2011-11-22

A Network Conditions Estimator for Voice Over IP Objective Quality Assessment

Carlos Daniel Nocito
University of Miami, c.nocito@umiami.edu

UNIVERSITY OF MIAMI

A NETWORK CONDITIONS ESTIMATOR FOR Voice Over IP OBJECTIVE QUALITY ASSESSMENT

By

Carlos Daniel Nocito

A THESIS

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science

Coral Gables, Florida

December 2011

UNIVERSITY OF MIAMI

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science

A NETWORK CONDITIONS ESTIMATOR FOR Voice Over IP OBJECTIVE QUALITY ASSESSMENT

Carlos Daniel Nocito

Approved:

_____	_____
Michael Scordilis, Ph.D. Research Assistant Professor Department of Electrical and Computer Engineering	Terri A. Scandura, Ph.D. Dean of the Graduate School
_____	_____
Manohar N. Murthi, Ph.D. Assistant Professor Department of Electrical and Computer Engineering	Subramanian Ramakrishnan, Ph.D. Associate Professor Department of Mathematics

NOCITO, CARLOS (B.S., Electrical Engineering)

A Network Conditions Estimator
For Voice Over IP Objective
Quality Assessment

(December 2011)

Abstract of a thesis at the University of Miami.

Thesis supervised by Professor Michael Scordilis.
No. of pages in text. (74)

Objective quality evaluation is a key element for the success of the emerging Voice over IP (VoIP) technologies. Although there are extensive economic incentives for the convergence of voice, data, and video networks, packet networks such as the Internet have inherent incompatibilities with the transport of real time services. Under this paradigm, network planners and administrators are interested in ongoing mechanisms to measure and ensure the quality of these real time services.

Objective quality assessment algorithms can be broadly divided into a) intrusive (methods that require a reference signal), and b) non intrusive (methods that do not require a known reference signal). The latter group, typically requires knowledge of the network conditions (level of delay, jitter, packet loss, etc.), and that has been a very active area of research in the past decade. The state of the art methods for objective non-intrusive quality assessment provide high correlations with the subjective tests.

Although good correlations have been achieved already for objective non-intrusive quality assessment, the current large voice transport networks are in a hybrid state, where the necessary network parameters cannot easily be observed from the packet traffic between nodes. This thesis proposes a new process, the Network Conditions Estimator (NCE), which can serve as bridge element to real-world hybrid networks.

Two classifications systems, an artificial neural network and a C4.5 decision tree, were developed using speech from a database collected from experiments under controlled network conditions. The database was composed of a group of four female speakers and three male speakers, who conducted unscripted conversations without knowledge about the details of the experiment. Using mel frequency cepstral coefficients (MFCCs) as the feature-set, an accuracy of about 70% was achieved in detecting the presence of jitter or packet loss on the channel.

This resulting classifier can be incorporated as an input to the E-Model, in order to properly estimate the QoS of a network in real time. Additionally, rather than just providing an estimation of subjective quality of service provided, the NCE provides an insight into the cause for low performance.

To my parents, my sister, and my nieces.

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude and appreciation to my thesis supervisor and chairperson of the committee, Dr. Michael Scordilis. Also, my thanks go to Dr. Manohar Murthi of the Department of Electrical and Computer Engineering, and Dr. Subramanian Ramakrishnan of the Mathematics Department, for accepting the appointment to the dissertation committee, as well as for their helpful suggestions and support. I extend my thanks to Professor Reuven Lask, Dr. Maria Martinez and Dr. James Modestino of the Department of Electrical and Computer Engineering for extensive support during my studies. I would like to thank my friends and colleagues whom I have met and known while attending the University of Miami. Finally, I extend my utmost gratitude to my family for their support, encouragement and love, which made this work possible.

Carlos Daniel Nocito

University of Miami

December 2011

TABLE OF CONTENTS

Page

LIST OF FIGURES vii

LIST OF TABLES viii

Chapter

1 INTRODUCTION 1
1.1 Introduction to VoIP Networks 2
1.1.1 Traditional TDM Networks 2
1.1.2 Voice over Packet Networks 2
1.1.3 Media Transport Protocol (RTP) 4
1.1.4 Codecs 4
1.2 Phenomena affecting Quality on VoIP 8
1.2.1 Delay and Jitter 9
1.2.1.1 Coder Delay (Fixed) 10
1.2.1.2 Packetization Delay (Fixed) 11
1.2.1.3 Queuing/Buffering Delay (Variable) 12
1.2.1.4 Propagation Delay and Network Switching Delay (Variable). 12
1.2.1.5 De-Jitter Delay (Fixed or Variable) 13
1.2.2 Overview of Packet Loss 13
1.3 Existing Correction Algorithms 14
1.3.1 Anti-Jitter Buffers 15
1.3.2 Forward Error Correction (FEC) and Packet Loss Concealment 16
1.4 Estimating Network Conditions by Observing PCM Signals 17
1.4.1 Motivation 17
1.4.2 Outline of Thesis 19
1.4.3 Contributions 20

2 RELATED WORK 22
2.1 Quality of Evaluation of VoIP 22
2.2 Objective vs. Subjective Quality Evaluation 22
2.3 Intrusive vs. Non-Intrusive Quality Evaluation 23
2.4 State of the Art for Objective Evaluation Algorithms 24
2.4.1 PSQM (Intrusive) 24
2.4.2 PAMS (Intrusive) 25
2.4.3 PESQ (Intrusive) 27
2.4.4 E-Model 29
2.4.5 Artificial Intelligence Approaches 31

3 PROPOSED APPROACH 33
3.1 Applications of Network Conditions Estimator 33
3.2 Theoretical Background 34
3.2.1.1 Artificial Neural Networks 34
3.2.1.2 C4.5 Decision Trees 35
3.2.2 Feature Extraction 36
3.2.2.1 Multi-Resolutions Filterbanks 37
3.2.2.2 Mel-Frequency Cepstral Coefficients (MFCC) 39
3.2.2.3 Linear Prediction Coefficients (LPC) 40
3.2.2.4 Linear Prediction Cepstrum Coefficients (LPCC) 41
3.3 Exploiting LPC Codecs for Feature Extraction 41
3.4 Training Considerations 43

4 EXPERIMENTAL SETUP 45
4.1 General Considerations for Experimental Setup 46
4.2 Asterisk PBX 47
4.3 Netem 48
4.4 Eyebeam Softphone 50
4.5 Monitoring, Analysis, and Capture Tools: Wireshark and Ping 51
4.6 Training Set Generation Procedure 55

5 EXPERIMENTAL RESULTS 56
5.1 Classifiers Used 57
5.2 MFCC Performance 58
5.2.1 Speaker Dependent Performance 58
5.2.2 Multi-Speaker Accuracy 59
5.2.3 Accuracy Discerning Between Different Network Deficiencies 60
5.3 LPCs and LPCCs Performance 61
5.3.1 Speaker Dependent Accuracy 61
5.3.2 Speaker Independent Accuracy 62
5.3.3 Accuracy Discerning Between Different Network Deficiencies 63
5.4 LPCs Extracted from the G.729 Codec Payload 63

6 CONCLUSION AND FUTURE WORK 65
6.1 Overview of Work Conducted 65
6.2 Conclusions 66
6.3 Future Work 67

REFERENCES…… 69

LIST OF FIGURES

Figure 1.1 Analysis by Synthesis Coder 6
Figure 1.2 Signal Flow at the CS-ACELP Decoder, from ITU-T Recommendation G.729E (January 2007) 8
Figure 1.3 Effects of Packet Loss 14
Figure 1.4 Mechanics of the Anti-jitter Buffer 15
Figure 1.5 Voice Routed Over a Heterogeneous Network 19
Figure 2.1 PSQM Model 25
Figure 2.2 PAMS Model 26
Figure 2.3 PESQ Model 28
Figure 3.1 Combination of the NCE with the E-Model 33
Figure 3.2 Neural Network Structure used for the Classifier 35
Figure 3.3 Sample Decision Tree 36
Figure 3.4 Typical Symmetric Filterbank 37
Figure 3.5 Typical Asymmetric Filterbank 38
Figure 3.6 Frequency Response of the Prototype Low-Pass Filter used in the Filterbank Analysis 39
Figure 3.7 Filter for Generating MFCCs, with Band-Limiting Between 300 Hz to 3400 Hz [Wong01] 40
Figure 4.1 Experimental Network Setup 48
Figure 4.2 Signaling and Media Packet Flow 48
Figure 4.3 Architecture of Emulator Server 49
Figure 4.4 Typical Scripted Jitter Emulation 49
Figure 4.5 The Code Necessary to Implement the Script 50
Figure 4.6 Codec Options for Eyebeam Softphone 51
Figure 4.7 Typical Wireshark Capture of Traffic 52
Figure 4.8 Typical Analysis Window in Wireshark 53
Figure 4.9 Wireshark Screen Allowing the Dump of Payload to a File 53
Figure 4.10 Output of the Ping Command During one of the Experiments 54
Figure 4.11 Audio Capturing Scheme 55

LIST OF TABLES

Table 1.1 Impact of One Way Transmission Delay on Voice Services Quality..... 9
Table 1.2 Typical Processing Delays for Various Codecs..... 10
Table 1.3 Typical Packetization Delays for Various Codecs 11
Table 2.1 Reference Values for MOS Results [RAAKE2007]..... 22
Table 2.2 Default E-Model Values for G.711 Codec 29
Table 3.1 Payload Content of a Transmitted RTP Packet [ITUT-TG.729E]... 42
Table 5.1 Comparison of Accuracy Between Neural Network Classifier and C4.5 Classification Tree When Classifying Between Normal Conditions and High Jitter..... 57
Table 5.2 Experimental Results for a Single Male Speaker 58
Table 5.3 Experimental Results for an Individual Female Speaker..... 59
Table 5.4 Experimental Results for a System Trained and Tested on a Male Speaker..... 62
Table 5.5 Experimental Results for a System Trained and Tested on a Female Speaker..... 62
Table 5.6 Summary of Performance Obtained with LPCs Features Extracted From The G.729 Codecs Payload 64

CHAPTER 1: Introduction

As the tendency towards network convergence increases year after year, Voice Over IP (VoIP) has became a critical technology for companies in the telecom sector (i.e., cable companies using VoIP to provide telephone service to their subscribers through their existing networks) [1]. VoIP is also a very active area of research, due to the inherent contradictions between the packet network architecture of the Internet and the real time requirements of voice communications [2].

Traditionally, Time Domain Multiplexing (TDM) circuits provided a dedicated link between subscribers, making issues such as delay or delay jitter (the variation of delay over time) negligible. Packet networks, on the other hand, cannot guarantee the transit time of packets, especially under high congestion scenarios. Additionally, the use of congestion control algorithms on routers introduces both delay and jitter that are much greater than those encountered on TDM circuits.

Several algorithms have been developed to deal with the problem of jitter, and are referenced in [3] and [4]. Such algorithms involve many non-linear operations, such as time scaling of the frames playback, frame interpolation, and prediction from past frames. This, in addition to the time varying nature of the channel, makes it very hard to find automated approaches to the evaluation of quality for VoIP. This thesis presents a novel process, the Networks Condition Estimator, which has the potential of being combined with existing algorithms to improve their dependency on certain attributes, such as reference signals or network parameters, which may not always be available.

1.1 Introduction to VoIP Networks

In this section, an overview of the technologies that make VoIP possible is included. In particular, the RTP protocol, the standard compression algorithms, and the most popular compression algorithms are reviewed in order to provide the necessary background to understand the challenges of VoIP.

1.1.1 Traditional TDM Networks

For many decades, TDM based Wide Area Networks (WANs) enabled the transport of a multitude of user applications. Under a TDM scheme, a circuit is divided into a continuous stream of time slots and multiple channels are multiplexed into the circuit. These circuits provide reliable and low delay dedicated connections for services such as voice, data and video transport. The basic channel bandwidth is 64 Kbits/second, which derives from the requirement to transmit a voice call sampled at 8 KHz and quantized with 8 bits [5].

Circuit switching technologies, such as TDM, provide a temporary dedicated connection between two stations, no matter how many switching devices the data are routed through. A connection is maintained until broken (when one side hangs up), which means bandwidth is reserved regardless of the nature of the information being transmitted on the channel (silence, audio, noise, etc).

1.1.2 Voice over Packet Networks

In recent years, the tendency towards network convergence has lead to an increased interest in the transport of real time services (such as voice or video) over shared packet

networks [6]. In Voice over Internet Protocol (VoIP) technologies, voice signals are digitized, packetized and then transmitted through a best effort packet network, such as the Internet. VoIP has many commercial advantages over TDM networks, as it makes use of the massive packet network infrastructure developed for the internet, effectively implementing a statistical multiplexing between different services, including voice, data and video.

Goode in [7] summarized the challenges of transporting voice service over packet networks, which arise from the higher delay, delay variation, packet loss, and typically high compression algorithms to conserve bandwidth. Typically, Internet applications use the TCP/IP protocol suite for their operation, where the IP protocol provides a connectionless best effort network communications protocol, and TCP protocol provides a reliable transport, which uses acknowledgments and retransmissions to ensure the delivery of information. This suite, nonetheless, is not fit for the real time transmission of speech, because the acknowledgment and retransmission process results in excessive end-to-end delay. For this reason, VoIP technologies use the UDP/RTP suite, where UDP provides an *unreliable* connectionless delivery service using IP to transport messages between end points. Real Time Transport Protocol (RTP), used in conjunction with UDP, provides end-to-end network transport functions for applications transmitting real time data, such as audio and video. However, RTP does not reserve resources and does not guarantee quality of service [7].

1.1.3 Media Transport Protocol (RTP)

The Real Time Media Transport Protocol was standardized in 2003 in Request for Comments 3550 [8], and is the technical foundation of VoIP. RTP services include:

- Payload type identification: Indicates the type of content being transported, such as audio or video and the codec type
- Sequence numbering: The Protocol Data Unit (PDU) number of the packet, to keep track of the order in which packet must be played out
- Time stamping: A time stamp, to allow synchronization and jitter calculations
- Payload: Binary representation of a number of coded audio frames

RTP by itself does not provide a mechanism to ensure timely delivery or Quality of Service (QoS, defined in ITU T Recommendation E.800 [9] as the ““collective degree of service performance””). Also, although the order of the packets is tracked, out of order delivery is possible on jittery network connections, as will be explained in later sections. The companion protocol RTCP does allow monitoring of a link, but most VoIP applications offer continuous stream of RTP/UDP/IP packets, without regard to packet loss or delay in reaching the receiver [7].

1.1.4 Codecs

The term codec refers to the combination of encoders and decoders used in order to reduce the bandwidth requirements over limited capacity channels. At the transmitter, the encodec takes an analog voice signal from an input such as a microphone and transforms the signal into a digital format that can be appended as the payload of an RTP packet. On the receiver, the decoder takes that payload content of the RTP packet and converts it

back to an analog signal for play out to the receiving user. Spanias provides a very comprehensive overview on the coding technologies in use [10].

The objective of speech coding is to obtain representations of the speech signal that maximizes perceptual quality but minimizes the number of required bits. The broadest division among speech coding algorithms is between waveform quantization and parametric quantization, where the first group represents the signal by quantization of the time samples, and the second group represents the signal by a binary representation of a speech model or spectral parameters [11].

The simplest nonparametric coding technique is Pulse Code Modulation (PCM), which is simply a quantizer of sampled amplitudes. ITU T Recommendation G.711 [12] standardizes the u Law and A Law compression algorithms, used in the United States and Europe, respectively. Both algorithms use logarithmic scale quantization and are considered both uncompressed and reference for quality comparison with other algorithms. Appendixes I (September 1999) and II (February 2000) of ITU T recommendation G.711 incorporated Packet Loss Concealment (PLC) and Discontinuous Transmission (DTX) mechanisms for the G.711 codec, making it fit for VoIP applications [13].

Parametric codecs, such as ITU T G.729 [14], typically involve an analysis synthesis process. In the analysis stage, speech is represented by a compact set of parameters that are encoded efficiently. At the synthesis stage, these parameters are decoded, and used in conjunction with a reconstruction algorithm to recreate a speech signal. The analysis stage can be open loop or closed loop. In the case of close loop analysis, the parameters are extracted by minimizing an objective metric, typically the

square of the error between the original signal and the reconstructed speech [10]. Since close loop analysis usually involves the reconstructed speech, it is also called analysis by synthesis. Figure 1.1 illustrates a typical analysis by synthesis coder, where we can observe that the signal X~[n], the reconstructed speech, is recursively compared with the original signal X[n], in order to find the correct excitation in a code book that minimizes of the perceptual difference.

Some of the most commonly used codecs in the industry today are ITU T G.729, ITU T G.723.1 [15] and iLBC [16], all of which use Linear Prediction Coefficients (LPC) for the representation of the speech signals, and construct the signal using the analysis by synthesis process described above. In the particular case of G.729, the autocorrelation coefficients are computed on 10 ms windows, and converted to LP coefficients using the Levinson Durbin algorithm. The exact equations and algorithms involved in this process are available in Chapter 3 of ITU T recommendation G.729E [14].

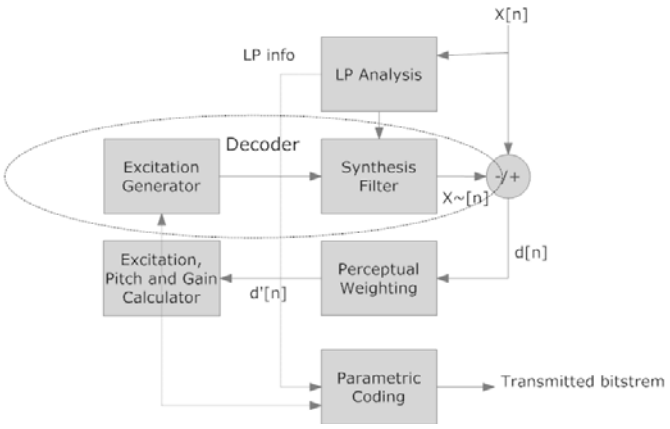

Figure 1.1: Analysis by Synthesis Coder

The function of the decoder consists then of decoding the transmitted parameters (LP parameters, adaptive codebook vector, fixed codebook vector, and gains) and performing synthesis to obtain the reconstructed speech, followed by a post processing stage, consisting of an adaptive postfilter and a fixed highpass filter [17]. The decoded

parameters consist of the excitation signal (both adaptive codebook and fixed codebook components), as well as the LP coefficients and additional integrity information. The adaptive codebook and fixed codebook components are combined in order to generate the excitation signal. The excitation is then filtered with the recovered LP coefficients, as seen in equation 1.1 [14]:

$$\tilde{s}[n] = u[n] + \sum_{i=1}^{10} \tilde{a}_i \cdot \tilde{s}(n-i) \qquad n = 0, \ldots, 39 \qquad (1.1)$$

Postprocessing consists of adaptive postfiltering and highpass filtering. The adaptive postfilter is the cascade of three filters: a long term postfilter , a short term postfilter , and a tilt compensation filter , followed by an adaptive gain control procedure [ITU T G.729]. The postfilter coefficients are updated every 5 ms subframe. The postfiltering process is organized as follows. First, the reconstructed speech is filtered through to produce the residual signal. This signal is used to compute the lag and gain of the long term postfilter. The signal is then filtered through the long term postfilter and the synthesis filter. Finally, the output of the synthesis filter is passed through the tilt compensation filter to generate the postfiltered reconstructed speech signal. An adaptive gain control is then applied to match the energies. The resulting signal is filtered with a 100 Hz highpass filter and multiplied by two to produce the output signal of the decoder. Figure 1.2 illustrates the complete decoding process.

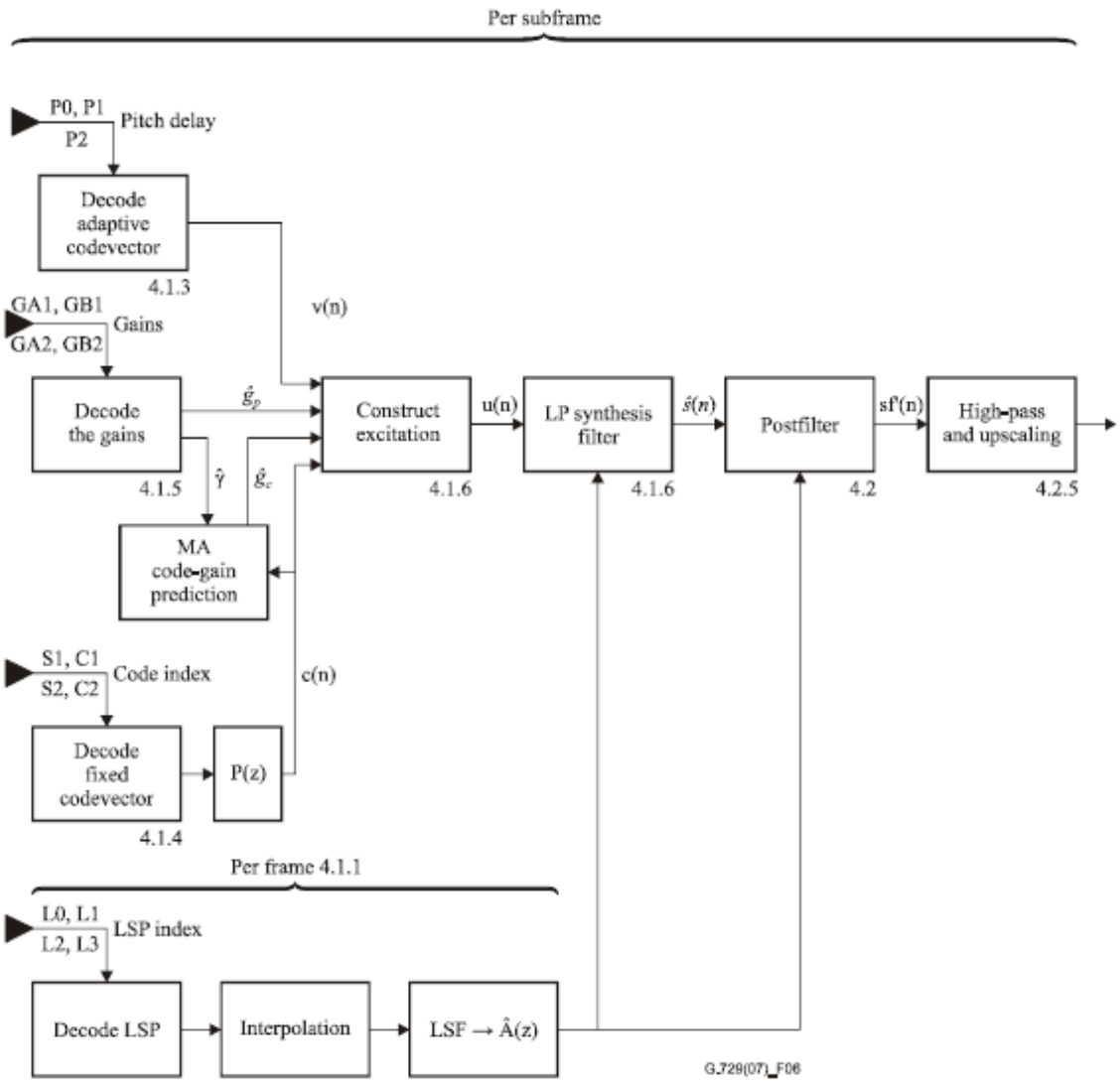

Figure 1.2: Signal flow at the CS ACELP decoder, from ITU T Recommendation G.729E (January 2007)

1.2 Phenomena affecting Quality on VoIP

Packet networks, such as the Internet, have some inherent incompatibilities with the transport of real time services, such as voice or video. This section covers in some detail these challenges, as well as existing algorithms in place to mitigate their effects. In

particular, a detailed review on the roots of delay and jitter is conducted, because these two network conditions are particularly difficult to manage for VoIP systems.

1.2.1 Delay and Jitter

The causes and effects of delay on packet networks are referenced in [18], which describes how each stage affects the overall delay on the VoIP channel. The total delay on the network is compounded by several independent sources, which can be broadly divided in two categories: fixed delay components and variable delay components.

Fixed delay components add directly to the overall delay on the channel. Variable delay (commonly called delay jitter or simply jitter), is caused by the queuing that occurs in the different egress trunks that interface the Local Area Networks (LANs) to the Wide Area Networks (WANs), and add a variable and unpredictable amount of delay. ITU recommendations G.114 [19] and G.131 [20] provide guidelines on the impact of delay on speech transport applications. Table 1.1 shows a summary of these documents, assuming echo cancellers are being used:

Range (ms)	Description
0-150	Acceptable for most user applications
150-400	Acceptable, provided that administrators are aware of the transmission time and the impact it has on the transmission quality of user applications.
Above 400	Unacceptable for general network planning purposes. However, it is recognized that in some exceptional cases this limit is exceeded.

Table 1.1: Impact of One-Way Transmission Delay on Voice Services Quality

1.2.1.1 Coder Delay (Fixed)

Coder delay refers to the time taken by the DSP processor to compress a block of PCM sample. This factor is evidently affected by the processing capability of the device performing the compression algorithm, the number of concurrent calls on this device, and, more importantly, by the algorithmic complexity and the frame size of the coder in use. Table 1.2 provides a comparative view of processing delays on Cisco Systems 2600 series Access Servers, assuming a best case scenario of one active channel per DSP chip, and a worst case scenario of 4 active channels per DSP chip:

Coder	Bit Rate	Required Sample Block	Best Case Coder Delay	Worst Case Coder Delay
ADPCM, G.726	32 Kbps	10 ms	2.5 ms	10 ms
CS-ACELP, G.729A	8.0 Kbps	10 ms	2.5 ms	10 ms
MP-MLQ,G.723.1	6.3 Kbps	30 ms	5 ms	20 ms
MP-ACELP, G.723.1	5.3 Kbps	30 ms	5 ms	20 ms

Table 1.2: Typical Processing Delays for Various Codecs

In addition to the coder delay, two other important factors that add to the overall delay are the decompression delay and the algorithmic delay. Decompression delay is roughly 10% of the compression delay, but it is multiplied by the number of frames sent in an RTP packet. Algorithmic delay refers to the necessary delay some coders required, in order to posses ““future knowledge”” of the signal. This delay depends on the number of future frames the coder or decoder needs to look ahead. [18] summarizes the total Lumped Coder Delay Parameter as the sum of the worst case compression delay, the number of blocks in a frame times the decompression time per block, and the algorithmic delay.

1.2.1.2 Packetization Delay (fixed)

Packetization delay is the time it takes to accumulate sufficient frames as needed by the designed payload size (number of frames in a single RTP packet). This delay is the product of the window size chosen for the compression algorithm (in ms) and the number of frames included in each RTP packet. Table 1.3 shows nominal values for this type of delay:

Coder	Bit Rate	Required Sample Block	Best Case Coder Delay	Worst Case Coder Delay
ADPCM, G.726	32 Kbps	10 ms	2.5 ms	10 ms
CS-ACELP, G.729A	8.0 Kbps	10 ms	2.5 ms	10 ms
MP-MLQ,G.723.1	6.3 Kbps	30 ms	5 ms	20 ms
MP-ACELP, G.723.1	5.3 Kbps	30 ms	5 ms	20 ms

Table 1.3: Typical Packetization Delays for Various Codecs

Lower payload sizes reduce delay but increase bandwidth utilization because a greater number of packets require a greater number of RTP and UDP headers to be appended. Although both algorithmic delay and packetization delay are always present they do not simply add together because VoIP gateways usually perform packetization and compression tasks in parallel, which greatly reduces the net effect of the algorithmic delay. In low speed links an additional factor called serialization delay is also present. This factor is purposely omitted, because for current links and practical frame sizes its effect is negligible [21], but reference tables are available for different frame sizes and different low speed links.

1.2.1.3 Queuing/Buffering Delay (Variable)

After the compressed voice payload is built, the header is added and the frame is queued for transmission on the network connection. Queuing delay is a variable delay and is dependent on the trunk speed and the state of the queue in the router. For example, assuming a 64 Kbps link, if a voice packet is queued behind one 48 bytes data frame and a 42 bytes voice frame, because the line is low speed a serialization delay of 3 ms is induced for the data frame and 5.25 ms induced for the speech frame, combining for a total delay of 8.25 ms [18]. Queuing delay refers to the queue inside the transmitting endpoint or gateway, and must not be confused with the propagation and network switching delay.

1.2.1.4 Propagation Delay and Network Switching Delay (Variable)

Finally, the most difficult to quantify source of variable delay is the Network Switching Delay. The propagation portion of the switching delay results from the physical distance that needs to be traveled by the transmitted packets. A common assumption, as standardized in ITU G.114 [22], is to estimate 10 us per kilometer between hops. Nonetheless, network congestion may alter the number of hops necessary to reach the destination points, so the physical distance can vary from packet to packet, making packets arrive in an order different to which they were transmitted. The other significant component is the sum of the various queuing delays that occur on the different hops of the WAN traveled. Typical carrier delays for US frame relay connections are 40 ms fixed and 25 ms variable [18]. The variation in delay is called the delay jitter, or simply jitter, and is a major factor affecting Quality of Service (QoS) on VoIP.

1.2.1.5 De Jitter Delay (Fixed or Variable)

Due to the routing of packets over different network paths and the asynchronous characteristics of the network, packets may arrive to their destination with varying delay. Jitter degrades speech quality significantly and has to be compensated for. This is usually done at the receiver side by applying anti-jitter buffers, which store packets for a static or dynamically managed amount of time before playback [23]. These mechanisms introduce additional delay and potential packet loss if packets arrive too late to be played out. The length of the jitter buffer can be fixed or adaptive, and it is explained in more detail in section 1.3.1.

1.2.2 Overview of Packet Loss

Under the scope of this thesis, packet loss refers to the unavailability of packets at the decoder. These packets may simply not be received, be received too late to be played out by the play out buffer, or contain bit errors in the payload. The main elements when evaluating the impact on quality of packet loss are: the packet loss distribution, the packet payload size, the recovery mechanisms in place at the packet level, the codec in use, and finally the error concealment algorithms at the receiver.

Because of several correction and concealment algorithms, lost packets do not simply disappear. Error correction algorithms make use of redundancy in order to reconstruct loss packets, and anti-jitter buffers will use various methods to replace packets that cannot be recovered. Finally, the performance of CELP coders depend on long term estimation loops, so a reconstructed packet may have adverse effects on the decompression of past or future frames. This process is illustrated in Figure 1.3.

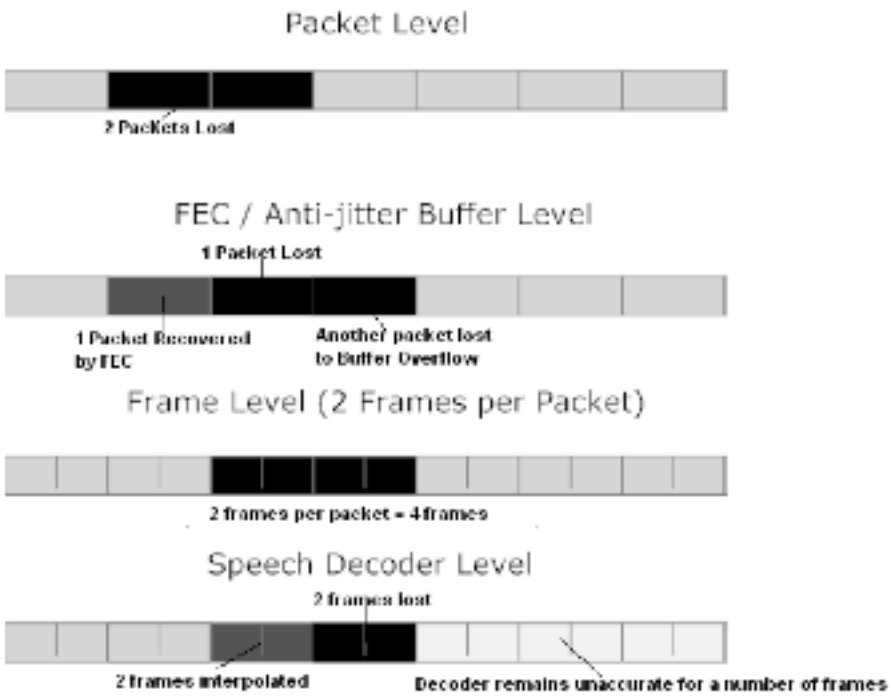

Figure 1.3: Effects of Packet Loss

1.3 Existing Correction Algorithms

An array of technologies was developed in response to the challenges described to the successful delivery of voice services over packet networks. In this section, a brief overview is provided of the most popular techniques applied in state of the art VoIP systems. Many of these techniques are re applications of existing techniques in traditional communications systems, but many of them exploit the unique characteristics of voice signals.

The analysis of the different techniques is of great interest for this research because it provides an insight in the observable artifacts that will be produced by delay, jitter and packet loss. It is important to mention that, under many circumstances, these algorithms will completely conceal any network deficiencies form the decoded signal, so classifiers

working after the receiver will not be able to accurately estimate the network conditions. Nonetheless, network conditions can be considered irrelevant under these circumstances, as far as the transport of voice services is concerned. The scope of this thesis is to detect phenomena on the PCM signal that may indicate that the correction algorithms in place are not successfully compensating for network deficiencies.

1.3.1 Anti-Jitter Buffers

Since speech is a constant bit rate service, the jitter induced by all the variable delays must be removed before the RTP packets can be decoded. This is achieved by buffering a number of packets in an anti-jitter buffer [24]. Effectively, an anti-jitter buffer will transform the jittery delay into a fixed delay, equal to the length of the buffer in frames times each frame's duration. The frame duration is dependent on the compression algorithm used, and is in the range of 5 milliseconds to 30 milliseconds, so it is evident that the anti-jitter buffer will be a major source of delay in the system. Figure 1.4 illustrates the mechanics of the anti-jitter buffer.

Figure 1.4: Mechanics of the Anti-Jitter Buffer

The length of the anti-jitter buffer is critical for the performance of a VoIP system. If the buffer length is too short, variations in delay can cause the buffer to be "under-run", and dropped packets will affect the quality of the received signal. If the length of the buffer is too long, the overall delay will be increased, which may result in unacceptable

quality. This necessary compromise made the study of anti-jitter buffers a very active area of research, which developed into the use of adaptive anti-jitter buffers.

1.3.2 Forward Error Correction (FEC) and Packet Loss Concealment

Packet loss behavior is bursty, and generally modeled using Markov chains, most specifically Gilbert Model [25]. Given this quality of packet loss, forward error correction algorithms with an open loop approach are favored over closed loop correction algorithms (usually referred to as correction algorithms), that involve the re transmission of packets. This is the case because closed loop algorithms introduce network delay as the feedback packets need to travel back to the transmitter. The most common closed loop correction alternative is Automatic Repeat Request (ARQ) that is native in the TCP protocol [24].

FEC algorithms, on the other hand, continuously transmit redundant information, with certain temporal restrictions to conserve bandwidth efficiency. FEC approaches add redundancy to the normal voice stream to protect it from the loss introduced by the network. A conundrum in FEC approaches is that packet loss is many times caused by bandwidth saturation, and adding redundancy packets effectively increase the bandwidth utilization. To this purpose, the number of redundant packets sent is controlled in FEC implementations, to limit bandwidth usage [24].

Media independent FEC includes protection algorithms such as Viterbi codes and Reed Solomon, which protect the integrity of the entire packet. When the FEC algorithms are media dependent, different bit rates are used for the original and redundancy packets. The redundancy packets are transmitted on a lower quality, but since they are interpolated

between higher quality frames these quality deficiency is most of the times not perceivable. The problem with this approach is that it requires additional processing at the sender and the receiver. Usually FEC is combined with packet loss concealment: The FEC will provide a protection against complete loss, while packet loss concealment will look at the characteristics of surrounding packets to interpolate an adequate signal [26], [27], [28].

This interpolation is many times achieved by means of simple silence insertion. The problem with this approach is that the longer the packet loss, the more annoying the effects of the interpolation to the user. Alternatively, surrounding frames can be replicated in order to replace of the lost frame. This approach has an equally low computational expense as silence and noise substitution. A refinement to this approach is the pitch waveform substitution, where the gap is filled by a pitch waveform generated from the pitch of the last successfully received frame [29], [30].

1.4 Estimating Network Conditions by Observing PCM Signals

1.4.1 Motivation

Objective evaluation of speech quality has been a very active area of research for the last decade. As VoIP technologies evolve, and the economic and managerial incentives grow for network convergence, transport of voice over packet networks has became a common practice among wholesale long distance carriers. At the present time, the international footprint of telephony networks is in a very peculiar hybrid state, where calls are routed through both TDM and packet networks around the world.

Typically, wholesale carriers and resellers route their traffic base on two criteria: least cost and best quality. Online minutes markets, such as Arbinet or I Basis, price the cost of minutes based on the quality of the route, which at the present time they evaluate with metrics such as Average Call Duration (ACD) or Answer Seizure Ration (ASR) [31]. Although these metrics are very effective in reflecting the QoS for a specific route, they have two fundamental flaws: a great number of calls are necessary to obtain adequate ACD and ASR metrics, and these metrics provide no insight into the root cause of low measurements.

Existing intrusive methods are not commonly use in international long distance QoS monitoring because they require physical presence at both ends of the channel. Non-intrusive methods, such as the E-Model, propose an alternative that tackle this issue, but depend on the proper measurement of network metrics, such as packet loss, delay and jitter. As shown in Figure 1.4, in the current hybrid state of international telephony, such metrics are not reliable, because of tandem elements such as TDM switches or media routers isolate the different nodes of the packet network. As seen in the figure, the originating gateway in Argentina may have very low delay and jitter to the reseller in Brazil. Nonetheless, if the reseller in Brazil experiences high jitter to the phone company in Italy, this information would effectively be concealed from the gateway in Argentina, because of the routing through the TDM network at the reseller's facilities.

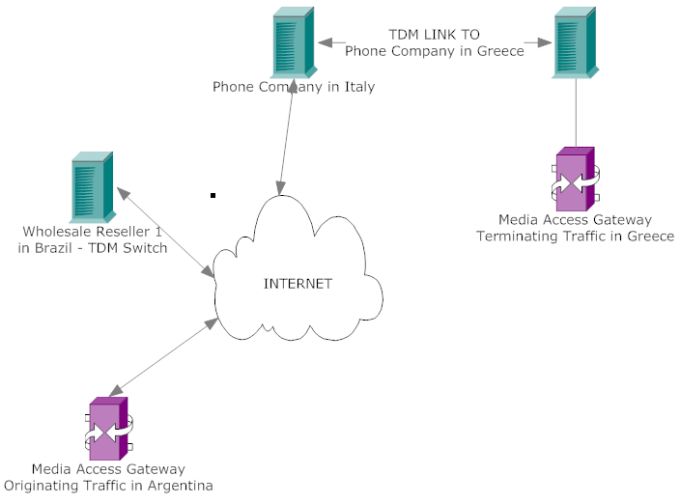

Figure 1.5: Voice Routed Over a Heterogeneous Network

1.4.2 Outline of Thesis

The objective of this thesis is to do a thorough study on the possible network impairments that can affect Voice over IP, and device classifiers that will identify these impairments by studying the decoded PCM signal at the receivers end. In this sense, there is a big overlap between the subject matter of this thesis and the existing research on quality of service (QoS), objective quality measurement, and QoS monitoring.

In this chapter, an overview of the main impairments (delay, jitter and packet loss) was conducted, along with the existing mechanisms in place to minimize their impact on quality (FEC, PLC, Adaptive Playout) [32], [33]. A description on the root causes of delay was conducted, with the objective of illustrating how in complex telephony networks, such as the ones encountered in international wholesale, the network conditions can be very hard to predict. Also, state of the art speech codecs were described, in order to gain understanding on the trade offs made in order to conserve bandwidth. One of the most significant tradeoffs is that LPC algorithms add a level of

sensibility to packet loss, due to long terms processes that get desynchronized by noise information at the receiver. It is important to note that under some conditions the enhanced PLC algorithms these codecs incorporate mitigate this sensibility.

In chapter 2 an overview on the existing research on quality measurement will be conducted. This chapter covers the state of the art methods for objective quality measurement algorithms, both intrusive and non-intrusive, as well as an overview of the latest literature, which for the most part attempt to train different sorts of artificial intelligence classifiers (such as neural networks, fuzzy classifiers, KD Trees, etc.) in order to estimate the QoS.

Chapter 3 presents the proposed architecture, which is a novel element in the field of QoS monitoring: a classifier that can estimate the network conditions from the received PCM signals. This classifier matches different anomalies in the PCM signal with specific network conditions. The training of the classifiers is done with the results of experiments conducted under a controlled environment, described in Chapter 4.

Chapter 5 presents the experimental results of the classifiers, and compares their performance under a variety of metrics. Chapter 6 presents the conclusions of this thesis.

1.4.3 Contributions

The main contribution of this thesis is the introduction of a new process, the Network Condition Estimator (NCE), which can be used in conjunction with existing technologies, such as the E-Model, to allow real time monitoring of QoS for VoIP networks. The payload content of LPC codec (G.729) was used for the training of the network. Since that coder already performed the analysis of the signal in terms of LPC

coefficients, using these coefficients, as well as the excitations, as features for the classification would drastically reduce the computational impact of the NCE on the equipment.

CHAPTER 2: Related Work

2.1 Quality Evaluation of VoIP

ITU T Recommendation E.800 defines Quality of Service (QoS) as "the collective effect of service performance which determines the degree of satisfaction of a user with the service." In 2004, ITU T Delayed Contribution D.197 introduced the term Quality of Experience (QoE), which is defined as "a measure of the overall acceptability of an application or service, as perceived subjectively by the end user". Under this new context, QoS refers to the quality of the network from an objective perspective, and QoE to the quality of the service as subjectively perceived by users, which obviously depends dramatically on the QoS. In the bibliography, the terms are used mostly interchangeably.

2.2 Objective vs. Subjective Quality Evaluation

Subjective measurements of QoS are carried out by groups of people, which in various fashions give their personal opinion about several aspects of the quality of a conversation (such as listening quality, listening effort and loudness preference) [34]. ITU T Recommendation P.800 [35] standardizes the procedures to obtain such measurements. The most common metric used to express to the results of subjective measurements is the Mean Opinion Score (MOS), which varies from 1 to 5, as shown in Table 2.1.

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very Annoying

Table 2.1: Reference Values for MOS Results [36]

Although subjective methods provide us with the best metric for quality assessment, they have many practical problems: they cannot be performed at the network planning face, they are very laborious, and also very expensive [6]. In response to these challenges, new methods that permit the estimation of MOS values from objective measures were developed.

Objective voice quality metrics replace the human testing subjects with various computational models or algorithms, which compute MOS values based on various parameters [37]. The objective of these systems is to predict the MOS values that would be produced by a panel of evaluators, under diverse speech distortion conditions. The accuracy of these algorithms is therefore measured in terms of their correlation with subjective MOS scores, where a value of 0.8 or greater is considered effective [38]. Depending on whether or not these algorithms require an unaffected reference signal, they can be classified as intrusive and non-intrusive, as seen in the next section.

2.3 Intrusive vs. Non-Intrusive Quality Evaluation

This distinction between intrusive and non intrusive algorithms refers to whether or not a reference signal is necessary in order to estimate the MOS. Intrusive quality measurement algorithms usually use two inputs signals: a reference signal and a degraded signal taken from the output of the receiver at the far end of the network [39]. Non intrusive algorithms, on the other hand, attempt to estimate the MOS from network impairment parameters, such as jitter, delay and packet loss, without a reference signal [40], [41], [42].

Most recent versions of intrusive E-Models work by transforming both the reference and the degraded signals into perceptual domains, which is a domain where the signal is characterized by its most relevant psychoacoustic features. Once both signals are in this domain, the distance between them is calculated, providing an estimation of perceptual similarity between the signals, and mapped into a MOS scale [38].

Non-intrusive methods can be broadly divided into signal based and parametric. Signal based models, which at the present time are in a very incipient level of development, estimate MOS by processing the output of a production system [43], [44], [45]. Many of these methods focus on speech production models, speech signal likelihood, or other perceptual aspects such as noise loudness. Non-intrusive parametric methods, on the opposite, typically do not take the audio signal as an input, but rather underlying properties of the underlying transport network and equipment [46]. In the case of VoIP networks, such properties include codec type, bit rate, delay and packet loss statistics [47].

2.4 State of the Art for Objective Evaluation Algorithms

2.4.1 PSQM (Intrusive)

The Perceptual Speech Quality Measurement (PSQM) method was standardized by the ITU T Rec. P.861 in 1998 [48], and is the result of the work of Beerends et al. [49], working at KPN Research in the Netherlands. The original PSQM algorithm was developed for the study of low bit rate codecs working in telephone band signals, and it was very ineffective for high distortion scenarios such as clipping or packet loss [50]. For this reason, PSQM+ was later developed, providing support for packet networks testing

[48]. The mathematical algorithm can be separated in three blocks depicted in Figure 2.1. The steps involved in the PSQM algorithm are the following:

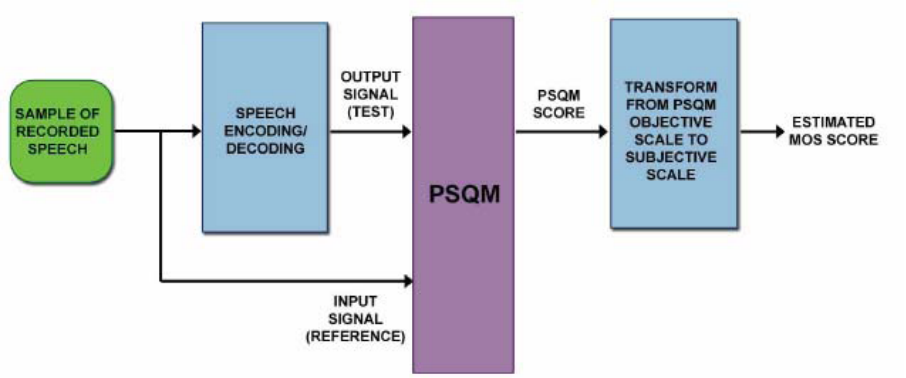

Figure 2.1: PSQM Model

1) Pre processing: The inputs signals are time aligned first, and then power normalized.

2) Perceptual modeling: The signals are converted to a perceptual domain, based on perceptually warped frequency bands and loudness sensitivities, observed on the human auditory system.

3) Perceptual distance calculation: A weighted distance between the reference and the degraded signals is estimated in the time domain, to produce a value from 0 (excellent quality) to practical upper bounds in the range of 15 to 20 [50], called the PSQM score.

The PSQM score is finally mapped to a MOS scale. As previously mentioned, PSQM was unable to properly predict MOS scores under clipping or packet loss scenarios, so PSQM+ must be used for VoIP quality assessment [51].

2.4.2 PAMS (Intrusive)

Perceptual Analysis Measurement System (PAMS) provides a voice quality objective measurement for a system affected by damaging factors such as time clipping,

packets loss, delay and distortion due to the codec usage [52]. Figure 2.2 illustrates the process, where the three outputs to the block are later combined in a non-linear mapping to obtain a MOS.

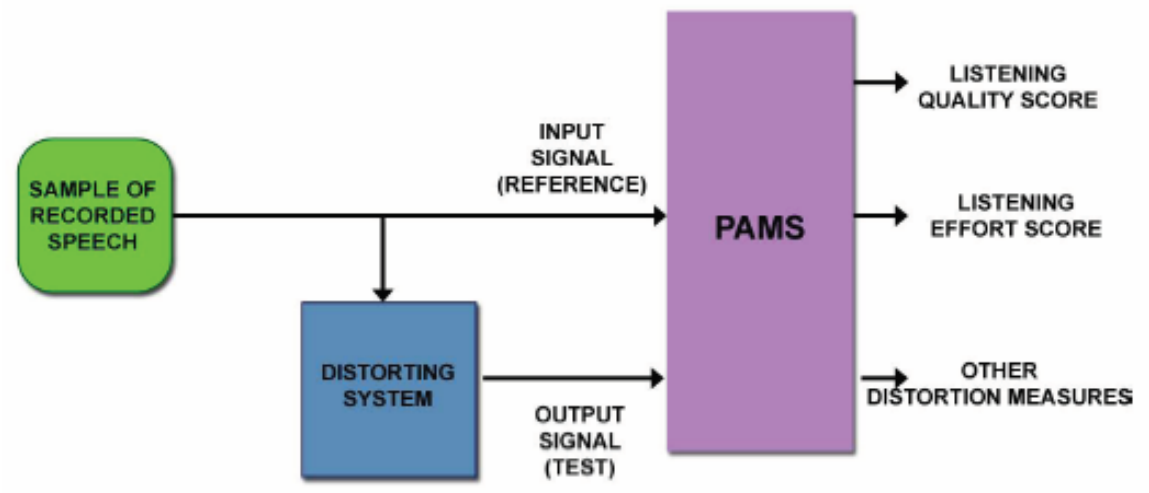

Figure 2.2: PAMS Model

PAMS greatest contribution was the time alignment algorithm, which made it insensitive to jitter, and better fit for VoIP quality evaluation. The PAM algorithm can be divided as follows:

1) Pre processing: before signals are compared, they are aligned and power equalized, to account for bulk delay, variable delay and linear filtering. Both signals are also filtered, to simulate the response of the response between the network handset to the inner ear.

2) Auditory transform: PAMS uses the auditory transform described on [38], [43] and [42]. The transform contains 19 band Bark spaced perceptual filterbanks, smoothing and downsampling, and perceptual warping to the phone and sone scale.

3) Error parameterization: the difference between the two signals is evaluated in the perceptual domain and parameterized to yield a Listening Quality Score and a Listening Effort Score.

The perceptual errors are mapped in a subjective quality scale. In particular, PAMS produces a Listening Quality Score and a Listening Effort Score. A non-linear mapping is necessary to derive MOS values from the output of the PAMS algorithm. The function used to obtain MOS values from the output of the PAMS algorithm was obtained by correlation with subjective experiments.

2.4.3 PESQ (Intrusive)

In 2001, ITU approved recommendation P.862 [53]for the Perceptual Estimation of Speech Quality algorithm [54], [55]. PESQ combines features of PSQM and PAMS, which were chosen by a designated study group from among several other candidates. ThE-Model is shown in Figure 2.3.

PESQ adds novel factors and methods to calculate signal distortion offering the possibility to use natural and artificial audio samples [56], and is the latest recommendation by the ITU for intrusive quality measurements. It is applied for the evaluation of the following factors:

- Transcoding
- Transmission Errors on the transmission channel
- Codec Errors
- Noise introduced by the system

- Packet loss
- Time clipping

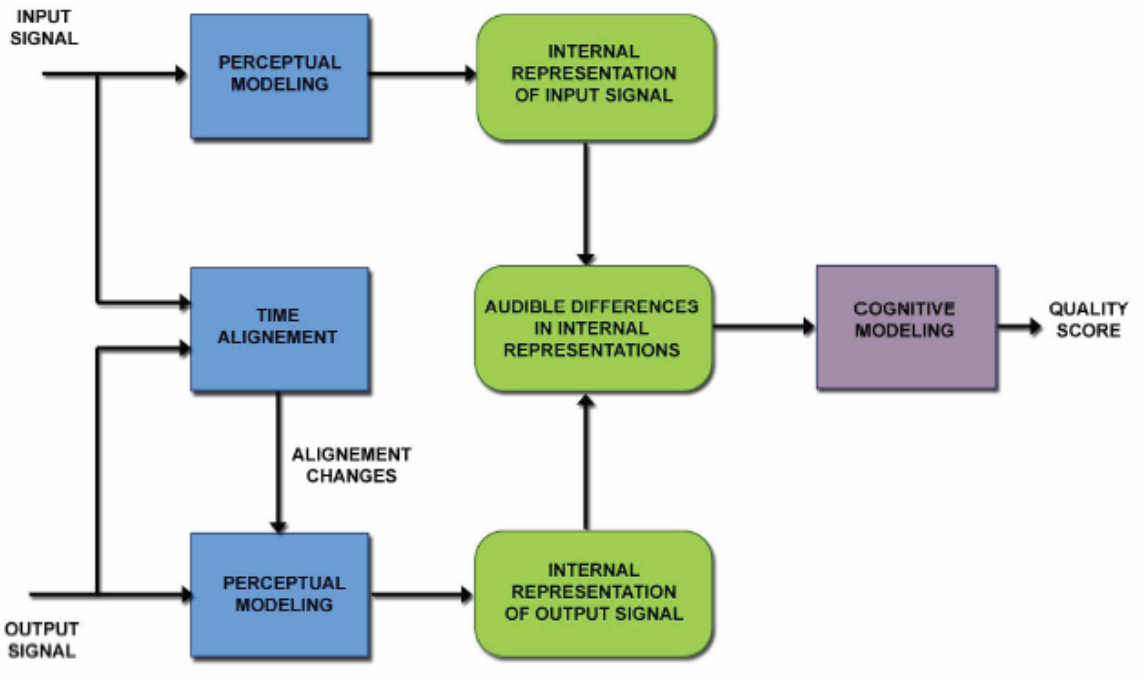

Figure 2.3: PESQ Model

The PESQ method can be structured as follows [53]:

1) Pre processing: this step includes the time and frequency alignment of the reference and the degraded signals.

2) Perceptual Modeling: The steps for the perceptual modeling are the same used in PAMS.

3) CognitivE-Modeling: in this phase the values that represent noise computation are evaluated. These values are then combined to provide a MOS score prediction. A difference between reference signal and distorted signal for each time frequency cell is calculated. A positive difference indicates the presence of noise, while a negative difference indicates a minimum noise presence such as codec distortion. ThE-Model

permits to discover the time jitter and to identify which frames are involved and which frames affected by the delay are erased in order to prevent a bad score.

2.4.4 E-Model

The most popular non-intrusive objective measurement measure is the E-Model [De Rango06]. E-Model is an abbreviation of "European Telecommunications Standards Institute (ETSI) Computational Model". The E-Model operates under the assumption that each quality degradation type is associated to a certain type of damaging factor [6], to produce a ""Rating Factor"" (R) [47]. Table 2.2, obtained from ITU T Recommendation G.711, shows all the different degradation types contemplated by the E-Model, as well as default values for a clear telephone channel with PCM coding.

Parameter	**Abbreviation**	**Unit**	**Default Value**
Sending Loudness Rating	SLR	dB	8
Receiving Loudness Rating	RLR	dB	2
Sidetone Masking Rating	STMR	dB	15
Listener Sidetone Rating	LSTR	dB	18
D-Factor Handset, Send Side	Ds	-	3
D-Factor Handset, Receive Side	Dr	-	3
Talker Echo Loudness Rating	TELR	dB	65
Weighted Echo Path Loss	WEPL	dB	110
Round Trip Delay in a 4-Wire Loop	Tr	Ms	0
Absolute Delay	Ta	Ms	0
Mean One-Way Delay	T	Ms	0
Number of Quantizing	Qdu	-	1

Distortion Units			
Equipment Impairment Factor	Ie	-	0
Circuit Noise relative to 0 dBr-point	Nc	dBm0p	-70
Noise Floor at Receive Side	Nfor	dBmp	-64
Room Noise at Send Side	Ps	dB(A)	35
Room Noise at Receive Side	Pr	dB(A)	35
Packet Loss Percentage	Pp	%	0
Packet Loss Robustness Factor	Bpl	%	1
Burst Ratio	BurstR	%	1
Advantage Factor	A	-	0

Table 2.2: Default E-Model Values for G.711 Codec

The E-Model operates under the assumption that different degradations, when transformed to a particular scale, can be added together to obtain the total degradation of the channel [47]. The transmission rating scale, or R Scale, is the output of the E-Model that expresses quality. It is obtained by equation 2.1, where is the speech quality due to the basic signal to noise ratio, is the simultaneous impairment factor, is the delayed impairment factor, is the effective equipment impairment factor (which was updated on ITU T Delayed Contribution D.027 in 2005 and ITU T Delayed Contribution D.044 in 2001 to encompass impairments related to VoIP and packet networks in general), and A is the Advantage Factor. In Appendix E, [36] provides a all necessary equations to derive the R rating from the parameters in table 2.2. Equation 2.2 provides the mapping from the R factor to the MOS scale [6].

$$R = R_0 - I_s - I_d - I_{e,eff} + A \tag{2.1}$$

$$MOS = \begin{cases} 1 & for\ R < 0 \\ 1 + 0.035\ R + R\ (R - 60)(100 - R) \cdot 7 \cdot 10^{-6} & for\ 0 \le R \le 100 \\ 4.5 & for\ R > 100 \end{cases} \tag{2.2}$$

At the present time, the E-Model represents the state of the art for non-intrusive objective quality evaluation.

2.4.5 Artificial Intelligence Approaches

Several terms are used in the literature to refer to objective non-intrusive quality estimators, out of which a persisting one is the "single ended" estimator. The previously described E-Model represents the standardized state of the art in this subset of classifiers, but extensive research has been conducted in ordered to provide alternatives. Most recent publications in this area use artificial intelligence classifiers, such as fuzzy logic, neural networks and Bayesian models, among others, which are trained on a variety of feature sets to estimate MOS.

Chivi et al. [42] presents the use of Perceptual Linear Prediction (PLP) vector quantizers as the feature set of choice. Based on this features, he evaluates three metrics for distortion: median minimum distance, transition probability distance, and combined distortion index. After testing this classifier against several test beds, he concludes that the technique is effective and robust against speaker dependency, audio content and distortion variation.

Chen et al. [57] introduces the use of speech perceptual spectral density features, extracted from the output audio to train a fuzzy logic neural network. This classifier was

also trained and tested against several test sets and showed a high correlation with PESQ results.

Falk et al. [58] conducted a broader survey on the use of various Machine Learning methods in order to derive a single ended classifier. Gaussian MixturE-Models, Support Vector Machines and Random Forest Evaluators were trained, using both clean and degraded speech signals. The resulting classifiers produce simultaneous estimations of the MOS, which are later combined using hard and soft decisions on a second classifier stage.

Mahdi [50] introduces a KD Tree based non-intrusive speech quality evaluator. This single ended classifier computes the auditory distance between the perceptual features (PLP) representing segments of the input speech signal and several reference signals contained in a reference book. The reference book is constructed by clustering a large number of feature vectors, obtained from a database of clean speech, using a KD Tree data structure. The auditory distance between the tested vectors and the references vectors is mapped into a listening quality scores domain, in order to obtain an estimation of MOS. The performance obtained is one of similar accuracy as ITU T 3SQM, but with significantly reduced computational expense.

Chen et al. [59] proposed the BM NiSQE algorithm, which combines Gaussian Mixture Density Hidden Markov Models (GMDHMM) with Bayesian inferences and minimum mean square error estimators, in order to reflect both the temporal variations of speech signal and the statistical characteristics of the perceptual feature space. The authors conclude in this paper that the method yields a high correlation with subjective quality scores, obtained using the PESQ algorithm.

CHAPTER 3: Proposed Approach

This chapter describes the development of the Network Conditions Estimator (NCE), which is a novel contribution to the objective evaluation of quality for VoIP networks. The objective of this process is to give an estimation of the actual network conditions of the packet networks involved on the transport of voice services, by examining the output PCM signal of the last codec involved. The need for such process arises from the existence of tandem devices in hybrid telephony networks, which effectively mask the transmission of QoS and monitoring information at the IP level.

3.1 Applications of Network Conditions Estimator

The network conditions estimator obtained after the training of the classifier has many potential applications with existing technologies. In terms of parametric non-intrusive algorithms like the E-Model, it can be used as the input for the parametric values of jitter and packet loss, as seen in Figure 3.1.

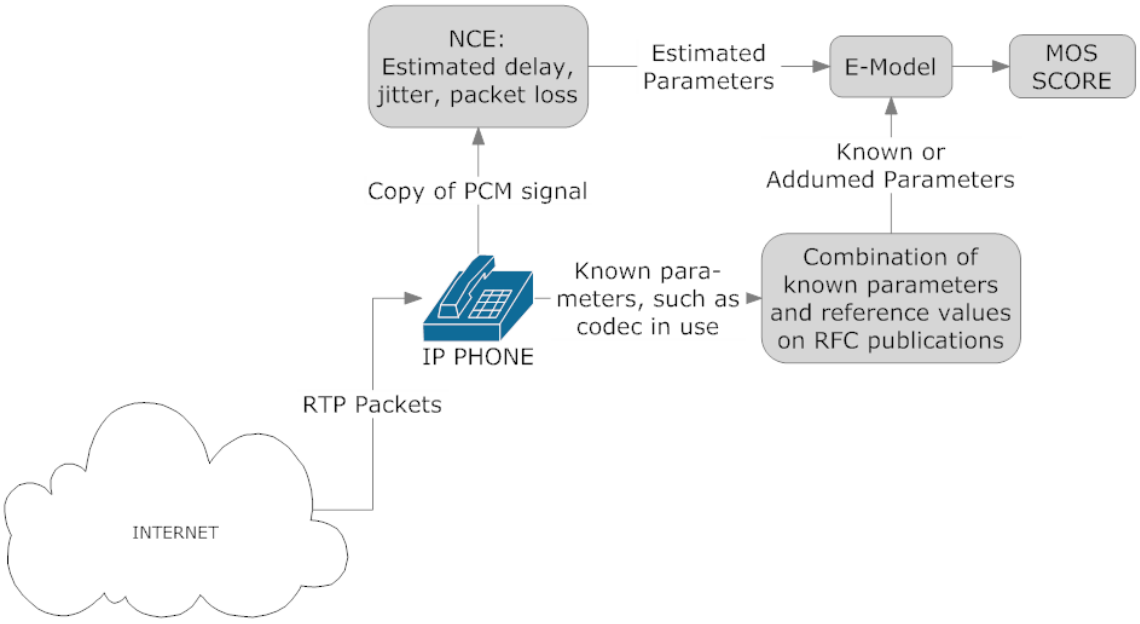

Figure 3.1: Combination of the NCE with the E-Model

Furthermore, given that the quality of a specific destination is known to be below requirements by other means, such as intrusive measurements, the NCE can help pin point the concrete underlying network condition that is causing a drop in quality.

3.2 Theoretical Background

In this section, the theoretical foundations of the NCE are covered. In order to model the mapping between the feature space vectors extracted from the PCM signal and the actual network conditions of the network, a neural network classifier was developed. Neural networks have the property of, giving certain assumptions, being able to mimic any linear or non-linear mapping. Although neural networks are particularly simple to train, they do not represent the state of the art for artificial intelligence classifiers. For this reason, alternative classifiers are also evaluated in Chapter 5 on the best performing feature set, such as Decision Trees.

3.2.1.1 Artificial Neural Networks

A multilayered feed forward Neural Network was trained with a Generalized Back propagation algorithm, following the architecture of Nayak et al. [60]. Figure 3.2 shows a generic neural network, with one hidden layer. The number of neurons was adjusted for each feature set, until the best performance was achieved. The number of features N varied according to the feature set in investigation.

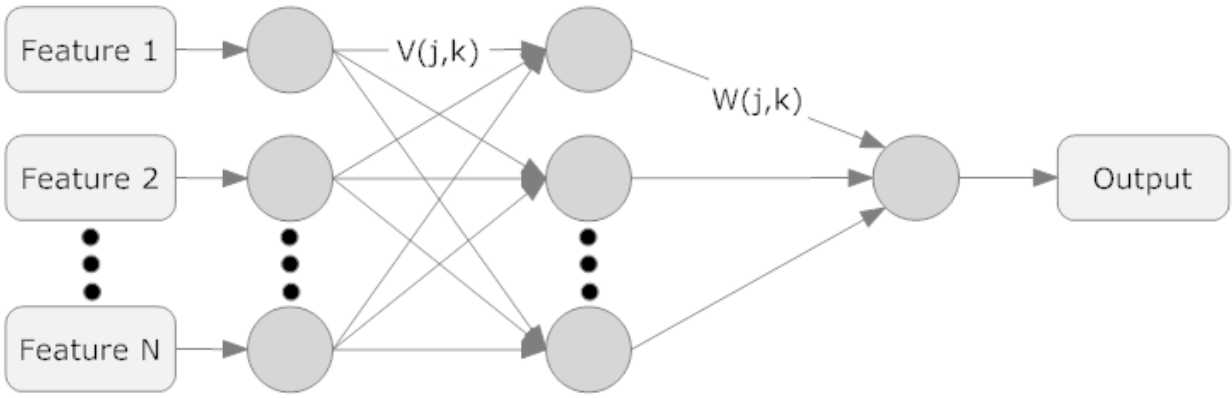

Figure 3.2: Neural Network Structure Used for the Classifier

3.2.1.2 C4.5 Decision Trees

A decision tree is a data structure consisting on decision nodes and decision leaves, where the nodes represent tests on a specific attribute, and a leaves represent class labels [61]. For each possible outcome for a test, a child node is present. In particular, for discrete values the outcome of a test on an attribute A has h possible outcomes:

$$A = d_1, A = d_2, \ldots, A = d_N.$$

In the case of continues attributes, there are two possible outcomes:

$$A \geq t \text{ or } A < t,$$

where t is a value determined at the node, called *threshold*. Figure 3.3 illustrates a sample decision tree, which decides whether a child should be allowed to go out to play, based on the attributes outlook, humidity and windy.

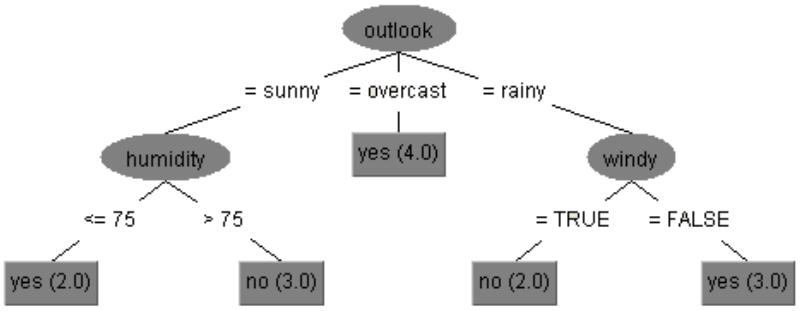

Figure 3.3: Sample Decision Tree

C4.5 decision trees recursively find the attribute that will split the training set into subsets providing the maximum information gain. The information gain for a set of cases is:

$$gain(a) = info(T) - \sum_{i=1}^{s} \frac{|T_i|}{|T|} \cdot info(T_i) \qquad (3.1)$$

$$info(T) = -\sum_{j=1}^{NClass} \frac{freq(C_j, T)}{|T|} \cdot log_2\left(\frac{freq(C_j, T)}{|T|}\right) \qquad (3.2)$$

In equation 3.1, *gain* (a) is the information gain of splitting the set *T* into sub sets $T_1, T_2, \ldots T_S$ with distinct values of *a. info (T)* is the entropy function, where *N Class* is the number of distinct classes found in the set *T*, and C_j is a specific class in the group.

3.2.2 Feature Extraction

A variety of features have been examined in the literature for the task of speech and speaker recognition. [62] conducts a thorough survey of some of the most popular features sets used to date: MFCCs, LPC and LPCC. For the purpose of this thesis, the

feature sets evaluated included all of these features, in addition to Multi-Resolution filterbanks. Additionally, several combinations of the various feature sets, as well as their rates of change, were included.

3.2.2.1 Multi-Resolution Filterbanks

The first feature set investigated was the energy bands obtained using filterbanks. Nguyen [63] gives an overview on how critical subband analysis can be achieved by means of dyatic filterbanks. Dyatic filterbanks can be implemented as n Level Symmetric (Figure 3.4) or Asymmetric filterbanks (Figure 3.5), the difference being that the first group will provide a decomposition of the signal into its coarser and more detailed components, while the second one will provide uniform spectral and temporal resolution. The energy in each subband was calculated for every 20 ms interval, in order to generate N features.

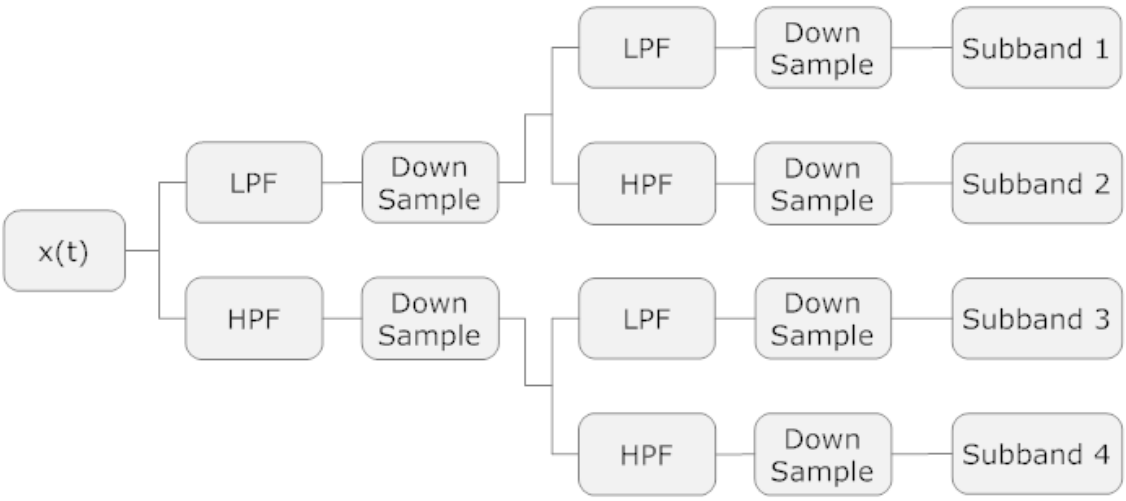

Figure 3.4: Typical Symmetric Filterbank

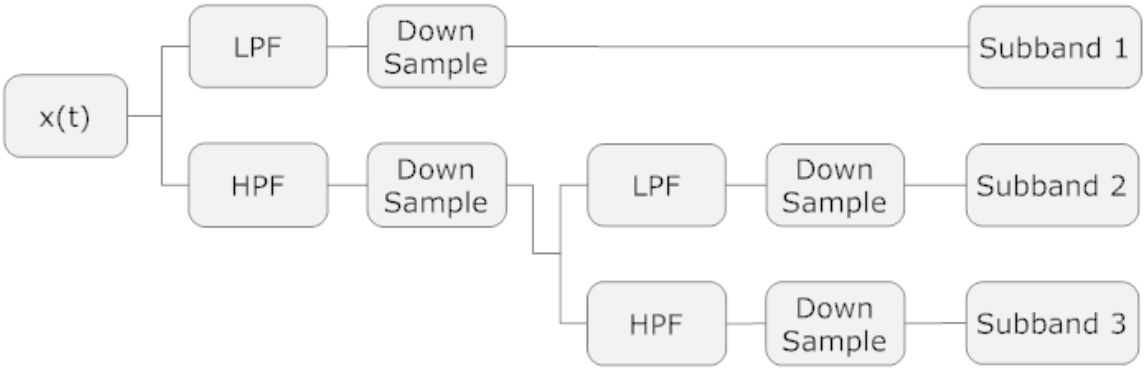

Figure 3.5: Typical Asymmetric Filterbank

For the particular task at hand, the use of asymmetric filterbanks was avoided, as it imposes an implicit bias on the importance of the different frequency bands that have more resolution. Only symmetric filterbanks were evaluated, since it was desired that the classifier found the relevant frequency bands by means of a training algorithm.

As shown in Figure 3.4, the implementation of a filterbank requires a number of high pass and low-pass filtering operations. Typically, a single low-pass filter $H_0(z)$ is used as the prototype, which for this study was of the 10^{th} order FIR filter. Figure 3.5 shows the frequency response of the low-pass filter prototype. The high pass filter was obtained from the low-pass filter, as seen in equation 3.3.

$$H_1(z) = H_0(-z) \qquad (3.3)$$

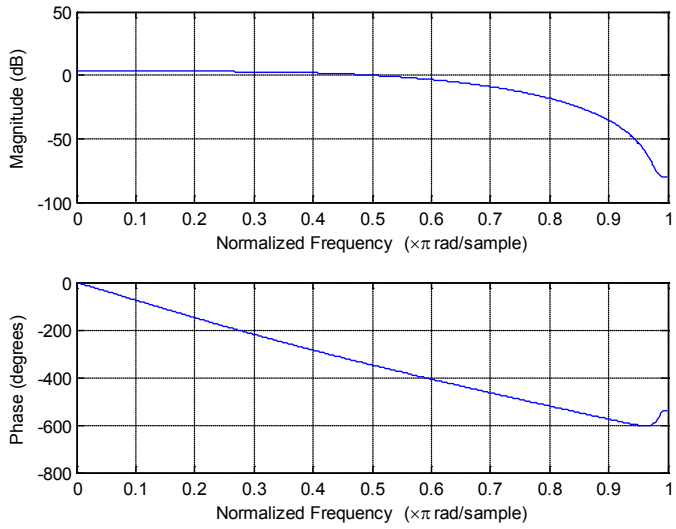

Figure 3.6: Frequency Response of the Prototype Low-Pass Filter used in the Filterbank Analysis

3.2.2.2 Mel-Frequency Cepstral Coefficients (MFCC)

Mel-Frequency cepstrum coefficients are commonly used in speech recognition tasks, due to the fact that the auditory response of the human ear resolves frequencies nonlinearly [64]. The warping from linear frequency to Mel-Frequency is given in Equation 3.4. Figure 3.6 shows this mapping with 20 triangular bandpass filters that are equally spaced along the Mel-Frequency scale, with band-limiting between 300 and 3400 Hz.

$$Mel(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \qquad (3.4)$$

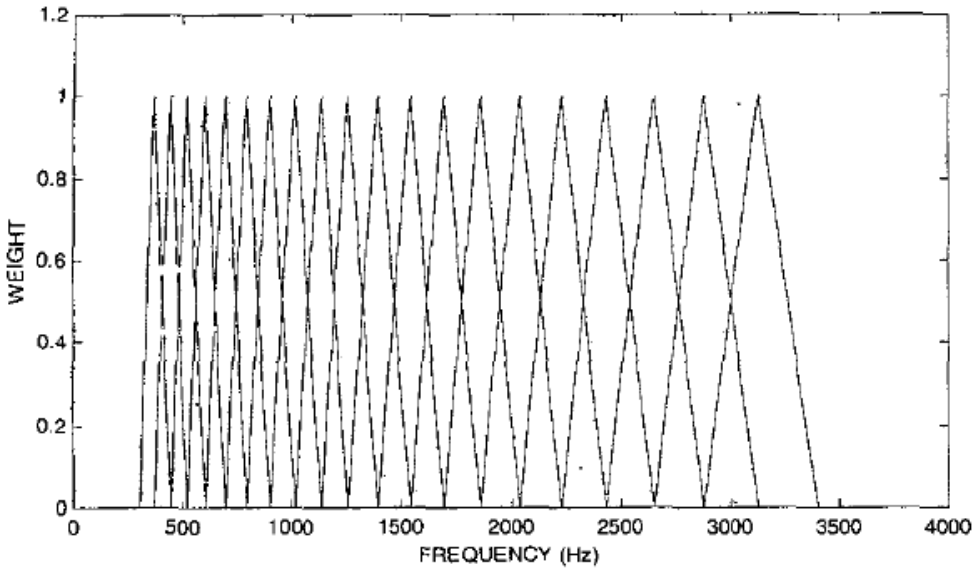

Figure 3.7: Filter for Generating MFCCs, with Band-Limiting Between 300 Hz to 3400 Hz [Wong01]

The MFCCs were computed using the Discrete Cosine Transform [65], as seen in Equation 3.5, where N is the number of bandpass filters m_j is the log bandpass filter output amplitudes. A small drawback is that MFCCs are more computationally expensive than LPCC due to the Fast Fourier Transform (FFT) at the early stages to convert speech from the time to the frequency domain.

$$MFCC_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j cos\left(\frac{\pi\, i}{N}(j - 0.5)\right) \qquad (3.5)$$

3.2.2.3 Linear Prediction Coefficients (LPC)

Linear Predictive Coding is a popular technique for speech compression and speech synthesis. Linear prediction models the human vocal tract as an infinite impulse response (IIR) system that produces the speech signal. The general form of the filter is given in equation 3.6.

$$s(z) = \frac{1}{1 - \sum_{k=1}^{p}(a_k\, z^{-k})} \qquad (3.6)$$

In terms of the training, the coefficients can be used for the training of the classifier. In particular, for the case of the training, the variation of the pole placements is of particular interest, since there are physiological constraints to the amount of change that can be naturally produced by a speaker.

3.2.2.4 Linear Prediction Cepstrum Coefficients (LPCC)

Linear Prediction Cepstrum Coefficients are Linear Prediction Coefficients (LPC) represented in the cepstrum domain [64]. LPCs work under the assumption that the characteristics of the vocal tract can bE-Modeled by an all pole filter, where the features extracted are simply the coefficients of this all pole filter, and are equivalent to the smoothed envelope of the log spectrum of the speech. LPC can be calculated either by the autocorrelation or covariance methods directly from the windowed portion of speech, and the LPCCs [66] are acquired from the LPC following Equation 3.7.

$$LPCC_i = LPC_i + \sum_{k=1}^{i-1}\left(\frac{k-i}{i}\, LPCC_{i-k}\, LPC_k\right) \qquad (3.7)$$

3.3 Exploiting LPC Codecs for Feature Extraction

LPC codecs were covered in certain detail in Chapter 2, because they present an opportunity for significant computational savings in the implementation of an NCE. The RTP payload of a G.729 packet is described in ITUT T G.729E. Table 4.1 shows the content of the information appended:

Symbol	Description	Bits
*L*0	Switched MA predictor of LSP quantizer	1
*L*1	First stage vector of quantizer	7
*L*2	Second stage lower vector of LSP quantizer	5
*L*3	Second stage higher vector of LSP quantizer	5
*P*1	Pitch delay first subframe	8
*P*0	Parity bit for pitch delay	1
*C*1	Fixed codebook first subframe	13
*S*1	Signs of fixed-codebook pulses 1st subframe	4
*GA*1	Gain codebook (stage 1) 1st subframe	3
*GB*1	Gain codebook (stage 2) 1st subframe	4
*P*2	Pitch delay second subframe	5
*C*2	Fixed codebook 2nd subframe	13
*S*2	Signs of fixed-codebook pulses 2nd subframe	4
*GA*2	Gain codebook (stage 1) 2nd subframe	3
*GB*2	Gain codebook (stage 2) 2nd subframe	4
NOTE – The bit stream ordering is reflected by the order in the table. For each parameter, the most significant bit (MSB) is transmitted first.		

Table 3.1: Payload Content of a Transmitted RTP Packet [ITUT T G.729E]

The received indices *L*0, *L*1, *L*2 and *L*3 of the Linear Spectrum Pairs (LSP) quantizer are used to reconstruct the quantized LSP coefficients using the procedure described in clause 3.2.4 of ITU T G.729E. A procedure is also provided in the document to recover the LPCs from the LSP coefficients. Although at first sight it may seem like extracting this parameters from the RTP packets defeats the purpose of estimating network conditions (because this information is directly observable from the RTP headers), this is not exactly the case. As described initially, tandem elements make it possible to not have deficiencies in the underlying network between the last two nodes, but significant impairments between previous nodes. Given this is the scenario, an NCE should take into account any network information obtained directly from the RTP packets

and add it to its output, but should not avoid further inspection of the output based features, such as LPCs.

A final consideration for the use of LPCs extracted from the payload of the RTP packets is that in the case that jitter or packet loss is directly observed in the network packets, until further research is conducted, the estimation should be made entirely based on the decoded signal only. This is the case, because it is not yet clear whether or not the cascading of jitter and packet loss across different nodes can simply be added together.

3.4 Training Considerations

Some special considerations were necessary in order to train the classifier. The first consideration was the number of frames necessary to make any useful observations on the time varying phenomena produced by jitter or packet loss. In terms of jitter, for example, typical encountered values of 25 ms to 100 ms may not be easily reflected in a single frame of audio. Since the frames extracted are of 20 ms duration, typically 10 frames to 20 frames were used for the training, as well as their respective velocities and accelerations in the feature space.

Also, many times a combination of different feature set yield better results that only one set of features. Hence, several experiments were conducted that combined subband energies, MFCCs, LPCs and LPCCs, as well as their respective velocities and accelerations. The metric used for the evaluation of the feature sets was strictly the performance under an objective function, such as Mean Square Error between the ground truth values and the predicted values.

Finally, the back propagation training algorithm used does not have a closed end solution, but it rather reiterates the weight's updating process until a certain maximum error is obtained, as measured by the objective function. Being this the case, as in many optimization problems, a subjective decision as to the number of training cycles used must be made. This arbitrary decision on the number of cycles makes it perfectly possible to stop training at a local minima, producing a less than optimum classifier.

CHAPTER 4: Experimental Setup

Several publications have carried out simulations on existing speech databases to generate the training and validation databases. Chen et al. [57], [59] conducted their experiments on the database provided by ITU T P series supplement 23 (1998), in order obtain 1338 audio files with their corresponding subjective MOS (MOS LQS). Falk et al. [58] followed the same approach, but expanded the scope to other pre-recorded databases, covering more codecs and noise scenarios, working with a total of 24256 audio files.

The experiments conducted for this thesis were all conducted with live speakers, unscripted conversations, and using devices of common commercial use. There were several reasons behind this decision:

1) A simple playback of pre recorded files through the network, although simpler to implement, does not reflect the totality of phenomena that occurs on a typical phone conversation, because bandwidth consumption is reduced by half, and all conversational or emotional phenomena are not reflected. Additionally, 60 minutes of unscripted conversations provide a diverse collection of words and phonemes, which can become challenging in a pre-recorded database.

2) A reduced number of call recordings may not provide sufficient data for quality assessment training, but do provide enough data for identification of network conditions. This is the case, because when a conversation is segmented in 20 ms frames of audio, 3 minutes of translate into 9,000 frames with various labels. A total of 60 minutes of audio were recorded for this thesis, so over 180,000 frames of audio were available for training and validation.

3) The implementations of the codecs, FEC and PLC on the chosen devices comply with all the necessary standards, and are of very common use in the marketplace [67], [68]. A custom implementation of such algorithms would have had induced an additional level of uncertainty in the results obtained.

4.1 General Considerations for the Experimental Setup

In order to train the different classifiers, it was necessary to develop an experimental environment, where complete control over the network conditions was possible. Furthermore, the ground truth values for the network conditions (delay, jitter, packet loss) needed to be recorded at all times, in order to be used as the training input to the classifiers.

Given the above mentioned requirements, the setup consisted on an isolated LAN, where two computers act as the end points, and a server acts as both the network emulator and the sip proxy. The two end points, two computers running the Eyebeam Softphone by CounterPath Technologies, were located in different rooms, so that participants could only communicate with each other through their respective headsets. Since all elements resided on the same switch, and no other traffic was present on that switch at the time of the experiments, it was safe to assume that any delay, jitter or packet loss present on the switch was due to the emulator in use. All participants in the experiments were acquaintances, so that they could conduct an unscripted conversation lasting several minutes.

4.2 Asterisk PBX

A Private Branch Exchange (PBX) is a private telephone network, which traditionally was used as a mean to connect to the PSTN with a minimized number of lines [69]. The original idea was that most of the calls within a company are between internal extensions, so having this private network could allow for significant savings. As time passed, PBX systems incorporated a variety of so called “Class 5” features, such as call transfer, call on hold, voice mail, and other such functionalities. Asterisk PBX is an open source software PBX, with native support for a variety of VoIP protocols, including SIP and RTP.

For the purpose of this thesis, Asterisk PBX was used as a SIP Proxy, SIP Registrar and Media Router (as specified in IETF RFC 3261 [70]), meaning that it simulates a completely functional phone central. Under this environment, all the endpoints involved in the experimental calls registered onto the same Asterisk PBX server. Asterisk PBX was installed on a Fedora Linux 8.0 operating system, with two separate network interfaces configured with different network addresses. Figure 4.1 illustrates the setup, and Figure 4.2 illustrates the typical signaling and media packet flow. As seen in the figures, Asterisk PBX is acting as a tandem element for the purpose of RTP traffic, which means that packet loss on one interface is concealed from the other, making all the typical non intrusive quality measurement algorithms ineffective, as any jitter or packet loss on one interface is not transmitted to the other one.

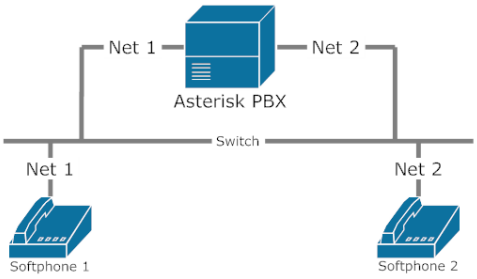

Figure 4.1: Experimental Network Setup

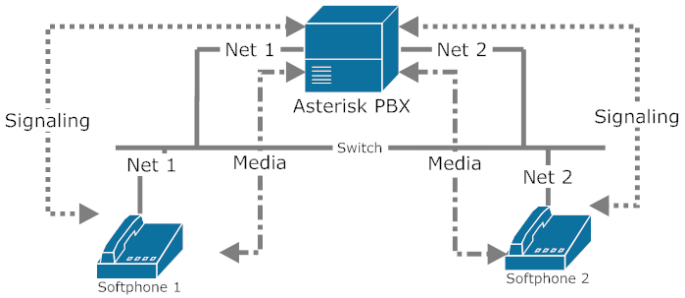

Figure 4.2: Signaling and Media Packet Flow

4.3 Netem

Netem is a Linux utility that allows a Linux server, acting as a router or bridge, to emulate the properties of a WAN. The latest version at the time this thesis was written allows emulating delay, jitter, packet loss, packet duplication and re ordering. Furthermore, the statistical properties of the delay and the packet loss can be specified. In the case of delay and jitter, the user can chose from normal, pareto or paretonormal distributions. In the case of packet loss, a correlation parameter can be set to control the burstiness of the loss.

Since the server had two Ethernet interfaces, and was effectively acting as a router, Netem was configured to act on one interface only, in order to best emulate the behavior of a tandem element (such as a TDM switch), as shown in Figure 4.3.

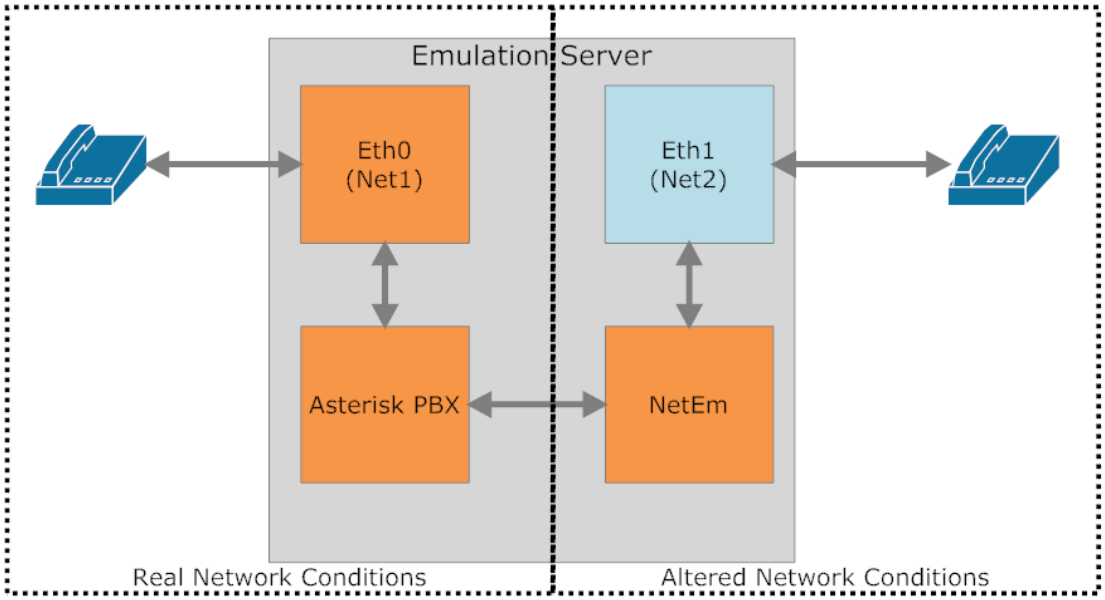

Figure 4.3: Architecture of Emulator Server

Another advantage of Netem is that it is controlled by the command line, in opposition to a GUI. This allows for the development of scripts that provide the time control necessary to properly label the extracted frames. Figure 4.4 shows a sample script, and Figure 4.5 shows the script used.

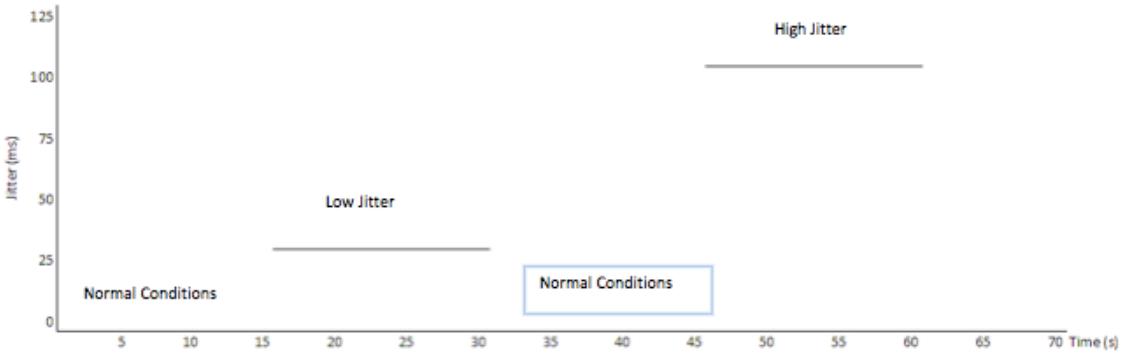

Figure 4.4: Typical Scripted Jitter Emulation

```
sleep 15
tc qdisc change dev eth0 root netem delay 150ms 25ms distribution normal
sleep 15
tc qdisc change dev eth0 root netem delay 150ms 0ms distribution normal
sleep 15
        tc qdisc change dev eth0 root netem delay 150ms 100ms distribution normal
```

Figure 4.5: The Code Necessary to Implement the Script

The same procedure was followed for packet loss emulation. It is important to note that a fix delay is necessary in order to add jitter to the connection, to avoid negative delay values. Also, even though the delay on the local switch was 0 for all practical purposes, this is not the case for any traffic routed through the internet, so a fix jitter component should be added in any case.

4.4 Eyebeam Softphone

Both end points ran instances of the Eyebeam Softphone, the commercial version of the X Lite softphone software by CounterPath Technologies. A Softphone is software that allows the making of telephone calls from a computer, without the use of any dedicated hardware. In general, most Softphones support the SIP protocol for signaling, and the RTP protocol for media transport. Eyebeam was chosen over other options, including its free version X Lite, because it has native support for a number of codecs, as seen in Figure 4.6. Under the scope of this thesis, G.729 was or particular interest, as it represents the most commonly used LPC codec.

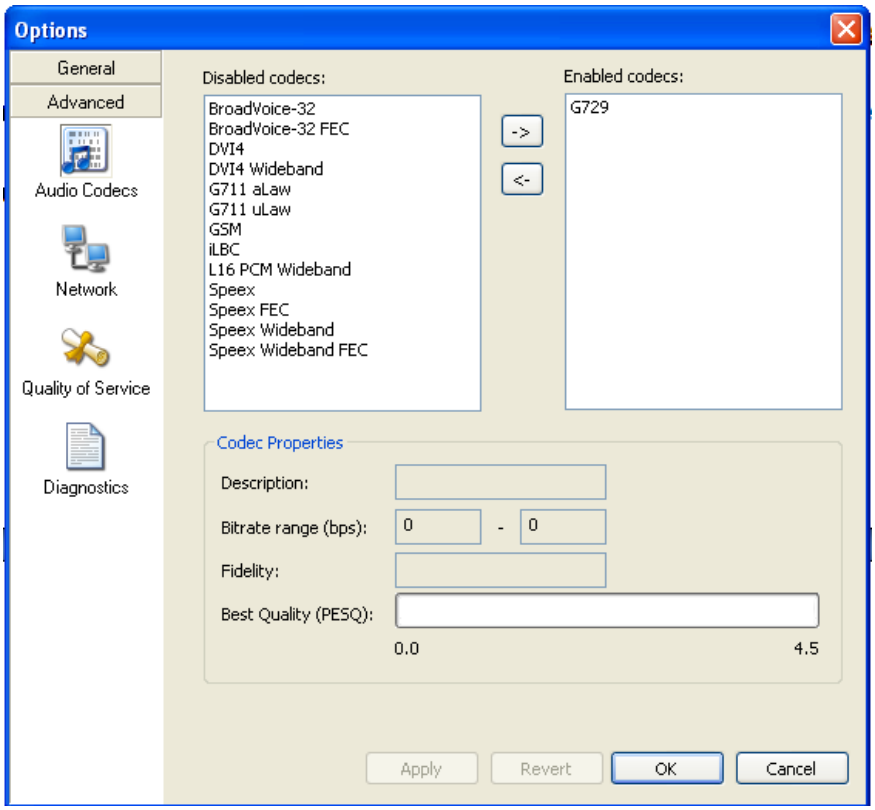

Figure 4.6: Codec Options for Eyebeam Softphone

A Softphone was preferred over dedicated hardware, because of availability, but also to enhance the comfort for the participants, who conducted all the test wearing headsets instead of handsets. Additionally, section 4.6 explains the advantage of using computers as the endpoints, in terms of data capturing.

4.5 Monitoring, Analysis and Capture Tools: Wireshark and Ping

Wireshark is a very popular open source packet sniffer. A packet sniffer is a type of software that captures the traffic on a network interface, and decodes it according to the relevant RFC . Wireshark [71] is a particularly powerful tool when it comes to VoIP calls analysis and troubleshooting, because it allows to capture all signaling and media traffic on the interface. Figure 4.7 shows a generic Wireshark capture of VoIP traffic, Figure 4.8

shows an analysis window for a typical call, and Figure 4.9 shows a RTP payload dump, which enable to capture all payload content on the RTP packets, and group it together on a single file for play out.

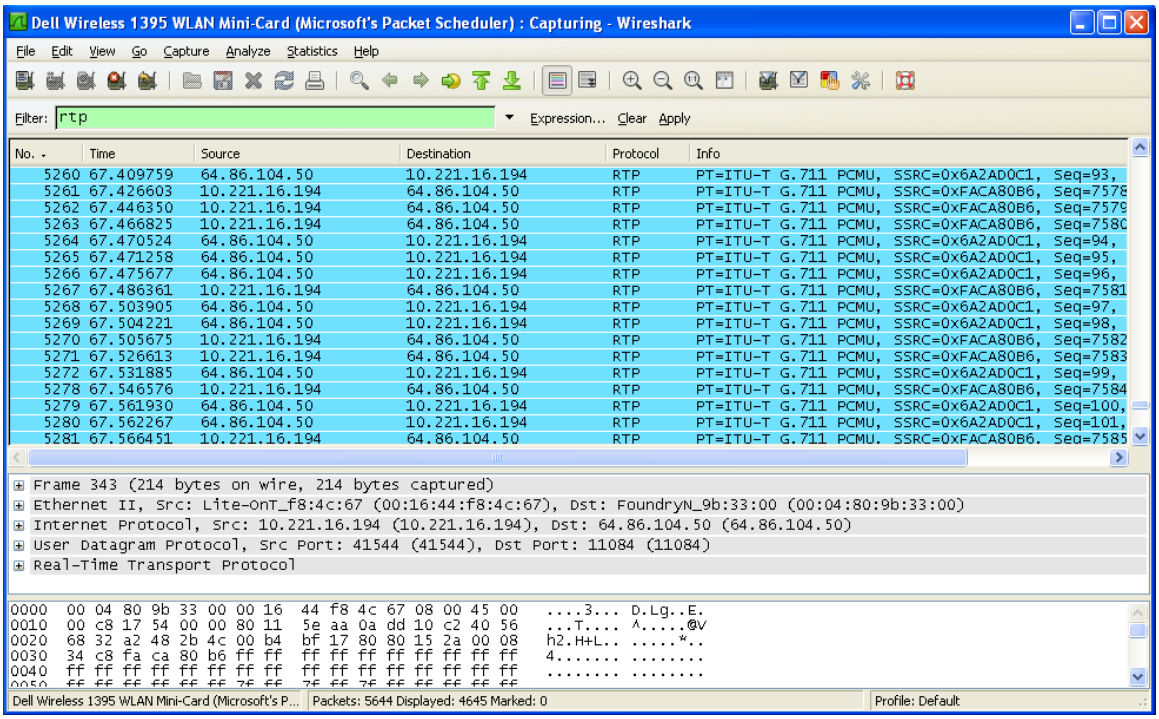

Figure 4.7: Typical Wireshark Capture of TCP Traffic

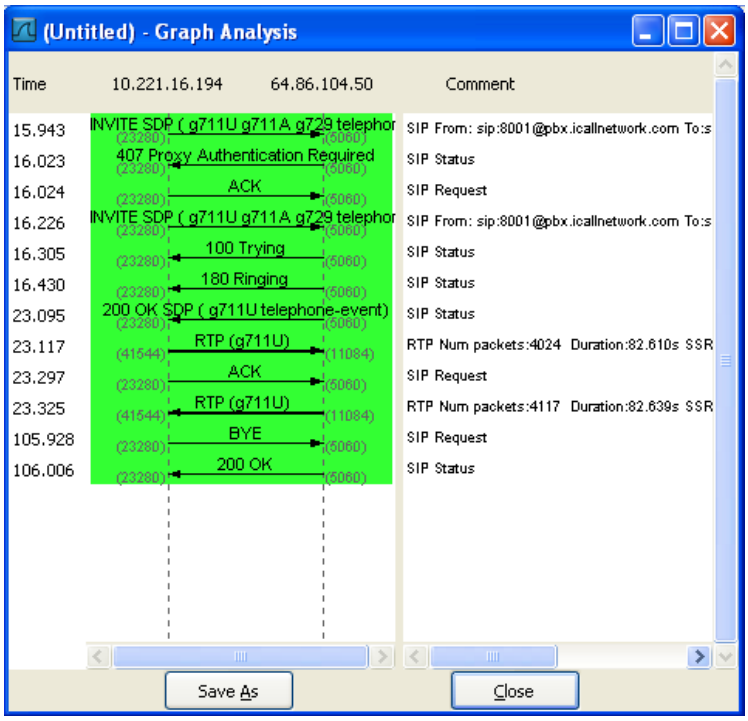

Figure 4.8: Typical Analysis Window in Wireshark

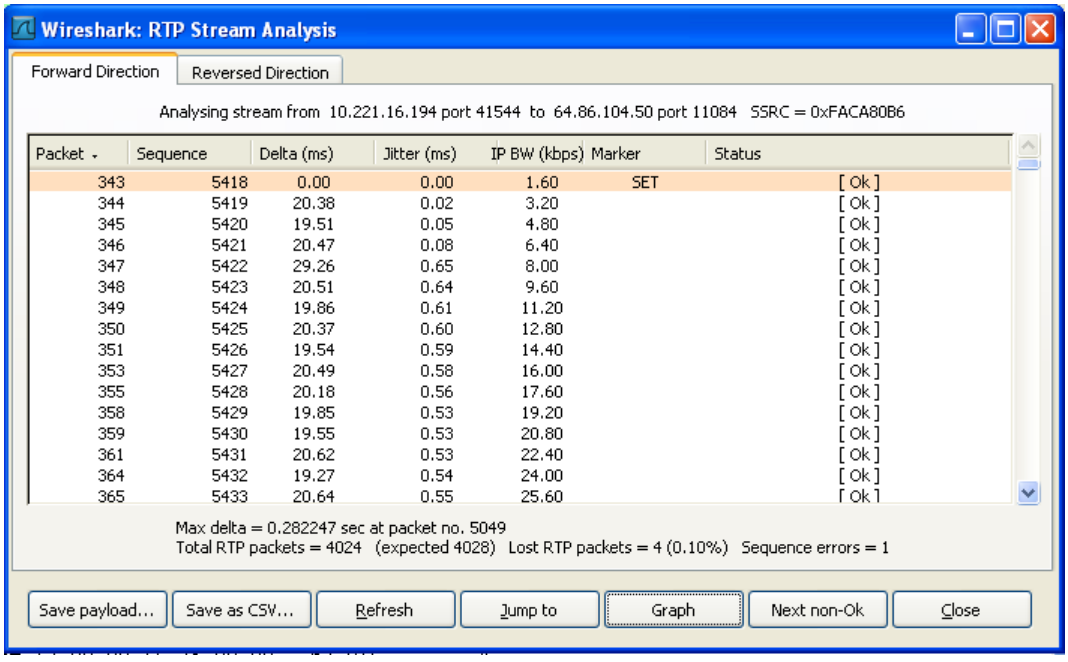

Figure 4.9: Wireshark Screen Allowing the Dump of Payload to a File

As we can see, Wireshark also provides the delay and jitter statistics in the Delta and Jitter columns of the Stream Analysis windows. This information was used

extensively, to corroborate that the emulator was performing as expected. Also, the Lost RTP packets statistic was extensively use, to insure the desired number of packets was lost during packet loss experiments. Since this statistic is only a summary of the packet s lost, but does not offer any insight into the distribution, the ping utility was used to keep track the network conditions in parallel with the experiments, as illustrated in Figure 4.10. The ping command will only indicate when packets are lost at the IP level, meaning that packet that arrive too late to be played out would not be reflected as lost with this utility. Nonetheless, it provides an intuitive idea of the network conditions between the end points and the emulator. Loss of continuity in the icmp_seq field indicate that a packet was lost, and the time field indicates the round trip delay taken for the packet to go from the end point to the network emulator. The jitter is observed as variations of the time field.

```
[root@asterisk1 ~]# ping 64.86.104.55
PING 64.86.104.55 (64.86.104.55) 56(84) bytes of data.
64 bytes from 64.86.104.55: icmp_seq=0 ttl=255 time=0.784 ms
64 bytes from 64.86.104.55: icmp_seq=1 ttl=255 time=0.725 ms
64 bytes from 64.86.104.55: icmp_seq=2 ttl=255 time=0.748 ms
64 bytes from 64.86.104.55: icmp_seq=3 ttl=255 time=0.644 ms
64 bytes from 64.86.104.55: icmp_seq=4 ttl=255 time=0.662 ms
64 bytes from 64.86.104.55: icmp_seq=5 ttl=255 time=0.631 ms
64 bytes from 64.86.104.55: icmp_seq=6 ttl=255 time=0.760 ms
64 bytes from 64.86.104.55: icmp_seq=7 ttl=255 time=0.899 ms
64 bytes from 64.86.104.55: icmp_seq=8 ttl=255 time=0.809 ms
64 bytes from 64.86.104.55: icmp_seq=9 ttl=255 time=0.880 ms
64 bytes from 64.86.104.55: icmp_seq=10 ttl=255 time=0.626 ms
64 bytes from 64.86.104.55: icmp_seq=11 ttl=255 time=0.852 ms
64 bytes from 64.86.104.55: icmp_seq=12 ttl=255 time=0.830 ms
64 bytes from 64.86.104.55: icmp_seq=13 ttl=255 time=0.764 ms
64 bytes from 64.86.104.55: icmp_seq=14 ttl=255 time=0.636 ms
64 bytes from 64.86.104.55: icmp_seq=15 ttl=255 time=0.677 ms
64 bytes from 64.86.104.55: icmp_seq=16 ttl=255 time=0.795 ms
64 bytes from 64.86.104.55: icmp_seq=17 ttl=255 time=19.1 ms
64 bytes from 64.86.104.55: icmp_seq=18 ttl=255 time=0.634 ms
```

Figure 4.10: Output of the Ping Command During one of the Experiments

4.6 Training Set Generation Procedure

Several alternatives where available for the capture of the audio necessary for the training of the classifier. Because tools such as Wireshark are of public domain, the simplest way to capture the voice stream was by capturing the RTP packets between the end points 1 and the unaffected interface Eth0 at the emulation server. This is a safe approach, because all packet loss, jitter, FEC and PLC algorithms are enacted on the Eth1 interface, connected to Net2. Figure 4.11 illustrates this process.

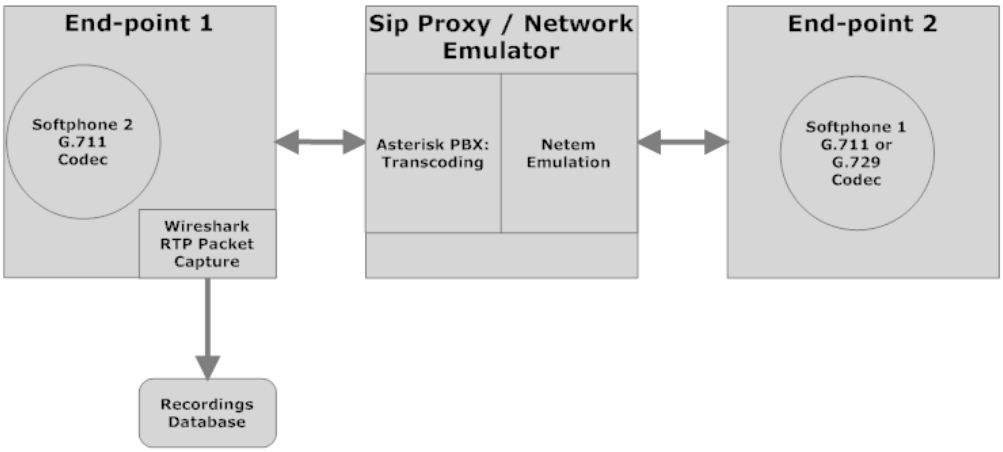

Figure 4.11: Audio Capturing Scheme

The received audio packets were then combined into a single large file, producing the different recordings. These recordings, along with the corresponding script indicating the conditions at each time, were processed by a Matlab script, to produce the final frames, which were used for the training process indicated in Chapter 3.

CHAPTER 5: Experimental Results

A large database of over 150,000 frames of audio was collected following the procedure shown in chapter 4. The data was collected from a group of 4 female and 3 male speakers, each of which conducted 5 conversations lasting 2 minutes each. Each conversation contained periods with various levels of jitter and packet loss, as well as periods without network disturbances, and were conducted using a standard implementation of the G.729 codec that does not incorporate Voice Activity Detection (VAD is only available in the G.729b codec). These frames were all labeled according to the corresponding network conditions, in order to be used as the training and testing databases for a series of artificial intelligence classifiers. To simplify the training process, the network conditions were discretized to the following labels:

1) Low Jitter: An arbitrary 60 ms of jitter
2) Medium Jitter: An arbitrary 120 ms of jitter
3) High Jitter: An arbitrary 320 ms of jitter
4) Low Packet Loss: An arbitrary 2% of packet loss
5) Medium Packet Loss: An arbitrary 10% of packet loss
6) High packet loss: An arbitrary 20% of packet loss
7) Normal: Negligible delay, jitter and packet loss

A number of systems were trained, and evaluated in terms of their accuracy when used to classify a testing database different to the database used for training (both in terms of percentage of correctly classified instances and the resulting confusion matrix). The summary of the results is shown in this chapter.

5.1 Classifiers Used

Only a few experiments were conducted using neural network classifiers, because the computational complexity of the back propagation training process was several times higher than the one for C4.5, and the improvement on classification accuracy was minimal if any. An outline of the few experiments conducted with the algorithm is shown in Table 5.1, along with the performance of the C4.5 classification tree. As we can see, the C4.5 algorithm's performance matches the one of the neural network classifier, but the training process takes a small fraction of the time. For this reason, C4.5 classification trees were used for most experiments.

Besides offering very similar accuracy at a significantly reduced training time, decision trees share a property with neural networks at the classification stage: their implementation is extremely computationally efficient, as it only requires simple comparisons between numerical values.

Classifier	Dataset	Average accuracy (%)
Neural Network	4 female speakers (3 for training, one for testing)	71.85
Neural Network	3 male speakers (2 for training, one for testing)	67.46
Neural Network	7 speakers combined (6 for training, one for testing)	65.39
C4.5	4 female speakers (3 for training, one for testing)	72.67
C4.5	3 male speakers (2 for training, one for testing)	69.82
C4.5	7 speakers combined (6 for training, one for testing)	68.20

Table 5.1: Comparison of Accuracy Between Neural Network Classifier and C4.5 Classification Tree When Classifying Between Normal Conditions and High Jitter

5.2 MFCC Performance

The best performance was obtained using a feature set composed of the Mel scale Cepstral Coefficients (MFCC). This set was composed of 60 features: 20 MFCCs coefficients, their delayed versions, and their twice delayed versions. The choice of 20 MFCCs was made based on trial and error, where less than 10 coefficients provided accuracies below 50%, and more than 25 coefficients showed a decline in accuracy. In the case of decision trees, the delayed and twice delayed version were replaced with the first and second differentials of the coefficients, because the C4.5 algorithm cannot induce differential relationships in the same way neural networks can (C4.5 will only look at the information gain obtained by splitting the feature space on a particular attribute).

5.2.1 Speaker Dependent Performance

Table 5.2 shows the performance obtained by training the system using the MFCCs feature set, on databases compiled from an individual male speaker, picked at random from the 3 available in the database. Two thirds of the database was used for training, and one third for testing.

Dataset	Accuracy	Confusion Matrix		
Normal vs. High Jitter	68.50%	A	B	
		572	132	A
		315	400	B
Normal vs. Medium Jitter	68.00%	A	B	
		592	97	A
		357	373	B
Normal vs. Low Jitter	62.86%	A	B	
		339	359	A
		168	553	B
Normal vs. High Packet Loss	67.65%	A	B	
		561	164	A

		295	419	B
Normal vs. Medium Packet Loss	63.78%	A	B	
		469	210	A
		304	436	B
Normal vs. Low Packet Loss	63.85%	A	B	
		450	238	A
		275	456	B

Table 5.2: Experimental Results for a Single Male Speaker Using C4.5

As we can see on the table, the classifier achieved good results in all scenarios, except for the Low Jitter level, where the confusion matrix shows incoherent values between the Low Jitter and the Normal Network. Table 5.3 shows representative values for a female speaker. We can observe that the system performed better in terms of jitter detection in the case of the female speaker.

Dataset	Accuracy	Confusion Matrix		
Normal vs. High Jitter	73.65%	A	B	
		515	175	A
		141	368	B
Normal vs. High Packet Loss	63.97%	A	B	
		592	97	A
		357	373	B

Table 5.3: Experimental Results for a Single Female Speaker

5.2.2 Multi-Speaker performance

Speaker dependency was also tested for the designed classifier. The following results were achieved when training for three male speakers, and classifying on the third. The process was repeated on the rotating all speakers as the testing set, obtaining the following results (average accuracy for all speakers):

- Normal vs. High Jitter: 69.88%
- Normal vs. Medium Jitter: 67.26%

- Normal vs. High Packet Loss: 66.79%
- Normal vs. Medium Packet Loss: 68.51%

After conducting the same tests for 3 female speakers, the following results were obtained

- Normal vs. High Jitter: 71.30%
- Normal vs. Medium Jitter: 70.11%
- Normal vs. High Packet Loss: 64.57%
- Normal vs. Medium Packet Loss: 63.72%

As we can see, the accuracy remains fairly steady regardless of speaker dependency. Finally, in the most generic case of a system trained with 6 speakers (3 male and 3 female), where 5 are used for training and 1 for testing, the performance of the classifier approximates the average of the performance for the male and female speakers, but still yields consistent results:

- Normal vs. High Jitter: 68.91%
- Normal vs. Medium Jitter: 68.33%
- Normal vs. High Packet Loss: 65.34%
- Normal vs. Medium Packet Loss: 64.77%

5.2.3 Accuracy Discerning Between Different Network Deficiencies

In addition to the accuracy classifying between normal network conditions and deficiencies such as jitter or packet loss, it is also of great interest to know whether or not the classifier can discern between the different types of network deficiencies. The following results illustrate the performance of a classifier, trained with both Jitter and

High Loss samples. As we can see, the classifier is more accurate than the one developed to classify against normal network conditions:

- High Jitter vs. High Packet Loss: 74.67% (on Multi-Speaker dataset)
- Medium jitter vs. Medium Packet Loss: 70.89% (on Multi-Speaker dataset)

5.3 LPCs and LPCCs Performance

After a long process of trial and error, it was evident that the best configuration in terms of linear prediction analysis was a combination of Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs). In addition to 10 linear prediction coefficients and 10 linear prediction cepstral coefficients, the feature set contained the delayed and twice delayed coefficients, for a total of 60 features. Although this configuration provided the best results, the performance was several percentage points lower than the one obtained for MFCCs across all the experiments.

5.3.1 Speaker Dependent Accuracy

Table 5.4 shows the results for a two male speakers. As we can see, for a male speaker, although the classifier is still effective, the performance is overall a few percentage points lower than the one obtained by MFCCs. In particular, when classifying for High Packet Loss, the confusion matrix shows an equal number of true negatives and false negatives (second row).

Dataset	Accuracy	Confusion Matrix		
Normal vs. High Jitter	65.61%	A	B	
		214	139	A
		101	244	B
Normal vs. High Packet Loss	63.36%	A	B	
		264	83	A
		175	177	B

Table 5.4: Experimental Results for a System Trained and Tested on a Male Speaker

Dataset	Accuracy	Confusion Matrix		
Normal vs. High Jitter	57.52%	A	B	
		226	120	A
		137	122	B
Normal vs. High Packet Loss	54.82%	A	B	
		217	129	A
		140	119	B

Table 5.5: Experimental Results for a System Trained and Tested on a Female Speaker

As seen in Table 5.5, by inspecting the confusion matrix, the classifier becomes ineffective at properly classifying Normal network conditions from High Packet Loss, in the case of female speakers, because the number of false negatives is greater than the number of true negatives.

5.3.2 Speaker Independent Accuracy

The following results show the performance of a classifier trained with the Multi-Speaker dataset of 3 male speakers and 3 female speakers. We can see that in this scenario the classifier is less effective than the MFCCs equivalent:

- Normal vs. High Jitter: 60.72%
- Normal vs. Medium Jitter: 57.05%
- Normal vs. High Packet Loss: 55.69%
- Normal vs. Medium Packet Loss: 51.23%

5.3.3 Accuracy Discerning Between Different Network Deficiencies

Finally, we are again interested in the performance of the classifier at discerning between jitter and packet loss. The following results show the performance of a classifier trained with both High Jitter and High Packet Loss frames on the Multi-Speaker dataset of 3 males and 3 females. As seen in the table, this classifier is unable to properly detect High Packet Loss.

- High Jitter vs. High Packet Loss: 54.65%
- Medium Jitter vs. Medium Packet Loss: 49.91%

5.4 LPCs Extracted from the G.729 Codec Payload

Even though the performance of the classifier was consistently better using the MFCCs features than the LPCs features, there are great potential practical advantages to the use of LPCs features, because many of them are readily available in the payload of certain codecs. For this particular study, the procedure detailed on ITU T G.729E was used to obtain the 10^{th} order LPCs from the received payload. Table 5.6 shows a summary of the performance obtained using these coefficients, their delayed versions, and their twice delayed version for a variety of scenarios. The last column indicates whether the number of true positives and true negatives is greater than the number of false positives and false negatives or not.

As seen in the table, the performance obtained with this feature was only effective at classifying a reduced number of scenarios, namely: High Jitter and High Packet Loss for male speakers and High Jitter for female speakers.

Dataset	Accuracy	Strong diagonal in confusion matrix
Normal vs. High Jitter, two male speakers	60.71%	Yes
Normal vs. High Packet Loss, two male speakers	57.23%	Yes
Normal vs. High Jitter, two female speakers	53.25%	Yes
Normal vs. High Packet Loss, two female speakers	49.8%	No
Normal vs. High Jitter, two male and two female speakers	56.25%	No
Normal vs. High Packet Loss, two male and two female spakers	52.23%	No
High Jitter vs. High Packet Loss	51.00%	No

Table 5.6: Summary of Performance Obtained with LPCs Features Extracted from the G.729 Codecs Payload

CHAPTER 6: Conclusion and Future Work

This thesis concentrated on the study of a particular issue in objective VoIP Quality Evaluation: how can parametric algorithms deal with the concealment of network conditions, caused by transcoding, various tandem elements, or hybrid TDM-VoIP networks. Although this is not a concern in current research publications, which deal mostly with native VoIP networks, the belief of the author is that a solution to this problem would be valuable for hundreds or thousands of whole sale carriers of voice services, who already have massive investments in their existing infrastructures. Needless to say, the transition to native packet networks, which properly relay all delay, jitter and packet loss information, will be rather progressive for such companies.

6.1 Overview of Work Conducted

This thesis aimed at achieving two main purposes: to provide a comprehensive overview of the challenges to the transport of voice services over packet networks, especially when tandem elements are involved, and to provide a method to estimate network conditions from the decoded PCM signal at the last decoder's output. In this sense, all the relative aspects of the RTP protocol and the most popular coding technologies were covered. Particularly, a brief description of the Code Exited Linear Prediction Coefficient (CELP) codecs was conducted, in order to illustrate the complexity of such algorithms, the valuable information that they contain for analysis purposes, and their particular sensibility to jitter and packet loss, due to their long term prediction processes.

A description of the delay, jitter and packet loss was also provided, giving particular emphasis to the buildup of delay and jitter across the different hops of the internet, which best illustrates the challenge of transporting voice over packet networks. Some of the most common correction and concealment algorithms were briefly described, in order to provide some insight into the roots for some of the audible artifacts users can perceive on VoIP calls where the underlying network is suffering of some particular deficiencies.

Finally, based on a comprehensive overview of the literature in the field of non intrusive objective quality evaluation, the motivation and strategy for the development of the Network Condition Estimator (NCE) was provided. The NCE is a novel element in the field, aimed at providing robustness algorithms, such as the E-Model. In addition, the NCE can potentially provide a diagnostic tool for the troubleshooting of VoIP networks: existing methods can provide a metric for the quality of a specific route with good correlation with subjective scores, but provide no justification for low scores. NCE, when combined with one of the existing algorithms, can provide the root cause for the low scores.

6.2 Conclusions

An advantage of training a system for specific network deficiencies is that classifiers can be trained based on data with extremely low levels of uncertainty, by following procedures such as the one detailed in Chapter 4. After a series of lengthy experiments, a significant amount of information was compiled to begin studying the possible classifiers. This process culminated in a configuration that provides consistent

acceptable levels of accuracy in many aspects. The classifier is speaker independent, gender independent, has sensibility to three different levels of jitter and delay, and can discern between the actual deficiencies being experienced. The last result is of particular relevance, because it implies that the several classifiers can be combined to provide accurate diagnostics of the underlying network's condition.

Although the best performance of the classifier was achieved using MFCCs, there is great motivation to use LPC features for the NCE, as they can many times be obtained without significantly reduced processing requirement from the payload content of RTP packets. For this reason, the experimental results of classifiers trained on LPC features sets were reviewed, but the conclusion is that with the current level of development their performance did not provide consistent results, especially when the LPCs were extracted from the payload of RTP packets containing CELP coded frames.

6.3 Future Work

Future study needs to be conducted into the root cause for the lower performance of the classifier when trained with LPCs extracted from the RTP payload, in opposition to the ones extracted by traditional means from the decoded signal. At first glance, the two possible reasons for the lower performance may be quantization level of the coefficients at the source, or an erroneous implementation of the extraction algorithm.

Beside the study of this inconsistency in the results, there are several aspects to be further studied for the development of NCEs. One of them relates to the discretization of the network conditions into three levels, which made it possible to use some powerful artificial intelligence technologies such as neural networks and C4.5 trees. Additional

work can be conducted, in order to find a more detailed estimation, possibly by means of linear regression or more complex non-linear mappings.

Also, the interaction between the NCE and existing algorithms such as the E-Model need to be studied further, in terms of the improvement attainable by the integration of the two algorithms, and their effect on the reliability and confidence of the predicted scores.

Finally, the jitter and packet loss phenomena were selected amongst the many affecting the transport of voice over packet networks, because they were relatively simpler to emulate on a closed environment, and produce very distinctive artifacts. Nonetheless, other phenomena such as delay and the effects of multiple transcoding stages can also be emulated and studied. In terms of transcoding, the codec in use is an element of particular weight in E-Model evaluation, so the detection of the most aggressive coder used in a transcoding sequence could significantly improve the accuracy of the estimations.

REFERENCES

[1] Ben Charny. (2005, January) CNET News. [Online]. http://news.cnet.com/Comcast pushes VoIP to prime time/2100 7352_3 5519446.html

[2] 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*, Montreal, Canada, 2004, pp. 273-280.

[3] Teck Kuen Chua and D.C. Pheanis, "QOS EVALUATION OF SENDER BASED LOSS RECOVERY TECHNIQUES FOR VOIP," *IEEE Network*, vol. 20, no. 6, pp. 14-22, December 2006.

[4] P. Gournay and K.D. Anderson, "PERFORMANCE ANALYSIS OF A DECODER BASED TIME SCALING ALGORITHM FOR VARIABLE JITTER BUFFERING OF SPEECH OVER PACKET NETWORKS," in *IEEE Conference On Acoustics, Speech And Signal Processing*, Tolouse, France, 2006, pp. 14-19.

[5] Gilbert Held, 2, Ed.: *John Wiley & Songs*, 1999, pp. 73-88.

[6] Floriano De Rango, Mauro Tropea, Peppino Fazion, and Salvatore Marano, "VOIP: SUBJECTIVE AND OBJECTIVE MEASUREMENT METHODS," *International Journal Of Computer Science And Network Security*, vol. 6, pp. 33-44, January 2006.

[7] B. Goode, "VOICE OVER INTERNET PROTOCOL (VOIP)," *Proceeding Of The IEEE*, vol. 90, no. 9, pp. 1495-1517, September 2002.

[8] IETF. (2003, July) RFC 3550: *A Transport Protocol for Real Time Applications*.

[9] ITU T. (2008, September) Recommendation E.800: “DEFINITIONS OF TERMS RELATED TO QUALITY OF SERVICE”.

[10] A. S. Spanias, "SPEECH CODING: A TUTORIAL REVIEW," *Proceeding Of The IEEE*, vol. 82, no. 10, pp. 1541-1582, October 1994.

[11] Wai C Chu, *Speech Coding Algorithms: Foundation And Evolution Of Standardized Coders*. Williamstown, MD: Wiley Interscience, 2003.

[12] ITU T. (1988, November) Recommendation G.711: "PULSE CODE MODULATION (PCM) OF VOICE FREQUENCIES".

[13] C. Hoene, A. Gunther, and A. Wolisz, "MEASURING THE IMPACT OF SLOW USER MOTION ON PACKET LOSS AND DELAY OVER IEEE 802.11B WIRELESS LINKS," in *28th Annual IEEE International Conference On Local Computer Networks*, Zurich, 2003, pp. 652-662.

[14] ITU T. (2007, January) Recommendation G.729: "CODING OF SPEECH AT 8 KBIT/S USING CONJUGATE STRUCTURE ALGEBRAIC CODE EXCITED LINEAR PREDICTION (CS ACELP)".

[15] ITU T. (2006, May) Recommendation G.723.1: "DUAL RATE SPEECH CODER FOR MULTIMEDIA COMMUNICATIONS TRANSMITTING AT 5.3 AND 6.3 KBIT/S".

[16] S.V. Andersen et al., "ILBC A LINEAR PREDICTIVE CODER WITH ROBUSTNESS TO PACKET LOSSES," in *IEEE Workshop On Speech Coding*, 2002, pp. 23-25.

[17] R. Salami et al., "DESIGN AND DESCRIPTION OF CS ACELP: A TOLL QUALITY 8 KB/S SPEECH CODER," *IEEE Transactions On Speech And Audio Processing*, vol. 6, no. 2, pp. 116-130, March 1998.

[18] Cisco Systems. (2006, February) *Understanding Delay In Packet Voice Networks*. Technical Publication.

[19] ITU T. (2003, May) Recommendation G.114: "ONE-WAY TRANSMISSION TIME".

[20] ITU T. (2003, November) Recommendation G.131: "TALKER ECHO AND ITS CONTROL".

[21] Matthew Stafford, *Signaling And Switching For Packet Telephony*. Boston, MA: Artech House, 2004.

[22] ITU T. (2003, May) *General Recommendations On The Transmission Quality For An Entire International*. Technical Publication.

[23] J. Rosenberg, Lili Qiu, and H. Schulzrinne, "INTEGRATING PACKET FEC INTO ADAPTIVE VOICE PLAYOUT BUFFER ALGORITHMS ON THE INTERNET," in *IEEE INFOCOM 2000. Nineteenth Annual Joint Conference Of The IEEE Computer And Communications Societies Proceedings*, Telaviv, Israel, 2000, pp. 1705-1714.

[24] M.Z. Santos, L.G.G. Kiatake, F. Meylan, S.T. Kofuji, and J.P. Courtiat, "SIMULATION AND ANALYSIS OF IP/ATM SWITCHING AND ROUTING," in *IEEE ICATM '99*, Colmar, France, 1999, pp. 267-275.

[25] J C Bolot, S. Fosse Parisis, and D. Towsley, "ADAPTIVE FEC BASED ERROR CONTROL FOR INTERNET TELEPHONY," in *IEEE INFOCOM '99*, New York, NY, 1999, pp. 1453-1460.

[26] J C., Garcia, A.V. Bolot, "THE CASE FOR FEC BASED ERROR CONTROL FOR PACKET AUDIO IN THE INTERNET," *ACM Multimedia Systems*, 1996.

[27] S. Praestholm, S.S. Jensen, S.V. Andersen, and M.N. Murthi, "ON PACKET LOSS CONCEALMENT ARTIFACTS AND THEIR IMPLICATIONS FOR PACKET LABELING IN VOICE OVER IP," in *IEEE ICME '04*, Taipai, Taiwan, 1003, pp. 1667-1670.

[28] B. Kovesi and S. Ragot, "A LOW COMPLEXITY PACKET LOSS CONCEALMENT ALGORITHM FOR ITU T G.722," in *IEEE ICASSP '08*, Honolulu, Hawaii, 2008, pp. 4769-4772.

[29] C. Perkins, O. Hodson, and V. Hardman, "A SURVEY OF PACKET LOSS RECOVERY TECHNIQUES FOR STREAMING AUDIO," *IEEE Network*, vol. 12, no. 5, pp. 40-48, September 1998.

[30] N. Aoki, "VOIP PACKET LOSS CONCEALMENT BASED ON TWO SIDE PITCH WAVEFORM REPLICATION TECHNIQUE USING STEGANOGRAPHY," in *IEEE TENCON*, vol. 3, Chiang Mai, Thailand, November 2004, pp. 52-55.

[31] ITU T. (1998, March) E.411: "OVERALL NETWORK OPERATION, TELEPHONE SERVICE, SERVICE OPERATION AND HUMAN".

[32] A. Kansal and A. Karandikar, "ADAPTIVE DELAY ESTIMATION FOR LOW JITTER AUDIO OVER INTERNET," in *IEEE Globecom '01*, San Antonio, TX, 2001, pp. 2591-2595.

[33] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "ADAPTIVE PLAYOUT MECHANISMS FOR PACKETIZED AUDIO APPLICATIONS IN WIDE AREA NETWORKS," in *IEEE INFOCOM '94*, Toronto, Canada, 1994, pp. 680-688.

[34] A. Lakaniemi, J. Rosti, and V.I. Raisanen, "SUBJECTIVE VOIP SPEECH QUALITY EVALUATION BASED ON NETWORK MEASUREMENTS," in *IEEE International Conference On Communications*, St. Petersburg, Russia, 2001, pp. 748-752.

[35] ITU T. (1996, June) P.800: "METHODS FOR SUBJECTIVE DETERMINATION OF TRANSMISSION QUALITY".

[36] Alexander Raake, *Speech Quality Of Voip: Assessment And Prediction*: John Wiley & Sons, 2007.

[37] A.W. Rix, J.G. Beerends, D. S. Kim, P. Kroon, and O. Ghitza, "OBJECTIVE ASSESSMENT OF SPEECH AND AUDIO QUALITY - TECHNOLOGY AND APPLICATIONS," *IEEE Transactions On Audio, Speech, And Language Processing*, vol. 14, no. 6, p. November, 2006 1890-1901.

[38] M. Karjalainen, "A NEW AUDITORY MODEL FOR THE EVALUATION OF SOUND QUALITY OF AUDIO SYSTEM," *IEEE ICASSP*, no. Texas, p. 608-611, March 1985.

[39] J. Anderson. (2001, October) *Methods For Measuring Perceptual Speech Quality*. Agilent Technologies White Paper.

[40] ETSI. (2000, October) EG 201: Speech Processing, Transmission & Quality Aspects (STQ); QoS Parameter Definitions And Measurements; Parameters For Voice Telephony Service Required Under The ONP Voice Telephony Directive.

[41] L. Sun and E. Ifeachor, "NEW MODELS FOR PERCEIVED VOICE QUALITY PREDICTION AND THEIR APPLICATIONS IN PLAYOUT BUFFER."

OPTIMIZATION FOR VOIP NETWORKS," in *IEEE International Conference On Communications*, vol. 3, Paris, France, June 2004, pp. 1478-1483.

[42] P. Gray, M.P. Hollier, and R.E. Massara, "NON-INTRUSIVE SPEECH QUALITY ASSESSMENT USING VOCAL TRACT MODELS," *IEEE Proceedings On Vision, Image And Signal Processing*, vol. 147, no. 6, pp. 493-501, December 2000.

[43] J. Liang and R. Kubichek, "OUTPUT BASED OBJECTIVE SPEECH QUALITY," in *IEEE Vehicular Technology Conference*, Stockholm, Sweden, 1994, p. 1719-1723.

[44] ITU T. (2004, March) Recommendation P.563: "SINGLE ENDED METHOD FOR OBJECTIVE SPEECH QUALITY ASSESSMENT IN NARROW BAND TELEPHONY APPLICATIONS".

[45] D. Picovici and A.E. Mahdi, "NEW OUTPUT BASED PERCEPTUAL MEASURE FOR PREDICTING SUBJECTIVE QUALITY OF SPEECH," in *ICASSP '04*, Quebec, Montreal, 2004, pp. 633-636.

[46] ITU T. (2007, December) Recommendation P.562: "ANALYSIS AND INTERPRETATION OF INMD VOICE SERVICE MEASUREMENTS".

[47] ITU T. (2000, May) Recommendation G.107: "THE E-MODEL, A COMPUTATIONAL MODEL FOR USE IN TRANSMISSION PLANNING".

[48] ITU T. (1996, July) Recommendation P.861: "OBJECTIVE QUALITY MEASUREMENT OF TELEPHONE BAND (300-3400 HZ) SPEECH CODECS".

[49] J. G. Beerends and J. A. Stemerdink, "A PERCEPTUAL SPEECH QUALITY MEASURE BASED ON A PSYCHOACOUSTIC SOUND REPRESENTATION," *Journal Of The Audio Engineering Society*, vol. 42, p. 115, April 1994.

[50] A.E. Mahdi, "Voice Quality Measurement in Modern Telecommunication Networks Systems," *EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, pp. 25-32, June 2007.

[51] A. P. Hekstra, A. W. Rix, and M. P. Hollier J. G. Beerends, "PERCEPTUAL EVALUATION OF SPEECH QUALITY (PESQ), THE NEW ITU STANDARD FOR END TO END SPEECH QUALITY ASSESSMENT. PART II - PSYCHOACOUSTIC MODEL," in *ICASSP '01*, Salt Lake City, UT, 2001, pp. 749-752.

[52] A.W. Rix and M.P. Hollier, "THE PERCEPTUAL ANALYSIS MEASUREMENT SYSTEM FOR ROBUST END TO END SPEECH QUALITY ASSESSMENT," in *ICASP '00*, Istanbul, Turkey, 2000, pp. 1515-1518.

[53] ITU T. (2001, June) Recommendation P.862: "PERCEPTUAL EVALUATION OF SPEECH QUALITY (PESQ): AN OBJECTIVE METHOD FOR END-TO-END SPEECH QUALITY ASSESSMENT OF NARROWBAND TELEPHONE NETWORKS AND SPEECH CODECS".

[54] P. Paglierani and D. Petri, "UNCERTAINTY EVALUATION OF SPEECH QUALITY MEASUREMENT IN VOIP SYSTEMS," in *IEEE International Workshop On Advanced Methods For Uncertainty Estimation In Measurement*, Trento, Italy, 2007, pp. 104-108.

[55] 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 *ICASSP '01*, Salt Lake City, Utah, 2001.

[56] ITU T. (February, 2001) 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".

[57] Guo Chen and V. Parsa, "NONINTRUSIVE SPEECH QUALITY EVALUATION USING AN ADAPTIVE NEUROFUZZY INFERENCE SYSTEM," *IEEE Signal Processing Letters*, vol. 12, no. 5, p. 403-406, May 2005.

[58] T.H. Falk and W. Y. Chan, "SINGLE ENDED SPEECH QUALITY MEASUREMENT USING MACHINE LEARNING METHODS," *IEEE Transactions On Audio, Speech, And Language Processing*, vol. 14, no. 6, p. 1935-1947, November 2006.

[59] G. Chen and V. Parsa, "OUTPUT BASED SPEECH QUALITY EVALUATION BY MEASURING PERCEPTUAL SPECTRAL DENSITY DISTRIBUTION," *IEEE Electronics Letters*, vol. 40, no. 12, pp. 783-785, June 2004.

[60] J. Nayak and P.S. Bhat, "IDENTIFICATION OF VOICE DISORDERS USING SPEECH SAMPLES," in *IEEE TENCON '03*, Bangalore, India, 2003, pp. 951-953.

[61] S. Ruggieri, "EFFICIENT C4.5 [CLASSIFICATION ALGORITHM]," *IEEE Transactions On Knowledge And Data Engineering*, vol. 14, no. 2, pp. 438-444, March 2002.

[62] Reynolds D.A., "EXPERIMENTAL EVALUATION OF FEATURES FOR ROBUST SPEAKER IDENTIFICATION SPEECH AND AUDIO PROCESSING," *IEEE Transactions*, vol. 2, no. 4, p. 639-643, October 1994.

[63] T.Q. Nguyen, "DIGITAL FILTER BANK DESIGN QUADRATIC CONSTRAINED FORMULATION SIGNAL PROCESSING," *IEEE Transactions*, vol. 43, no. 9, p. 2103-2108, September 1995.

[64] E. Wong and S. Sridharan, "COMPARISON OF LINEAR PREDICTION CEPSTRUM COEFFICIENTS AND MEL-FREQUENCY CEPSTRUM COEFFICIENTS FOR LANGUAGE IDENTIFICATION," in *IEEE Proceedings International Symposium On Intelligent Multimedia, Video And Speech Processing*, Kowloon Shangri La, Hong Kong, 2001, p. 95-98.

[65] S. Young, *The HTK Book: For HTK Version 2.1*. Cambridge, England: Cambridge University Press, 1997.

[66] J. D. Markel and A. H. Gray, *Linera Prediction Of Speech*. New York, NY: Springer Verlag, 1976.

[67] Digium Technologies. www.asterisk.org.

[68] Counterpath Technologies. www.counterpath.com.

[69] K.S. Chava and J. How, "INTEGRATION OF OPEN SOURCE AND ENTERPRISE IP PBXS," in *IEEE TridentCom 2007*, Budapest, Hungary, 2007, pp. 1-6.

[70] IETF. (2002, June) RFC 3261: "SIP: SESSION INITIATION PROTOCOL V.2.0".

[71] http://www.wireshark.org/.