	0.04920	0.18350	0.04460	0.21640
0.05170	0.23100	0.06380	0.17950	0.05650	0.16330	0.05460	0.21440
0.05470	0.18550	0.05250	0.18440	0.05450	0.19030	0.05660	0.21170
0.05710	0.17310	0.04780	0.25810	0.05380	0.14990	0.05370	0.18340
0.05070	0.16370	0.05410	0.21020	0.06090	0.15040	0.05510	0.17920
0.05260	0.20210	0.04740	0.20130	0.05350	0.20310	0.05510	0.32660
0.05740	0.17500	0.05410	0.15040	0.05080	0.17000	0.05330	0.16240
0.05210	0.18200	0.05610	0.15990	0.05520	0.25300	0.04990	0.19090
0.05640	0.15880	0.06000	0.19920	0.04920	0.23400	0.05750	0.18770
0.06510	0.19370	0.05620	0.14920	0.05790	0.17780	0.05220	0.17600
0.05810	0.12640	0.05630	0.18640	0.05680	0.21650	0.05910	0.15860
0.06030	0.21120	0.04810	0.20180	0.05270	0.17930	0.06090	0.14290

Table C.25: Training and Testing MSE for 97 to 100 hidden neurons

References

[1] V. Abreu-Sernandez and C. Garcia-Mateo. Adaptive Multi-Rate Speech Coder for VoIP Transmission. In *Electronics Letters*, volume 36, pages 1978–1980, 9 November 2000.

[2] S. Agrawal, J. Ramamirtham, and R. Rastogi. Design of Active and Passive Probes for VoIP Service Quality Monitoring. In *12th International Telecommunications Network Strategy and Planning Symposium, 2006. NETWORKS 2006.*, pages 1–6, November 2006.

[3] Mousa Al-Akhras and H. Zedan. Quality Estimation for Streamed VoIP Services. In *The 2nd European Young Researchers Workshop on Service Oriented Computing*, 2007.

[4] I. Aleksander and H. Morton. *An Introduction to Neural Computing.* Van Nostrand Reinhold Co., New York, NY, USA, 1990.

[5] J. Allnatt. Subjective Rating and Apparent Magnitude. *International Journal Man -Machine Studies*, 7:801–816, 1975.

[6] R. Arora. Voice over IP : Protocols and Standards. Technical report, Washington University in St. Louis.

[7] R. Arora. Voice over IP: Products, Services and Issues. Technical report, Washington University in St. Louis.

[8] A. Asosheh and M.S. Bahaei. Adaptive Voice over IP Network with Selective Dropping Mechanism. In *4th International Conference on 3G Mobile Communication Technologies, 2003. 3G 2003.*, pages 404–408, 25-27 June 2003.

[9] B. Atal and J. Remde. A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates. In *IEEE International Conference on*

Acoustics, Speech, and Signal Processing, ICASSP '82, volume 7, pages 614–617, May 1982.

[10] B. Atal and M. Schroeder. Stochastic Coding of Speech at Very Low Bit-Rate. In *International Conference Communication*, page 16101613, Amsterdam, The Netherlands, 1984.

[11] A. Bearden, L. Denby, B. Karacali, J. Meloche, and D.T. Stott. Experiences With Evaluating Network QoS for IP Telephony. In *Tenth IEEE International Workshop on Quality of Service, 2002.*, pages 259–268, 15-17 May 2002.

[12] F. Beritelli, S. Casale, and G. Ruggeri. New Speech Processing Issues in IP Telephony. In *International Conference on Communication Technology Proceedings, 2000. WCC - ICCT 2000.*, volume 1, pages 652–656, 21-25 August 2000.

[13] A.A. Bilhaj and K. Mase. Endpoint Admission Control Enhanced Systems for VoIP Networks. In *International Symposium on Applications and the Internet, 2004. Proceedings. 2004*, pages 269–272, 2004.

[14] M.S. Borella, D. Swider, S. Uludag, and G.B. Brewster. Internet Packet Loss: Measurement and Implications for End-to-End QoS. In *Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing: Proceedings of the 1998 ICPP Workshops on*, pages 3–12, 14 August 1998.

[15] L. Bos and S. Leroy. Toward an All-IP-Based UMTS System Architecture. *IEEE Network*, 15(1):36–45, January-February 2001.

[16] C. Boutremans and J.-Y. Le Boudec. Adaptive Joint Playout Buffer and FEC Adjustement for Internet Telephony. In *IEEE Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2003*, volume 1, pages 652–662, 30 March-3 April 2003.

[17] M.C. Caesar, S. Balaraman, and D. Ghosal. A Comparative Study of Pricing Strategies for IP Telephony. In *IEEE Global Telecommunications Conference, 2000. GLOBECOM '00.*, volume 1, pages 344–349, 27 November-1 December 2000.

[18] D. Cassing. C6x Solutions for Voice over IP Gateway. In *Northcon/98 Conference Proceedings*, pages 74–85, 21-23 October 1998.

[19] Xiuzhong Chen, Chunfeng Wang, Dong Xuan, Zhongcheng Li, Yinghua Min, and Wei Zhao. Survey on QoS Management of VoIP. In *International Conference on Computer Networks and Mobile Computing, 2003. ICCNMC 2003.*, pages 69–77, 20-23 October 2003.

[20] Fong Loong Chong, I.V. Mcloughlin, and K. Pawlikowski. A Methodology for Improving PESQ Accuracy for Chinese Speech. In *TENCON 2005 2005 IEEE Region 10*, pages 1–6, November 2005.

[21] H.M. Chong and H.S. Matthews. Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems. In *IEEE International Symposium on Electronics and the Environment*, 2004.

[22] Switzerland Christian Ghisler, C. Ghisler & Co. Total commander 5.53.

[23] Cisco.com. Understanding Delay in Packet Voice Networks.

[24] Daniel Collins. *Carrier Grade Voice over IP*. McGraw-Hill Companies, 2nd edition, 2003.

[25] A.E. Conway. A Passive Method for Monitoring Voice-over-IP Call Quality with ITU-T Objective Speech Quality Measurement Methods. In *IEEE International Conference on Communications, 2002. ICC 2002.*, volume 4, pages 2583–2586, 28 April-2 May 2002.

[26] A.E. Conway. Output-based Method of Applying PESQ to Measure the Perceptual Quality of Framed Speech Signals. In *IEEE Wireless Communications and Networking Conference, 2004. WCNC. 2004*, volume 4, pages 2521–2526, 21-25 March 2004.

[27] S.D. Crocker. Internet-Based Collaboration in 2010. *IEEE Internet Computing*, 4(1):53–55, January-February 2000.

[28] H. Cruickshank, A. Sanchez, Zhili Sun, and B. Carro. Voice over IP over Satellite Links. In *The 8th IEEE International Conference on Electronics, Circuits and Systems, 2001. ICECS 2001.*, volume 1, pages 473–476, 2-5 September 2001.

[29] Data-Compresssion.com. Vector quantization.

[30] L. Ding and R.A. Goubran. Assessment of Effects of Packet Loss on Speech Quality in VoIP. In *Proceedings. of the 2nd IEEE Internatioal Workshop on Haptic, Audio and Visual Environments and their Applications, 2003. HAVE 2003*, pages 49–54, 20-21 September 2003.

[31] L. Ding and R.A. Goubran. Speech Quality Prediction in VoIP Using the Extended E-Model. In *IEEE Global Telecommunications Conference, 2003. GLOBECOM '03.*, volume 7, pages 3974–3978, 1-5 December 2003.

[32] B. Duysburgh, S. Vanhastel, B. De Vreese, C. Petrisor, and P. Demeester. On the Influence of Best-Effort Network Conditions on the Perceived Speech Quality of VoIP Connections. In *Proceedings. Tenth International Conference on Computer Communications and Networks, 2001.*, pages 334–339, 15-17 October 2001.

[33] R. El-Haddadeh, G.A. Taylor, and S.J. Watts. Towards Scalable End-to-End QoS Provision for VoIP applications. In *IEE Telecommunications Quality of Services: The Business of Success, 2004. QoS 2004.*, pages 132–135, 2-3 March 2004.

[34] A. Estepa, R. Estepa, and J.M. Vozmediano. On the Suitability of the E-Model to VoIP Networks. In *Proceedings of Seventh International Symposium on Computers and Communications,2002. ISCC 2002.*, pages 511–516, 1-4 July 2002.

[35] ETSI. ETSI Tech. Report (ETR) 250 - Speech Communication Quality from Mouth to Ear of 3.1 kHz Handset Telephony Across Networks. Technical report, European Telecommunications Standards Institute, 1996.

[36] ETSI. Digital Cellular Telecommunications System (Phase 2);Adaptive Multi-Rate (AMR) Speech Transcoding. Technical report, European Telecommunications Standards Institute, 2000.

[37] ETSI. Digital Cellular Telecommunications System (Phase 2);Enhanced Full Rate (EFR) Speech Transcoding. Technical report, European Telecommunications Standards Institute, 2000.

[38] ETSI. Digital Cellular Telecommunications System (Phase 2);Half Rate Speech. Technical report, European Telecommunications Standards Institute, 2000.

[39] ETSI. Digital Cellular Telecommunications System (Phase 2);Full Rate Speech. Technical report, European Telecommunications Standards Institute, 2001.

[40] T. Ezaki, H. Kawakami, and K. Asatani. A New VoIP Call Admission Control Scheme with Use of Alternate Routing for Low Call Loss Probability. In *IEEE International Conference on Communications, 2004*, volume 4, pages 2209–2213, 20-24 June 2004.

[41] T.H. Falk and Wai-Yip Chan. Nonintrusive Speech Quality Estimation Using Gaussian Mixture Models. *IEEE Signal Processing Letters*, 13(2):108–111, February 2006.

[42] T.H. Falk and Wai-Yip Chan. Single-Ended Speech Quality Measurement Using Machine Learning Methods. *IEEE Transactions on Audio, Speech & Language Processing*, 14(6):1935–1947, 2006.

[43] P.K. Fellbaum. Human Speech Production Based on a Linear Predictive Vocoder. Technical report, Brandenburg Technical University of Cottbus.

[44] Behrouz A. Forouzan. *Data Communications and Networking*. McGraw-Hill, Inc., New York, NY, USA, 3rd edition, 2003.

[45] Qiang Fu, Kechu Yi, and Mingui Sun. Speech Quality Objective Assessment Using Neural Network. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP '00.*, volume 3, pages 1511–1514, 5-9 June 2000.

[46] P. Galiotos, T. Dagiuklas, and D. Arkadianos. QoS Management for an Enhanced VoIP Platform using R-factor and Network Load Estimation Functionality. In *5th IEEE International Conference on High Speed Networks and Multimedia Communications*, pages 305–314, 3-5 July 2002.

[47] I.A. Gerson and M.A. Jasiuk. Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kbps. In *International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90*, pages 461–464, 3-6 April 1990.

[48] J.D. Gibson and B. Wei. Tandem Voice Communications: Digital Cellular, VoIP, and Voice over Wi-Fi. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, volume 2, pages 617 – 621, 29 November-3 December 2004.

[49] J. Glasmann, W. Kellerer, and H. Muller. Service Development and Deployment in H.323 and SIP. In *Sixth IEEE Symposium on Computers and Communications, 2001.*, pages 378–385, 3-5 July 2001.

[50] I. Goetz. IT & 3GPP Evolution Convergence: Opportunities for Reduced Costs. In *Fifth IEE International Conference on 3G Mobile Communication Technologies, 2004. 3G 2004.*, volume 3, 2004.

[51] I. Goetz. VoIP Interconnect and Quality Impacts on 3G Core Network Evolution. In *Fifth IEE International Conference on 3G Mobile Communication Technologies, 2004. 3G 2004.*, pages 534–536, 2004.

[52] G. Gonzalez Lopez, Q. Wang, M.A. Abu-Rgheff, and A. Akram. A MIP-SIP Macro-Mobility Management Scheme for VoIP Across Wired and Wireless Domains. In *IEE Telecommunications Quality of Services: The Business of Success, 2004. QoS 2004.*, pages 114–118, 2-3 March 2004.

[53] B. Goodman. Internet Telephony and Modem Delay. *IEEE Network*, 13(3):8–16, May-June 1999.

[54] P. Gray, M.P. Hollier, and R.E. Massara. Non-Intrusive Speech-Quality Assessment Using Vocal-Tract Models. *IEE Proceedings Vision, Image and Signal Processing*, 147(6):493–501, December 2000.

[55] E.J. Ha and J.T. Park. Voice Traffic Multiplexing Scheme with Guaranteed QoS between VolP Access Routers Using DiffServ. In *Electronics Letters*, volume 37, pages 828–829, 21 June 2001.

[56] David Haccoun, Paul Cohen, and Hoang Hai-Hoc. An Experimental Investigation of the Active-Idle Pattern of Speech over Land Mobile Radio Telephone Channels. *IEEE Transactions on Vehicular Technology*, 32(4):260–268, November 1983.

[57] F. Hammer, P. Reichl, and T. Ziegler. Where Packet Traces Meet Speech Samples: An Instrumental Approach to Perceptual QoS Evaluation of VoIP. In *Twelfth IEEE International Workshop on Quality of Service, 2004. IWQOS 2004.*, pages 273–280, 7-9 June 2004.

[58] A. Haojun, Zhang Xinchen, Hu Ruimin, and Tu Weiping. A Wideband Speech Codecs Quality Measure Based on Bark Spectrum Distance. In *Proceedings of*

2004 International Symposium on Intelligent Signal Processing and Communication Systems, 2004. ISPACS 2004., pages 155–158, 18-19 Nov. 2004.

[59] Simon Haykin. *Neural Networks: a Comprehensive Foundation.* Prentice Hall, 2nd edition, 1998.

[60] Simon Haykin and Barry Van Veen. *Signals and Systems.* Wiley, October 2002.

[61] Jialong He. Calling c functions. Technical report, Arizona State University.

[62] F.P. Heiman. A Wireless LAN Voice over IP Telephone System. In *Northcon/98 Conference Proceedings*, pages 52–54, 21-23 October 1998.

[63] T. Hoshi, K. Tanigawa, and K. Tsukada. Proposal of a Method of Voice Stream Multiplexing for IP Telephony Systems. In *Internet Workshop, 1999. IWS 99*, pages 182–188, 18-20 February 1999.

[64] IEC. *Voice and Fax over Internet Protocol (V/FoIP).* International Engineering Consortium.

[65] IETF. *SIP: Session Initiation Protocol.* Internet Engineering Task Force (IETF), May 2002.

[66] ITU-T. *Recommendation G.711 - Pulse Code Modulation (PCM) of Voice Frequencies.* International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), November 1988.

[67] ITU-T. *Recommendation G.726 - 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM).* International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), December 1990.

[68] ITU-T. *Recommendation G.728 - Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction.* International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), September 1992.

[69] ITU-T. *Recommendation G.165 - Echo Cancellers.* International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 1993.

[70] ITU-T. *Recommendation G.723.1 - Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 1996.

[71] ITU-T. *Recommendation G.729 - Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 1996.

[72] ITU-T. *Recommendation P.800 - Methods for Subjective Determination of Transmission Quality*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), August 1996.

[73] ITU-T. *Recommendation P.50 Appendix I - Artificial Voices, Appendix I: Test Signals*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), February 1998.

[74] ITU-T. *Recommendation P.861 - Objective Quality Measurement of Telephoneband (300-3400 Hz) Speech Codecs*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), February 1998.

[75] ITU-T. *Recommendation G.109 - Definition of Categories of Speech Transmission Quality*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), September 1999.

[76] ITU-T. *Recommendation G.113 Appendix I - Provisional Planning Values for the Equipment Impairment Factor Ie*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), September 1999.

[77] ITU-T. *Recommendation P.50 - Artificial Voices*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), September 1999.

[78] ITU-T. *Recommendation G.107 - The E-model, a Computational Model for use in Transmission Planning*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 2000.

[79] ITU-T. *Recommendation P.862 - Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of*

Narrow-Band Telephone Networks and Speech Codecs. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), February 2001.

[80] ITU-T. *Recommendation G.113 Appendix I - Provisional Planning Values for the Equipment Impairment Factor Ie and Packet-Loss Robustness Factor Bpl*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), May 2002.

[81] ITU-T. *Recommendation G.114 - One-Way Transmission Time*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), May 2003.

[82] ITU-T. *Recommendation G.114 Appendix II - Guidance on One-Way Delay for Voice over IP*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), September 2003.

[83] ITU-T. *Recommendation P.800.1 - Mean Opinion Score (MOS) Terminology*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 2003.

[84] ITU-T. *Recommendation G.107 - The E-model, a Computational Model for use in Transmission Planning*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 2005.

[85] ITU-T. *Recommendation P.862 - Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs, Amendment 2: Revised Annex A: Reference Implementations and Conformance Testing for Recommendations P.862, P.862.1 and P.862.2*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), November 2005.

[86] ITU-T. *Recommendation P.862.1-Mapping Function for Transforming P.862 Raw Result Scores to MOS-LQO*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), March 2005.

[87] ITU-T. *Recommendation G.723.1 - Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*. International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), May 2006.

[88] J.H. James, Bing Chen, and L. Garrison. Implementing VoIP: A Voice Transmission Performance Progress Report. *IEEE Communications Magazine*, 42(7):36–41, July 2004.

[89] W. Jiang and H. Schulzrinne. Modeling of Packet Loss and Delay and their Effect on Real-Time Multimedia Service Quality. In *Proc. NOSSDAV*, 2000.

[90] I. Johansson, T. Frankkila, and P. Synnergren. Bandwidth Efficient AMR Operation For VoIP. In *Proceedings. IEEE Workshop Speech Coding, 2002*, pages 150–152, 6-9 October 2002.

[91] Hong-Goo Kang, Hong Kook Kim, and Richard V. Cox. Improving the Transcoding Capability of Speech Coders. *IEEE Transactions on Multimedia*, 5(1):24–33, 2003.

[92] A. Kansal and A. Karandikar. Adaptive Delay Estimation for Low Jitter Audio over Internet. In *IEEE Global Telecommunications Conference, 2001. GLOBECOM '01.*, volume 4, pages 2591–2595, 25-29 November 2001.

[93] S. Kaukonen, J. Sauvola, and T. Ojala. Agent-Based Conferencing Using Mobile IP-Telephony. In *IEEE 3rd Workshop on Multimedia Signal Processing, 1999*, pages 593–598, 13-15 September 1999.

[94] B. Keepence. Quality of Service for Voice over IP. In *IEE Colloquium on Services Over the Internet - What Does Quality Cost? (Ref. No. 1999/099)*, pages 4/1–4/4, 23 June 1999.

[95] Bhumip Khasnabish. *Implementing Voice over IP.* Wiley-Interscience, 2nd edition, 2003.

[96] Doh-Suk Kim and A. Tarraf. Perceptual Model for Non-intrusive Speech Quality Assessment. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, volume 3, pages iii–1060–3, 17-21 May 2004.

[97] Doh-Suk Kim and A. Tarraf. Enhanced Perceptual Model for Non-Intrusive Speech Quality Assessment. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006*, volume 1, pages I–I, 14-19 May 2006.

[98] Hyogon Kim, Inhye Kang, and Eenjun Hwang. Measurement-Based Multi-Call Voice Frame Grouping in Internet Telephony. *IEEE Communications Letters*, 6(5):199–201, May 2002.

[99] M. Kodialam, T.V. Lakshman, and S. Sengupta. Traffic-Oblivious Routing for Guaranteed Bandwidth Performance. *Communications Magazine, IEEE*, 45(4):46–51, April 2007.

[100] A. M. Kondoz. *Digital Speech Coding for Low Bit Rate Communication Systems.* John Wiley and Sons Ltd, New York, NY, USA, 2nd edition, 2004.

[101] A. Lakaniemi, J. Rosti, and V.I. Raisanen. Subjective VoIP Speech Quality Evaluation Based on Network Measurements. In *IEEE International Conference on Communications, 2001. ICC 2001.*, volume 3, pages 748–752, 11-14 June 2001.

[102] N. Laoutaris and I. Stavrakakis. Adaptive Play out Strategies for Packet Video Receivers with Finite Buffer Capacity. In *IEEE International Conference on Communications, 2001. ICC 2001.*, volume 3, pages 969–973, 11-14 June 2001.

[103] L.-A. Larzon, H. Hannu, L.-E. Jonsson, and K. Svanbro. Efficient Transport of Voice over IP over Cellular links. In *IEEE Global Telecommunications Conference, 2000. GLOBECOM '00.*, volume 3, pages 1669–1676, 27 November-1 December 2000.

[104] Moon-Keun Lee, Sung-Kyo Jung, Hong-Goo Kang, Young-Cheol Park, and Dae-Hee Youn. A Packet Loss Concealment Algorithm Based on Time-Scale Modification for CELP-type Speech Coders. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, volume 1, pages I–116–I–119, 6-10 April 2003.

[105] F.F. Li. Speech Intelligibility of VoIP to PSTN Interworking - A Key Index for the QoS. In *IEE Telecommunications Quality of Services: The Business of Success, 2004. QoS 2004.*, pages 104–108, 2-3 March 2004.

[106] W. Li and R.F. Kubichek. Output-based Objective Speech Quality Measurement Using Continuous Hidden Markov Models. In *Proceedings of Seventh International Symposium on Signal Processing and Its Applications, 2003.*, volume 1, pages 389–392, 1-4 July 2003.

[107] Y.J. Liang, E.G. Steinbach, and B. Girod. Multi-stream Voice over IP Using Packet Path Diversity. In *IEEE Fourth Workshop on Multimedia Signal Processing, 2001*, pages 555–560, 3-5 Oct. 2001.

[108] Wanjiun Liao. Mobile Internet Telephony: Mobility Extension to H.323. *IEEE Transactions on Vehicular Technology*, 50(6):1403–1414, November 2001.

[109] Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications, [legacy, pre - 1988]*, 28(1):84–95, January 1980.

[110] Linktionary.com. QoS (Quality of Service).

[111] Hong Liu and P. Mouchtaris. Voice over IP Signaling: H.323 and Beyond. *IEEE Communications Magazine*, 38(10):142–148, October 2000.

[112] C. Low. The Internet Telephony Red Herring. In *IEEE Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity*, pages 72–80, 18-22 November 1996.

[113] G. Madre, E.H. Baghious, S. Azou, and G. Burel. Design of a Variable Rate Algorithm for CS-ACELP Coder. In *IEEE International Conference on Communications, 2003. ICC '03.*, volume 1, pages 600–604, 11-15 May 2003.

[114] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessment of VoIP Quality over Internet Backbones. In *IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. INFOCOM 2002*, volume 1, pages 150–159, 23-27 June 2002.

[115] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessing the Quality of Voice Communications over Internet Backbones. *IEEE/ACM Transactions on Networking*, 11(5):747–760, October 2003.

[116] K. Mase. Toward Scalable Admission Control for VoIP Networks. *IEEE Communications Magazine*, 42(7):42–47, July 2004.

[117] K. Mase and Y. Toyama. End-to-End Measurement Based Admission Control for VoIP Networks. In *IEEE International Conference on Communications, 2002. ICC 2002.*, volume 2, pages 1194–1198, 28 April-2 May 2002.

[118] K. Mase, Y. Toyama, A.A. Bilhaj, and Y. Suda. QoS Management for VoIP Networks with Edge-to-Edge Admission Control. In *IEEE Global Telecommunications Conference, 2001. GLOBECOM '01.*, volume 4, pages 2556–2560, 25-29 November 2001.

[119] M. Masugi. QoS Mapping of VoIP Communication Using Self-Organizing Neural Network. In *IEEE Workshop on IP Operations and Management, 2002*, pages 13–17, 2002.

[120] MathWorks. Levenberg-marquardt (trainlm)-speed and memory comparison.

[121] MathWorks. Mx array manipulation.

[122] H.S. Matthews. Energy Impacts of Wired and Wireless Networks. In *IEEE International Symposium on Electronics and the Environment*, San Francisco, CA, 2002.

[123] P. Mayorga and L. Besacier. Voice over IP and Vocal Recognition. In *3rd International Conference onElectrical and Electronics Engineering, 2006*, pages 1–4, September 2006.

[124] P. Mayorga, L. Besacier, R. Lamy, and J.-F. Serignat. Audio Packet Loss over IP and Speech Recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003*, pages 607–612, 30 November-3 December 2003.

[125] J. W. McGowan. Burst Ratio: A Measure of Bursty Loss on Packet-Based Networks. United States Patent 6,931,017 B2, August 2005.

[126] Microsoft.com. Quality of Service Defined.

[127] A. Miloslavski, V. Antonov, L. Yegoshin, S. Shkrabov, J. Boyle, G. Pogosyants, and N. Anisimov. Third-party Call Control in VoIP Networks for Call Center Applications. In *2001 IEEE Intelligent Network Workshop*, pages 161–167, 6-9 May 2001.

[128] S. K. Mitra. *Digital Signal Processing: A Computer Based Approach.* McGraw-Hill, Inc., New York, NY, USA, 1997.

[129] Samir Mohamed, Gerardo Rubino, and Martín Varela. Performance Evaluation of Real-Time Speech Through a Packet Network: A Random Neural Networks-Based Approach. *Performance Evaluation*, 57(2):141–161, 2004.

[130] Y.S. Moon, C.C. Leung, K.N. Yuen, H.C. Ho, and X. Yu. A CRM Model Based on Voice over IP. In *2000 Canadian Conference on Electrical and Computer Engineering*, volume 1, pages 464–468, 7-10 March 2000.

[131] J.K. Muppala, T. Bancherdvanich, and A. Tyagi. VoIP Performance on Differentiated Services Enabled Network. In *Proceedings. IEEE International Conference on Networks, 2000. (ICON 2000)*, pages 419–423, 5-8 September 2000.

[132] Nachiappan Nachiappan and Fredrik Sjoqvist. Survey of Voice over IP (VoIP). Technical report, Standford University, 2004.

[133] M. Narbutt and L. Murphy. Improving Voice over IP Subjective Call Quality. *IEEE Communications Letters*, 8(5):308–310, May 2004.

[134] M.E. Nasr and S.A. Napoleon. On Improving Voice Quality Degraded by Packet Loss in Data Networks. In *Proceedings of the Twenty-Second National Radio Science Conference, 2005. NRSC 2005.*, pages 465–472, March 15-17, 2005.

[135] Elias Nemer. Handling VoIP Speech Coding Challenges: Part 1, October 2002.

[136] Omid Omidvar and Judith Dayhoff, editors. *Neural Networks and Pattern Recognition*. Academic Press, Inc., Orlando, FL, USA, 1997.

[137] OPTICOM GmbH and PsyTechnics Limited. Pesq project.

[138] Jr. Ortiz, S. Internet Telephony Jumps off the Wires. *Computer*, 37(12):16–19, December 2004.

[139] F. Palmieri. Improving the Performance in Multimedia Streaming Networks: A Differentiated Service Approach. In *4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications, 2003.*, volume 2, pages 841–849, 2-5 July 2003.

[140] C. Perkins, O. Hodson, and V. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Audio. *IEEE Network*, 12(5):40–48, September-October 1998.

[141] D. Picovici and A.E. Mahdi. Output-Based Objective Speech Quality Measure Using Self-Organizing Map. In *Proceedings of IEEE International Conference*

on Acoustics, Speech, and Signal Processing, 2003. (ICASSP '03), volume 1, pages I–476–I–479, 6-10 April 2003.

[142] D. Picovici and A.E. Mahdi. New Output-based Perceptual Measure for Predicting Subjective Quality of Speech. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. (ICASSP '04)*, volume 5, pages V–633–6, 17-21 May 2004.

[143] Zizhi Qiao, L. Sun, N. Heilemann, and E.C. Ifeachor. A New Method for VoIP Quality of Service Control Use Combined Adaptive Sender Rate and Priority Marking. In *2004 IEEE International Conference on Communications*, volume 3, pages 1473–1477, 20-24 June 2004.

[144] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals.* Englewood Cliffs: Prentice Hall, 1978.

[145] Adil Raja, R. Muhammad Atif Azad, Colin Flanagan, Dorel Picovici, and Conor Ryan. Non-Intrusive Quality Evaluation of VoIP Using Genetic Programming. In *1st Bio-Inspired Models of Network, Information and Computing Systems, 2006.*, pages 1–8, December 2006.

[146] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks. In *INFOCOM '94. Networking for Global Communications. 13th Proceedings IEEE*, pages 680–688, 12-16 June 1994.

[147] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. Perceptual Evaluation of Speech Quality (PESQ)-A New Method for Speech Quality Assessment of Telephone Networks and Codecs. In *Proceedings. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. (ICASSP '01)*, volume 2, pages 749–752, 7-11 May 2001.

[148] B. Rohani and H.-J. Zepernick. Frame Erasure Pattern Feedback for Real-Time Perceptual Quality Estimation. In *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia*, volume 1, pages 110–113, 15-18 December 2003.

[149] B. Rohani and H.-J. Zepernick. An Efficient Method for Perceptual Evaluation of Speech Quality in UMTS. In *Proceedings Systems Communications, 2005.*, pages 185–190, 14-17 August 2005.

[150] J. Rosenberg, J. Lennox, and H. Schulzrinne. Programming Internet Telephony Services. *IEEE Network*, 13(3):42–49, May-June 1999.

[151] R. Salami, C. Laflamme, and J.-P. Adoul. 8 kbit/s ACELP Coding of Speech with 10 ms Speech-Frame: A Candidate for CCITT Standardization. *icassp*, 3:97–100, 1994.

[152] P.M. Santos, R. Balbinot, J.G. Silveira, and F.C. Castello. Analysis of Packet Loss Correction and Concealment Algorithms in Robust Voice over IP Environments. In *IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2003. PACRIM.*, volume 2, pages 824–827, 28-30 August 2003.

[153] M. Schroeder and B. Atal. Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '85*, 1985.

[154] H. Schulzrinne and J. Rosenberg. Signaling for Internet Telephony. In *Proceedings of the Sixth International Conference on Network Protocols, 1998.*, pages 298–307, 13-16 October 1998.

[155] H. Schulzrinne and J. Rosenberg. The IETF Internet Telephony Architecture and Protocols. *IEEE Network*, 13(3):18–23, May-June 1999.

[156] H. Schulzrinne and E. Wedlund. Application-Layer Mobility Using SIP. In *IEEE Service Portability and Virtual Customer Environments, 2000*, pages 29–36, 1 December 2000.

[157] J. Scourias. Overview of the Global System for Mobile Communications. Technical report, University of Wateloo, Canada.

[158] Jeong Wook Seo, Se Jeong Woo, and Keun Sung Bae. A Study on the Application of an AMR Speech Codec to VoIP. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. ICASSP '01*, volume 3, pages 1373–1376, 7-11 May 2001.

[159] A. Shallwani and P. Kabal. An Adaptive Playout Algorithm with Delay Spike Detection for Real-Time VoIP. In *Canadian Conference on Electrical and Computer Engineering, 2003. IEEE CCECE 2003.*, volume 2, pages 997–1000, 4-7 May 2003.

[160] Xueshan Shan and J.J. Li. A Case Study of IP Network Monitoring Using Wireless Mobile Devices. In *Proceedings. Tenth International Conference on Computer Communications and Networks, 2001.*, pages 590–593, 15-17 October 2001.

[161] C. Shao and M. Bouchard. Efficient Classification of Noisy Speech Using Neural Networks. In *Proceedings. Seventh International Symposium on Signal Processing and Its Applications, 2003*, volume 1, pages 357–360, 1-4 July 2003.

[162] A.S. Spanias. Speech Coding: A Tutorial Review. *Proceedings of the IEEE*, 82(10):1541–1582, October 1994.

[163] SPSS.com. SPSS v 12.0.

[164] Ion Stoica and Hui Zhang. Providing Guaranteed Services Without Per Flow Management. In *SIGCOMM*, pages 81–94, 1999.

[165] L. Sun. *Speech Quality Prediction for Voice over Internet Protocol Networks.* PhD thesis, School of Computing, Communications and Electronics, University of Plymouth, U.K., January 2004.

[166] L. Sun and E.C. Ifeachor. Perceived Speech Quality Prediction for Voice over IP-Based Networks. In *IEEE International Conference on Communications, 2002. ICC 2002.*, volume 4, pages 2573–2577, 28 April-2 May 2002.

[167] L. Sun and E.C. Ifeachor. Prediction of Perceived Conversational Speech Quality and Effects of Playout Buffer Algorithms. In *IEEE International Conference on Communications, 2003. ICC '03.*, volume 1, pages 1–6, 11-15 May 2003.

[168] L. Sun and E.C. Ifeachor. New Models for Perceived Voice Quality Prediction and their Applications in Playout Buffer Optimization for VoIP Networks. In *IEEE International Conference on Communications, 2004*, volume 3, pages 1478–1483, 20-24 June 2004.

[169] L. Sun and E.C. Ifeachor. Voice Quality Prediction Models and their Application in VoIP Networks. *IEEE Transactions on Multimedia*, 8(4):809–820, August 2006.

[170] L. Sun, G. Wade, B. Lines, and E.C. Ifeachor. Impact of Packet Loss Location on Perceived Speech Quality. In *Proceedings of 2nd IP-Telephony Workshop (IPTEL 01), Columbia University*, page 114 122, 6-10 April 2001.

[171] K. Svanbro, J. Wiorek, and B. Olin. Voice-over-IP-over-Wireless. In *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on*, volume 1, pages 24–28, 18-21 September 2000.

[172] H.P. Sze, S.C. Liew, J.Y.B. Lee, and D.C.S. Yip. A Multiplexing Scheme for H.323 Voice-over-IP Applications. *IEEE Journal on Selected Areas in Communications*, 20(7):1360–1368, September 2002.

[173] A. Takahashi. Opinion Model for Estimating Conversational Quality of VoIP. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, volume 3, pages iii–1072–5, 17-21 May 2004.

[174] A. Takahashi, A. Kurashima, and H. Yoshino. Objective Assessment Methodology for Estimating Conversational Quality in VoIP. *IEEE Transactions on Audio, Speech & Language Processing*, 14(6):1984–1993, 2006.

[175] A. Takahashi, H. Yoshino, and N. Kitawaki. Perceptual QoS Assessment Technologies for VoIP. *IEEE Communications Magazine*, 42(7):28–34, July 2004.

[176] Gaurav Talwar, Robert F. Kubichek, and Hongkang Liang. Hiddenness Control of Hidden Markov Models and Application to Objective Speech Quality and Isolated-Word Speech Recognition. In *Fortieth Asilomar Conference on Signals, Systems and Computers, 2006. ACSSC '06.*, pages 1076–1080, October-November 2006.

[177] Telegeography.com. Skype Registers 100 Million Users.

[178] Kuo-Kun Tseng, Yuan-Cheng Lai, and Ying-Dar Lin. Perceptual Codec and Interaction Aware Playout Algorithms and Quality Measurement for VoIP

Systems. *IEEE Transactions on Consumer Electronics*, 50(1):297–305, Feb 2004.

[179] Kuo-Kun Tseng and Yin-Dar Lin. User Perceived Codec and Duplex Aware Playout Algorithms and LMOS-DMOS Measurement for Real Time Streams. In *International Conference on Communication Technology Proceedings, 2003. ICCT 2003.*, volume 2, pages 1666–1669, 9-11 April 2003.

[180] A. Tyagi, J.K. Muppala, and H. De Meer. VoIP Support on Differentiated Services Using Expedited Forwarding. In *Conference Proceeding of the IEEE International Performance, Computing, and Communications Conference, 2000. IPCCC '00.*, pages 574–580, 20-22 February 2000.

[181] H. Uzunalioglu, K.P. Das, and D.J. Houck. An Architecture and Admission Control Algorithm for Multi-Precedence Voice over IP (VoIP) Calls. In *IEEE Military Communications Conference, 2004. MILCOM 2004.*, volume 2, pages 906–912, 31 October-3 November 2004.

[182] G. Van Hoey, S. Van den Bosch, P. de La Vallee-Poussin, H. De Neve, and G. Petit. Capacity Planning Strategies for Voice-over-IP Traffic in the Core Network. In *IEEE Workshop on High Performance Switching and Routing*, pages 109–112, 29-31 May 2001.

[183] P. Van Mieghem. A Lower Bound for the End-To-End Delay in Networks: Application to Voice over IP. In *IEEE Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration.*, volume 4, pages 2508–2513, 8-12 November 1998.

[184] R. Venkatesha Prasad, A. Sangwan, H.S. Jamadagni, M.C. Chiranth, R. Sah, and V. Gaurav. Comparison of Voice Activity Detection Algorithms for VoIP. In *Seventh International Symposium on Computers and Communications, 2002. Proceedings. ISCC 2002.*, pages 530–535, 1-4 July 2002.

[185] S. Voran. Objective Estimation of Perceived Speech Quality-Part I: Development of the Measuring Normalizing Block Technique. *IEEE Transactions on Speech and Audio Processing*, 7(4):371–382, July 1999.

[186] S. Voran. Objective Estimation of Perceived Speech Quality-Part II: Evaluation of the Measuring Normalizing Block Technique. *IEEE Transactions on Speech and Audio Processing*, 7(4):383–390, July 1999.

[187] S. Vuong and Y. Bai. A Survey of VoIP Intrusions and Intrusion Detection Systems. In *The 6th International Conference on Advanced Communication Technology*, 2004.

[188] Toshihiko Wakahara, Yao Meng, Kamugisha Kazaura, Mitsuji Matsumoto, and Takao Shimizu. A Voice Quality Measurement Method for PHS over IP Communication. In *AINA '04: Proceedings of the 18th International Conference on Advanced Information Networking and Applications*, page 261, Washington, DC, USA, 2004. IEEE Computer Society.

[189] Chungyih-Yih Wang, Ching-Chi Hsu, and Yennun Huang. VORAL: a System for Voice over IP Routing in Application Layer. In *Proceedings of Seventh IEEE Real-Time Technology and Applications Symposium, 2001.*, pages 165–170, 30 May-1 June 2001.

[190] Runsheng Wang and Xiaorui Hu. VoIP Development in China. *Computer*, 37(9):30–37, September 2004.

[191] Wei Wang, S.C. Liew, Qixiang Pang, and V.O.K. Li. A Multiplex-Multicast Scheme that Improves System Capacity of Voice-over-IP on Wireless LAN by 100%. In *Proceedings Ninth International Symposium on Computers and Communications (ISCC), 2004.*, volume 1, pages 472–477, 28 June-1 July 2004.

[192] Wikipedia.org. Skype, 2005.

[193] Wipro.com. Speech codecs-a white paper. Technical report, Wipro.com, 2001.

[194] Jason Woodard. 64 kbits/s PCM Codecs. Technical report, University of Southampton.

[195] Jason Woodard. ADPCM Codecs. Technical report, University of Southampton.

[196] Jason Woodard. Hybrid Codecs. Technical report, University of Southampton.

[197] Jason Woodard. Source Codecs. Technical report, University of Southampton.

[198] Jason Woodard. Speech Coding. Technical report, University of Southampton.

[199] Jason Woodard. The 4.8 kbits/s DoD CELP Codec. Technical report, University of Southampton.

[200] Jason Woodard. The Basic Properties of Speech. Technical report, University of Southampton.

[201] Jason Woodard. The G728 Low Delay CELP Codec. Technical report, University of Southampton.

[202] Jason Woodard. The GSM Codec. Technical report, University of Southampton.

[203] Jason Woodard. Waveform Codecs. Technical report, University of Southampton.

[204] Xiaotao Wu and H. Schulzrinne. Location-Based Services in Internet Telephony. In *IEEE Second Consumer Communications and Networking Conference, 2005. CCNC. 2005*, pages 331–336, 3-6 January 2005.

[205] Yichuan Wu, Jianping Zheng, and Kui Huang. The Performance Analysis and Simulation of Header Compression in VoIP Services over Cellular Links. In *The 9th Asia-Pacific Conference on Communications, 2003. APCC 2003.*, volume 2, pages 644–647, 21-24 September 2003.

[206] Yang Xu, M. Westhead, and F. Baker. An Investigation of Multilevel Service Provision for Voice over IP Under Catastrophic Congestion. *IEEE Communications Magazine*, 42(6):94–100, June 2004.

[207] T. Yensen, M. Parperis, R. Goubran, and I. Lambndaris. Echo Target Determination Using Acoustic Round Trip Delay for Voice Over IP Conferences. In *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999. ISCAS '99.*, volume 4, pages 348–351, 30 May-2 June 1999.

[208] T. Yensen, M. Parperis, I. Lambadaris, and R. Goubran. Determining Acoustic Round Trip Delay for VoIP Conferences. In *IEEE Second Workshop on Multimedia Signal Processing, 1998*, pages 161–166, 7-9 December 1998.

[209] X. Yu, J.W. Modestino, and X. Tian. The Accuracy of Gilbert models in Predicting Packet-Loss Statistics for a Single-Multiplexer Network Model. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2005.*, volume 4, pages 2602–2612, 13-17 March 2005.

[210] S. Zeadally and F. Siddiqui. Design and Implementation of a SIP-Based VoIP Architecture. In *18th International Conference on Advanced Information Networking and Applications*, volume 2, Los Alamitos, 2004. IEEE COMPUTER SOC.

[211] Hongbing Zhang, Liehua Xie, Joonbum Byun, P. Flynn, and Choon Shim. Packet Loss Burstiness and Enhancement to the E-Model. In *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005.*, pages 214–219, 23-25 May 2005.

[212] Yuan Zhang. SIP-Based VoIP Network and its Interworking with the PSTN. In *Electronics & Communication Engineering Journal*, volume 14, pages 273–282, December 2002.

[213] Xin Zhong, J. Arrowood, A. Moreno, and M. Clements. Multiple Description Coding for Recognizing Voice over IP. In *Proceedings of IEEE 10th Digital Signal Processing Workshop and the 2nd Signal Processing Education Workshop*, pages 383–386, 13-16 October 2002.

[214] Hua Zou, Hongman Wang, Wenxin Mao, Bai Wang, S. Focant, K. Handekyn, D. Chantrain, and N. Marly. Prototyping SIP-Based VoIP Services in Java. In *International Conference on Communication Technology Proceedings, 2000. WCC - ICCT 2000.*, volume 2, pages 1395–1399, 21-25 August 2000.

[215] E.E. Zurek, J. Leffew, and W.A. Moreno. Objective Evaluation of Voice Clarity Measurements for VoIP Compression Algorithms. In *Proceedings of the Fourth IEEE International Caracas Conference on Devices, Circuits and Systems, 2002.*, pages T033–1–T033–6, 17-19 April 2002.