Université d'Ottawa • University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	BOISVERT, Mario
ADRESSE POSTALE / MAILING ADDRESS:	257 RUE DES PEUPLIERS MASSON-ANGERS QC J8M1T8
GRADE / DEGREE:	**ANNÉE D'OBTENTION / YEAR GRANTED**
M.A.Sc. (Electrical Engineering)	2003
TITRE DE LA THÈSE / TITLE OF THESIS:	Minimizing State Error Propagation in Low-Bit Rate Speech Codec for Voice over IP

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photographie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

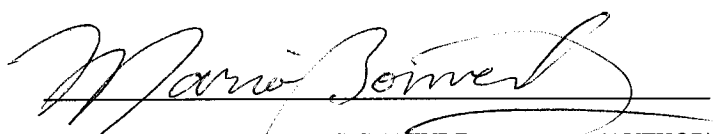
The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

16 May 03

DATE



(AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

BOISVERT, Mario

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Minimizing State Error Propagation in Low-Bit Rate Speech Codec for Voice over IP

Tyseer Aboulnasr

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

R. Goubran

M. Bouchard

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Minimizing State Error Propagation in Low-Bit Rate Speech Codec for Voice over IP

By

Mario Boisvert, B. Comp. Sc.

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements of the degree of
Master of Applied Sciences
in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

12 May 2003

Abstract

Today's Internet and intranets are packet-switched networks where the Internet Protocol (IP) is the most widespread of all network protocols in use. Such network protocol poses serious constraints on the real-time transmission of packets such as in the case of voice applications, namely Voice over IP (VoIP) applications. The "best effort" delivery mechanism and the lack of guarantee of Quality of Service (QoS) of packet networks are known to cause packet arrival problems that result in packet losses. Obviously, the loss of packets impairs the quality of the speech at the receiving end.

In this thesis, we focus on two main areas in the implementation of VoIP systems. Initially, we review speech compression techniques to better understand the operation and characteristics of speech codecs. This allows us to select a promising speech codec, namely, the ITU-T G.729A speech codec that will be used during our demonstrations and investigations. Secondly, we review the operation of packet-switched networks, more specifically IP networks, for the purpose of understanding the degradation effects they cause to VoIP systems. With this knowledge, we formalize a set of constraints and requirements that allows us to properly analyze the effect of packet losses over IP networks.

Finally, we propose a closed-loop over the network method to assist the codec in improving its speech quality performance in periods of packet losses. The method, named State Error Correction (SEC), is described in detail and its performance is assessed through simulations.

Table of Contents

List of Figures

List of Tables

Acronyms

ABS	Analysis-By-Synthesis
ACELP	Adaptive Code Excited Linear Prediction
ACR	Absolute Category Rating
ADPCM	Adaptive Differential Pulse Code Modulation
BFI	Bad Frame Indicator
CBR	Constant Bit Rate
CCR	Comparison Category Rating
CELP	Code-Excited Linear Prediction
CMOS	Comparison Mean Opinion Score
CNG	Comfort Noise Generator
CPU	Central Processing Unit
CS-ACELP	Conjugate-Structure Algebraic CELP
DCR	Degradation Category Rating
DMOS	Degraded Mean Opinion Score
DOD	Department Of Defense
DPCM	Differential Pulse Code Modulation
DSP	Digital Signal Processing
DTMF	Dual Tone Multi-Frequency
FDDI	Fiber Distributed Data Interface
FEC	Forward Error Correction
FFT	Fast Fourier Transform
IP	Internet Protocol
LAN	Local Area Network

ISP	Internet Service Provider
ITU	International Telecommunication Union
LD-CELP	Low-Delay CELP
LKG	Last Known Good
LMS	Least Mean Square
LP	Linear Prediction
LPC	Linear Prediction Coefficients
LSF	Line Spectrum Frequencies
LSP	Line Spectrum Pairs
MA	Moving Average
MAN	Metropolitan Area Network
MBSD	Modified Bark Spectral Distortion
MELP	Mixed-Excitation Linear Prediction
MIPS	Million of Instructions per Second
MNB	Measurement Blocks
MOS	Mean Opinion Score
MSE	Mean Square Error
NIC	Network Interface Card
PC	Personal Computer
PCM	Pulse Code Modulation
PESQ	Perceptual Evaluation of Speech Quality
POTS	Plain Old Telephone Service
PSQM	Perceptual Speech Quality Measure
PSTN	Public Switched Telephone Network

QoS	Quality of Service
RAM	Random Access Memory
RELP	Residual-Excited Linear Prediction
ROM	Read Only Memory
RPE	Regular Pulse Excitation
RSVP	Resource reSerVation Protocol
RTCP	Real-Time Control Protocol
RTP	Real-Time Protocol
RST	Reset-to-Defaults
SE	Squared Error
SEC	State Error Correction
SIP	Session Initiation Protocol
SNR	Signal-to-Noise Ratio
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VAD	Voice Activity Detector
VBR	Variable Bit Rate
VELP	Voice-Excited Linear Prediction
VQ	Vector Quantization
VWB	Voice WorkBench
VoIP	Voice over IP
WAN	Wide Area Network

Acknowledgements

I would like to express my overwhelmed appreciation to my thesis supervisor, Dr. Tyseer Aboulnasr, for her continuous kindness and encouragements. Her guidance through words of wisdom and keen insights was central in making this experience exciting and memorable.

Above all, special thanks to my spouse Guylaine and daughter Gabrielle for their continuous encouragements and support over those long years. Without their love and support I would not have succeeded in completing all the requisites for the Master Diploma.

Special thanks to Ms. Lucette Lepage, Graduate Studies Academic Assistant, for her superb support that assisted me comply with the administrative demands of the program. As a part-time student, it was very difficult to tract the mandatory administrative steps for the completion of the program.

Finally, I would like to thank my family and friends that shared encouragement words during the course of my study program.

1.0 INTRODUCTION

1.1 BACKGROUND

In the last decade, the exponential growth in the telecommunication industry has contributed to the widespread use of packet-switched networks in the majority of corporate intranets and the Internet, as we know it today. The flexibility associated with the digital representation of communication signals quickly brought packet-switched networks to compete against the traditional circuit-switched networks currently used in telephony. Through this communication revolution [74], we are witnessing the convergence of voice and data information common infrastructures onto a single infrastructure. This opens up a myriad of services such as e-mail, collaborative tools, conferencing applications, Voice over IP[1] (VoIP), and other such communication services, providing flexibility and mobility to the end users. Experts [6] in the field of speech and language communications services predict major changes in the areas of networks such as network access and devices that connect to networks and provide network services. Another associated communication novelty is the use of cell phones by the general public. Nowadays, it would be inconceivable for anyone not to use at least one of these communication services.

Large corporations usually gain access to the General[2] Switched Telephone Network (GSTN) through their own or rented Private Branch Exchange (PBX). The PBX equipment is often more than a decade old and its speed has already been surpassed by a factor of two to three orders of magnitude. The faster operation of packet-switched networks allows for the development of new

---

[1] As the name suggests, Voice over IP is the transmission of voice data over the Internet Protocol. IP is the most common network protocol in use today, hence, the popularity of VoIP. However, similar implementations over other network layers such as Voice over Frame Relay and Voice over ATM exist.

[2] Public Switched Telephone Network (PSTN) is now called the General Switched Telephone Network (GSTN) following the CRTC decision to deregulate the Public Switched Telephone Network (PSTN)

and highly integrated services. Hence, it logically makes sense to merge PSTN services with packet network applications. Advances in technology, particularly in the area of network services convergence, have made it possible for voice communications to be carried over packet-switched networks that were previously dedicated to data transmission. VoIP is perceived to be the way of the future by many, based on the assumption that substantial monetary savings are possible. In fact, long distance calls savings initiated the attraction of VoIP systems. However, long distance calls savings were associated with a lower speech quality than existing telephony services over the GSTN and not everyone was willing to accept the tradeoff. This loss of quality may be acceptable for intercontinental social communications but is often inadequate for a corporation relying on accurate communications. The eventual cost savings associated with managing and maintaining a single network infrastructure rather than two (i.e. data and voice transport on a single network) became the second attraction factor toward VoIP. Other opportunities to significantly reduce the operating costs and benefits of moving to the new technology are often advertised by hardware and software providers of the technology, but we perceive that the most attractive benefit of all is the almost limitless set of services that can be implemented when compared with GSTN. Despite the perceived cost savings, the deployment of VoIP technology did not take off as expected and we believe it is due to the fact that this new technology requires an initially large investment to ensure the same services and speech quality as those offered by the GSTN is achieved and maintained.

The GSTN is a sound architecture with its dedicated communication infrastructure that has evolved over the years to its current state. It uses control and signaling functions to deliver quality speech to its users.

By contrast, the IP network (e.g. Internet or corporate intranet) is the communication infrastructure used in VoIP systems. This infrastructure is not new but it used to be dedicated to data

communications. Consequently, there are fundamental differences between the two infrastructures as highlighted in Table 1.1.

GSTN Infrastructure	*IP Network Infrastructure*
Built for: Voice.	Built for: Data.
Delay sensitive: Any segments of a conversation convey information that must be played at a specific moment in time (i.e. real-time) to fully represent the communicated meaning.	Delay insensitive: Late data still contains its content that can be perused time after time.
Long hold time: Exclusive use of the circuit. Non-efficient utilization of the infrastructure.	Short hold time: Bursty data traffic. Efficient utilization of the infrastructure.
Narrow bandpass: Every call has dedicated bandwidth.	Wide bandpass: Every request or use generates bandwidth contention.

Table 1.1 – Fundamental differences between GSTN and IP Networks

The VoIP architecture is relatively new and must implement control and signaling functions to deliver its services, as well as interoperate with the GSTN. IP network protocols, namely real-time and multimedia protocols are used to implement the voice architecture. While signaling and control functions can be implemented without much difficulty, they are all sources of additional traffic and, when combined to existing traffic, have the net effect of increasing the bandwidth contention, thereby negatively affecting real-time delivery of voice streams by generating network delays, inter-packet arrival delay variance or jitter, and packet losses. Therefore, a VoIP application must address all these factors for it to be successful.

1.2 MOTIVATION

This thesis deals with the enhancement of a low-bit rate speech codec for minimizing the state error propagation in VoIP communication systems. The growing interest in real-time audio and video transmissions over the Internet [6], as well as over private networks is creating a demand for VoIP systems. As explained earlier, the transmission of real-time voice data on a packet-switched network (i.e. IP network in this case) that was originally designed to transfer non real-time data poses some challenges. The Internet poses even more serious problems due to the pervasive or piece-wise

approach used in its design. On the other hand, better design control can be exercised with private packet-switched networks that typically have high data rates and ensure reliable network services such as VoIP to their users. To be successful, VoIP systems must constantly provide a speech quality level as good as the General Switched Telephone Network does. This poses a major challenge: the scalability of the VoIP architecture to ensure its survivability. The driving factors relate to the need to scale to available bandwidth (e.g. scalable systems). They originate from:

- Generally speaking, users constantly request more network services such as videos, conferencing, interactive games, etcetera, that all have the potential to increase the bandwidth contention, hence, the potential to degrade the speech quality from time to time;
- It is predicted [6] that the new era will basically allow access to network services from anywhere, but often using low-rate access; and
- From a corporation's point of view, adding users should not cause immediate bandwidth contention, forcing infrastructure upgrades.

The bottom line is bandwidth conservation until such a time Quality of Service mechanisms, such as the Resource reSerVation Protocol (RSVP), become widely implemented in IP networks.

There is a requirement to provide speech codecs that are robust to errors and perform well at different bit rates. Some codecs can already switch the operating bit rate or bandwidth on the fly, allowing a user to adapt to changing network conditions while maintaining the quality of speech. On the other hand, robustness to network impairments also contributes at maintaining quality speech. Current low bit rate codecs have limited capability to repair severely damaged streams. Even worse, they often propagate a residual error known as state error, which builds up and provides

significant speech quality degradation of subsequent speech segments. Current Packet Loss Recovery (PLC) methods do not consider the repair of the residual or state error. The effects of the state error propagation are not well understood. We are therefore interested in the study of those effects applied to a VoIP system. The questions we want to answer are:

- Can we avoid memory state error propagation?
- Can the network react quickly enough to correct the memory states de-synchronization?

We will be able to answer these questions once the following objectives or fulfilled.

1.3 OBJECTIVES

The main objective of this thesis is to propose an algorithm aimed at enhancing the codec robustness to packet losses through re-synchronization of its memory state. The proposed algorithm will be fully described and experimental data provided to substantiate its merits.

Secondly, a set of supportive objectives is required to ensure the author has the necessary knowledge to understand and identify issues and requirements for the proposed algorithm. They are:

- Provide a literature review of speech compression principles and methods to understand their most desired characteristics when used in Voice over IP applications. This is important to understand their effect on the IP infrastructure and their impact on the VoIP architecture.
- Provide a literature review of packet-switched networks as they apply to IP networks. VoIP systems rely on an IP network infrastructure to deliver VoIP services. It is important to properly understand the problems associated with this infrastructure.
- Investigate the effect of packet losses generated by the IP networks on the quality of the speech.

- Provide an in depth look at packet loss recovery and concealment techniques that currently exist.
- Identify and use standard methods to evaluate speech quality.

Finally, a prototype application implementing the selected codec and proposed enhancement will be developed. This application will be necessary to evaluate the merits of the proposed algorithm.

1.4 THESIS ORGANIZATION

This section provides a summary of each chapter in this thesis.

Chapter 2: Speech Compression

In chapter 2, we introduce the characteristics of the human speech production and hearing systems, and some of the characteristics of spoken sounds. We follow by presenting the basic principles behind the digitization of speech signals. Then, an extensive review of compression methods is provided with the objective of establishing a common understanding of compression approaches. Three families or classes of speech compression methods are presented: waveform-based, source-model-based, and hybrids methods. Our attention will mainly focus on the hybrid compression methods. Standard speech compression methods or codecs are presented with their associated advantages and disadvantages, helping us understand the acceptable tradeoffs and limitations when selecting a compression algorithm for a VoIP application. Methods to subjectively and objectively evaluate the speech quality of codecs are introduced. Finally, for the purpose of running simulations for this thesis, we describe the ITU-T G.729A algorithm in more details.

Chapter 3: IP Networks

Chapter 3 provides a review of the fundamental packet-switched networks theory associated with the operation of IP networks as well as the associated protocols enabling VoIP. As discussed in the background section of Chapter 1, VoIP systems are in fact characterized by an architecture that uses a communication infrastructure, namely, the IP network. IP network implementations were historically aimed at carrying non real-time data in a reliable manner but that mechanism is inadequate for the real-time transfer of voice data. We explain the main problems associated with IP networks and describe the best effort delivery mechanism used in VoIP. We close the chapter by introducing multimedia protocols such as RTP/RTCP, H.323, and SIP, which are widely used in VoIP system implementations. We discuss their most important functions, as they are major components of the VoIP architecture.

Chapter 4: VoIP Systems Issues and Requirements

Chapter 4 builds on material presented in chapters 2 and 3 to identify the major causes of speech quality degradation in an IP network: the delays associated with a best effort delivery service lacking Quality Of Service (QoS) guarantees, and packet losses as a side effect of using a packet-switched network. We then describe the effects of the network on the processed streams, and investigate the effect of lost packets. We come to a first observation, that speech quality is a direct result of the ability to repair error, thus, leading us to a brief review of packet repair methods commonly known as sender-based recovery and receiver-based concealment techniques. With the knowledge acquired so far, we provide a detailed description of speech quality degradations with respect to the error associated with the loss of packets, that is, the concealment error (directly associated with a packet loss) and the memory state error (where, following the first concealment of a packet, the encoder and decoder memories are no longer the same).

Chapter 5: Memory State Error Correction (SEC) Algorithm

At the heart of the thesis, Chapter 5 proposes an algorithm to minimize the propagation of the memory state error. The method aims at supplementing the codec algorithm (add-on that is easily adaptable to other low bit rate codecs), in this case the ITU-T G.729A, a low-bit rate codec that must maintain the encoder and decoder memory states in synchronization for its optimal operation. As a consequence of a missing packet, the decoder memory state drifts from its intended state and generates state errors when decoding subsequent frames. We describe a closed-loop over the network method, called State Error Correction (SEC), that re-initializes internal encoder and decoder memory values with the objective of re-synchronizing the memory states, hence, minimizing the propagation of the memory state error following the loss of one or more packets.

Two re-initialization schemes aimed at re-synchronizing the memory states will be described, namely the Reset-to-defaults (SEC-RST) scheme and the Last-Known-Good (SEC-LKG) scheme. Thorough experimentations of the proposed algorithm have been performed, and results show that under a given set of conditions (e.g. network delay and packet losses), *marginal* speech quality improvement is achieved at the cost of little added complexity and slight increase in bit rate. The chapter concludes with a discussion about the simulation results obtained.

Chapter 6: Discussion and Conclusion

This chapter concludes the thesis by summarizing key areas and results obtained in the context of verifying that the thesis objectives were meet. It highlights the benefits of this thesis, and identifies future work directions based on the experience acquired during this thesis work.

2.0 SPEECH COMPRESSION

2.1 OVERVIEW

Data compression has been an on-going field of research for many years now. In a simple way, data compression accepts a source data stream and transforms it to a smaller output data stream (e.g. a different representation smaller in size). Data compression has been very popular in two main areas: data storage and data transmission [74].

Real-time and non real-time data compression is based on the principle that compression can be achieved by removing information redundancy from the source data. To do so, the structures inherent to the source data are exploited and the source data is re-structured using more compact representations. However, data sources are not all of the same type and require different approaches for their efficient compression. For example, text documents, faxes, images, and sounds, all present characteristics and structures that are specific to each other. A generic compression method may compress one type of data very well but may perform poorly on another one. This would in fact depend on how well the method would exploit the redundancies in the structure of the source data. Another important factor to consider is how close the reconstructed signal must be to the original source data. In general, data compression methods fall in either two categories: lossless compression or lossy compression. While both methods aim at removing redundancies in the digital information, lossy compression does not reconstruct an exact replica of the source information presented. Consequently, lossy compression methods are more permissive than lossless compression methods and provide better information reduction then the lossless methods while attempting to ensure that differences between the original and reproduced signal are not perceptible by the eventual user. Data compression is such a wide field that it is not possible to cover it all, nor do we want to cover more than what is really in the context of this thesis (i.e. speech compression).

This being said, the interested reader will find some excellent reviews on the topics of lossless and lossy compression in general [74][72], for image compression [40][67], using wavelet and subband coding [84], and other complementing literature [59][5].

In this chapter, we first introduce some of the key characteristics of the speech production and hearing systems as well as the terminology associated with those systems. Then we identify and describe the format of the input and output signals of an encoder/decoder (i.e. codec), which establishes the signal basis to be compressed. A review of compression methods is presented in some detail to establish a common understanding of speech compression standards. We then investigate an advanced set of compression techniques that explains the research path that enabled low bit rate speech codecs to eventually offer near toll quality performance. To facilitate discussions, we will categorize the quality of the reproduced speech as synthetic quality, communication quality, toll quality, or commentary quality [48] , associated to poor, fair, good, and very good speech quality respectively.

We then tabulate the attributes of some of the most common codecs. Using the tabulated view, we will rationalize our choice of a specific codec to be further studied in this thesis. Given that the ultimate goal is to improve speech quality, we will also discuss subjective and objective speech quality assessment tools. We conclude this chapter with a detailed description of the ITU-T G.729A codec since we chose to follow through on it.

2.2 CHARACTERISTICS OF THE HUMAN SYSTEMS ASSOCIATED TO VOICE COMMUNICATION

This section will highlight some characteristics of the human speech production and hearing systems, which are important in a VoIP system. Speech and hearing work in tandem, that is, when

you speak, it is hoped it will be heard by one or more person, in this way communicating information.

Speech system

The human speech system is composed of lungs, vocal cords, nasal and mouth cavities, the tongue and lips, that are dynamically altered during the production of sounds (i.e. speech). Everything past the vocal cords is generally referred to as the vocal tract [74]. Speech is produced when air is forced from the lungs through the vocal cords and along the vocal tract. The use of the vocal tract introduces resonance frequency called formants, which carry speech information, generally voiced information. When air is forced through the open vocal cords it produces a noise-like turbulence resulting in unvoiced sounds. Voiced and unvoiced sounds will exhibit short-term correlations due to the vocal tract slow opening and closing, but long-term periodicity will mostly be present in voiced sounds (e.g. the pitch as a consequence of vibrations of the vocal cords).

Classification of produced speech

Speech can generally be classified as *voiced* or *unvoiced*. The spectrum of a voiced speech sample presents periodic peaks while the spectrum for unvoiced sounds is typically flat. Therefore, voiced speech is characterized by high-energy low frequencies while unvoiced speech contains higher frequencies and low-energy. Segments of speech that carry meaningful sounds are called *phonemes* and last on average 80 ms [13][66]. A phoneme is composed of voiced or unvoiced speech segments or a combination of the two (e.g. transition form voiced to unvoiced, and vice-versa), represented by one or several sequences of vocal tract shapes.

There are also sounds called *plosives* that are generated by creating a pressure build up in the vocal tract and releasing it suddenly. However, for the purpose of this thesis we will focus on voiced and unvoiced sounds since they provide sufficient generalization.

Hearing system

Of importance to this thesis is the modeling of auditory perception. Without going into the mechanics of the ear, the ear acts as a microphone that sends signals to the brain. Contrary to the microphone, the human ear is very sensitive and can distinguish 1500 timbres in sound intensities ranging from 0 db to 140 db [74][9]. The hearing process bears a tight dependence on the brain that analyzes and makes sense out of the sounds reaching the ear. Sounds are characterized by pitch, loudness, and timbre, associated with the perception of frequency, intensity, and spectrum respectively. Of these perceptions, the frequency perception has been adopted as a way to model the human hearing system. The human hearing model can be based on the *Absolute Hearing Threshold*, which establishes the performance of the ear to frequency tones in quiet environment [13][85]. The absolute hearing threshold was measured following an extensive testing effort and provides the basis for the perceptual principle used in modern compression methods. It is approximated by the following formula:

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 \ (dB\ SPL) \qquad \text{(Eq. 2.1)}$$

Where applied to signal compression, $T_q(f)$ is interpreted as the maximum allowable energy level for a given frequency f. Another important characteristic of the ear is the *masking* effect. This is a phenomenon where the ears mask tones that are adjacent. However, we do not want to expand on this topic and the interested reader can find a good review in [59].

2.3 DIGITIZATION OF THE SPEECH SIGNAL

Human speech is analog in nature. To transmit speech, we first need to transform it into an equivalent electrical signal. This is done with the use of a transducer, in this case a microphone, which translates the physical sound waves displacements to their equivalent electrical signal displacements. Then, follows an analog-to-digital conversion stage (A/D converter where the

electrical signal is digitized in time and amplitude (i.e. sampling and quantizing respectively). In the case of speech systems such as in telephony systems, the analog signal is filtered before conversion in such a way that most of the energy lies within an allowed bandwidth, that is, from 300Hz to 3400Hz [20][1][46]. Hence, a sampling frequency of 8KHz will fulfill Nyquist requirements [64] and will provide a new sample every 125 microseconds ($T = 1/F$). The linearly quantized samples will have a discrete value that may or may not be a rounded version of the analog value. The difference between the analog and its discrete representation is termed quantization noise and this quantization noise is inversely proportional to the number of quantization levels used (i.e. number of bits used) resulting in a trade-off between the signal quality and the bit rate requirements. For example, a 16 bit/sample representation will most resemble the analog signal but will also require a high transmission rate of 128 kbps while an 8 bit/sample representation will provide a degraded version of the signal but will only require 64 kbps. The conversion of the analog signal discussed thus far produces a digitized version termed Linear PCM.

Advanced applications may require additional information about the signal, thus they will require sampling the signal at higher rates, conveying more information and consequently allowing, in principle, better quality signals to be reconstructed. The following table provides such examples.

Type	Transmitted bandwidth	Sampling frequency	Number of bits in D/A and A/D converters	Bit rate in kbit/s	Main application
Telephone Speech	300-3400 Hz	8kHz	8 to 16	64 to 128	PSTN, ISDN networks, digital cellular
Wide band Speech (and audio)	50-7000 Hz	16 kHz	14 to 16	224 to 256	Video and audio conferencing, FM radio
High Quality Speech and Audio	30-15000Hz 20-20000Hz 10-22000 Hz	32 kHz 44.1 kHz 48 kHz	16 to 24	512 706 1152	Digital sound for analogue TV (NICAM) Audio CD player Professional audio

Table 2.1 – Common families of audio signals (reproduced in part from [20])

As alluded to earlier, this thesis will only consider signals digitized for standard telephone systems (whose characteristics are summarized in the first row of Table 2.1).

2.4 SPEECH COMPRESSION TECHNIQUES AND STANDARDS

Digital speech brings flexibility but requires higher data rates. Communication bandwidth conservation identifies a need, or requirement, for speech compression. Most effective speech compression techniques are lossy since speech encoding is mainly concerned with representing the speech with as few bits as possible while maintaining its perceptual quality.

There are two classes of speech codecs, namely, *waveform-based* and *source model-based* codecs. Waveform-based codecs are essentially waveform follower that will exploit the correlation in the signal representation (e.g. the shape of the signal). They offer the best speech quality performance at the expense of high bit rates. On the other hand, source model-based codecs achieve low speech quality and low bit rate. Source codecs use a model of the human speech system to replicate the speech signal. A third class of codecs, known as a hybrid class, has also been widely accepted. This class attempts to exploit best options/components from both waveform and source codecs methods [13][79][66] and merges those together to form hybrid codecs. Hybrid-based codecs provide better speech quality than source codecs with lower bit rates than waveform codecs.

In general, compression methods are applied on speech samples, on a sample-by-sample basis or on a block of samples basis. For streaming applications, such as in the case of VoIP, the samples are normally grouped into blocks of samples representing source signal segments or units[3] of data

[3] A unit is defined as a time interval of audio data as manipulated by the encoding/decoding algorithm [64]. It is the smallest audio entity where one or more of these units can be encapsulated within the transmission packet.

[61][14]. In this thesis, these units of data will be referred to as *speech frames*. While processing speech frames enables better information reduction with more advanced encoding methods, they also induce a processing delay and, an associated increase in codec complexity.

The quantization technique discussed so far is named *scalar quantization* and quantizes one sample at a time. Quantization is not only used in the conversion of analog signal as described earlier, but it is also used in intermediate stages of a compression system. Another important technique, *vector quantization,* works on a group of samples at a time in order to effectively reduce the bits required to represent the information vector. A good treatment of the subject is found in [74][72].

In this section, we will mostly review source and hybrid codecs. On the other hand, waveform-based codecs are very well represented in literature [74][72][41], consequently, we will only present its principle and identify two standards commonly used in VoIP.

2.4.1 WAVEFORM CODECS

Speech signals are considered slowly varying signals that show a high degree of predictability. In general, waveform codecs attempt to exploit the correlation or predictability between successive samples in order to reduce the bit rate.

There is a large variety of waveform compression techniques. The simplest ones exploit the redundancies in the time-domain while other look at the signal representation in the frequency-domain. Speech waveforms can simply be characterized by their amplitude distribution looking at inter-sample relation or their autocorrelation function in the time domain. Waveform codecs can

also exploit redundancies such as their power spectral densities in the frequency domain. Consequently, waveform codecs are divided into time-domain and frequency-domain codecs.

In this section, we will focus on two common waveform codecs used in VoIP applications. We will provide a general description for each.

PCM

The ITU-T G.711 Recommendation [25] is the simplest and most widely used codec in telephony. This recommendation provides two different encoding algorithms. The first one called μ-law is primarily used in North America while the second, A-law, is used in the European countries. The G.711 is in reality a logarithmic encoding law that provides finer resolution of the weaker signals [9], namely a compander [74][62] (μ-law or A-law), that has the advantage of maintaining an approximate 35 db signal-to-quantization noise ratio. It provides very good speech quality but at a very high bit rate of 64 kbps.

ADPCM

The ITU-T G.726 Recommendation [28] can be used at bit rates of 16/24/32/40 kbps and provides poor, fair, good, and very good speech quality respectively. This Adaptive Differential Pulse Code Modulation (ADPCM) codec is often used as a reference codec to evaluate other lower bit rate codecs. One of its major advantages is its bit rate scalability that assists the management of traffic in periods of congestion. It is reported to be the most used for terrestrial and submarines cables [20].

2.4.2 SOURCE CODECS

2.4.2.1 Overview

Source[4] codecs are often referred to as vocoders (Voice Coders) in the literature. The concept behind source codec is to model the physiological characteristics of the human speech production system. As discussed in Section 2.2.2, speech is the response of the vocal tract to one or more excitation signals. Pushing air out of the lungs through the vocal cords produces an excitation signal that is naturally filtered by the vocal tract, thus producing voiced and unvoiced sounds. An important part of source codecs is the modeling of the vocal tract as a short-term filter. Therefore, the excitation generator and the short-term filter are the major components implemented in Linear Predictive Coding (LPC) codecs.

Source codecs are designed for very low bit rate channels, 2.4 kbps and lower, and are known to produce intelligible speech being considered of synthetic quality. This type of codec comes in different flavors such as frequency-domain vocoders (cepstral analysis), time-domain vocoders (filter banks), and Linear Prediction (LP) vocoders [48][79]. Research on these techniques started many years ago when physical machines were built to synthesize the human voice [13]. Fully mechanical voice machines were then replaced by electrical analog circuitry and nowadays by digital circuits. Already at the beginning of the digital age, the LPC codec had a great advantage. Its very low bit rate allowed secure encryption of its encoded digital representation for transmission over the analog telephony system using modems. In those early days, it was impossible to transmit PCM (64 kbps) encoded speech through those same analog circuits. The most important of those [75][55][79], the LPC codec, has been an enabler for more advance methods and it will be explained in more details in the following paragraphs.

---

[4] Source codecs as the name suggests, are designed for specific input data, in this case speech signals, hence, the nickname vocoder.

2.4.2.2 Linear Predictive Coding Model

The simplest LPC codec is a two-state machine that produces voiced and unvoiced sounds. The codec uses a linear, quasi-stationary[5] model of speech where independent voiced and unvoiced speech segments are generated in an interlaced fashion, thus producing an estimate of the uncompressed speech signal. Linear Prediction is performed on a group of samples at a time. The high-level analysis-synthesis components required for linear predictive encoding and decoding of speech signals are shown in Figure 2.1 and Figure 2.2 respectively. A convenient approach to understand the operation of the LPC codec is to start with the description of the decoder, or synthesis side, of this tandem.

The decoder receives four parameters from the encoder, namely the *pitch period*, *Voiced/Unvoiced (V/U) indicator*, *gain*, and the *filter coefficients*. For voiced sounds, the decoder uses a periodic pulse generator to supply the excitation signal, while for unvoiced sounds a random noise generator is used. During the production, or synthesis, of speech, voiced and unvoiced signal generation are mutually exclusive, representative of a two-state excitation model.

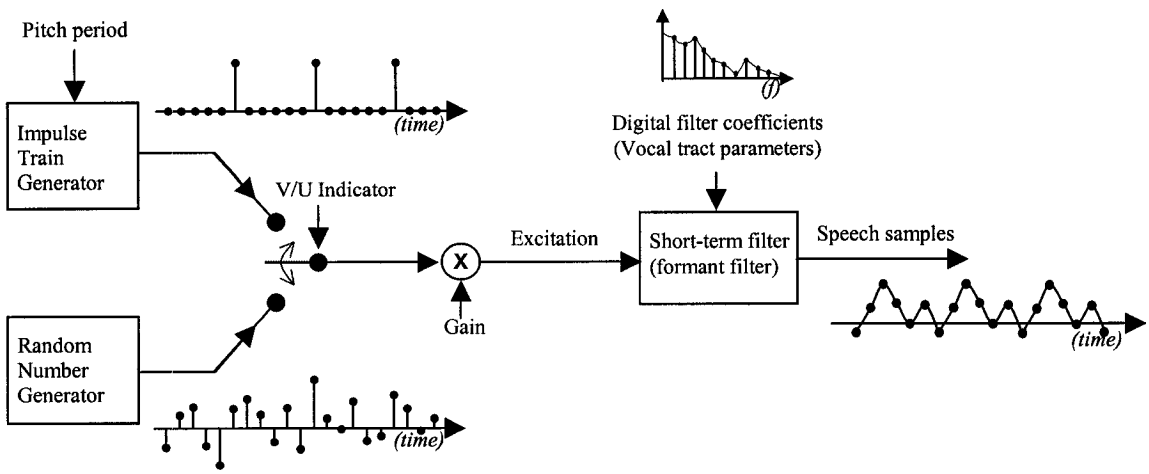


Figure 2.1 – The classical two-state LPC decoder

[5] Speech is considered stationary over short segments, typically a segment between 5 to 20 ms in duration [5][50][63].

The output of the periodic pulse generator (or the noise generator) is then amplified or attenuated to form the excitation signal. LPC relies on an excitation being fed through a short-term filter. Figure 2.1 shows a sample sketch of a short-term spectral envelope estimate representative of a voiced segment. The filter is an all pole filter (typically of order between 8-14 poles) that is used to reconstruct the spectral envelope estimate of the original signal short-term power spectrum. This sequence of events is repeated for each consecutive encoded speech frame received. A major part of many speech codecs is the modelling of the vocal tract as a short-term filter. The vocal tract produces formant frequencies, or resonant frequencies, modulating the excitation signal for each segment of speech being inputted to the encoder. As the shape of the vocal tract varies relatively slowly; the transfer function of its modelling filter needs to be updated only relatively infrequently (typically every 5 to 20ms as suggested before). The short-term filter is usually very well characterized by the first three or four formants modulating the envelope. Therefore, from a time-domain perspective, the resulting speech signal is the result of the excitation signal being convolved with the short-term filter [75][55].

Source codecs are often referred to as parametric codecs as they produce a parametric model for the reconstruction of the speech signal [79]. In Figure 2.2, the encoder side of the LPC codec has to determine the parameters to be sent to the decoder. The encoder analyzes the source signal and extracts associated parameters for transmission to the decoder. The encoder repeats the analysis process for each incoming speech frame. The pseudo-code describing that process is:

a. Determine the voiced or unvoiced state;

b. Determine the pitch period;

c. Determine the gain; and

d. Determine the formant filter coefficients.

The figure below provides a high-level overview of this process. The interested reader should consult [79] for LPC-10 or FS 1015 encoder and decoder block diagrams.

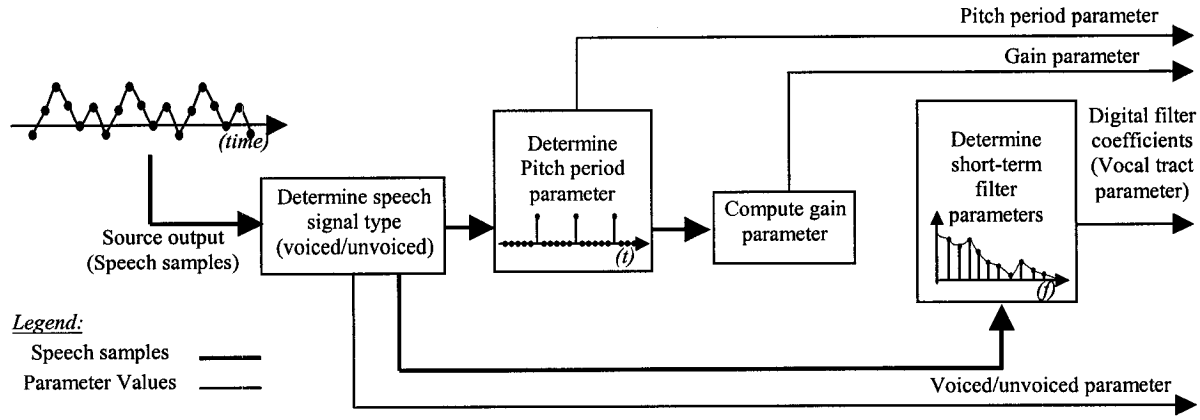


Figure 2.2 – The classical two-state LPC encoder

For explanation purposes, the blocks will be nicknamed the V/U block[6], the pitch block, the gain block, and the synthesis filter block. They appear from left to right, respectively, in the above figure. *The V/U indicator*, the *pitch period*, and the *gain* form the excitation signal parameters while the digital filter coefficients are the coefficients that model the vocal tract of the human speech system model. As mentioned earlier, there are two possible states. For voiced signals, all blocks are used and each block determines its parameter to be transmitted to the receiver. For unvoiced signals, only the V/U block, gain block, and the synthesis filter block are used since the pitch block would not produce any useful information (i.e. would produce a random value). In the next section, we will provide more details on how each block determines its parameters. The last step is the quantization of the parameters before they are transmitted to the receiver. When the receiver gets the parameters, the decoder synthesizes an estimate of the speech signal through the application of the parameters to the two-state human speech production model previously described.

The speech signal reconstructed by a two-state model is characterized as having good intelligibility but often sounds buzzy and of poor quality for non-speech sources [48][79].

2.4.2.3 Mixed-Excitation Linear Prediction Model

As discussed in Section 2.2, spoken words or phrases contain speech transitions that in general may include a voiced segment softly melding into an unvoiced segment and vice-versa. Thus, the two-state LPC codec presented does not perform very well during those transitions. This two-state model is shown in Figure 2.1 where the switch represents the mutually exclusive voiced and unvoiced states. The Mixed-Excitation Linear Prediction (MELP) model attempts to provide better speech quality by generating a mixed excitation signal that is a combination of the outputs of both periodic generator and noise generator.

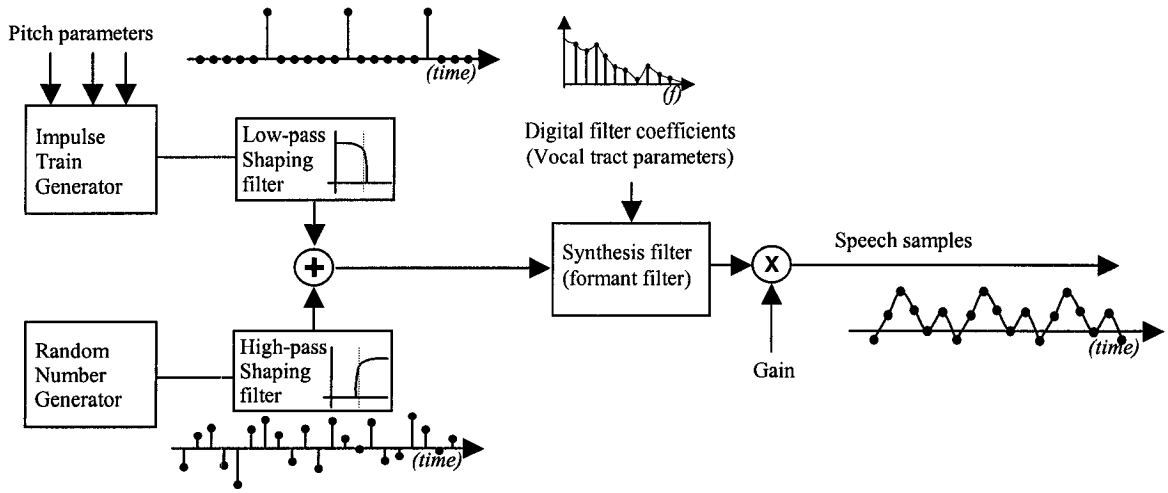


Figure 2.3 – The Mixed-Excitation Linear Prediction (MELP) decoder

Figure 2.3 shows this principle where the switch of Figure 2.1 is replaced by an adder. This model presented by Makhoul *et al.*, reported in [79], describes a low-band and high-band pass filters following the impulse and noise generators respectively. The objective is to flatten the overall excitation spectrum. Their model provided better speech quality by reducing the buzzy sounds associated with synthetic speech. Other enhancements such as a pulse position jitter for the synthesis of weakly periodic or aperiodic voiced speech were introduced for better control of the impulses

[6] To note here, is that the voiced/unvoiced detection is often a by product of the pitch detection algorithm [13], hence often part of the pitch block.

generation. The new 2.4 kbps codec Federal Standard (FS-MELP) for the DOD [72] is based on that principle.

2.4.3 HYBRID CODECS

Waveform codecs are generally simple to implement, they provide very good speech quality but at relatively high bit rates (>16kbps). On the other hand, source codecs are relatively complex. They provide very low bit rates (< 2.4kbps) but are associated with synthetic speech quality (intelligible but lacking speech naturalness). Hybrid codecs operate at bit rates between 2.4 kbps to 16 kbps. They provide toll quality speech but will generally require an increase in complexity that varies depending on the implementation. Hybrid codecs achieve their bit rates and speech quality objectives by borrowing the best techniques from waveform codecs and source codecs. Almost all hybrid codecs rely on the human speech system model, as presented in the context of the LPC codec. Contrary to the LPC codec where only parameters were transmitted to the receiver, hybrid codecs will also transmit some waveform representation, compressed or not, to the receiver.

In this section we will provide an historical perspective on the evolution of hybrid codecs, which contributed to the design of CELP codecs (and variants) as we know them today. For example, the simplest hybrid codec, the Residual-Excited Linear Prediction (RELP) codec, where low-frequency components conveying most of the information are waveform coded and the high-frequency components are coded using source modeling coding techniques. However, as we will see, the gain in speech quality is obtained at the price of added complexity and higher bit rate. Techniques to decrease the rate, such as adding a long-term filter to extract the periodic information and a weighting filter to effectively distribute quantization noise, are used to further decrease the bit rate and increase the speech quality. However, both methods will result in an increase in complexity.

The most successful and most commonly used hybrid codecs rely on an Analysis-by-Synthesis (ABS) technique. An important difference is that it uses a closed-loop approach to provide a better estimate of the signal. More details on the ABS technique will follow shortly.

2.4.3.1 Residual Excited Linear Prediction Model

The two-state models presented thus far still exhibit synthetic speech quality. By closely studying those models, we clearly see that the independent approach to generate the excitation (i.e. periodic impulse and noise generators) forms a loose-coupling between the encoder and decoder. That is, any residual information (when the encoder determines the parameters) is lost, hence, not conveyed to the decoder. Intuitively, it results in a reconstructed speech lacking important traits. The idea probably came from the ADPCM principle where RELP replaces the two-state excitation by building a residual component that will also be transmitted to the receiver along with the vocal tract parameters. The principle of Linear Prediction theory is shown in Figure 2.4. Exciting the LP synthesizer with the error signal *e(n)* reproduces the input signal *s(n)*. This is the basic principle behind RELP codecs.

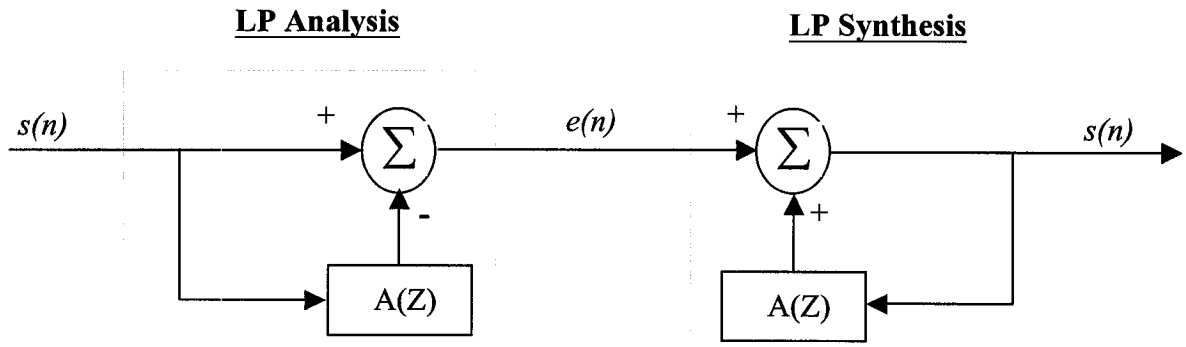


Figure 2.4 – Linear prediction Analysis-Synthesis model

Therefore, in RELP, the residual signal is low pass filtered and classically encoded in PCM. This provides an excitation signal that is more similar to the one at the encoder used to generate the

parameters. As we have discussed at the beginning of this chapter, simply quantizing a signal to a given accuracy requires substantial bandwidth to transmit the signal to the receiver. The two-state LPC codec transmits parameters only and avoids sending any signal representation to minimize the transmission rate. RELP codecs, on the other hand, will transmit a compressed version of the error signal that will be conditioned at the decoder to become a close estimate of the real error signal. Figure 2.5 shows a residual/excitation signal being transferred to the receiver. This first hybrid codec or compression method adds perceptual components that were not present before, thus improving speech quality but with an increased bit rate requirement.

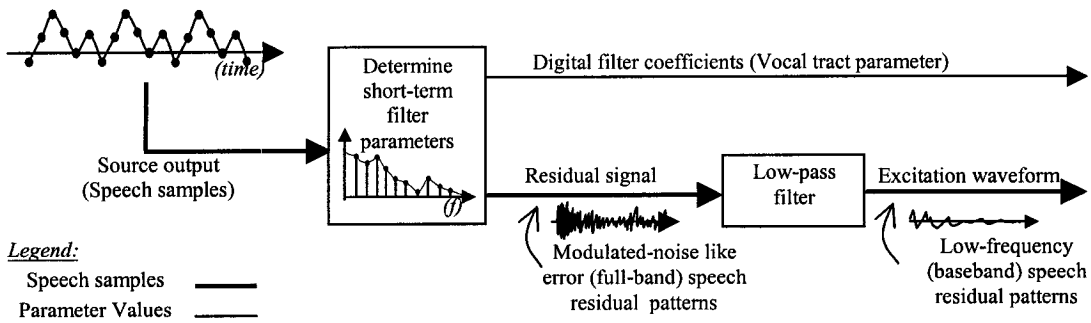


Figure 2.5 – Residual Excited Linear Prediction (RELP) encoder principle

Important to note at this point, is that the low-pass filter will receive as input an error signal which may or may not contain periodic information depending on whether it is a voiced or unvoiced input signal. For voiced segments, the residual signal will contain signal redundancy, namely pitch information that could be removed, thereof, further reducing the rate (long-term periodicity will be covered in a later section). For unvoiced segments, the residual signal will mostly resemble a white-noise signal but will be tightly coupled to the vocal tract filter. The basic idea will be to remove further signal redundancy and provide a spectrally flat error signal or residual signal to be used as an excitation signal by the synthesizer. Speech quality from the RELP codec at rates of 4.8 kbps and above is superior to the one from the simple two-state LPC models presented. More details with respect to the method to compress the residual signal can be found in [74][79]. Voice-Excited Linear

Prediction (VELP) codecs use a similar approach as that presented for RELP but uses a compressed version of the input speech as excitation [74].

Many hybrid codecs use linear predictive modeling of the speech production system to reconstruct the speech samples using a synthesis filter excited by a residual signal. Those codecs use short-term and sometimes long-term linear prediction to derive a difference signal often referred to as the residual signal. Several techniques have been experimented with, such as baseband coding of the residual using a feed-forward approach. Latest development uses an ABS scheme, which has been shown to provide better results.

As explained previously, coding efficiency is achieved by extracting as much signal information as possible from the samples (removing the redundancy). Since correlation is an indication that there is some remaining information in the samples, we wish to end up with a residual signal that looks like white noise (e.g. no more correlation, where samples are independent from one another). Once this objective is achieved, the residual can be most efficiently quantized (i.e. PCM) for its transmission to the decoder.

Associated with the quantized residual, a set of parameters representative of filter coefficients, pitch lag and amplitude are also efficiently coded and transmitted to the decoder.
So far, we have described what a hybrid codec is by studying the RELP codec as a specific example. We now need to present some of the most popular concepts that are used to decrease the bit rate requirement while maintaining or increasing the speech quality. Three concepts are presented in the following sections, namely, redundancy reduction through the long-term periodicity extraction, optimized quantization through distribution of quantization noise using perceptual weighting, and better signal reconstruction using an ABS approach.

Long-term periodicities

So far we have discussed the short-term filter where adjacent speech samples correlation is removed. Additional coding gain is achieved by the use of a Long-Term (LT) predictor. The basic idea is to remove the remaining periodicity in the residual signal, thereby, further removing the redundancy. The transmitted waveform representation (e.g. a residual signal) may exhibit large quasi-periodic variations due to periodicity in a voiced speech segment. Typically, the correlation between adjacent subframes defines peaks representing the pitch period (or pitch lag between 20-140 samples [20]). The redundancy between adjacent subframes is removed on a subframe basis using a pitch predictor that is represented mathematically as:

$$B(z) = 1 - \sum_{i} \beta \, z^{-P-i} \qquad \text{(Eq. 2.2)}$$

where P represents the pitch period, or lag, and β is an associated scalar gain. We can see that the periodic contribution at P is subtracted from the current speech samples using a multi-taps LT filter. A more advanced procedure will find a fractional lag by upsampling the signal thus providing a finer representation of the structure of the speech spectrum. The significant improvements in speech quality [20] clearly justifies the increase in complexity. The use of a LT filter contributes to further flattening the spectral shape of the residual signal. This in turn will allow quantization gain to be achieved or equivalently, provide a better speech quality performance.

Perceptual weighting

Music compression generally requires a higher sampling rate (32 kbps-48 kbps) resulting in a wider band than speech compression. Perceptual codecs are widely used for the coding of digital audio [59]. The principle is based on a psycho acoustic model of the human auditory perception to discriminate between relevant and irrelevant signal components. It provides a mean to remove sounds that would otherwise be inaudible to the human ear. As discussed in Section 2.2.2, the

hearing characteristics or the ear frequency sensitivity or performance is described using the Absolute Hearing Threshold. This approach has been adapted to speech compression where a perceptual model is used to relocate noise or distortion in areas inaudible by the ear. We exploit the ear's response by performing a distribution of large quantization noise into regions of high speech energy and the redistribution of smaller quantization noise in regions where it is most susceptible to being heard. This distribution of the error is usually carried out on the residual signal using a weighted perceptual filter.

Analysis-by-Synthesis

Analysis-by-Synthesis (ABS) was initiated in 1982 by Atal and Remde [79]. The source codecs covered so far are characterized as open-loop since the encoder simply analyzes the input speech signal and determines the coding parameters in a straightforward manner. On the other hand, ABS operates in a closed-loop fashion. Figure 2.7 shows the components of a typical ABS system. It consists of a perceptually weighted filter, an error minimization measurement, an excitation generator, a synthesis filter, and a simple adder. The excitation generator and synthesis filter operate as before. Typically, the decoder has not been modified and its operation is the same. The encoder on the other hand has additional blocks composing its system.

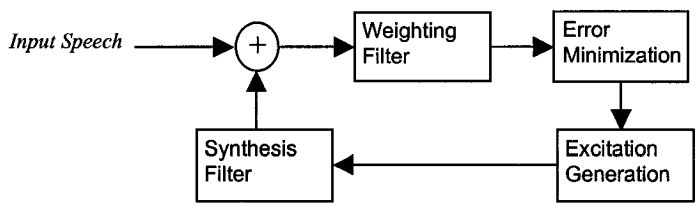


Figure 2.6 – The Analysis-by-Synthesis (ABS) principle

Its operation is now focused on a closed-loop error minimization approach that will perform several iterations until it has minimized the error between the source input speech and the synthesized one. ABS can be categorized as a waveform follower in the sense that it is attempting to replicate the

input signal as accurately as possible. Once again, the input speech samples are grouped together to form speech frames. Each input frame is analyzed in order to extract the synthesis filter coefficients and associated excitation parameters. The distinguishing feature is in the synthesis of the parameters by the local synthesis building block. The output of the synthesis filter is compared against the input speech signal using some error measure[7]. Usually, the Mean Squared Error (MSE) is used as the estimator. Its value is compared from one iteration to the next until the difference or residual error between the synthesized speech and the input speech falls below a minimal error or threshold. This implies that a large number of iterations (all possible combinations of the excitation filter) are required, therefore drastically increasing the numerical complexity but with the benefit of providing very good speech quality at low bit rates.

Excitation Signal Representations

When describing RELP earlier, the PCM representation of the residual signal required a significant number of bits. Several approaches for encoding the residual or excitation signal have been designed to reduce the overall complexity. The most common are the Multi-Pulse Excitation (MPE), the Regular Pulse Excitation (RPE), and the Code-Excited Linear Prediction (CELP) techniques. The following figure highlights the differences in the excitation signal representations.

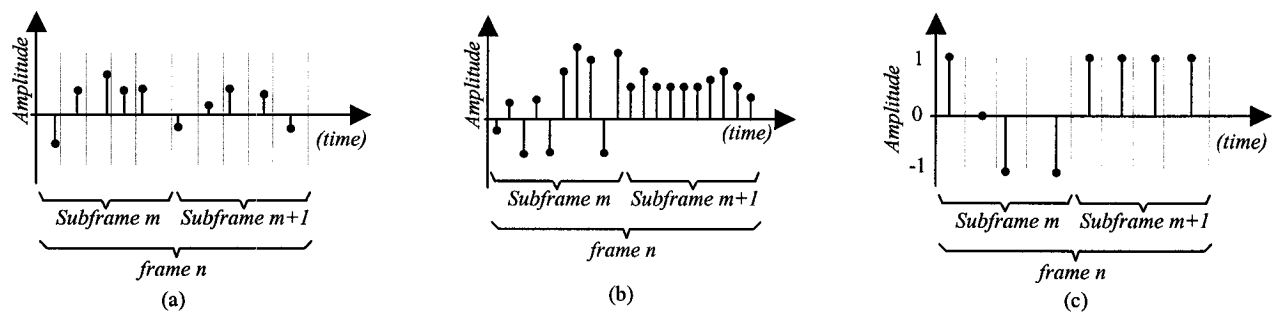


Figure 2.7 – Excitation signals models for (a) MPE, (b) RPE, and (c) CELP codecs

[7] The measure is usually performed on a short-term basis (e.g. on a subframe basis) comparing the input source signal with the encoder locally synthesized speech signal.

- *Multi-Pulse Excitation (MPE)*: The algorithm generates an excitation sequence that is composed of several pulses each characterized by a temporal position and amplitude (Figure 2.8 (a)). Typically, a subframe is associated with 4-6 such pulses that are spaced in a non-uniform manner. The subframes are usually divided into time slots each holding a single pulse. In MPE, pulses position and amplitude are determined sequentially one after the other and pulses location are differentially encoded relative to the location of the first pulse. This method produces good speech quality at rates around 10 kbps. However, a reported [79] problem is that its performance deteriorates in the presence of high pitch signal.

- *Regular Pulse Excitation (RPE)*: Contrary to MPE, RPE pulses are positioned at regular intervals (Figure 2.8 (b)). RPE only specifies the location of the first pulse and the uniform spacing for subsequent pulses within that subframe (typically 10-13 pulses per 5 ms segment [79]). The location of the first pulse is usually coded using the absolute position. Coding gains are attributed to the coding of the remainder of the locations where these are coded relative to the first pulse location (i.e. spacing index). The amplitudes are computed using a set of linear equations. Compared with the MPE method, this method effectively reduces the number of bits required to represent the excitation signal, or equivalently, allows using more pulses (for a comparable MPE rate) producing a light increase in speech quality. However, the increased number of pulses is associated with an increase in processing complexity.

- *Code Excitated Linear Prediction (CELP)*: MPE and RPE transmit information relative to each of their pulses. They both provide good speech quality at rates around 10 kbps. CELP and variant of the CELP method operates in the range of 4.8 kbps to 10 kbps and offer good up to toll quality speech. In CELP, the position and amplitude of the pulses is determined using

codebooks indexes (Figure 2.8 (c)). Therefore, depending on the codebook structure used, only few codebook indexes need be transmitted to the decoder for the generation of the excitation signal that produces the best match to the original input signal. This substantial saving is attributable to vector quantization techniques and optimization of the codebooks as well as the speeding up of searches.

2.4.3.2 Code Excited Linear Prediction

One of today's most popular hybrid codecs is the CELP codec. These codecs provide good speech quality at low bit rates (e.g. 4.8 kbps– 8 kbps).

The design of CELP codecs is based on the human speech production system model (as presented for the source codec in section 2.4.2). Remember, signal redundancies are first removed from the source signal by passing it through the short-term and long-term filters, to model the vocal tract and periodicity, respectively. This leaves a residual signal, which in this case represents the excitation to model. The objective in CELP codec is to efficiently model the excitation signal. To do so, CELP relies on vector quantization principles where codebooks containing vectors are joined to generate the excitation signal. Typically, CELP selects the excitation signal from using an ABS error minimization loop that allows finding the best residual vector estimate. This best estimate is found by iteratively searching the vector space for the vectors that provides the minimum error between the synthesized signal and the original signal.

The decoder does not have the ABS minimization loop, hence, is less complex than the encoder but can include a post filter that will improve the quality of the speech. The post filter increases the response at the pitch frequency and reduces the noise between the spectrum peaks (e.g. at the

harmonic frequencies and between formant frequencies respectively), where the coefficients for this filter are derived from the LP coefficients that were perceptually determined at the encoder.

CELP codecs complexity has been a major drawback for real-time implementation for many years [20][11]. This was mainly due to the added complexity to search the codebooks indexes and associated gains for the best vectors. In the following section, we look at some of the most common methods that are used to allow practical implementations of CELP.

Variants of CELP

Since the introduction of the first CELP codec by Schroeder and Atal [11], many variations to the initial design were developed to reduce the excessive computational requirements or to reduce the large codec delays. For example, one of the early attempt at *cutting processing requirements*: rather than performing the weighting of each sequence individually, a global implementation of the perceptual weighting function is implemented at the codebook level [11]. This approach effectively reduces the number of iterations the ABS must perform to generate the best estimate of the signal, hence, reducing the complexity. Other modifications to the basic CELP were implemented over the years, such as (however, the list is not exhaustive), self-excitation, special codewords, two-stage VQ, backward filtering, algebraic codes, linear codebooks, frequency domain approaches [91], trellis excitation [45], combination of basis vectors [78], and other [12][7][71] researched approaches.

As we will see in section 2.7, the CS-ACELP ITU-T G.729A codec has implemented some of those performance enhancements or designs.

2.5 SPEECH CODECS SELECTION

In this section we will identify main desirable codec characteristics to assist in selecting the best codec based on the application's requirements and implementation constraints. The emphasis will be on characteristics associated with multimedia applications, more specifically, with the transmission of voice over packet networks. Currently, many standard and non-standard compression algorithms are used for compressing voice data for telephony systems and multimedia systems. This variety of algorithms allows for selecting the best codec on the basis of trade-offs between low delay, low rate, low complexity, and good quality characteristics [5].

Standard codecs characteristics will be summarized and compared against each another. Then, the two contenders for the general public use on the Internet [20][1][79], the G.723.1 and G.729 standards will be discussed from the point of view of trade-offs between the bit rate, the codec or algorithmic delay, the processing complexity, and the speech quality performance.

2.5.1 SPEECH CODECS SELECTION ATTRIBUTES

Several codecs have been designed and used over the years. The experience thus acquired allows one to select a codec on the basis of well-established key attributes. These attributes are presented in Table 2.2 below.

Name of Characteristic	Description of Characteristic
Bit rate	The output rate of the coder, or synonymously, the required operating rate. This effectively translates to the minimum bandwidth required for the proper operation of the codec in real-time.
Algorithmic delay	The delay incurred by the encoder and decoder from the time the source signal enters the encoder to the time the reconstructed signal exists the decoder but excluding the transmission delay.
Processing complexity	The description of required resources required to implement the codec.
Speech Quality	The perceived quality of the reconstructed speech. Several quality measures exist but the most common is the Mean Opinion Score (MOS) that is determined by subjective testing.

Table 2.2 – Speech codec key characteristics

2.5.1.1 Bit rate

The most common speech codecs are Constant Bit rate (CBR) codecs but Variable Bit rate (VBR) codecs, although not as popular, are also used. Therefore, we will assume CBR codecs in this thesis unless otherwise specified. Research in the field of speech compression has evolved along the years and new codecs with lower bit rates are regularly introduced. In general, the bit rates are grouped under four commonly accepted families [6][79]:

Rate Family	Range
High-rate	Bit rate > 16 kbps
Medium-rate	8kbps < Bit rate ≤ 16 kbps
Low-rate	2.4 kbps < Bit rate ≤ 8 kbps
Very low-rate	Bit rate ≤ 2.4 kbps

Table 2.3 – Bit rate categories

Fundamentally, the bit rate is a direct result of how well a compression algorithm reduces the number of bits required to represent the signal. On the other hand, the compression ratio is a good indicator of how well the compression algorithm reduces the information. It is calculated as follows:

$$Compression\ ratio = \frac{size\ of\ output\ stream}{size\ of\ input\ stream}$$

The compression ratio is a measure most appropriate when discussing storage requirements, while the bit rate is most appropriate when specifying a transmission bandwidth requirement.

The bit rate is an important characteristic when selecting a codec for VoIP. As will be discussed in Chapter 3, network access points are necessary to connect to the packet-network infrastructure. Low, medium, or high rate network access points will specify a maximum transmission rate that will present a constraint for the selection of a codec.

2.5.1.2 Algorithmic Delay

The algorithmic delay is an important factor when selecting a codec for real-time operation. The algorithmic delay relates to the time it takes to receive or gather the source speech samples for

encoding, on one hand, and the time it takes to gather the encoded representation at the decoder, that is the time difference between the reception of the first bit and last bit of the encoded frame. Those delays added together are often very important when end-to-end delay is an issue. At the encoder, the algorithmic delay is the sum of the source frame size and the look-ahead delay.

- ***Frame size:*** This parameter represents the length of the voice traffic measured in time. Compression algorithms such as PCM operate on a sample-by-sample basis. Therefore for a signal sampled at 8000 samples per seconds, the delay associated with capturing a sample would be 125 μs. If we were to group samples to form a frame such as is commonly done (e.g. to implement predictors that rely on inter-sample correlation), say 80 samples per frame as in the case of the ITU-T G.729 algorithm, then the frame delay would then be 80 X 125 μs = 10 ms.
- ***Look-ahead delay:*** The look-ahead delay occurs when the codec requires some knowledge of the future frame. The idea of look-ahead is to take advantage of the correlation between successive speech frames in order to improve the speech quality.

Contrary to the encoder, the decoder does not compute parameters but rather interprets those. Furthermore, the decoder may initiate the decoding process in a piece-wise manner as sub-grouping of bits is received. It is therefore very difficult to calculate this delay. It is often considered negligible and excluded from the algorithmic delay value presented.

2.5.1.3 Processing Complexity

The processing complexity includes key constraints affecting resources required to operate the codec. The key constraints are computing speed, memory required, and power consumption.

- ***Computing Speed:*** This is the CPU performance in terms of Millions of Instructions Per Seconds (MIPS). This measure will vary according to the CPU and architecture used.

- ***Memory Size:*** This is the amount of physical memory required for the implementation of the codec. Constant and variable values and structures used in computations or stored in physical memory (e.g. ROM and/or RAM).
- ***Power Consumption:*** This is the power required to operate the circuits on which the codec is implemented. Usually, faster CPU will consume more power, which is not necessarily good for mobile devices.

Another consideration, if implementation on a desktop PC is envisioned, is the performance of floating-point architecture versus fixed-point architecture. The former is preferable for desktop PCs.

2.5.1.4 Speech Quality

Of all attributes presented thus far, speech quality is difficult to quantify objectively. However, formal methods to determine the subjective and objective speech quality performance of codecs will be introduced in Section 2.6. As we will see, many factors may impair the quality of the reproduced speech, namely, the presence of background noise or music in the source signal, the spoken language, and most importantly, the presence of lost frames, or more precisely the lack of frames (i.e. missing packet). Nevertheless, we need to briefly introduce the Mean Opinion Score (MOS) as it will be used in the following section. Basically, the MOS is widely used to quantify speech codec quality. A numerical value between 1 and 5, representing a *bad* up to an *excellent* quality assessment respectively, is used to assess the speech quality performance for a particular codec.

2.5.2 COMMON SPEECH COMPRESSION STANDARDS

In this section we wish to provide an overview at the most common codecs by providing the attributes of each. Table 2.4 presents those codecs, of which we will select a codec that we will use

in this thesis for demonstration and investigation purposes. The selection will be on the basis of trade-offs with respect to the attributes presented in the previous section.

Codec Name	Compression Method	Bit rate (kbps)	Speech Quality (MOS)	Frame Size (ms)	Look-ahead delay (ms)	Processing Speed (MIPS)	Memory Size (16 bit Words)
ITU-T G.711	PCM	64	4.2	0.125	0	> 0.5	> 100
ITU-T G.721	ADPCM	32	4.1	0.125	0	2	?
ITU-T G.723.1	ABS: ACELP ABS: MP-MLQ	5.3 6.3	3.7 3.9	30 30	7.5 7.5	14.6 16	2200
ITU-T G.726	ADPCM	40/32/24/16	4.2/4/3.2/2	0.125	0	2	?
ITU-T G.728	ABS: LD-CELP	16	3.85	0.625	0	30	?
ITU-T G.729	ABS: CS-ACELP	8	3.92	10	5	20	3000
ITU-T G.729A	ABS: CS-ACELP	8	3.7	10	5	10.5	2000
ITU-T G.729D	ABS: CS-ACELP	6.4	< 3.7 [2]	10	5	?	?
ITU-T G.729E	ABS: CS-ACELP	11	> 3.92 [2]	?	?	?	?
DOD FS 1015	Vocoder:LPC 10e	2.4	2.3	22.5	90	20	?
DOD FS 1016	ABS: CELP	4.8	3.2	30	7.5	16	?
DOD FS 10??	ABS: MELP	2.4	3.2	22.5	20	40	?
IS-54	ABS: VSELP	7.95	3.45	20	5	13.5	?
IS-96	ABS: CELP	8/4/2/0.8	3.5	20	5	10	?
GSM 6.60	ABS: ACELP	12.2	4.1	20	0	15.4	?
GSM 6.10 (Full-rate)	ABS: RPE-LTP	13	3.6	20	0	5	?
GSM 6.20 (Half-rate)	ABS: VCELP	5.6	3.5	20	0	17.5	?

General Note: Consolidation from references [57][20][1][6][47][5][79]. The reported values were averaged when much difference existed between references. Otherwise, the values most often reported (recurring values) were taken. Also, unknown values at time of thesis submission are marked with a question mark '?'.
Note 1: The information could not be located.
Note 2: Assumed statement since information could not be located.
Unknown values at time of thesis submission are marked with a question mark '?'.

Table 2.4 – Characteristics of common speech codecs

For example, an obvious trade-offs would be speech quality versus low operating bit rate, as it would apply to applications using low-rate access points or networks with large latency. Another example would be with respect to a limited complexity budget where we would have to settle for a medium-rate codec to maintain a given speech quality. In general, the application requirements and/or constraints will form the selection basis for a particular codec.

Selecting a codec for this thesis

We first wish to eliminate few codecs from the list by identifying the codecs' current field of application. First, some of the codecs can be categorized under two domains of application, namely, the U.S. DOD Secure Telephony and the cellular telephone standards. The U.S. DOD FS-1015, FS-1016, and the latest FS-MELP codecs are mainly dedicated to provide low-rate encrypted communication between parties. These standards provide low-rate but also offer least speech quality. Then, ETSI GSM 06.10, GSM 06.20, and GSM 06.60 are cellular telephone standards used in Europe. Similarly, the IS-54 and IS-96 are cellular telephone standards from the TIA USA standard body. These ETSI and TIA standards offer better speech quality performance for a slight increase in the bit rate but the speech quality offered is still hardly comparable to the quality found on the current telephone systems.

The ITU, an international standard body, has offered a series of speech codecs for diverse applications. The internationally accepted ITU-T G.711 PCM [25] codec is used across the world in the current telephone systems [6]. The ITU-T G.726 ADPCM [28] is used for undersea cables and satellite links [1]. The ITU-T G.722 Wide Band [26] and ITU-T G.728 [29] are used in video conferencing over ISDN or frame relay connections [6]. While offering very good speech quality, they all require medium to high bit rates for the proper operation.

Of the remainder, the ITU-T G.723.1 [27], ITU-T G.729 [30], and ITU-T G.729A [31], and looking at their attributes in Table 2.3, we select the G.729A. The ITU-T G.729A Recommendation has been the center of interest for simple reason that it operates at low-rate, introduces a small delay, provides tool quality speech, and low complexity compared against the other. These are key factors for the use of a codec aimed at the general public Internet or where low rate network access is the only option. For those reasons, we elected for the ITU-T G.729A Recommendation and a detailed description of the recommendation follows at the end of this chapter.

2.6 SPEECH CODECS QUALITY PERFORMANCE EVALUATION

2.6.1 OVERVIEW

Speech quality implies a measure of fidelity. As described in Chapter 2, the human auditory system is very complex. The sound waves are received by the ear and translated to electrical impulses that are interpreted by the brain. The human brain is in fact the one actually judging the assessment of the quality [3][85]. Therefore, it is difficult to qualitatively measure the fidelity of speech because human perception is involved.

From a communication's engineering point of view, speech quality assessment is a key factor in selecting the right codec. Methodologies to evaluate the quality of speech reproduced in communication systems are classified under two categories, namely subjective and objective speech quality assessment methods. Since communication systems are designed for mankind, subjective testing obtains a statistically averaged rating from a group of listener by empirically recording their feedback or response to the different listener or conversational tests. It is a well-known fact that subjective speech quality assessment, although essential, is expensive and time-consuming [20][1][79]. Therefore, business sense dictates that some cheaper evaluation means shall first be used to substantiate the need for subjective testing. Recently, considerable effort has gone into objective methods, which are based on automated methods for the evaluation (e.g. accurately reflects similar results as those of subjective tests) of speech codecs used in VoIP systems.

In this section, we will summarize the key components of the internationally accepted standards (i.e. ITU Recommendations) that define subjective testing methodologies for speech codecs. We will then follow on and identify some of the most common objective methods. Of those objective

methods, the Perceptual Evaluation of Speech Quality (PESQ) standard, namely the ITU-T P.862 [39] will be described in more details since it will be used to assess the results of the algorithm proposed in this thesis.

2.6.2 SUBJECTIVE SPEECH QUALITY EVALUATION

The international standard ITU-T P.800 [36] provides the details for conducting the subjective evaluation of transmission system quality performance while the ITU-T P.830 [37] is used for the subjective testing of telephone-band and wideband speech codecs. The two are tightly related to the point where ITU-T P.830 relies heavily on ITU-T P.800. The main problem with such tests is that they are very laborious and time consuming to perform as they involve testing by human subjects in a tightly controlled environment with significant equipment and facility constraints, consequently, they are very expensive [20][79]. Another problem resides in the difficulty to correlate test sessions since they are performed by different listeners. Listeners may not attribute the same scores, when they re-assess over time, due to the subjective nature of those tests.

The ITU-T P.830 recommendation prescribes the listening-only and/or conversational methods, the later depending on a real-time implementation of the codec being available. For each of the methods, a number of prescribed steps must be followed. They are summarized here from [37]:

1. ***Preparation of source materials, including recording of talkers:*** A set of constraints on how to record the source material and also defining the recording content.
2. ***Selection of experimental parameters to exercise the features of the codec that are of interest:*** It suggests a set of *Codec* and *Reference* conditions for the evaluation of the codec. The *codec conditions* range from the input level of speech, though the asynchronous and synchronous tandeming of codecs, through interoperability with other

codecs, and background noise and music conditions, to only name a few. On the other hand, *reference conditions* are used to define the test environment used for testing and also to use a reference codec against which performance can be judge in terms of parameters.

3. ***Design of the experiments:*** Consists of designing and planning how the parameters identified in step 2 above will be tested.

4. ***Selection of a test procedure and conduct of experiment:*** This step follows the guidelines provided by the ITU-T P.800 where the *testing methods, test procedures,* and the *conduct* of listening tests, are described in detail.

5. ***Analysis of results:*** The consolidation and/or rationalization of the results allowing for the assignment of a numerical value to the codec representing its speech quality performance.

Scale Type	Scale Description				
Listening-quality	(1) Bad	(2) Poor	(3) Fair	(4) Good	(5) Excellent
Listening-effort	(1) No meaning understood with any feasible effort	(2) Considerable effort required	(3) Moderate effort required	(4) Attention necessary; no appreciable effort required	(5) Complete relaxation possible; no effort required
Loudness-preference	(1) Much quieter than preferred	(2) Quieter than preferred	(3) Preferred	(4) Louder than preferred	(5) Much louder than preferred
Disturbance (e.g. noise)	(-3) Intolerable	(-1) Rather loud (-2) Loud	(0) Moderately audible	(1) Slightly audible (2) Just audible	(3) Inaudible
Degradation category	(1) Degradation is very annoying	(2) Degradation is annoying	(3) Degradation is slightly annoying	(4) Is audible but not annoying	(5) Inaudible
Comparison category	(-3) Much worse	(-1) Slightly worse, (-2) Worse	(0) About the same	(1) Slightly better, (2) Better	(3) Much better

Table 2.5 – Ranking scales for the evaluation of speech quality performance

The main testing methods described in ITU-T P.800 are the Absolute Category Rating (ACR), the Degradation Category Rating (DCR), and the Comparison Category Rating (CCR). All the testing

methods have their associated testing procedure and use one or more of the scale types presented in Table 2.5. The table provides an overview of the scale types that can be used and a meaningful description of the scale and associated numerical value.

Next, we present the subjective test procedure for ACR, DCR, and CCR.

- ACR is the simplest to conduct where a test speech file is played and the listeners ranked this one according to the Listening-quality scale, the Listening-effort scale, the Loudness-preference scale, or another scale as directed by the experimenter. Several listeners conduct the tests and their score is averaged to provide the Mean Opinion Score (MOS). The main disadvantage of this test method is that it tends to lead to low sensitivity when we need to distinguish between almost similar performances.
- DCR is basically a refinement of the ACR method to allow us to evaluate similarly good speech quality performances. In this case two test speech files are listened to every time. The reference is listened first before listening the speech file of interest. The Degradation category scale is used for this method and results in the Degradation Mean Opinion Score (DMOS).
- CCR – Is basically the same as DCR with the exception that the test files (i.e. the reference and the target) are not always presented in a same order. This method uses the Comparison category scale and results in a Comparison Mean Opinion Score (CMOS).

Due to the number of details to consider, and the fact that those same details must be repeated every time a similar device need to be tested, subjective testing relies on certified testers or agency to be meaningful.

2.6.3 OBJECTIVE SPEECH QUALITY EVALUATION

As mentioned, subjective testing is difficult to conduct, tedious, time consuming, very expensive, and difficult to replicate. Objective measures such as the Squared Error (SE) [54], the Mean Squared Error (MSE) [74], and the Signal-to-Noise Ratio (SNR) [54] have been used to objectively measure the performance of legacy telephone systems, and are still used [86] in initial studies to complement more modern methods. They are still very useful for evaluating the performance of waveform codecs on a sample-by-sample basis (when samples are aligned), but are often of limited used when testing the new generation of low bit rate codecs. However, it is now well known that the SE, MSE, and SNR measures do not adequately predict subjective quality when applied to components of packet networks [6][53][79]. While these traditional objective measures can provide somewhat meaningful results under a set of conditions[8], it is recognized that more powerful methods should provide more accurate results without incurring the time and expense of subjective testing. Modern methods will model the human hearing system (recall the perceptual characteristics of the human system presented in section 2.2) and judgment logic to provide an objective evaluation of speech quality.

ITU-T has been investigating a variety of objective quality measures in the last two decades. The main ones were the LPC Cepstrum Distance Measure (CD), the Information Index (II), the Coherence Function (CHF), the Expert Pattern Recognition (EPR), and the Perceptual Speech Quality Measure (PSQM) and PSQM+, an enhancement to PSQM targeted at evaluating codecs used in wireless environment. Another objective method modeling the human hearing and judgment to estimate the perceived quality of speech is called Measuring Normalizing Blocks (MNB) and is

[8] Alignment of the samples is the most important condition and better time resolution (i.e. segmented SNR for say) a benefit to its usefulness [50].

described in [85][86] and the appendix of P.861 [38] . However, the method is compared against PSQM/PSQM+, EMBSD, and PESQ, and reported [42] to saturate with particular test sets.

Until recently, the ITU-T P.861 [38] commonly know as PSQM/PSQM+ was used to evaluate speech codec quality and was recognized to best correlate to results from subjective testing presented in the previous section. Lately, a new standard, the ITU-T P.862 [39] Recommendation, was designed to replace PSQM/PSQM+. PSQM/PSQM+ was not intended to measure real-time audio stream through an IP network since it had no capability to time align the original and uncompressed source signals (e.g. did not account for network delays). PESQ method integrates new algorithms such as transfer function equalization, time alignment, and distorting averaging over time, which makes it the tool of choice to test speech codecs aimed at VoIP systems. The P.862 is reported [42] to correlate best to subjective assessment results.

This recommendation will nicely add to experimentation performed in this thesis since it was tested with packet loss and packet loss concealment of CELP codecs, which is at the heart of this thesis.

2.7 ITU-T G.729A RECOMMENDATION

2.7.1 OVERVIEW

Annex A [31] to the ITU-T G.729 [30] recommendation named "Reduced complexity 8 kbit/s CS-ACELP speech codec" describes the reduced complexity Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP) implementation. The algorithm accepts a 16-bit linear PCM source input streams sampled at 8 kHz. The algorithm reduces the source input stream bandwidth requirement from 128 kbps to 8 kbps (a 16:1 compression ratio) while providing toll quality speech. The algorithm is said to perform well with background noise and under error conditions [70]. It

belongs to the hybrid class and is based on the human speech production model as described before. It takes advantage of ABS, long-term periodicities, and perceptual weighting principles, as described earlier.

In this section we rely on the description of CELP covered earlier in Section 2.4.3.2 since it provides its basic operation. We will focus on describing the new constructs to provide a high-level description of the encoder and decoder operation. Then, we introduce some of the enhancements associated to this standard and we present a summary of some of the performance criteria the standard was subjected to before its acceptance.

2.7.2 ITU-T G.729A ENCODER OPERATION

As before, the encoder uses the human speech system model to generate the short-term filter coefficient and ABS minimization loop (based on perceptual weighted measurements – i.e. weighted MSE) to determine the best excitation signal. This algorithm uses a variant of the original CELP algorithm to parameterize the excitation signal by removing the signal periodicities and selecting a stochastic sequence using indexes to codebooks. The adaptive codebook is used to represent the signal periodicities while the stochastic (sometime referred to as the innovation codebook) codebook deals with the residual signal components that are not predictable. Table 2.6 describes the parameters generated from the encoder. The table shows the parameter label, its bit allocation and number of bits, the subsystem of interest, and associated description. The first section, going top to bottom, represents the parameters for the LPC coefficients. The other 2 represent the excitation parameters for subframe 1 and subframe 2 respectively.

ID	Bit Allocation	No. of bits	Subsystem	Description
LPC Section:				
L0	0	1	1st codebook (1 bit)	Switched MA Predictor of LSF Quantizer
L1	1-7	7	1st codebook (7 bit)	1st stage LSF VQ
L2	8-12	5	2nd codebook (5 bit)	2nd stage VQ, first half
L3	13-17	5	2nd codebook (5 bit)	2nd stage VQ, second half
1st subframe:				
P1	18-25	8	Pitch period (absolute)	Pitch delay, 1st subframe
P0	26	1	Parity check on 1st period	Parity bit for pitch delay
C1	27-39	13	Stochastic Codebook index1 (positions)	Fixed codebook for 1st subframe
S1	40-43	4	Stochastic Codebook index2 (signs)	Signs of fixed codebook for 1st t subframe
GA1	44-46	3	Pitch and Innovation gains (3 bit)	Gains codebook, stage 1, for 1st subframe
GB1	47-50	4	Pitch and Innovation gains (4 bit)	Gains codebook, stage 2, for 1st subframe
2nd subframe:				
P2	51-55	5	Pitch period (relative)	Pitch delay, 2nd subframe
C2	56-68	13	Stochastic Codebook index1 (positions)	Fixed codebook for 2nd subframe
S2	69-72	4	Stochastic Codebook index2 (signs)	Signs of fixed codebook for 2nd subframe
GA2	73-75	3	Pitch and Innovation gains (3 bit)	Gains codebook, stage 1, for 2nd subframe
GB2	76-79	4	Pitch and Innovation gains (4 bit)	Gains codebook, stage 2, for 2nd subframe

Table 2.6 – ITU-T G.729A codec parameters description

The algorithm accepts 10 ms speech frames (e.g. 2 subframes of 5 ms) and the short-term analysis is based on a 10^{th} order linear prediction filter. The LP coefficients are determined using the Levinson-Durbin algorithm [19][66]. The LP coefficients are then converted to Line Spectral Pairs (LSP) and subsequently converted to Line Spectral Frequencies (LSF) [44] as the LSF are not as sensitive to quantization. The LSF values are pushed on the switched 4^{th} order MA predictor where the residue of this one is quantized by an efficient two-stage vector quantization procedure (conjugate structure).

The conjugate structure of Figure 2.8 shows the first stage as a 10-dimensional VQ codebook and the second stage is 10-dimensional split VQ codebook. This setup allows several combinations to be made while keeping the size of the codebook relatively small. The selection process is performed twice, for each of the *L0* selections (1 bit), and indexes *L1, L2,* and *L3* (17 bits) that provide the best match are transmitted to the decoder.

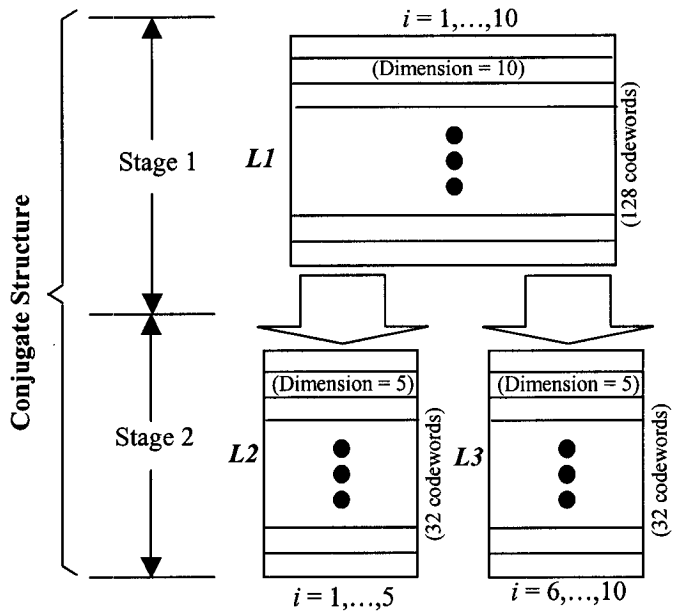


Figure 2.8 – The G.729 conjugate structure block diagram

The excitation signal or sequence is composed of an adaptive codebook vector that represents the periodicity in the voiced speech and an innovation codebook vector that represents the stochastic component of unvoiced speech. An appropriate gain is used for each vector to represent the speech level appropriately. We find the excitation signal parameters for each subframe. These will be selected after several ABS iterations:

- An index in the stochastic codebook (innovation vector).
- A gain for the innovation vector.
- A pitch delay (integer or fractional) in the adaptive codebook
- A gain for the pitch delay vector.

The pitch delay[9] and gain (the pitch parameters) are found using a two-step approach to minimize complexity. The open loop pitch analysis is done once per frame to find a coarse estimation of the pitch delay (i.e. a limited range). Then, the joint selection of the four parameters is performed by a

[9] Referred to as pitch delay rather then pitch period since it is a relative delay (i.e. speech is considered quasi-periodic over small segments).

closed loop search (i.e. ABS) in the neighborhood of the coarse estimate. The adaptive codebook vector is chosen during the ABS procedure. The stochastic codebook consists of building the vector samples during the ABS procedure (selecting 4 pulses location and associated gains) as shown in Figure 2.7(c). Since only four non-zero pulses are in the innovation vector, very fast search procedures are possible (e.g. Four nested loops corresponding to each pulse are used).

In the ABS procedure, the gains for the adaptive vector and the stochastic vector are jointly vector quantized using 7 bits. The pitch gain is around 1, but the innovation vector gain varies much more. To perform bit rate reduction, this wider range is quantized using a 4^{th} order MA predictor, which predicts the fixed codebook gain by considering the sequence of previous fixed codebook excitation vectors.

Overall, the LPC parameters (18 bits) and the excitation parameters (33 bits for the 1^{st} subframe and 29 bits for the 2^{nd} subframe) are transmitted to the decoder.

2.7.3 ITU-T G.729A DECODER OPERATION

The decoder is much simpler than the encoder (e.g. there is no ABS procedure). It relies on compression principles presented in this chapter. As before, the decoder operation is based on the human speech production model system but uses codebooks to generate its values. The decoder includes an adaptive post filter that consists of a cascade of 3 filters: the long-term post filter, the short-term post filter, and the tilt compensation filter, to improve the perceived quality of the reconstructed speech. The long-term post filter emphasizes the signal at the pitch frequency, the short-term post filter aims at reducing noise between the spectrum peaks (e.g. between formant

frequencies), and the tilt compensation adjust some of the distortion generated by the short-term post filter itself [70].

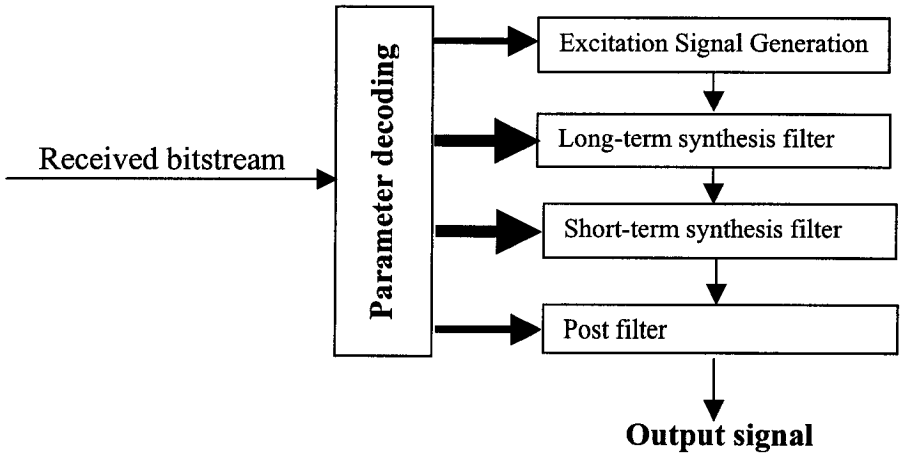


Figure 2.9 - ITU-T G.729A Decoder block diagram (reproduced from [31])

The pseudo-code for the decoding operation is, for each frame:

1. Determine if the received frame is good or bad frame;
2. Decode the LSPs;
3. Interpolation of the LPC;
4. Compute subframe based values (2 sub-frames):
    a. Decode the pitch delay;
    b. Decode the innovation code;
    c. Decode the pitch and innovation gains;
    d. Find the excitation; and
    e. Reconstructed the speech segment.

The resulting reconstruction of the source signal provides toll quality or a MOS of 3.7 as described in table 2.4.

2.7.4 ENHANCEMENT TO THE RECOMMENDATION

2.7.4.1 PACKET LOSS CONCEALMENT

A Packet Loss Concealment (PLC) strategy is contained in the description of the ITU-T G.729A Recommendation. In a nutshell, PLC is an algorithm that is embedded in the decoder side only, which is aimed at replacing lost packets with an estimate for the reconstruction of the speech signal. PLC effectively allows minimizing speech quality degradation associated with the loss of packets. Chapter 3 discusses the loss of packets through packet networks. The detailed description of the ITU-T G.729A PLC algorithm will be presented in Chapter 4 as part of the section that discusses methods to repair packet losses.

2.7.4.2 SILENCE COMPRESSION

Silence compression is an important consideration for VoIP. A conversation between people will most likely contain many periods of silences. Using silence compression is another method that assists conserving bandwidth. In this section we do not wish to detail the operation of silence compression but rather highlight some key observations. Annex B, ITU-T G.729B [32], is the recommendation describing the implementation of silence compression. Silence compression is achieved by combining the operation of three modules, namely, a Voice Activity Detection (VAD), The Discontinuous Transmission (DTX), and the Comfort Noise Generator (CNG).

Silence periods will often be associated with an unvoiced segment that falls below a given threshold. VAD logic will detect those segments and allow for transmission economy. There is a problem associated with the use of VAD. Since the VAD makes a voiced/unvoiced decision after evaluating the spectral distortion, the energy difference, the low band energy difference, and zero crossing difference, processing latency causes temporal clipping of the signal. This relates to the switching time from a silence-suppression mode to a voice-activity mode. Another problem with VAD is with

respect to the silence-suppression threshold. Louder background noise may be present which would be interpreted as voice-activity unless the background noise threshold is increased, in which case, will suppress lower level signals. Therefore silence suppression may not be usable in all environments.

2.7.5 PERFORMANCE EVALUATION

Since we elected for the ITU-T G.729A Recommendation, it is worth going through some of the performance criteria that were articulated before its formal acceptance. The ITU-T G.729A (CS-ACELP) algorithm was required evaluation against the ITU-T G.726 algorithm (32 kbps ADPCM) where the CS-ACELP codec should not perform worse than the reference ADPCM codec. Some of the quality speech performance criteria evaluations are:

- *Random bit errors* - under the presence of 10^-3 random bit errors;
- *Frame error lost* – allowable degradation of no more than 0.5 the MOS affected by random frame erasures (single frame and bursts of lost frames at the time); and
- *Background noise* – to present comparable performance as of the reference.

The ITU-T P.830 and P.831 provide several parameters or quality speech performance criteria that could also be investigated. Since the G.729A is a currently accepted standard, we can only assume the performance evaluation was successful.

2.8 SUMMARY

In this chapter, we have reviewed many topics associated to speech compression for Voice over IP. We reviewed the key characteristics of the human speech production and hearing systems. We identified the common input signal format for speech signals. We thoroughly reviewed the speech

compression techniques and associated approaches and principles. It was shown that hybrid codecs present a class of speech compression algorithm that are well suited for VoIP. Consequently, much of the review was focused at presenting a chronological history of speech compression principles, concepts, and approaches, from a basic two-state vocoder to the CELP codecs.

We described the key attributes, namely, the bit rate, the algorithmic delay, the complexity, and the speech quality, which are commonly used to select speech codecs. Common codec standards were tabulated, presenting their attributes in a consolidated manner for us to evaluate the speech codec we would select for this thesis. We selected to G.729A codec as it provides low delay, acceptable complexity, low-bit rate, and most importantly good speech quality.

Next, we covered a topic of high relevance to VoIP, the subjective and objective evaluation methods used to evaluate the speech quality of codecs. We believe it is important to re-iterate the fact that subjective testing is difficult to conduct, tedious and time consuming, and very expensive. ITU-T P.800 provides several criteria that must be followed for the subjective testing to be conclusive. Failure to follow some of the conditions will ultimately qualify the conducted tests as informal rather than formal. ITU-T P.830 should be seen as complementing ITU-T P.800 when subjective testing of speech codecs and associated devices is of interest.

Finally, we described the ITU-T G.729A algorithm in more details. We built on the new understanding of CELP technology and explained the CS-ACELP differences (i.e. structures and processes) for the purpose of implementing the algorithm to run simulations for this thesis.

3.0 PACKET-SWITCHED NETWORKS

3.1 OVERVIEW

A phone call in circuit switched networks follows a fixed and dedicated communication path used between users in a conversation. The connection is hardware, through switches, and transmit (receives) an analog representation of voice that is relayed through synchronous and asynchronous data and signaling channels. Control signaling is required to setup the connection but once established, the system only needs to monitor for the end of the connection. Circuit switching is a dedicated connection between two end users that is characterized by low overhead and low network latency, but this architecture is recognized as inefficiently managing the bandwidth usage [69][21]. On the other hand, packet networks are characterized as having non-dedicated communication paths rather than dedicated communication paths as in the case of circuit switched networks [21]. This presents the advantage that data is only transmitted when required, through the network devices (mostly routers and gateways), but to the cost of storing the data before forwarding it, adding transmission delays until the final destination is reached.

In VoIP systems, we are concerned with the exchange of real-time data (e.g. where a data segment is representative of a real-time speech segment – speech frame) over packet-switched networks. In this subsection we will present the network components enabling VoIP. The key components will be presented at a high abstraction level since packet network concept, techniques and technology is a field of research on its own. IP network operation is an important component of this thesis but complementary communication technologies such as ATM, frame relay, as well as Radio Frequency (RF networks for cellular phones) also play a significant role in the large-scale instauration of a flexible and reliable VoIP system. Discussions will also be surrounding network protocols, as they are the intrinsic glue that allows packet network to exchange data between the source system, the

transmitter, and the destination system, the receiver. We direct the interested reader seeking elaborate details with respect to packet network technology to consult the following literature [65][69][63][82][81][80][16][10][48] and [46] since this section will only be looking at a narrow view of packet-switched networks, mainly focusing on enablers and issues surrounding packet network architecture, operation, protocols, and Quality of Services (QoS). This will allow us to acquire a basic understanding of VoIP systems implementation challenges.

3.2 PACKET NETWORKS

Nowadays, IP networks represent the most widely deployed and commonly used communication technology for the exchange of data, thereof providing for the sharing of information everywhere across the globe. As we will see, the challenge of transmitting voice through packet networks is bigger than for transmitting voice through the public switched telephone network. In packet-switched networks, real-time or streaming data transmission generally relies on a best effort delivery mechanism.

3.2.1 BASIC ARCHITECTURE

The basic function of a packet-switched network is to provide communication between one or more transmitters and one or more receivers. The most prominent of packet-switched networks is the Internet that is referred to as an IP network. The popularity of the Internet or IP network has gone beyond the Internet boundaries, to the industry where IP networks form intranets of all sizes that are interoperable with the Internet. This section is aimed at formulating a basic understanding of the concepts, techniques, and other peculiarities surrounding packet-switched network operation. We begin by introducing the Open Systems Inter-connect (OSI) Model. The OSI model is a model of computer communications architecture with the initial objective of providing a framework for the

effective development of computer communications architecture. The model describes a seven layers framework that is aimed at developing highly modular protocols [65][80]. This layered arrangement is commonly referred to as the protocol stack. Each of those layers is assigned a set of basic functions they must support. An example protocol stack is presented in Figure 3.1.

<table>
<tr><th>OSI Layers</th><th colspan="6">Protocol Implementation Examples</th><th>DARPA Layers</th></tr>
<tr><td>Application</td><td>File Transfer</td><td>Electronic Mail</td><td>Terminal Emulation</td><td>Client/Server</td><td>Network Management</td><td>Control & Signaling</td><td rowspan="3">Application or Process</td></tr>
<tr><td>Presentation</td><td rowspan="2">File Transfer Protocol (FTP) RFC 959</td><td rowspan="2">Simple Mail Transfer Protocol (SMTP) RFC 821</td><td rowspan="2">TELNET Protocol RFC 854</td><td rowspan="2">Network File System Protocol (NFS) RFC 1014, 1057, and 1094</td><td rowspan="2">Simple Network Management Protocol (SNMP) RFC 1157</td><td rowspan="3">Internet Control Message Protocol (ICMP) RFC 792</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport</td><td colspan="3">Transmission Control Protocol (TCP)
RFC 793</td><td colspan="2">User Datagram Protocol (UDP)
RFC 768</td><td>Host-to-host</td></tr>
<tr><td>Network</td><td colspan="5">Internet Protocol (IP)
Address Resolution ARP RFC 826 RARP RFC 903
RFC 791</td><td>Internet</td></tr>
<tr><td>Data link</td><td colspan="5">Network Interface Cards (NIC):
Ethernet (IEEE 802.x), StarLAN, Token Ring, ARCNET described in RFC 894, RFC 1042, and RFC 1201, X.25 in RFC , SLIP and PPP described in RFC 1055 and RFC 1171</td><td rowspan="2">Network Interface</td></tr>
<tr><td>Physical</td><td colspan="5">Transmission Media:
Twisted Pair (RJ45), Coax, Fiber Optics, Wireless Media, Satellite microwave, etc.</td></tr>
</table>

Figure 3.1 – Organization of the protocol stack

Figure 3.1 is divided in three sections. The leftmost and rightmost columns represent two packet network communication models. The two models are considered equivalent and are presented here to consolidate terminology. The Internet model is a 5 layers model that is very similar to the DARPA model [65]. A representation is given in Figure 3.3. All those models logically map to one another. Different application examples are laid across the top of Figure 3.1 to show the relationships that exist with lower layers of the model. The following presents a brief explanation of functions associated to each layer of the OSI model.

- *Application layer*: Network applications are being designed and implemented to provide specific functionality to the users according to user's requirements. If a requirement stipulates the exchange of information with other users, then some control and exchange mechanism will be required at the application level to support the exchange of information. The application usually contains a module that implements communication calls and structures. Some examples shown in Figure 3.1 are a file transfer, an electronic message, client/server, and a network management application that all requires data exchange to fulfill their purpose. The information exchange may be constrained by be real-time requirements as in the case of a VoIP system.

- *Presentation layer*: The presentation layer usually offers the first level of data independence between the application and the communication protocols. Of importance, is the capability of both the transmitter and the receiver to have a common understanding of the data presentation. Good software developers will choose a structure that is reusable among many applications [65].

- *Session layer*: The session layer is usually the layer through which the application establishes connection setup and termination, establishes their common capabilities, and manages both the data exchange and session characteristics between end users. For example, common multimedia protocols such as RTP/RTCP, ITU-T H.323 and IETF SIP are accepted standards aimed at supporting application layer development by providing a set of primitive calls and structures. No matter the protocol, standardized or not, common calls and structures must be applied to both the transmitter and the receiver for the network application to operate correctly.

- *Transport layer*: The transport layer is responsible to transfer the data between end users or more precisely end applications. The main transport protocols are the Transmission Control

Protocol (TCP) and the User Datagram Protocol (UDP). TCP is a connection oriented transport protocol. TCP maintains a handshake or control that detects and compensate packet losses by retransmission of the lost packet. The protocol ensures that all the information is transferred and correctly received at destination. This is very adequate to delay insensitive applications such as FTP, e-mails, and other non-interactive applications, but the added delay to retransmit the lost packet (or the delay incurred by the hand-shake alone) is not tolerable for real-time voice traffic. Consequently, VoIP networks use UDP, which is a connectionless transport protocol that does not guarantee delivery (i.e. best effort delivery mechanism).

- *Network layer*: The network layer, of the ones discussed thus far, is the layer of interest in this thesis. The network layer for the Internet is called the IP layer. The IP layer is the layer from which the computer communication architecture becomes independent of the underlying hardware forming the communication infrastructure (the DARPA layer model has recognized this independence when looking at the right side of Figure 3.1). Therefore, the network layer and lower layers are responsible for the end-to-end delivery of data to the end systems/terminals through network devices such as Network Interface Cards (NIC), routers, switches, as well as interfaces to other network topologies/types via bridges and gateways.

- *Data link control layer*: The data link control layer is responsible for the reliable transfer of information over the physical link. Data link control protocols are often associated to the topology connecting the end terminals. For example, the IEEE 802.3 protocol is a bus topology protocol. The common functions associated with this layer are link management, flow control via Automatic Repeat Request (ARQ), and error control via Cyclic Redundancy Check (CRC).

- *Physical layer*: The physical layers or physical link or circuit, as its name implies, deals with the transmission of the raw information (i.e. bits) at the electrical and mechanical level. Digital symbols are transformed to some electrical representations for their proper transmission over the selected medium. For example, each LAN device uses some Access Control Protocol as with the most common, the bus contention protocol (CSMA-CD, IEEE 802.3). Nowadays, available medium are numerous and have their characteristics such as data rate, transfer speed or latency, carrying distance, error rate, etcetera. Figure 3.1 identifies some of these.

3.2.2 BASIC OPERATION

So far we have defined a narrow view of the protocol stack used in IP networks. We will use Figure 3.2 to explain the operation of the Internet or intranets. Labels have been placed on Figure 3.2 to facilitate the description. A legend is also provided which describes the different components contained in the figure. Figure 3.2 is laterally divided in two where a tight association exists between the upper and lower halves. We start with label 1 that illustrates the OSI model and associated layers. Two blocks containing the 7 OSI model layers are presented side by side. The two blocks are representative of end terminals such as a PC or dedicated IP devices such as an IP phone that connect to a topology as indicated by label 5 (see legend). The dashed lines associated with label 4, represent the need for equivalent level layers to understand one another (e.g. forms a logical channel). That is, the information exchange flows from the application layer through the lower layers before being broadcasted through the local network (e.g. the small cloud). If the end destination is part of the same cloud, it directly receives the communication broadcast and establishes the information exchange. On the other hand, if the destination terminal is part of a different LAN, then the information would require routing through network devices (e.g. routers at label 7 and 8) before reaching the destination LAN and destination workstation (the rightmost

protocol stack below label 1). Figure 3.2 is shown to represent a segment of the Internet or a large intranet. Label 7 is associated with a typical LAN that maybe contained in a same building. Label 8 shows several LANs that usually form a MAN. Label 7 and 8 are sometimes referred to as an end network and intermediate Internet Service Provider (ISP) respectively. Label 9 shows a major ISP that generally provides high data access for intermediate ISPs. Label 10 shows the very high-speed link to the rest of the Internet or intranet.

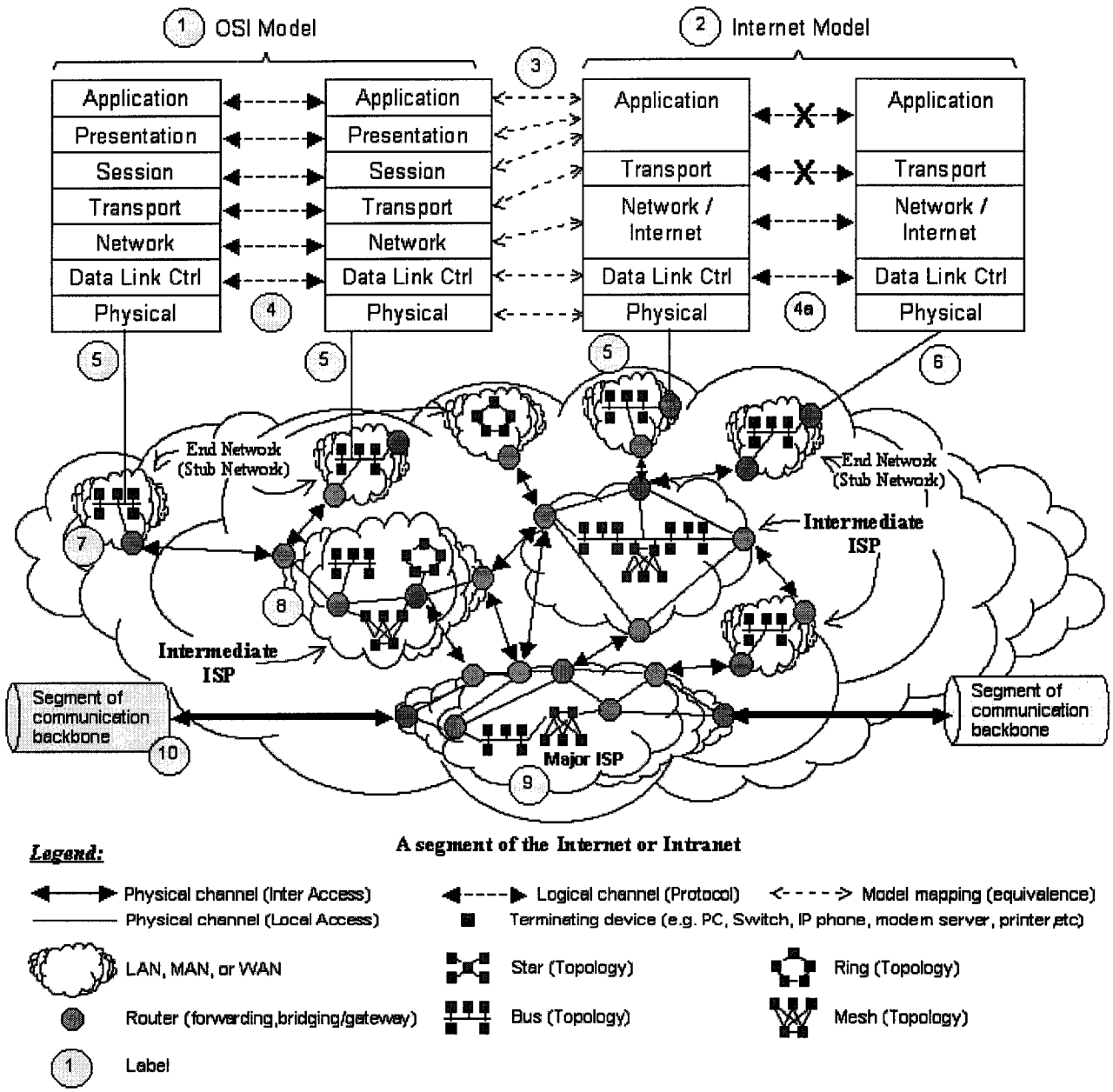


Figure 3.2 - A conceptual view of packet-switched network (Internet/intranets)

In fact, this diagram shows that VoIP communication may occur locally as part of a same end network (probably the best scenario), and may be delivered from node-to-node (i.e. many hops) before reaching destination. In the case where many hops are traversed, the network delay for end-to-end communication is ought to increase. Label 2 shows the Internet layer model as discussed earlier. In this case, the application layer is not used, as the end devices are routers. A router is essentially a store-and-forward device that only uses the lower layers of the model to deliver the information between networks. On the other hand, gateways (i.e. another type of network devices enabling interoperability of networks) usually interconnect disparate networks and will have an application layer performing some conversion of the information to be exchanged. Finally, label 3 shows the mapping or relation that exist between the ISO Model layers and the Internet Model layers. Fundamentally, they perform the same functions but those functions are grouped under different modules respectively.

3.2.2.1 ROUTING

IP networks deliver the information from router-to-router or node-to-node in a store-and-forward manner. This is necessary to overcome the bursty nature of data networks. Assuming the application has broken the data into segments, or packets, and transmitted the packets through its protocol stack, as explained before, then, packets are reaching neighbor routers to deliver the packets beyond its LAN. In fact, the router can have multiple paths that are managed through the routing table (i.e. the table is dynamically updated from information received from adjacent routers according to a routing protocol such as the Router Information Protocol (RIP) [65][80] or the open Shortest Path First (OSPF) [65][80], etcetera). Therefore, redirection or forwarding of the packets to the right path is a decision the router must make using its routing table. No matter the routing technique used, each

router involves a processing overhead and possibly a transmission overhead [65] that together form a routing delay that contributes to the overall transmission delay incurred for the current exchange of data.

The unpredictability of the traffic on the Internet or the intranet can create surges at the routers. In fact, a surge is when the router receives more transmission requests than it can handle (e.g. beyond its bandwidth), creating a buffer overflow, which implies packets are dropped. A router is internally composed of input and output buffers or queues that are used to store-and-forward each stream toward its destination, thereof, *longer queues will help prevent the lost of packets during surges but will also introduce longer delays*. For a network lacking QoS, the router usually provides First-In First-Out (FIFO) queues where packets are sent in the same order they are received. This implies that data, real-time or not is all treated equally. This causes some problems to VoIP streams since they must wait their turn in the queues, consequently accumulating more transmission delay. The problems associated to routing will further be explained in Section 3.4.

3.3 MULTIMEDIA PROTOCOLS

As discussed in Section 1.1, GSTN provides signaling that allows for the management, control, and signaling of the voice channels. Similarly, multimedia protocols are providing management, control, and signaling services for the delivery of packets containing real-time information (i.e. audio, video, or interactive data). The Real-Time Protocol and Real-time Control Protocol (RTP/RTPC) [77] are the most widely used protocol for the delivery of real-time data in VoIP. Additionally, we introduce protocols such as the ITU-T H.323 standard [20][52] and the Session Initiation Protocol (SIP) [76] since they are architecture specifications that support the end-to-end calls by providing signaling, control, and data delivery mechanisms.

3.3.1 REAL-TIME PROTOCOL

The RTP [77] is an important component of any VoIP applications as it provides a framework or protocol for real-time data transmission. It runs on top of UDP and provides the delivery mechanism and structures augmenting the application tolerance to packet jitter and packet losses. Because of the unpredictability of network impairments there is no guarantee about the delivery performance and reliability. RTP performs the end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. It assigns a sequence number and time stamp to each packet that allows tracking missing packets. Also, RTP packets may arrive late or out-of-order, in which case, the time stamp will assist reordering the sequence before playback at destination.

The Real-Time Control Protocol (RTCP) [77] is used to provide administrative support or services to RTP. Those services include payload type identification, user information, and network statistics. For example, RTCP transmits statistical messages that assist end terminals manage their jitter buffer.

3.3.2 ITU-T H.323 Standard

The Packet-Based Multimedia Communication Systems or the ITU-T H.323 recommendation [35] is an architecture specification that establishes a set of functions for the operation of multiparty real-time audio and video systems over packet-switched networks. The primary role of the standard is to inter-operate with other multimedia terminals. The standard describes the major components of its architecture as terminals, gateways, gatekeepers, and multipoint control units.

- ***Terminals:*** A desktop PC or dedicated device used for bi-directional multimedia communications. Usually an application that is H.323 capable/enabled.

- ***Gateway:*** An H.323 gateway provides connectivity between dissimilar networks (e.g. H.323 network vs. non-H323 networks). For example, a gateway is used to connect an H.323 terminal to a GSTN terminal. Translating protocols are used to establish call setup and call release procedures and format the data accordingly for its exchange.
- ***Gatekeeper:*** The gatekeeper is a focal point for all calls in the VoIP system. It provides important services such as addressing, component authentication and authorization, call accounting, and other such services.
- ***Multipoint Control Units (MCU):*** MCU is the central management point for conferences. It manages resources, negotiates setup parameters, and may optionally manage the streams.

The standard presents an architecture aimed to facilitate interoperability between multimedia applications in similar IP networks as well as disparate networks (i.e. IP network interoperability with GSTN). The configuration[10] parameters have to be synchronized between communicating entities and is a function/component of the control processes based on ITU-T H.225 and ITU-T H.245 Standards.

Figure 3.3 shows the relationships of the application of interest versus the two packet network communication models presented earlier. The four bottom layers of the model are the common communications layers found in the Internet and corporate intranets. The remainder of the layers, namely, the Session, the Presentation, and the Application layers show the member protocols associated to the H.323 architecture.

---

[10] Information exchange (i.e. capability information) between entities occur before the synchroniziation of configuration parameters occurs.

<table>
<tr><th>OSI Model Layers</th><th colspan="6">System Implementation Examples</th><th>DARPA Model Layers</th></tr>
<tr><td rowspan="2">Application</td><td>Audio Application</td><td>Video Application</td><td colspan="4" rowspan="2">Application/System Control and Management</td><td rowspan="4">Application or Process</td></tr>
<tr><td>Voice Codecs G.711, G.723.1, G.729A, etc.</td><td>Video Codecs H.261, H.263, etc.</td></tr>
<tr><td>Presentation</td><td colspan="2" rowspan="2">Real-Time Protocol (RTP) RFC 1889</td><td rowspan="2">Real-Time Control Protocol (RTCP) RFC 1889</td><td rowspan="2">H.225 RAS</td><td rowspan="2">H.225 Call Control</td><td rowspan="2">H.245 Media Control</td><td rowspan="2">T.120 Data</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport</td><td colspan="2">User Datagram Protocol (UDP) RFC 768</td><td colspan="4">Transmission Control Protocol (TCP) MIL-STD-1778, RFC 793</td><td>Host-to-host</td></tr>
<tr><td>Network</td><td colspan="6">Internet Protocol (IP) MIL-STD-1777 RFC 791</td><td>Internet</td></tr>
<tr><td>Data link</td><td colspan="6">Network Interface Cards (NIC): Ethernet (IEEE 802.x), StarLAN, Token Ring, ARCNET RFC 894, RFC 1042, RFC 1201</td><td rowspan="2">Network Interface</td></tr>
<tr><td>Physical</td><td colspan="6">Transmission Media: Twisted Pair (RJ45), Coax, Fiber Optics, Wireless Media, etc.</td></tr>
</table>

Figure 3.3 – H.323 Protocol Architecture

The Session Initiation Protocol (SIP) is an industry standard protocol proposed by IETF RFC 2543. It is used in IP networks and provides similar function as of ITU-T H.323 but is said to be simpler, more efficient, scalable, and more flexible, than the H.323 standard [20][1]. These are in fact competing standards for VoIP.

3.4 NETWORK PROBLEMS

In this section, the effects of network impairments, namely, the transmission delay (or network latency) and packet losses are reviewed. We have just seen that if a network is not congested, then, the network usually provides very low delays and very low packet losses. However, if the network becomes congested, the transmission delay increases and potentially reverts to losing packets once its buffers are filled to capacity, causing network impairments in either case.

In this section, we first attempt to characterize the network impairments. Then, we look at techniques that assist minimization of network impairments.

3.4.1 PACKET LOSS

Packet loss is the first most important cause of network impairments found in IP networks. The rate of losses or missing packets is very difficult to predict or establish since it varies according to the capacity of the network, the network load, the network services provided, etcetera, which would require a multidimensional mathematical model based on non-deterministic values. However some studies and associated experiments [42][2][14][17][4] were conducted and some of the values and key observations are reported below for the benefit of this thesis.

- Loss rate is usually in the range of 2-5% for single packets lost.
- Loss occurrences decreases an order of magnitude for larger bursts but the loss rate may not necessarily decrease (i.e. loss bursts of size 10 and above).
- Correlation between errors exists (i.e. the probability the next packet will be lost given this one is lost).

In VoIP, lost packets are directly associated with the delivery mechanism used to deliver real-time voice traffic, namely, UPD messages (recall that TCP segments would be retransmitted following the detection of errors). Network congestion occurs at a router when its queues/buffers are reaching beyond their capability. When the storage space (e.g. queues and buffers) gets filled, the next transmission requests will force an overflow causing the loss of one or more packets. Another instance where a lost packet may occur is when using the Carrier Sense Multiple Access with Collision Detect (CSMA-CD) principle used by Ethernet (e.g. IEEE 802.3) protocols. In this case, the Data Link Control layer and the Physical layer implementing CSMA-CD would detect a collision and the packet would automatically be discarded.

3.4.2 PACKET DELAY

The transmission delay is the second most important cause of network impairments found in IP networks. The packet delay is the sum of the delays accumulated from the source terminal, through routing or queuing delays, and to the end terminal. The source and end terminals delays are attributable to hardware resources contention and/or software execution time present in all devices used to communicate the information. As seen earlier, the queuing delay is mostly due to the transmission of the information from node-to-node in IP networks (i.e. store-and-forward). The queuing delay is non-deterministic as it varies according to the network load, which is highly dependent on the network usage (the variance in delay from packet to packet is called *jitter*).

Terminal delay

We define this delay to be the period from the time the information is ready to be transmitted to the time it gets transmitted. Three delays characterize the terminal delay.

- Packetization delay: This delay is the period of time required to form a packet. This is application dependent and is usually not significant (<< 1 ms). However, this is not always the case in VoIP. For example, if multiple speech frames are grouped into a single packet to reduce the header overhead, then, a substantial delay may be incured while waiting for the additional frames (> 10 ms in most cases).
- Access delay: This is the elapsed time before a terminal successfully connects to the communication infrastructure (i.e. once per session). For example, the V.34 modem standard requires 20-40 ms to access the network [42]. In the case of access via Ethernet (i.e. IEEE 802.3), the access delay depends on the medium contention but assuming sufficient bandwidth is available, this delay is relatively small (<< 1 ms).

- Serialization delay: The time it takes for the modem or the network device to transmit the data on the network (e.g. the bits, one after the other). This is a critical consideration for low-speed interfaces since it is a function of the amount of information to transmit over the rate of transmission (e.g.1500 bytes/10 Mbps = 1.2 ms, 10 bytes /10Mbps = 8 μs, 1500 bytes/ 56 kbps = 214 ms, and 10 bytes / 56 kbps = 1.43 ms).

Routing Delay

We define this delay to be the accumulated waiting period associated to the storing-and-forwarding of the information through the network. The information is queued in each router along the path for forwarding (e.g. decision) onto an out queue, before departing the device. The accumulated waiting period will be accounting for all routers traversed. Two delays characterize the routing delay.

- Queuing delay: Two queues exist. The in-queue that stores the information before processing or more precisely routing, and the out-queue that buffers the information before its serialization on the medium. If the network device uses FIFO queues with no QoS, then, non real-time data already in the queues will have to be exhausted (e.g. serialization) before the real-time data can be processed and serialized. This may be a serious problem in VoIP systems if low-speed interfaces are used. This delay is maintained between 30 ms and 100 ms in some well tuned network infrastructures but can exhibit several hundred milliseconds in other network infrastructures [6][42].
- Decision delay: The time it takes the network device to decide on the best path to use to forward the information to destination. This is a function of CPU speed that is usually negligible (<< 1 ms).

Transmission Delay

The transmission delay is the sum of the terminal delays (serialization at the transmitter and receiver) and the routing delays. From the values provided in the previous paragraphs, we can see that serialization is probably the most serious threat to a real-time network application such as VoIP, from a delay budget perspective, especially in the case of low speed links/connections and a mix of large non real-time data payloads and real-time payloads. Using a VoIP application on a desktop connected to a low speed link may still be workable assuming only the VoIP application is used. However, if routers or other network devices are interconnected with low speed links, this may not be acceptable because of delays induced by the transfer/serialization of non real-time data.

Delay Variations

As explained before, IP networks use routers to perform end-to-end delivery of information. The information is transmitted from one router to another where each router will induce some amount of delay according to the local network conditions (e.g. traffic conditions changing over time). This implies that consecutive packets from a same source will experience different delays and these inter-packet delay variances are called *jitter delay* or simply *jitter*. Another cause of jitter is when consecutive packets are transiting different paths from a same source to a same destination terminal, thus inevitably inducing different packet inter-arrival delays.

Inter-packet delay variances or jitter is not damaging for the transmission of non real-time data but it is not the case for real-time applications producing packets at regular intervals such as in a VoIP system. The jitter delay introduces a temporal distortion that can degrade speech quality beyond intelligibility. A jitter buffer along with the RTP/RTCP protocol discussed earlier provide functions and structures to control the effect of jitter on a real-time stream. More details will be provided in the next chapter.

3.4.3 MINIMIZING THE NETWORK IMPAIRMENTS

The network should provide sufficient bandwidth and a stable routing environment (gateways included) in order to minimize network impairments. Network impairments are usually introduced when traffic peaks affect the routing environment or the network out-grown the initial capability, thereby affecting the performance of real-time applications such as VoIP. Optimizing the data throughput efficiency should be contemplated rather than forcing a bandwidth upgrade. The most serious cause of latency is associated with the serialization delay incurred by the transmission of packets when traversing all devices (end terminals included). Some possible solutions to managing network impairments are listed and described below.

- Minimize the size of headers;
- Minimize serialization delays;
- Prioritize flows; and,
- Reserve bandwidth for real-time applications.

Header Compression

Figure 3.4 shows that a network packet is mainly composed of header information and a very small portion represents the encoded speech frame. Therefore, the header information has an impact on the transmission delay and on the required bandwidth for real-time applications. Header compression has been proposed [51] to reduce this overhead. The Compressed Real-time Transport Protocol (CRTP), RFC 2508, is based on a similar technique used in TCP/IP header compression.

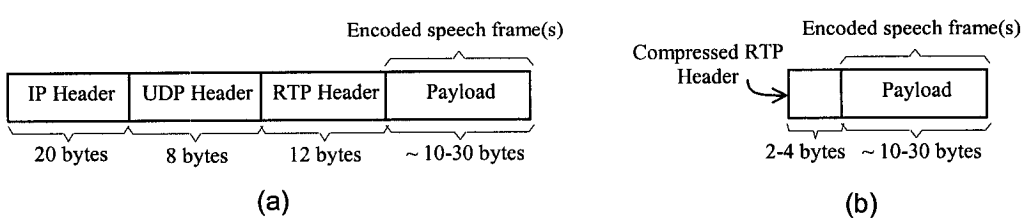


Figure 3.4 – RTP header compression format

Figure 3.4.a shows an uncompressed network packet that is constituted of an overhead of 40 bytes while Figure 3.4.b shows its equivalent compressed version that provides an overhead of 4 bytes. Assuming a 10 bytes payload (e.g. a single ITU-T G.729A encoded speech frame), the required bandwidth is effectively reduced from 40 kbps to 11.2 kbps. This technique is used for point-to-point links allows for substantial bandwidth savings effectively reducing network latency for VoIP traffic.

Fragmentation and Interleaving

Fragmentation is a mean to minimize the impact of serialization delays thereof reducing network latency and jitter. The idea is to fragment large packets into several smaller packets such that the serialization delay is reduced. In IP networks, data is generally configured to carry a 1500 bytes payload. By decreasing the size of the maximum payload, the serialization delay for each packet would be reduced, which would allow fast processing of smaller packets, such as in the case of VoIP. However, there must be an interleaving mechanism that will allow voice packets to be placed in between the fragmented packets for this method to be effective. A drawback to fragmentation is the additional number of headers required to transmit the same information.

Queuing Policy

Recall that almost all devices in a packet network perform queuing of packets for their transmission on the IP network. By default, fair queues are FIFO queues temporarily store packets until the transmitting interface can send them. Packets are sent in the order they were received, which causes some concerns when real-time voice traffic is mixed with data traffic. This means that a voice packet could potentially wait for one, two, or more data packets transmission before their transmission. During congestion periods, packets are dropped without regards to their content.

An initial QoS feature is the use of *priority queuing* in order to provide preferential treatment to more important traffic flows. Typically, arriving packets are dispatched to four queues: low, normal, medium, and high priority. Packets are then serviced according to their assigned priority and lower priority traffic is only serviced once higher priority queues have exhausted. A major limitation is the starvation of lower priority queues when priority queues never exhaust.

Traffic policing or traffic shaping and other queuing approaches exist. They provide for the construction of an adequate QoS implementation for IP networks, which would provide a performing VoIP infrastructure but their study is beyond the scope of this thesis.

Quality of Service

The Resource reSerVation Protocol (RSVP) is standardized by the Internet Engineering Task Force (IETF), and is one of the fundamental mechanisms to provide QoS [20][50]. RSVP is a transport level signaling protocol for reserving resources for unreliable IP-based networks. RSVP is a control and signaling protocol that is implemented at the IP layer like the Internet Control Management Protocol (ICMP) and the Internet Group Management Protocol (IGMP). Consequently, RSVP enables routers to dynamically allocate bandwidth for specified flows that require special service. It is designed to work with multicast or unicast traffic, and uses the underlying routing protocols to determine the next hops toward the destination.

When using RSVP, H.323 terminals can effectively reserve resources for real-time traffic streams that are based on a QoS requirement. Two models enabling the establishment of a QoS architecture are the *Integrated Services* (IntServ) and the *Differentiated Services* (DiffServ). These models

usually include an associated bandwidth management scheme that allows them to dynamically provision bandwidth capacity while managing allocated bandwidth to the current traffic [51].

The IntServ model proposes a *Guaranteed service* class for applications with a fixed delay constraint and a *Controlled load service* for applications requiring some reliability beyond the best effort delivery service typical in IP networks. IntServ is resource intensive, as all devices in the path must maintain state information for the session (each session requires their unique set of state information). Consequently, the model is associated with many flows to be managed.

The latest proposed model is the *Differentiated Services* (DiffServ) where a large bandwidth is allocated for its common use by all the traffic flows requiring QoS. This model offers two service classes. The first class, the *quantitative DiffServ* class where the throughput, the delay, the jitter, and the loss performance are specified deterministically or statically. The second class, the *priority-based DiffServ* class specifies the access priority to network resources.

These bandwidth management schemes provide an effective QoS architecture for the delivery of real-time voice data associated with VoIP systems.

3.5 SUMMARY

In this chapter, we highlighted some technological difficulties that are currently affecting the transport reliability of real-time speech using IP networks. We reviewed the fundamental packet-switched networks infrastructure that currently provides best effort delivery with no Quality of Service to VoIP applications. Its operation was briefly explained using the well-known OSI Layered model (mapping to the DARPA and the Internet Layered model was also introduced). Then, we

presented router operation where we could see that they are responsible to generate most of the impairments in packet-switched networks (gateways alike). We followed with a brief introduction to multimedia protocols such as RTP/RTCP, H.323, and SIP that are widely used in VoIP system implementations as they are commonly used to implement VoIP architectures and because they also provides mechanisms and structures to alleviate some of the network impairments. At that time we were in a good position to thoroughly investigate sources of network impairments in order to characterize them. Finally, we presented some of the most commonly QoS implementation features that can potentially assist IP networks minimizing the network impairments.

4.0 VOIP SYSTEMS

4.1 OVERVIEW

The flexibility associated with the digital representation of communication signals quickly brought packet-switched network to compete against the traditional circuit-switched network currently used in GSTN. The main attraction of packet-switched networks is that they can support an almost limitless set of voice services [6].

VoIP technology relies on the active areas of speech compression and packet network research. Both disciplines were covered in Chapter 2 and 3 respectively. VoIP services use end devices such as IP phones, multimedia PCs, servers, gateways and other similar devices that combined together compose the VoIP architecture. The vast majority of current VoIP systems are built over IP networks that provide best-effort delivery. Those networks are known to provide no guarantee with respect to the timely delivery of packets to destination. As discussed in Chapter 3, IP networks were initially designed for the delivery of data only. In convergent networks, voice streams are treated in the same way as data streams. During peak periods, more vulnerable areas of the network maybe overwhelmed with transmission requests to the point of severely delaying the packet transmission (beyond the useful time-domain – delay budget). Another problem associated with packet network, is the case when the internal buffers of the network device (i.e. router and gateways) are about to overflow and that the network device drops any new incoming packet transmission request, actually observed as a packet loss. It is likely that some encoded speech packets will be dropped during transmission over the network. When one or more network packets containing phonemes are lost, the perceived quality of the speech maybe severely degraded. It is reported [61][87] that the human brain is capable of reconstructing a few lost phonemes in speech but that too many missing packets makes a voice unintelligible. Consequently, a speech codec must provide robustness against lost

packets to ensure the perceptual quality of the reconstituted speech segment will not deteriorate significantly.

In telephony, such as the case of VoIP, we are apriori concerned with narrow-band (e.g. voice band) communication between two end terminals. The communication infrastructures, in this case the IP network, must be scaled to fit the system architecture of interest, such that, sufficient bandwidth[11] must be available across its links. Reliability is also of concern. For example, if link errors are not maintained to a low level, then the advertised rates will not be achieved. Once the infrastructure is known, you can then look at a speech compression algorithm that will fulfill communication requirements. For instance, if the scope of a VoIP system is to the LAN and that no dial-in VoIP sessions are to take place, than, chances are that an algorithm such as ITU-T G.711 will allow for simultaneous communications to take place while offering toll quality calls. On the other hand, the rate associated with the ITU-T G.711 algorithm does not allow using it for low rate links such as for dial-in (i.e. <56 kbps access rate). However, a low bit rate codec, such as the ITU-T G.729A, is adequate for low access rates while still providing toll quality communication.

In this section we will identify the requirements ensuring a viable VoIP system implementation. This will be achieved by investigating the issues or constraints within which a VoIP system must operate. We study the effect of these identified constraints using a generic network model. Then, we explore common sender-based, receiver based, and sender/receiver-based packet loss repair methods as they are used to overcome some of the network impairments. Of interest is the G.729A algorithm that is used in this thesis. We describe its PLC algorithm, its internal memory state, and the effect of

[11] Bandwidth is usually an expensive commodity. Consequently, network architecs and designers rarely over estimate the bandwidth requirements.

packet losses to its operation. Finally, we focus our effort to understand the problems associated with the memory state error induced following packet losses.

4.2 ISSUES IN VoIP COMMUNICATION

As discussed in Chapter 3, IP networks were designed for the transfer of data (non-interactive data). TCP is a connection oriented transport protocol that ensures all the information is transferred and correctly received at destination. TCP maintains a handshake or control that constantly consumes bandwidth but also detects and recovers packet losses by retransmission of the lost packets. This is adequate to delay insensitive applications such as FTP, e-mails, and other non-interactive applications, but the added delay to retransmit the lost packet (or the delay incurred by the hand-shake alone) is not tolerable for real-time voice traffic. Consequently, VoIP networks use UDP, which is a connectionless transport protocol that does not guarantee delivery (i.e. best effort delivery mechanism) but provides low transmission overhead. While IP networks using UDP allows for the transmission of real-time data such as voice over IP, those networks, from time-to-time, almost inevitably suffer impairments that degrade the performance of VoIP applications. In this section, we present problems associated with these network impairments, namely, the end-to-end delay or Round Trip Time (RTT), inter-packet arrival delay (jitter), and packet losses that affect the performance of VoIP systems and attempt to characterize associated constraints.

4.2.1 DELAY

In Chapter 3, we mentioned that IP networks could experience large delays in their normal operation. In contrast, the traditional circuit-switched network phone system or GSTN does not suffer those delays because it establishes an end-to-end circuit that maintains small propagation delays through devices and the medium (i.e. the distance divided by the speed of light). Therefore,

the implementations of VoIP systems present important challenges, especially if toll quality calls are required.

Issues involved with long delays

Voice is a continuous phenomenon that carries information with an important timing relation. This timing relation is important to characterize expressions of the speaker as well as to maintain a conversation. When delays are present in the communications system, unnatural silence segments tend to separate syllables or words that changes the rhythm of the voice (i.e. temporal distortion). This distortion negatively affects the perception of the voice by the end users. Studies characterizing the effect of delays in conversations were reported [1][6][5] and findings indicate that a 300 ms end-to-end delay was acceptable for most people. Beyond that threshold, losses in conversational efficiency of about 25% and 40% are associated to delays of 500 ms and 800 ms respectively. At those delays, the conversation begins to feel like half-duplex and is still acceptable for few people when necessary but hardly comparable to the GSTN performance. It is suggested that one-way delay between 200-800 ms is acceptable if only for a brief period and that such occurrences are far apart, which is certainly beneficial to VoIP systems.

To deliver quality speech or conversations, the end-to-end delay or Round-trip Delay (RTT) must be maintained to around 300 ms or less. Another important reason to maintain the delay below a threshold is to mitigate acoustical echo in the system. The delayed echo may become very annoying during a telephone conversation [57][20][47]. However, we do not cover these effects in this thesis as it could be the subject for a thesis on its own.

The description that follows is based on the assumption that at least two end devices (e.g. IP phones, or multimedia PCs, or gateways, or any combinations) are connected to the IP network (e.g. Internet or intranet) via a network access point. Figure 4.1 shows a typical two-way communication system

with internal details. Each end system is composed of a transmitter and a receiver that exchange voice data via a communication medium (e.g. IP network). A speaker at terminal A (the transmitter) processes and transmits voice information to terminal B (the receiver).

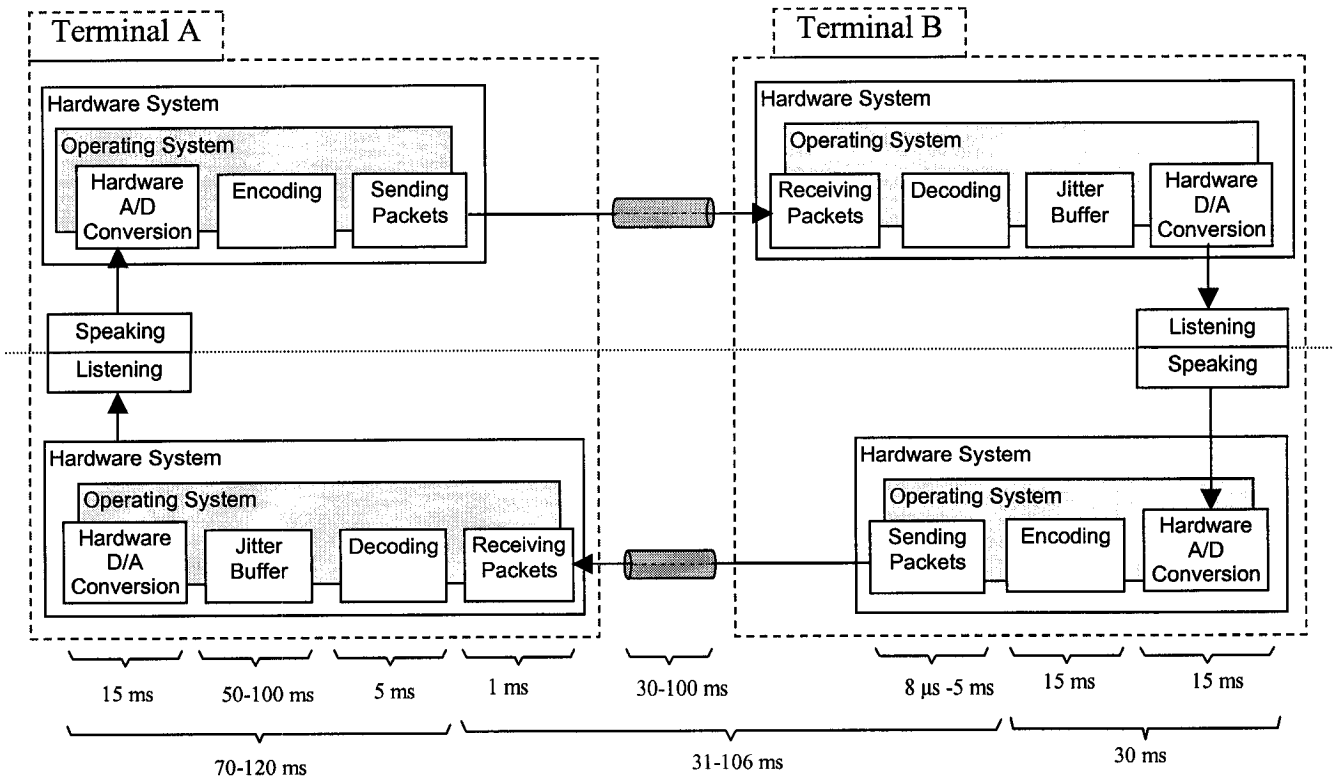


Figure 4.1 – A typical codec generated delays in a communication system

The receiver receives the information (usually source compressed, optionally encrypted and channel coded), processes the information (usually the reversed process that was performed at the transmitter), and finally presents the information to the listener. In general, a communication system implementation is performed on some hardware and associated firmware and/or software that will allow processing of the source information, under the control of an operating system, before it is transmitted to the destination end of the system via a medium.

Before we can minimize the end-to-end delay in a VoIP system, we must first understand what those delays are. The most significant sources of delays are the codec delay, the serialization delay,

network delay, jitter buffer delay, and packetization delay. Other less important sources of delays will also be briefly described. Figure 4.1 shows the delays associated with the delay description or identifications that follows.

Codec Delay

Codec or algorithmic delay is the delay associated with the time required to collect samples for processing by the speech codec. This delay was identified as the combination of the *Frame Size* and the *Look-ahead Delay* as described in Section 2.5. This delay is codec dependent and is generally longer for low bit rate codec (on average 15-40 ms - see table 2.4 – in our case 15 ms and 5 ms for the G.729A encoder and decoder respectively).

Transmission delay

Network delays as explained in Chapter 3. These delays are unpredictable and dependent on many combined delays such as the routing delays (30-100ms), the terminal or serialization delays (8 μs–5 ms) for IEEE 802.3 at 10 Mbps (e.g. V.35 modem delay (35-200ms)).

Jitter Buffer

The jitter buffer mitigates packets inter-arrival delay variances at the cost of additional delay. The causes of jitter were explained in Section 3.4.2. Jitter delay is caused by the variance of the network delays and the jitter buffer assist at smoothing the flow of the playback in real-time applications such as in VoIP applications. This delay provides some flexibility to accept late packets but introduces a temporal distortion that may degrade the communication. Consequently, common buffer sizes for the jitter buffer range from 50-100 ms [18]. If real-time interactivity is not important then a larger jitter buffer will allow receiving/attending all delayed frames for their eventual playback. This is a tradeoff between high latency (low loss) versus low latency (high loss).

Packet Size

Packet size, in this case, refers to the number of speech frames in a same packet. For example, the G.729A deals with frames of 10 ms. Assuming each packet was to hold three frames, it would imply an additional delay of 30 ms. Our discussions only consider packets containing a single speech frame.

Other sources of delay

The most common source of sporadic delay in a PC is the Operating System. Hardware and software contention are other sources of delay that are difficult to track. It is assumed that these delays are negligible and are not considered in this thesis.

End-to-end delay

To simplify the explanations, we elected to group the delay values under 3 main groups representing the collection and formation of the source speech data, its transmission through the IP network, and its reconstruction at destination for its playback. The bottom of Figure 4.1 shows 3 main braces representing these delays. We need to compute 4 combinations in order to find the minimum and maximum delays associated with this model. For example, the rightmost brace represents a fix delay of 30 ms that is induced when capturing and preparing the source speech data. The leftmost side represents a delay range between 70-100 ms that is induced by the jitter delay, the reconstruction delay, and the playback delay. Finally, the IP network will induce delays ranging from 30 ms to 100 ms. Therefore, a first combination, 30 ms + 70 ms + 31 ms = 131 ms represents the minimum unidirectional delay. Similarly, 30 ms + 70 ms + 106 ms = 206 ms, 30 ms + 120 ms + 31 ms = 181 ms, and finally, the maximum delay is 30 ms + 120 ms + 106 ms = 256 ms. Assuming the delays are

the same on either directions, the calculated end-to-end delays are 262 ms, 412 ms, 362 ms and 512 ms respectively.

The results obtained identify the delay constraints under which a typical VoIP implementation would operate. Therefore, a VoIP system based on best delivery mechanism with no QoS will encounter communication difficulties for end-to-end delays between 300 ms and 512 ms (e.g. during increased network traffic) but as discussed before, these may still be acceptable for most users. As suggested in Section 3.4.3, some IP network features may minimize the end-to-end delay that will maintain the adequate performance of the VoIP system.

4.2.2 LOST PACKETS

Packet loss is a major problem for VoIP systems because it can severely degrade the quality of speech. Perkins *et al.* [61] reported that several studies have attempted to quantify packet losses. On average, 2-5% packet losses will be experienced but more important losses occur less frequently. Another study [53] shows that small variations in the daily averaged round-trip delay corresponds to a significant increase in the percentage packet losses (an increase in delay of 15 ms shows lost rates climbing to 25%).

Issues involved with the loss of packets during the real-time transmission of streams

Packet losses in a VoIP system can be experienced in two ways. Firstly, the interactive nature of VoIP implies that the packet must arrive within a determined delay budget. If the packet arrives beyond that delay, it will be discarded, as it will arrive too late to be useful. Secondly, the routers drop packets during periods of intense congestion. In Section 3.2.2, we explained the operation of routers and offered reasons why packet losses are experienced when this one is congested. An

important correlation [42][2] was made between the bandwidth used and the amount of losses experienced. It suggests that *packet losses are temporally correlated and occur consecutively (in bursts) rather than randomly*. Since losses are generally due to router buffer overflow, sending packets more frequently during that time will lead to more consecutively loss packets, or equivalently, a higher probability of losing the next packet, which is most likely inevitable for a real-time streaming application.

4.3 EFFECTS OF NETWORK ON VOIP

VoIP systems can use several network devices to exchange voice information through packet networks. In the case of Figure 4.2, two PCs are used for that purpose, where several nodes in the packet network need to be traversed. Packet networks working on a best effort delivery model for streaming data would transmit each encoded speech frame to the receiving end by traversing the network from node to node until reaching destination. As discussed before, there is no guarantee that packets always traverse the same set of nodes during a session, thus presenting a possibility for the packets to be received unordered.

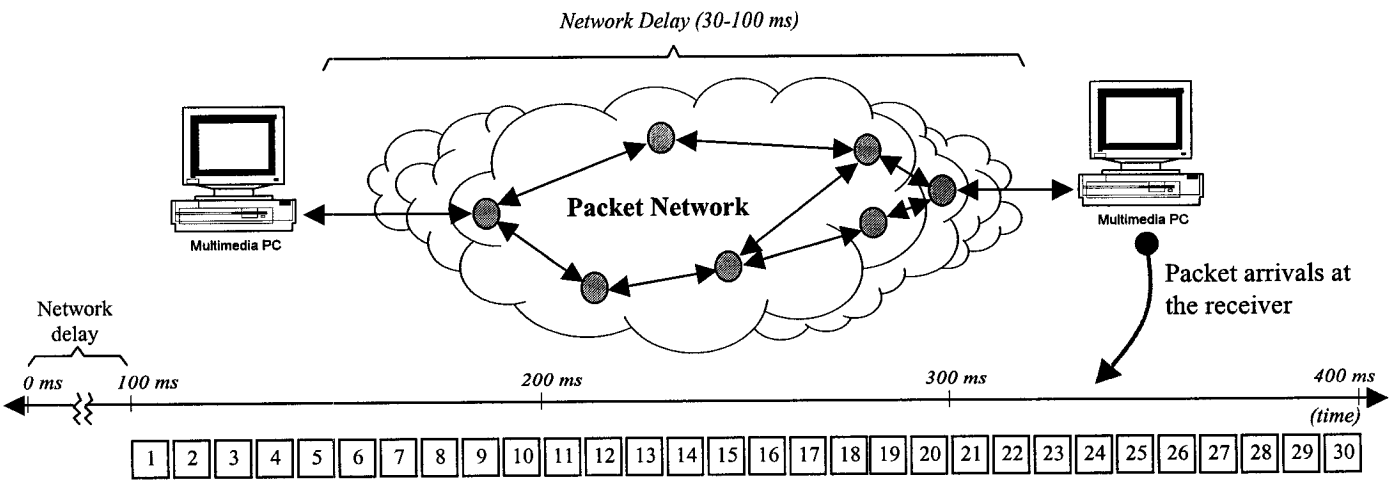


Figure 4.2 – PC-to-PC communication via packet network

A more important problem for VoIP relates to network congestion that is causing packets to be dropped or induces considerable delays to the packets in transit. Those delays may lead speech frames to arrive too late to be of any use for real-time applications. The real effect is that the packets or more precisely the coded speech frames, are considered lost in such instances.

As discussed earlier, the average network delay associated with the Internet is approximately between 30 ms and 100 ms [47]. Figure 4.2 demonstrates the effect of the packet network on the real-time stream. The top part of the figure represents a generic VoIP exchange between two PCs over a packet network. The cloud represents the packet network where several nodes may be traversed by the coded speech streams before arriving to destination. The time from when a coded speech frame is packetized and transmitted to the packet network until it reaches the destination PC where it will be depacketized is assumed to be 100 ms in this example. This implies that the first transmitted packet will reach the destination after a delay of 100 ms. This is depicted by the graph in the lower half of the figure where numbered boxes represent the arrival of transmitted packets at the receiver. What is more important is that for real-time streams, packets will be transmitted one after another implying some packet buffering throughout the packet network. Assuming a packet leaves the transmitter every 10 ms would imply 10 packets would be transiting[12] the packet network after 100 ms has elapsed. Each packet is exposed to transiting the network using a different path depending on network latency or congestions and the timing of the routing information exchange (e.g. updating of routing tables as discussed in Chapter 3) between routers. For example, a congestion in the lower path of Figure 4.2 may trap packets 3,4,5,and 6, just before a routing information exchange updates the cost of the paths, hence, directing packets 7, 8, 9, ..., to borrow the

[12] Ignoring other delays - the figure shows an optimistic idealization where constant delays throughout the network is observed. This is not representative of the real world but demonstrativeof a worse case scenario – it is assumed that a lower number of packets would be present in the network under real world conditions and that inter-packet delay (a.k.a. jitter delay) would be present and variable.

higher path of Figure 4.2. This would lead to the reception of an out of order sequence at the receiver (e.g. reception order 3, 7, 8, 4, 9, ..., which is one of the ordering possibilities). In this thesis, we assume the receiver re-orders the packets effectively and manages jitter delays, both within marginal delays. The focus will be with respect to packet losses, single losses as well as a series of consecutive losses that will be referred to as an error burst.

4.3.1 EFFECT OF PACKET LOSS

Figure 4.3 shows a voice segment lasting over 160 ms. The ITU-T G.729A algorithm is used to compress and decompress the e_f01s04.wav test speech file with no packet lost. The figure shows the 16-bit linear PCM encoded source stream (a) and the ITU-T G.729A reconstructed counterpart (b) where a voiced segment starting at speech frame 128 demonstrates the original operation of the algorithm under no packet lost. Of interest in this figure is the third plot that represents the Squared Error (SE) between the source and reconstructed signal (c). We can observe that even in periods of no packet lost, the SE between the original and reconstructed signals is apparent. The ITU-T G.729A algorithm itself causes this reconstruction error where speech synthesis or speech signal reconstruction relies heavily on minimization loops that iteratively select a best estimation of the excitation signal out of a limited set of vectors contained in codebooks.

Lets now demonstrate the effect of packet loss through the following example. The insertion of 3 frames lost (error burst of size 3) from the 10 ms marker is represented in Figure 4.4 where the associated reconstructed signal (a) and squared error (b) are shown. The study of the time-domain representation of the reconstructed signal reveals obvious signal distortion when compared to the reconstructed signal of Figure 4.3.(b). This distortion was consistently observed and its effect was also perceivable by the ear during the playback of the various speech test files.

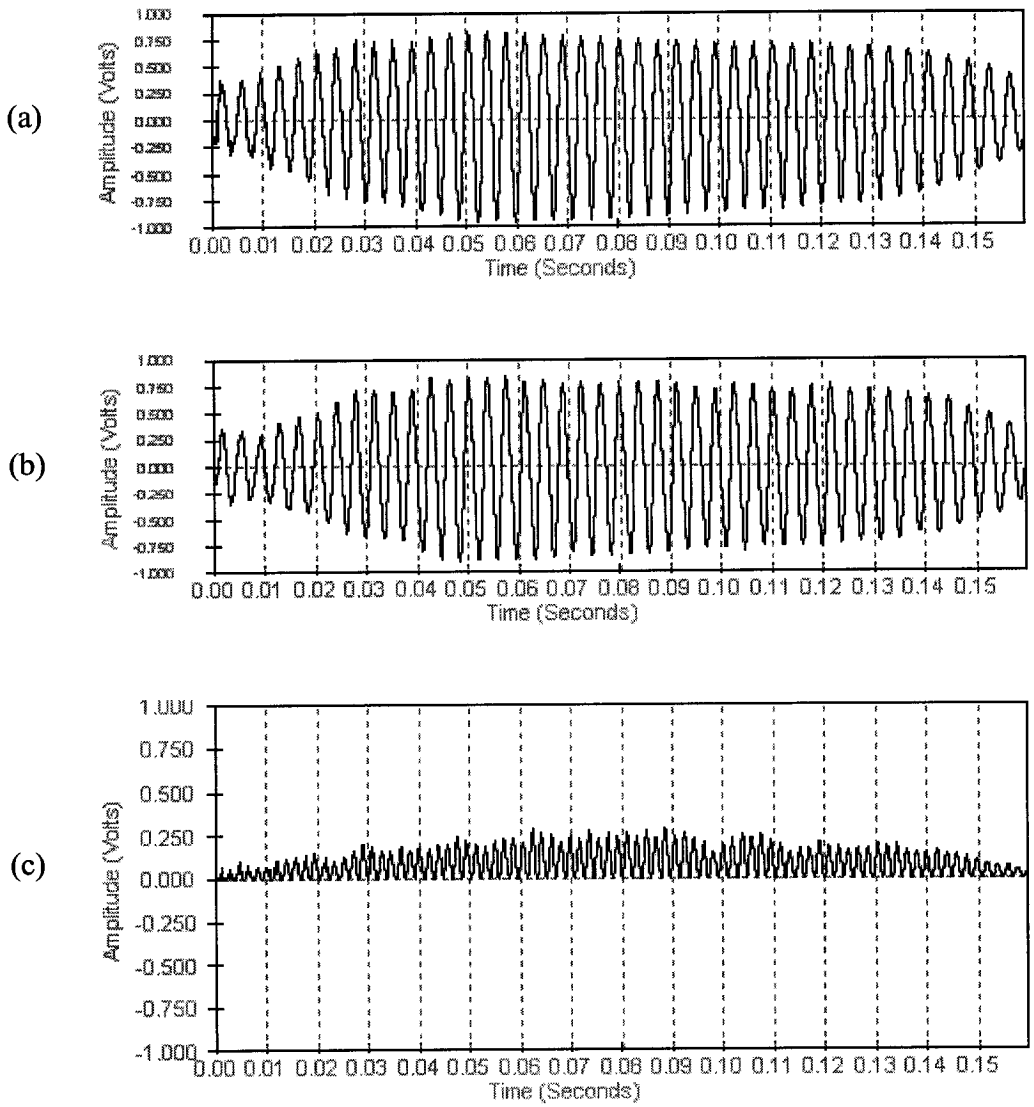
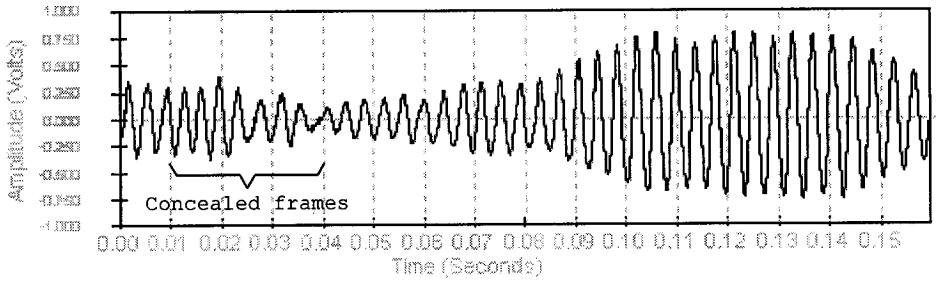


Figure 4.3 – ITU-T G.729A signal reconstruction with no frame loss
(a) original, (b) reconstructed, and (c) the squared error

The reconstructed signal of Figure 4.4 shows the concealed speech frames at 10, 20, and 30 ms. The ITU-T G.729A embedded Packet Loss Concealment (PLC) algorithm was used in the reconstruction of the speech signal of Figure 4.4.(a). Concealment methods will be described in Section 4.4.3 (however, some introduction to the method is required at this time). The first frame being concealed is basically a repeat of the last good frame. Subsequent concealment will resume at scaling down excitation signal gains and computing replacement values that estimate the missing speech frame.

(a)



(b)

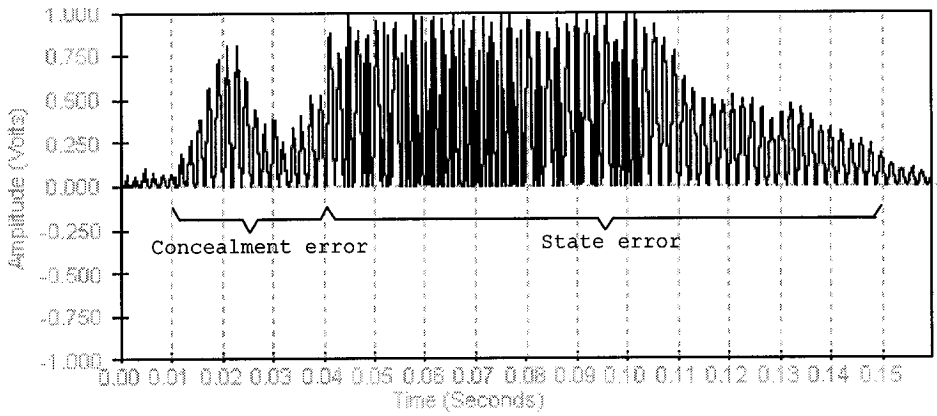


Figure 4.4 – ITU-T G.729A after a 3 packet error burst at the 129th frame
(a) reconstructed signal, and (b) squared error

Part (b) of the figure shows the SE between the original and the reconstructed speech segment. Two different errors are associated with the figure, namely, the concealment error and the memory state error. The concealment error is the result of the generation of a best estimate to replace the missing speech frame and the memory state error is the result of memory state de-synchronization between the encoder and decoder. The memory state error follows the replacement or substitution of the first bad packet. This type of error has been identified in literature [70][59] as being the difference between memory states or amount of de-synchronization. Observations were made where lost packets were inserted in groups of one packet loss up to six consecutive packet losses. Above burst of six losses, it was obvious that the reconstructed signal was overwhelmed with distortion and provided very poor speech quality. Additional to hearing the effect of packet losses, we observed the effect of losses in the time-domain through a set of plots as provided by Figure 4.3 and Figure 4.4. Our initial observation was that the effect of packet loss is signal dependent. As described in Section

2.2, speech signals are categorized as voiced, unvoiced, or silence segments typically exhibiting high, moderate, and low energy characteristics respectively. Voiced signals are further characterized as being quasi-periodic over short periods while unvoiced signals tend to be characterized by noise-like signals, but both contributing to forming an overall energy envelope in the time-domain as well as the frequency domain. For example, an error burst over a silence segment may have no noticeable impact on the overall quality of the reproduced speech segment. However, voiced and unvoiced segments are severely impacted by such error bursts. Single error bursts are hardly noticeable to the ear. However, the degradation in performance becomes noticeable as the error burst size increases. This effect correlates with the literature where several authors [57][20][6][70][79] identified the limitation of the G.729A PLC algorithm to be robust to single frame loss. From a visualization of the time-domain point of view, the distortion generated by any size error burst is easily noticeable and worse as the length of the error burst increases. As the stream is affected by error burst, we can notice a decrease in the values of the PESQ measurements (PESQ was described in Section 2.6.3), which is equivalently perceived by the ear.

Other observations were made where error bursts were inserted in the middle of voiced and unvoiced segments, at the ramp-up or transition toward phonemes as well as the ramp-down or transition toward silence. An error burst at the beginning of a transition seemed to generate the worst degradation of all. Error bursts in middle transitions (i.e. in the transition from one phoneme toward another, associated to a change in pitch) were easily noticeable where the ear seemed to be able to repair the loss in the middle of voiced/unvoiced transition. Finally, transitions toward silence seemed to be the least affected segments since concealment at the tail of a phoneme is fading subsequent speech frames anyway. The following section will present mechanisms to repair packet losses effect in the reconstruction process of speech segments.

4.4 PACKET LOSS RECOVERY AND CONCEALMENT METHODS

4.4.1 OVERVIEW

As discussed previously, it is very likely that some encoded speech packets will be dropped within the network. When one or more network packets containing phonemes are lost, the perceived quality of the speech maybe severely degraded. It is reported [61][87] that the human brain is capable of reconstructing a few lost phonemes in speech but that too many missing packets makes a voice unintelligible. Consequently, a speech codec must provide robustness to lost packets to be useful.

In this section we will familiarize ourselves with some of the methods used to repair the missing speech segments as an outcome to packet losses. The repair methods are grouped under three main classes, namely, sender-based repair, receiver-based repair, and sender/receiver-based repair. Sender-based and receiver-based methods can be used to complement each other, in which case they will be referred to as sender/receiver-based [61]. This last method aims to achieve the best possible trade-off between the speech quality performance and the conservation of bandwidth.
Sender-based methods typically add redundant information to the transmitted information, thereof increases the required bandwidth (equivalently increases the operating bit rate). The receiver-based method or concealment technique of the ITU-T G.729A algorithm is of interest and will be discussed with receiver-based methods. Concealment techniques attempt to reconstruct an estimate of the missing packet from previous information contained in the decoder. In other words, the sender does not assist the receiver when repairing or concealing the packet loss. Contrary to sender-based techniques, receiver-based will typically produce degradation in the speech quality but will not require additional bandwidth.

4.4.2 SENDER-BASED RECOVERY METHODS

In general this class of repair methods attempts to make provisions for the event where a packet would not reach destination. Closed-loop and open-loop mechanisms [60], respectively corresponding to *Active* and *Passive* techniques used to transmit or retransmit redundant information that will assist repairing the lost packet at the receiving end. While *Retransmission* is the only known active technique, passive techniques use *Interleaving* and *Forward Error Correction (FEC)*.

Open-loop techniques are the most commonly used whereas closed-loop methods are not well adapted to interactive sessions due to the latency of the packet network. As discussed in Chapter 2, the main focus of compression algorithms is to remove as much redundant information as possible from the speech signal so we can compress it more effectively. Consequently, sender-based methods inhenrently conflict with the compression objectives. While compression attempts to remove redundancy in the stream, sender-based recovery methods intentionnally add redundancy to increase robustness to packet losses.

Active Sender-based Techniques

Active sender-based techniques are in fact closed-loop mechanism where the receiver is requesting assistance from the transmitter following the detection of missing information. In the case of packet networks, the missing information refers to one or more lost packets where each may contain one or more encoded speech frames[13]. Such mechanism is implicit when using TCP/IP but when using UPD/IP, like in the case of IP telephony, the mechanism has to be implemented at the session or application layer [80]. For example, if applied to the presentation/session layer using RTP/RTCP, then a RTCP message may be transmitted from the receiver to the transmitter requesting information

[13] In this context, we are referring to distinct speech frames.

retransmission. The transmitter then acknowledges the request by retransmitting the information to the receiver. The amount of network latency usually plays a critical role in such methods. It may cause the retransmitted packet to arrive too late at the receiver. Additionally, the bit rate requirement will slightly increase for each retransmission request. As the name suggests, the *retransmission* technique is the actual retransmission of the information in order to effectively recover from the lost information (e.g. missing speech frames). Retransmission is very popular for non-interactive applications because it will ensure information completeness at destination [60]. However, when real-time exchange of information is necessary, such as in VoIP systems, retransmission is not a practical solution due to the long and variable network delay. A retransmission request must first reach the sender for the sender to resend the encoded speech frame. For this method to be effective in a real-time mode would require low network latency and some assurance that the retransmission (i.e. added bandwidth requirement) would not worsen the network response by adding traffic to an already congested node.

4.4.2.1 PASSIVE SENDER-BASED TECHNIQUES

Passive sender-based techniques are open-loop mechanisms between the transmitter and the receiver. These repair techniques are initiated from the transmitter where redundant information is sent to the receiver to allow for recovery of a packet in the eventuality of a packet loss. This technique does not require any signaling between the transmitter and receiver since a redundant copy (or copies) of the information is regularly sent by the transmitter. However, these methods have a substantial impact on the bit rate and/or delay required to operate. Two techniques are presented, namely, the Interleaving technique and the Forward Error Correction (FEC) techniques.

Interleaving reduces the effect of packet losses by reordering or sequencing the frames before transmission. The aim is to segregate the effect of the loss. To do so, the method waits for several

frames and then interleaves them (e.g. an ordered shuffling of the packets) before transmission. This method is not adequate for VoIP because it requires the buffering of several frames at a time that induces additional delay. However, it is well suited for non-interactive applications since it is a robust method that has the advantage of not increasing the bandwidth [60]. The principle in FEC is to send additional repair data with the stream. Two methods are applied, namely, *media-independent FEC* and *media-dependent FEC*. The first method does not discriminate the content to add the repair data, media-independent FEC methods are based on the traditional bit errors correction principles [69][63][80][64][59][62][42] used in channel coding, such as parity coding and Reed-Solomon coding currently proposed as a RTP payload format [89]. Rather than applying the technique to bits it is applied to frames. These methods are not as prohibitive as duplicating each packet and effectively increase the packet delivery reliability. The second method takes advantage of the source content knowledge to improve the repair process. In this FEC approach, the amount of redundancy and the block size determines the robustness to losses, but introduces a delay that is dependent on the block size. For the purpose of bandwidth conservation, the amount of FEC redundancy should be limited and the selected block size small enough to minimize the delay. In FEC, we use a simple notation to describe the parity scheme used. FEC parity code *(n,1)* implies a block size of *n* packets and redundancy of *1/(n-1)*. For example, a block of 2 packets containing a redundant packet would imply 100% redundancy. The FEC redundancy is usually piggybacked. A different form of FEC is low-bit rate redundancy. It works by transmitting a lower bit rate version of the same speech signal, but piggybacked to later packets. If a packet with the main speech codec is lost, the lost speech information is substituted with a lower quality version. It can be viewed as a form of loss concealment, which replaces the lost waveform with an approximation. Typically, the redundant bit-stream should not be coded at a higher bit rate than the original encoded frame (may worsen the congestion problem).

4.4.3 RECEIVER-BASED CONCEALMENT METHODS

Receiver-based concealment (a.k.a. frame-error-concealment) methods are basically attempting to mitigate the effect of packet losses by constructing an estimate of the missing packet using the previous information stored at the receiver. Concealment performance is signal dependent and will rely to a great extent to known signal information before the frame loss occurs. Concealment techniques are grouped under three main families, namely, the *Insertion* methods, the *Interpolation* methods, and the *Regenerative* methods. They range in complexity from very simple to complex respectively. The *Insertion* methods could be characterized as being the easiest to implement. *Interpolation* methods are more complex but also provide better speech performance. Finally, the best speech quality performance is associated with the *Regenerative* methods. The later methods often come at the cost of increased complexity.

4.4.3.1 INSERTION BASED TECHNIQUES

Insertion-based techniques present the advantage of being simple to implement and offer good results for low loss scenario (typically below 2 % lost) and for short frame length. *Splicing* hides the loss by removing the time space associated with the missing frame. In other words, if frame *n* is lost, the method plays frame *n+1* immediately after frame *n-1,* ignoring that frame *n* ever existed. An alternative to the splicing method is *silence substitution* where the gap created due to the lost packet is filled with silence (e.g. no energy – replacing the loss packet with zero-amplitude samples [56][61][14]) thus keeping the timing relationship of the transmitted stream. It is reported [61] that silence substitution can be effectively applied when speech frame length are short (< 4ms) and the frame loss rate is low (< 1%). While it allows the destination to continue play the stream without any timing disruptions, the perceived speech quality is deteriorated by the presence of choppiness that creates unnatural effects. The adequacy of this approach lies primarily in the simplicity of

implementation but its viability is easily surpassed in the case where packets contain speech frames larger than 16 ms or when loss rates reaches 1% [56][61][14]. *Noise substitution* fills the gaps with some background noise (i.e. white noise). This noise insertion method is often referred to as comfort noise generation. Perkins and al. [61] are reporting that studies of the human perception of interrupted speech have been conducted and that contrary to silence substitution, the human brain tends to better conceal the missing speech segments when noise substitution is used. Consequently, intelligibility improvements are achieved when noise substitution is used. This approach is relatively simple and requires few operations where random samples with energy equivalent to the averaged energy of the previous speech frames are substituted. More elaborate approaches could also be implemented where blending or smoothing of the comfort noise is used to minimize high frequency tonal artifacts such as hisses. *Frame Repetition* method simply replaces the missing frame with the previous one. Frame repetition provides the best speech quality results of all insertion based techniques but also necessitates additional memory to store the previous frame. The complexity is lower than for the noise substitution method. The Global System for Mobile communications (GSM) uses such an approach to conceal loss speech frames where the first missing speech frame is replaced by repeating last known good speech frame and subsequent missing frames are estimated by scaling down the first estimate of the burst.

The use of repetition with fading is a good tradeoff between the poorly performing insertion-based concealment techniques and the more complex interpolation-based and regenerative-based concealment methods.

4.4.3.2 INTERPOLATION BASED TECHNIQUES

A number of frame erasure concealment techniques attempt to build an estimate of the missing speech frame by interpolating the good frames surrounding the missing frame. These methods are usually more complex than the insertion methods but have the advantage of considering the changing characteristics of the signal. These methods are known to perform better than the insertion techniques presented earlier [14] but are still subject to acute distortion when the loss duration is to long (e.g. more than the phonemes length, typically 32 ms) or when the loss occurs during a speech segment which represent a speech transition.

Goodman et al. [14] studied the use of *waveform substitution using a pattern matching approach and a pitch waveform replication approach*. The first approach searches the previously received frames for a similar pattern. The second approach uses a pitch pattern template to locate the most similar or suitable segment to substitute the lost frame. The pitch detection approach demonstrated marginally better results than the waveform substitution. Other proposed pitch detection methods [21][14] are in use, namely, the later, the ITU-T G.711 appendix I where the synthesized replacement signal does not replace the whole frame at once but rather extract the last pitch period. The pitch period is then concatenated one after the other for the duration of the loss.

These methods generally perform one-sided or two-sided Overlap-and-Add (OLA) at the boundaries of the missing frame or gap, where the one sided simply repeats the pattern across the gap and the two-sided interpolates the gap from both sides. The two-sided OLA usually offers better artifacts reduction [14]. Furthermore, a two-sided approach [88] performing interpolation using the previous and next correctly received packet is reported to perceptually improve the quality of speech.

4.4.3.3 REGENERATIVE BASED TECHNIQUES

Regenerative repair techniques rely on the knowledge of the codec algorithm such that a replacement of the lost frame can be synthesized. Two regenerative techniques are presented in this section. The first technique relies on past values and memory state information to synthesize an estimate of the missing frame and is called *Interpolation of transmitted state.* The second technique is called *Model-based Recovery* and this one relies on previously synthesized samples to maintain a model-based algorithm that is used to synthesize an estimate of the missing frame. Both methods are computationally intensive but provide the best results of all concealment methods presented.

The ITU-T G.723.1[27] and the ITU-T G.729 [30][31] codecs both offer a PLC technique of the *interpolation of transmitted state* category. We will explain the G.729A PLC technique as it is of interest in this thesis. The G.729A embedded PLC algorithm assists the decoder repair the missing speech frame from the information it maintained while decoding previously received good speech frames. The algorithm interpolates the last known good values (from parameters) with the decoder maintained memory state values to synthesize an estimate and subsequently update the decoder memory state. The algorithm synthesizes the first frame estimate by interpolating a repeat of the previous values. Any consecutive synthesis to the first missing frame will extract the previous LSP, will scale down the previous signal amplitude and will slightly modify the pitch before performing the interpolation that generates the replacement frames. It is hoped that this algorithm produces relevant estimates for a duration of 5-6 frames after which the repeated scaling of the values eventually forces the values toward zero, hence, generating less relevant frames (i.e. zeroed frames). The advantage of this technique is that there is no boundary artifact because the interpolation ensures a smooth transition. Furthermore, since the decoder uses the same building blocs to synthesize speech frames during its normal operation, the processing complexity remains approximately constant.

Contrary to the interpolation techniques, regenerative techniques generally offers better artifacts reduction at the boundaries of the missing frame because interpolation is performed with the basis values of the regenerative model.

In *Model-Based Recovery*, the codec is assisted by another model-based codec that is embedded on the decoder side of the main codec. Recently, annex A to the ANSI T1.521a-2000 [83] proposed and algorithm based on this approach ([15] proposes a similar algorithm). This PLC algorithm resides on the decoder side only and is based on the Linear Prediction model of the human speech production system (discussed in Section 2) that is widely used for low bit rate codecs. The technique typically *operates on the last received set of samples* to extract the vocal tract characteristics and associated excitation signal used to provide an estimate of the lost frame. In other words, the PLC algorithm is composed of an analysis and a synthesis module contained at the decoder and that are used when the synthesis of an estimate to the missing frame is required. When burst losses occurs, the algorithms generally repeat the first missing frame and gradually fades the values for subsequent concealment. However, a substantial increase in processing complexity is required as well as an additional algorithmic delay (caused by the independent manner in which the algorithm is implemented).

4.4.4 SENDER/RECEIVER-BASED METHODS

In receiver-based methods, the receiver had to repair the missing packet the best it could from the knowledge it had from the previous packet. As discussed, this method works well as long as the missing frame is similar to its predecessor. Sender/receiver based methods are reported [73][2] to be more effective because the sender can inform the receiver of some of the attributes of the missing packet, thus allowing for a better estimate.

Many sender and receiver based repair methods are commonly known and many more are being pursued to enhance the robustness of codecs to packet losses. For example, a method named Speech Property-Based FEC (SPB-FEC) [73] seeks to discriminate between relevant and not as relevant speech frames to support the repair methodology. The algorithm provides network resources economy by only sending FEC redundancy to protect more relevant speech frames and letting the decoder concealment algorithm handle the recovery of less relevant speech frames. Despite the bandwidth saving compared with traditional FEC methods, the redundant data still contributes to network congestion. This is a recurring problem that can potentially be solved by allowing the receiver to send information to the sender with respect to the current session. This means the receiver could inform the sender about loss conditions and allow the sender to adapt the amount of redundancy accordingly. For example, a legacy approach to prevent losses, the Packetized Voice Protocol (PVP) or ITU-T G.764 introduced in [1], is able to relieve congestion by modifying the size of the speech packet. The algorithm allows for the discarding of lower priority bits or less sensitive bits rather than facing a congested network device that will be forced to drop those specific packets. In Chapter 5, we propose a closed-loop algorithm that could be considered a quasi-sender/receiver-based recovery method since it aims at correcting the error rather than the speech signal itself.

4.5 ITU-T G.729A ALGORITHM BEHAVIOR UNDER PACKET LOSS CONDITIONS

4.5.1 OVERVIEW

In Section 2.7 we covered the operation of the ITU-T G729A recommendation. We described the encoder and decoder operation in some detail without considering the effect of missing frames as described in Section 4.3.1. It was also mentioned earlier that packet losses are inevitable in VoIP

systems or more precisely in IP networks. For that reason, the G.729 PLC algorithm was just introduced in Section 4.4.3 where its operation was explained.

In this section, we will investigated the state error to understand its behavior during the lost of packets. We will first describe the G.729A memory used in both the encoder and decoder in order to faithfully reconstruct the signal at destination. We will then provide a new notation that will assist us to understand the behavior of the memory state in periods of no error as well as during periods of errors. Then, we will investigate the convergence of the state error, that is, the time it takes the decoder memory state to return to a synchronized memory state with the encoder.

4.5.2 MEMORY STATE DESCRIPTION

We need to understand the memory state behavior if we want to minimize its propagation. As discussed in Section 2.7.3, state memory is present in both the encoder and decoder. They fundamentally fulfill the same set of functions, that is, to synthesize a speech signal estimate. The difference only lies in the encoder where the ABS mechanism calls for many iterations to be performed for the generation of parameters to be transmitted to the decoder (the state memory is left unchanged from iteration to iteration – it is only updated once the speech frame parameters have been determined). In the decoder, the memory state and the decoded parameters are used to synthesize the speech estimate in a single iteration. The G.729A memory state structures have been identified in literature [70][58]. In this section, we will build on and emphasize the description of these memory state structures when possible. In this thesis, we will refer to each independent memory state structures forming the overall memory state, as *memory state elements*. The five memory state elements are:

a) The LSF 4th order MA predictor for the reconstruction of the short-term filter coefficients (the variable representing memory a is m^a):

- This memory structure holds the last 4 vectors of the Line Spectral Pairs (LSP) 4th order MA predictor.
- Each vector represents 10 LSP (each LSP value is represented by a 16 bit word – 10 X 16-bit = 160 X 4 = 640-bit).
- At start up, the LSF vector values are set to $v_i = i\pi/11$ for $i = 1,\ldots,10$ and each vector is scaled by fixed MA prediction coefficients held in a 4 X 10 matrix of coefficients.
- The 4th order MA predictor is updated following the decoding of a good frame received as well as after a missing frame is concealed.

b) The past excitation signal (the variable representing memory b is m^b):

- Represents the past 154 16-bit samples (154 X 16-bit = 2464 bits) of the overall excitation.
- At start up, all values are initialized to zeros.
- Resides in past values of overall excitation buffer.
- These are used to form the adaptive codebook vector.
- Is updated on a frame basis following the decoding of a good frame as well as after a missing frame is concealed.

c) The synthesis filter coefficients (10th order) (the variable representing memory c is m^c):

- This memory structure holds the 10th order all-pole LPC (synthesis filter).
- Represents the past 10 16-bit coefficients of the synthesis filter (10 X 16-bit = 160-bit).
- At start up, all values are initialized to zeros.
- This past vector is used to interpolate with the LSP vector of the next frame. More precisely, the interpolation to form the next LPC for the 1st subframe.
- Is updated on a sub-frame basis following the decoding of a good frame as well as after a missing frame is concealed.

d) The fixed codebook energies for the past four frames, which are used to predict the fixed codebook gain (the variable representing memory d is m^d):

- Represents the past 4 16-bit inputs to the predictor (4 X 16-bit = 64-bit).
- Resides in memory of the 4th order MA predictor used to predict the value of the fixed codebook gain.
- Is updated on a sub-frame basis following the decoding of a good frame as well as following the concealment of a missing frame.
- At startup, the gains are set to minimum gain value (e.g. –14 db) and each is scaled by fixed MA prediction coefficient, 0.68, 0.58, 0.34, and 0.19 respectively.

e) The adaptive codebook gain from the previous frame, which is used to generate the harmonic filter used on the fixed codebook excitation (the variable representing memory e is m^e):

- A single 16-bit word updated on a frame basis.
- This gain is chosen to be the adaptive codebook gain of the previous frame.
- At startup, the pitch sharpening value is set to its minimum (e.g. 0.2.).

Figure 4.5 represents the encoder (left) and decoder (right) showing the interaction between their components in an encoding/decoding cycle. The ABS process is shown in the transmitter side where this one uses the current memory state information to decide on a set of parameters that will best represent the source speech signal. The parameter labels in the figure associate with the same labels from the description of these parameters in Table 2.6 and the memory state elements were recently identified (also see Table 4.1).

By default, every frame coming out of the encoder is a good frame. During normal operation, the receiver will decode the received packet. The values thus acquired, along with memory state values are used to synthesize the speech frame before playback.

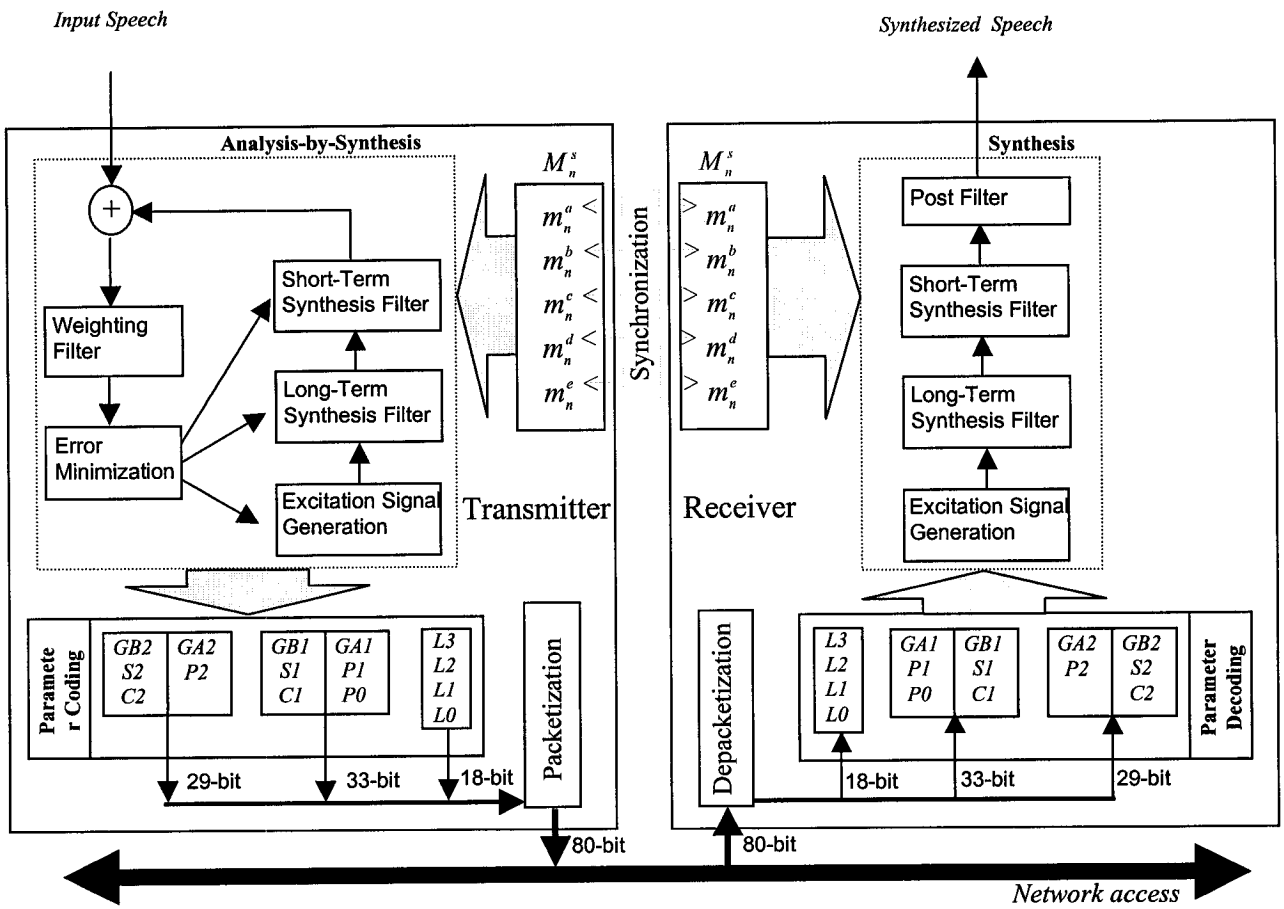


Figure 4.5 –Memory state synchronization concept in the G.729A codec

As discussed before, encoded speech frames may be lost or may be delayed beyond the maximum allowable delay (e.g. packet loss). In all packet loss scenarios, memory state between the encoder and decoder will become automatically out of synchronization as will be explained in Section 4.5.3. The time required for the algorithm to naturally converge back to a synchronized memory state between the encoder and decoder is dependant on the source information being compressed and the duration of the loss. Section 4.5.4 provides more details with respect to the meaning of the memory state elements and their convergence. The remainder of this chapter will rely on Figure 4.5 and will assist us to understand the effect of packet losses on the memory state.

4.5.3 MEMORY STATE EFFECTS

The memory state effects can be better explained through example scenarios. For example, Figure 4.6 shows a sequence of 16 speech frames being processed by the encoder (from left to right), transmitted and received, with no losses in this case, for processing by the decoder (from top to bottom). This process is repeated for each column while each row is representative of a process domain with respect to time (e.g. on a frame-by-frame basis). The leftmost column of the figure categorizes the rows as belonging to an encoder or decoder process or more precisely a transmitter and receiver process respectively.

Encoder	Source signal	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
	State memory	M_0^0	M_1^0	M_2^0	M_3^0	M_4^0	M_5^0	M_6^0	M_7^0	M_8^0	M_9^0	M_{10}^0	M_{11}^0	M_{12}^0	M_{13}^0	M_{14}^0	M_{15}^0
	Parameters	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	Xmit/Rcv	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Decoder	Parameters	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	State memory	M_0^0	M_1^0	M_2^0	M_3^0	M_4^0	M_5^0	M_6^0	M_7^0	M_8^0	M_9^0	M_{10}^0	M_{11}^0	M_{12}^0	M_{13}^0	M_{14}^0	M_{15}^0
	Reconstructed signal	R_0^G	R_1^G	R_2^G	R_3^G	R_4^G	R_5^G	R_6^G	R_7^G	R_8^G	R_9^G	R_{10}^G	R_{11}^G	R_{12}^G	R_{13}^G	R_{14}^G	R_{15}^G

Figure 4.6 - Memory state effects under no packet loss

The variables of interest associated to the different processes are described in the table below. The variables are supported by the following set of indexes; n: positive integer value, s: positive integer value, and c: positive integer value.

Variable	Description
S_n	Represents source speech frame n (16-bit linear PCM).
M_n^s	Represents the internal memory state at the encoder and decoder associated to the generation of speech frame n. The superscript s indicates the memory state domain to which the memory state belongs.
P_n	Represents the encoder and decoder parameters for speech frame n.
$\downarrow$	Implies the parameters were transmitted from the encoder and successfully received by the decoder.
V_n^c	Represent substituted values during a lost packet n. Generated values from previously known good parameters are first repeated, then interpolated, or scaled to support the synthesis of speech frame n and assist generate follow on memory state $\overline{M}_{n+1}^s$. The superscript c represents the index of consecutively concealed frame from the same previously known good values.
$\overline{M}_n^s$	Represents the internal memory state at the decoder for the reconstruction of speech frame n. This memory state was created using substituted values from previous known good values. It is considered a memory state error that grows, as consecutive packets are lost. Superscript s represents the PLC iteration from which the memory state was updated.
$\tilde{M}_n^s$	Represents decoder internal memory state generated after good parameters are received following a period of packet lost. However, the encoder and decoder memory states are still desynchronized but are most likely converging to a synchronized state. Superscript s represents the same contiguous memory state error domain.
R_n^T	Represent the reconstructed (16-bit linear PCM) speech frame n. The superscript T indicates the type of reconstruction, namely, from a Good, a Concealed, or State error affected set of values, represented by the letters G, C, or S respectively.
E_n^T	Represent the reconstruction error for speech frame n. The superscript T indicates the type of reconstruction error, namely, from a Good, a Concealed, or State error affected synthesis, represented by the letters G, C, or S respectively.

Table 4.1 - Description of variables

Let's go through this example in detail. The variables subscripts always represent the frame number within that sequence. The coding of the first speech frame S_0 using initial memory state M_0 generated the associated parameters P_0 and memory state M_1. While memory state M_1 will be used to generate the next set of parameters for the following speech frame, the parameters P_0 on the other hand, are transmitted to the receiving end of the communication system. At the decoder, the decoded parameters P_0 are applied to the system with memory state M_0 (identical to that of the encoder) to synthesize a reproduction of S_0, that is R_0^G, the reconstructed signal. Under no packet losses, the encoder and decoder memory states remain synchronized and provide the desired result. It is important to note that the ITU G.729A algorithm is in fact a lossy compression technique and

inherently generates a reconstruction error, E_n^G. As covered before, the CS-ACELP algorithm compresses speech frames using a human speech production system model. Multiple ABS decisions through open and closed loops are made to compress the signal using sets of adapted and fixed vectors from different codebooks and associated gains. Therefore, many combinations may take place for the generation of speech frame reproductions that make the distortion or error difficult to track mathematically. However, the error is easily traceable from a black box perspective where $E_n^G = S_n - R_n^G$. This natural or algorithm error provides a basis against which we can compare the codec performance. In this case we want to be able to correlate this error with the perceived quality of the reconstructed speech. As discussed in the Section 2.6, several measures or estimators can be used to assess the speech quality.

Next, we consider an example where error bursts of size one (single packet lost at a time) are introduced (Figure 4.7). In this example the standard ITU-T G.729A PLC algorithm is subsequently activated following the detection of the packet loss. The first two packets are correctly received and decoded at the decoder as explained in the first example. However, the packet containing parameters P_2 does not reach the decoder and forces the decoder PLC algorithm to synthesize a replacement speech frame R_2^C from associated state memory M_2^0 and predicted or interpolated values V_2^1 obtained from repeating previous values from previous speech frame synthesis.

Encoder	Source signal	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
	State memory	M_0^0	M_1^0	M_2^0	M_3^0	M_4^0	M_5^0	M_6^0	M_7^0	M_8^0	M_9^0	M_{10}^0	M_{11}^0	M_{12}^0	M_{13}^0	M_{14}^0	M_{15}^0
	Parameters	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	Xmit/Rcv	↓	↓		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Decoder	Parameters	P_0	P_1	V_2^1	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	State memory	M_0^0	M_1^0	M_2^0	$\overline{M}_3^1$	$\tilde{M}_4^0$	$\tilde{M}_5^0$	$\tilde{M}_6^0$	$\tilde{M}_7^0$	$\tilde{M}_8^0$	$\tilde{M}_9^0$	$\tilde{M}_{10}^0$	$\tilde{M}_{11}^0$	$\tilde{M}_{12}^0$	$\tilde{M}_{13}^0$	$\tilde{M}_{14}^0$	$\tilde{M}_{15}^0$
	Reconstructed signal	R_0^G	R_1^G	R_2^C	R_3^S	R_4^S	R_5^S	R_6^S	R_7^S	R_8^S	R_9^S	R_{10}^S	R_{11}^S	R_{12}^S	R_{13}^S	R_{14}^S	R_{15}^S

Figure 4.7 - Memory state effects following a single packet loss

The past values or last correctly decoded values are the previous LSP coefficients, the previous pitch delay value, the adaptive (pitch) and fixed (innovative) codebook gains, and the previous synthesis filter coefficients. The M_2^0 memory state comes from the synthesis of the previous frame and is equivalent to the encoder memory state. However, the memory state being constructed for the next frame will start drifting away from its sibling encoder memory state values. This is represented by variable $\overline{M}_3^1$. In this case, we refer to the difference between memory states M_3^1 and $\overline{M}_3^1$, at the encoder and decoder respectively, as the *drift distance* or memory state error. In fact the drift distance is an error indicating how close or how far astray memory state $\overline{M}_3^1$ is from the memory state M_3^1 that would have been generated would parameters P_2 been correctly received. Similarly, another drift distance is attributable to the concealment or predicted values V_2^1 replacing the values that would have been generated would parameters P_2 been decoded and is referred to as the concealment error. In the example of Figure 4.7, the generated reconstruction R_2^C is assumed to proceed with an erroneous set of values V_2^1 but a correct memory state M_2^0. The associated concealment error is represented by $E_2^C = S_2 - R_2^C$. From this point the reader not accustomed to compression techniques embedding memory state would assume the system would revert to normal operation but it is not the case. Subsequent packets are well received and decoded but it turns out synthesis of speech frames is dependent on the memory state. Since the memory state is incorrect (i.e. not synchronized with the encoder memory state), the synthesized speech frames are also incorrect. As the decoder memory states converge toward the memory states of the encoder, the drift distance affecting the synthesized speech frames will also minimize and eventually produce the desired reconstruction.

The following speech frame S_3 is encoded using memory state M_3^0 to generate encoder parameters P_3. This time, the parameters are transmitted and received at the decoder where they are correctly decoded. However, memory state $\overline{M}_3^1$ is now used to synthesize speech frame R_3^S that introduces the memory state error contribution. From that point, all processed frames will be impaired by the memory state error effects until such a time the encoder and decoder memory state become synchronized again. The memory state is represented as $E_3^S = S_3 - R_3^S$.

The following sample (Figure 4.8) will extend the basic understanding of the ITU-T G.729A algorithm by exposing it to a longer error bursts. In this example we decided to extend the previously explained behavior under single packet error to include the following packet to be part of the error. In other words, parameters P_3 will also be lost. This composes an error burst of size 2.

Encoder	Source signal	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
	State memory	M_0^0	M_1^0	M_2^0	M_3^0	M_4^0	M_5^0	M_6^0	M_7^0	M_8^0	M_9^0	M_{10}^0	M_{11}^0	M_{12}^0	M_{13}^0	M_{14}^0	M_{15}^0
	Parameters	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	Xmit/Rcv	↓	↓			↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Decoder	Parameters	P_0	P_1	V_2^1	V_3^2	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	State memory	M_0^0	M_1^0	M_2^0	$\overline{M}_3^1$	$\overline{M}_4^2$	$\tilde{M}_5^0$	$\tilde{M}_6^0$	$\tilde{M}_7^0$	$\tilde{M}_8^0$	$\tilde{M}_9^0$	$\tilde{M}_{10}^0$	$\tilde{M}_{11}^0$	$\tilde{M}_{12}^0$	$\tilde{M}_{13}^0$	$\tilde{M}_{14}^0$	$\tilde{M}_{15}^0$
	Reconstructed signal	R_0^G	R_1^G	R_2^C	R_3^C	R_4^S	R_5^S	R_6^S	R_7^S	R_8^S	R_9^S	R_{10}^S	R_{11}^S	R_{12}^S	R_{13}^S	R_{14}^S	R_{15}^S

Figure 4.8 - Memory state effects following dual packet loss

We have seen in the previous example that R_2^C was constructed with intermediate values V_2^1 and memory state M_2^0. We also mentioned that during that process memory state $\overline{M}_3^1$ was generated and used for the synthesis of the next frame. Assuming the packet holding P_3 is also lost implies that the concealment values V_3^2 would be generated by the PLC algorithm. Those values could be perceived as being a second order PLC set of intermediate values issued from values generated during the last known good packet just before the identification of the lost packets, namely from P_1 and M_1^0. As

before, we assume the presence of a drift distance affecting the intermediate values (e.g. last known good values are slightly altered and scaled down as per PLC algorithm discussed in section 4.4.3). The term drift distance was introduced as it represents well the cumulative effect (e.g. values drifting further astray) when consecutive packets are lost. The drift distance is signal dependent and is considered important in most cases except in the case where the signal is naturally fading.

We can observe an important difference with respect to the reconstruction of R_3 for single and dual packet error burst. In the first case, R_3^S is synthesized using values from good parameters but erroneous memory state $\overline{M}_3^1$. In the second case, R_3^C is synthesized using estimated values V_3^2 and erroneous memory state $\overline{M}_3^1$. Therefore, the first concealment of a lost packet introduces a pure concealment error but consecutive concealment introduces a concealment error including a memory state error component (also called concealment error for simplicity as the true concealment error is considered the major contributor to this error). This explains the good performance of the PLC algorithm to single packet loss but also explains why there is a marked difference in speech quality for error bursts of 2 or more packet losses as reported in similar studies [70][58].

4.5.4 MEMORY STATE CONVERGENCE

We now understand the mechanism through which the state error will propagate. In this section we will analyze the memory state element identified earlier in order to better understand their behavior. The convergence of the first set of memory state elements was easy to determine looking at the source code. Memory state element m^a or the 4th order MA predictor, holds the last 4 vectors of the Line Spectral Pairs (LSP). This predictor is used on a frame basis where each vector represents the LSP prediction at *n-4*, *n-3*, *n-2*, and *n-1* that are now used to generate prediction for the current frame *n*. These vectors are scaled to provide more importance to later values, thus requiring 4 good

frames to generate a synchronized memory state element. Next, memory state element m^d or the 4th order MA predictor, holds the last 4 predicted values of the fixed codebook gain. The prediction consists of scaled past values that are updated on a subframe basis, thus requiring 2 good frames to generate a synchronized memory state element. Finally, memory state element m^e or the pitch sharpening value, is updated using the adaptive codebook or pitch gain from the previous subframe. Therefore, this value gets synchronized on the following good frame. However, we found that both memory state elements m^b and m^c are speech dependent. That is, memory state element m^b takes the past excitation values and memory state element m^c takes the past short-term coefficients. Finding the convergence for these two memory state elements is not easy and requires us to run some simulations to study their behavior.

We opted for a manual approach to investigate the changes in memory states as errors were introduced in the encoded stream. The VWB application, which will be described in Chapter 5, was used to generate a report where all values contained in the encoder and decoder memory state elements were copied in a formatted file for easy comparisons. Two sets of files were generated. The first set representing the process with no error, and the second set the process with errors. This allowed us to open the file side by side and investigate the differences at error insertion points. We investigated several of these reports and confirmed that memory state elements m^a, m^d, and m^e, took 4, 2, and 1 good frame to converge to a synchronized state. On the other hand, signal dependent memory state elements m^b and m^c took on average 27 and 25 good frames respectively to converge to a synchronized state (51 and 47 good frames respectively without applying any threshold). This was determined by estimating group of values at the time. We assumed that any value showing less than a 10% difference (e.g. threshold) was a converged value. We averaged the values of 20 error insertion points to establish those convergence measures. These results correlate to similar results obtained in [73][70][58].

4.6 SUMMARY

In this chapter, we investigated some of the issues and constraints applicable to VoIP systems. The main difficulties are associated to the lack of transport reliability or best effort delivery of packets across IP networks, which subjects the receiver to packets delivery delays and packet losses (packets not delivered or not delivered on time) that impairs the quality of the reconstructed speech. A complete end-to-end communication model was presented to thoroughly investigate the main sources of delays in a VoIP system. Overall, the end-to-end delay is expected to fluctuate between 262 ms and 512 ms, which goes beyond the 300 ms end-to-end delay requirement that was identified. Nevertheless, we made the assumption that the network is well-tuned with sufficient bandwidth and that unacceptable delays are prevented most of the time. We then established that the network was in fact acting like a buffer from the end terminals point of view and that packet losses were inevitable. We studied the effect of missing packets and realized that when using low bit rate speech codecs, a missing packet is generating an error due to the missing packet (a concealment error) and an error due to the memory state differences between the encoder and decoder (memory state error).

In Section 4.4 we explored existing methods to recover from packet losses (codec robustness). Sender-based recovery usually adds redundant speech information that requires additional bandwidth for the transmission of the speech frames. Receiver-based concealment methods were presented where each were attempting to find the proper replacement frame to substitute in place of the missing one. Key to receiver based techniques and also their *raison-d'être* is that they do not rely on any intervention from the sender side of the system, hence saving bandwidth and minimizing the impact of potential delay problems associated with packet networks. This also infers that

receiver-based techniques rely on a priori information that has been preserved by the decoder. The G.729A PLC algorithm was covered to some length as it is extensively used in this thesis.

Then a notation and associated figures were used to allow us better understand the effect of packet losses on the memory state error. From the notation, we were able, in theory, to demonstrate the reason and rationalize the good performance of the PLC algorithm to individual packet losses compared to the performance for longer consecutive losses.

Finally, we studied the convergence of the memory state error after good packets are received. We determined that 5 memory state elements were responsible for the codec overall memory state and that elements m^a, m^d, and m^e, took 4, 2, and 1 good frames to converge to a synchronized state while memory state elements m^b and m^c took on average 27 and 25 good frames respectively.

5.0 MEMORY STATE ERROR CORRECTION (SEC) ALGORITHM

5.1 OVERVIEW

The previous chapter explained the operation of the ITU-T G.729A algorithm under conditions of packet loss. The explanation focused primarily on the behavior of the memory states between the encoder and the decoder. It was determined that their synchronization was a key factor for the optimal operation of the algorithm. In this section, we propose a closed loop algorithm in order to minimize the propagation of the memory state error. The algorithm will be simply referred to as State Error Correction (SEC) in this thesis. As mentioned earlier, some open loop algorithms having the same objective in mind were proposed [58]. The main disadvantage of open loop approaches is that they are applied whether or not packet losses occur. This impacts the reconstructed signal quality even in periods of no loss. Another serious drawback of those algorithms happens when, for some reason, the mechanism loses its synchronization. This situation would cause an even more important signal degradation since open loop also implies that there will be no way for the encoder or the decoder to know that they have become unsynchronized. To protect against this sort of degradation, a synchronization mechanism needs to be adopted to ensure optimal encoder and decoder operation. This being said, we believe that there is a potential for a closed-loop algorithm to minimize the propagation of state error despite the difficulties such algorithm will encounter due to the unpredictable network delay.

In the following discussion, we explain the envisioned operation of SEC and attempt to clearly identify what would constitute a beneficial implementation by taking into consideration the network delay. We will also investigate re-synchronization schemes and their effect on the speech quality performance. Then, we will describe the set of experiments conducted and present the results obtained. Finally, a conclusion will summarize our findings.

5.2 SEC OPERATION OVERVIEW

The proposed algorithm is intended to supplement compression methods that use memory state as part of their algorithm. Such is the case of the ITU-T G.723.1 [27], and the ITU-T G.729A [30][31] algorithms to name a few. As discussed previously, a side effect to the loss of packets is the generation of a state error between the encoder and the decoder and the associated distortion in the reconstructed speech signal. The proposed algorithm is only applicable when at least one memory state element is source signal dependent and requires considerable time to re-synchronize. The proposed SEC algorithm is not intended to conceal the lost packets. Rather, the objective of SEC is to minimize the propagation of the state error *following* the detection of errors and their concealment. As discussed earlier, the memory state error may take some time to return to a synchronized memory state between the encoder and decoder following one or more transmission errors. Since memory states between the encoder and decoder are unpredictable following the concealment of missing packets, we must address the problem of how the memory state can be re-synchronized in a most effective manner. A memory state re-initialization scheme [58] was reported as being an acceptable scheme on which we can build. For any closed-loop method, an exchange is required between the encoder and the decoder.

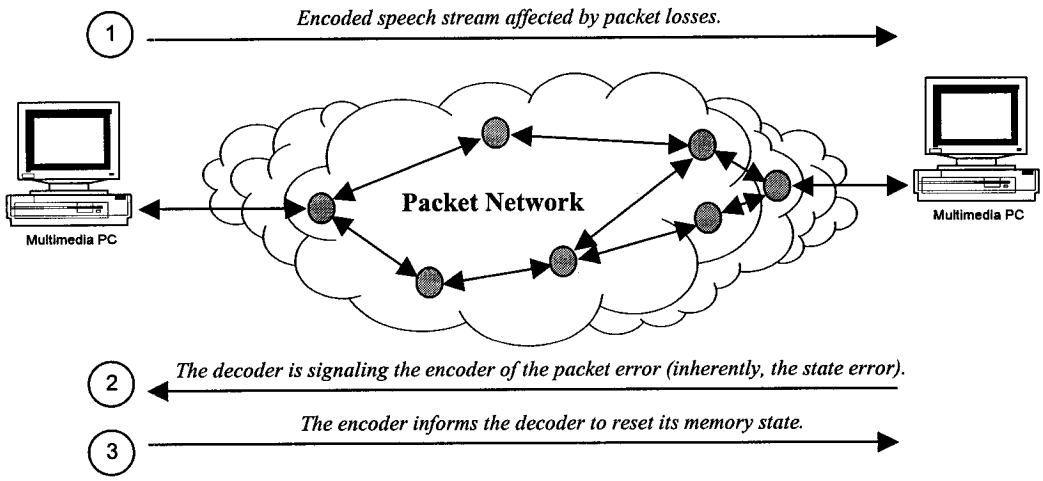


Figure 5.1 – SEC operation functional diagram

The operation of SEC is summarized in three steps as depicted by Figure 5.1. Firstly, the receiving PC detects the loss of a packet in the stream sent from the encoder. Secondly, the receiving PC immediately issues a memory state re-synchronization request to the encoder and subsequently informs the decoder of the error by directing it to conceal the error as per the embedded G.729A standard PLC algorithm. Thirdly, as the encoder receives the request to re-synchronize, the encoder re-synchronizes its internal memory state according to the selected scheme (investigated schemes will be described shortly) and subsequently informs the decoder that it performed the requested re-synchronization. In turn, as the decoder receives the message from the encoder, the decoder re-synchronizes its internal memory state prior to decoding the upcoming parameters representing the next encoded speech frame. Thus, there will be a delay between the detection of the missing packet and the re-synchronization of the memory states.

Figure 5.2 and Figure 5.3 describe in detail the proposed SEC algorithm using the same notation presented before. The associated description of the variables used can be found in Table 4.1. Using Figure 4.10 as a reference figure, a packet error is introduced at P_2 corresponding to speech frame S_2. Assuming a network round-trip delay of approximately 60 ms (six frames), evenly split over both directions, the algorithm's behavior to the missing packet is identical to the reference case in Figure 4.9 until six frames have elapsed since the error was detected. Then, before the encoder processes the following speech frame, a memory state re-synchronization occurs and the encoder and decoder enter a new synchronized memory state domain[14] identified by M_9^1 in this example. Let us review this example in more detail.

---

[14] Notice the memory state domain change indicated by the superscript (see table 4.1).

Encoder	Source signal	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
	State memory	M_0^0	M_1^0	M_2^0	M_3^0	M_4^0	M_5^0	M_6^0	M_7^0	M_8^0	M_9^1	M_{10}^1	M_{11}^1	M_{12}^1	M_{13}^1	M_{14}^1	M_{15}^1
	Parameters	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	Xmit/Rcv	↓	↓		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Decoder	Parameters	P_0	P_1	V_2^1	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	State memory	M_0^0	M_1^0	M_2^0	$\overline{M}_3^1$	$\widetilde{M}_4^0$	$\widetilde{M}_5^0$	$\widetilde{M}_6^0$	$\widetilde{M}_7^0$	$\widetilde{M}_8^0$	M_9^1	M_{10}^1	M_{11}^1	M_{12}^1	M_{13}^1	M_{14}^1	M_{15}^1
	Reconstructed signal	R_0^G	R_1^G	R_2^C	R_3^S	R_4^S	R_5^S	R_6^S	R_7^S	R_8^S	R_9^G	R_{10}^G	R_{11}^G	R_{12}^G	R_{13}^G	R_{14}^G	R_{15}^G

Figure 5.2 - Single packet loss with SEC after a delay of 6 speech frames (approx. 60 ms).

The event triggering the SEC is the detection of a missing packed by the receiver (missing parameters P_2). It is assumed that the declaration of the packed loss occurs when the jitter buffer logic has declared the packet as being lost. This triggers two actions. Firstly, a *SEC Request* message is issued to the encoder in order to re-synchronize the memory state. Secondly, the decoder receives the BFI directing it to conceal the missing speech frame. As indicated before, network latency acts as a queue where packets are already in transit through the network. In this example, it is assumed that approximately three frames are in the queue while the lost packet P_2 is being concealed (the encoder is about to encode speech frame S_6). Therefore, if the *SEC Request* message reaches the encoder within the next 30 ms, then the encoder memory state would re-synchronize according to some scheme. Remember that a common scheme was needed to re-synchronize the encoder and decoder memory states. The encoder would then acknowledge the *SEC Request* by issuing a message to the decoder before encoding speech frame S_9 using its new memory state domain M_9^1. Packet P_9 is then transmitted and received by the decoder. The decoder, on reception of the *SEC Request* acknowledgement from the encoder, resumes its operation by re-synchronizing to the new decoder memory state domain M_9^1 (details to be discussed later). Consequently, packet P_9 would be decoded using a synchronized memory state M_9^1 and would produce a "good" reconstructed signal segment R_9^G. Any packets received following P_9 will be decoded using the

correct respective memory state unless another packet loss is detected forcing the SEC algorithm into another cycle.

The SEC algorithm offers an advantage when many consecutive error bursts occur. For example, looking at Figure 4.9, if an error burst shortly follows a first error burst, a single *SEC Request* is required to correct the state error. This time, assuming a network round-trip delay of approximately 100 ms (10 frames) and 7 frames are already in the network queue when the first packet in error is detected. If the encoder receives the *SEC Request* after 30 ms (three additional frames are now in the queue for a total of 10 frames) then, the algorithm would typically instantiate a new state domain for the reconstruction of speech frame R_{13}^{G}.

Encoder	Source signal	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{14}	S_{15}	S_{16}
	State memory	M_0^0	M_1^0	M_2^0	M_3^0	M_4^0	M_5^0	M_6^0	M_7^0	M_8^0	M_9^0	M_{10}^0	M_{11}^0	M_{12}^0	M_{13}^1	M_{14}^1	M_{15}^1
	Parameters	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	Xmit/Rcv	↓	↓		↓	↓			↓	↓	↓	↓	↓	↓	↓	↓	↓
Decoder	Parameters	P_0	P_1	V_2^1	P_3	P_4	V_5^1	V_6^2	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	State memory	M_0^0	M_1^0	M_2^0	$\overline{M}_3^1$	$\widetilde{M}_4^0$	$\widetilde{M}_5^0$	$\overline{M}_6^1$	$\overline{M}_7^2$	$\widetilde{M}_8^0$	$\widetilde{M}_9^0$	$\widetilde{M}_{10}^0$	$\widetilde{M}_{11}^0$	$\widetilde{M}_{12}^0$	M_{13}^1	M_{14}^1	M_{15}^1
	Reconstructed signal	R_0^G	R_1^G	R_2^C	R_3^S	R_4^S	R_5^C	R_6^C	R_7^S	R_8^S	R_9^S	R_{10}^S	R_{11}^S	R_{12}^S	R_{13}^G	R_{14}^G	R_{15}^G

Figure 5.3 - SEC under consecutive error bursts

In this case, parameters P_2, P_5, and P_6 are lost. We already know the memory state will go astray once a missing packet is detected and that an associated state error will be a component of the perceived distortion. It is assumed that some state error is still present when the second packet loss occurs, and that this second loss will lead to a poor concealment and a worse state error. Again, a *SEC Request* is issued on the first missing packet of an error burst being detected. If we assume that the first *SEC Request* is still unanswered when the second error burst is detected, the algorithm would not issue another *SEC Request* and the second error burst would fall under the same state

error repair scope. This advantages the second error burst as the repair would occur in a shorter timeframe and possibly present a substantial speech quality performance gain.

So far we looked at the operation and effect of SEC without focusing our discussion on a specific state error repair scheme. We now investigate different SEC repair schemes in an attempt to better understand tradeoffs that exist. The identification of tradeoffs should provide sufficient insight to allow us to select or propose a SEC repair scheme. The envisioned SEC schemes are: reset to defaults (RST) and reset to Last Known Good (LKG).

- *Reset to defaults (RST)*: This scheme is the simplest of all. In this scheme we wish to re-synchronize both the encoder and decoder according to some pre-specified or initial values. However, this scheme does not recognize or consider any useful information that may still reside in the memory state before the error occurred.

- *Reset to last known good (LKG):* This scheme is somewhat simple but requires additional memory and a larger bandwidth. It aims at exploiting any information that may still be useful even after some elapsed time. Since phonemes last on average 80 ms [42] then some of the memory state may be pertinent and provide better results (e.g. compared to resetting both memory states to their default values). Furthermore, voice is characterized by many traits of the speaker. It is assumed that the memory state carries some correlation of those traits from phoneme to phoneme.

More details with respect to the description and operation of those schemes will be presented in section 5.3.2.

5.3 SEC DETAILED OPERATION

Packet loss errors are detected outside of the ITU-T G.729A codec algorithm. Remember that we discussed the Bad Frame Indicator (BFI) in Section 4.4.3.3. The BFI is considered as a side information or an out-of-band signal from the codec perspective where the H.323 or SIP, or more precisely RTP/RTCP provides the structure to detect the missing packet through a missing sequence number in the sequence of received packets. We elected for the SEC algorithm to use side information for the exchange between the encoder and decoder through an out-of-band RTP/RTCP channel (the RTP/RTCP protocols have provision for user defined reports that can be used to exchange any desired information). The main reason to proceed in that direction is that some implementations may use several encoded speech frames in the same packet for transmission over the packet network. In that case, the lost packet would imply many encoded speech frames being lost from a single network packet loss. Since it was already identified that it would be ineffective to contain loss frame information within the encoded source (e.g. a good example is the BFI external to the ITU-T G.729 and G.723.1 recommendations), the same should hold true for the bit or bits needed to convey the SEC signaling or control information, specially that they are only used following an error. Consequently, we assume the proposed algorithm will use the presentation and session layer protocol, namely the RTP/RTCP protocol to exchange signaling information between the encoder and the decoder. The SEC information thus conveyed by RTP/RTCP will enable the effective implementation of the closed-loop. However, this thesis only presents the principle and the exact details of a robust implementation shall be based on best practices as suggested by advances in that field of research.

5.3.1 THE CLOSED-LOOP STRUCTURE

The re-synchronization as per the selected repair scheme needs to be performed in a coordinated fashion between the encoder and decoder. As covered earlier, the SEC algorithm is initiated from the decoder side as an outcome to the detection of packet errors. The closed-loop is maintained by adding some signaling between the encoder and decoder. Depending on the selected scheme, a small increase in bandwidth requirement[15] is expected. As will be covered later, the repair scheme implementation will have a different implementation complexity and memory requirements. Figure 5.4 shows the encoder and decoder SEC implementation logic diagram and the SEC signaling required to establish the closed-loop.

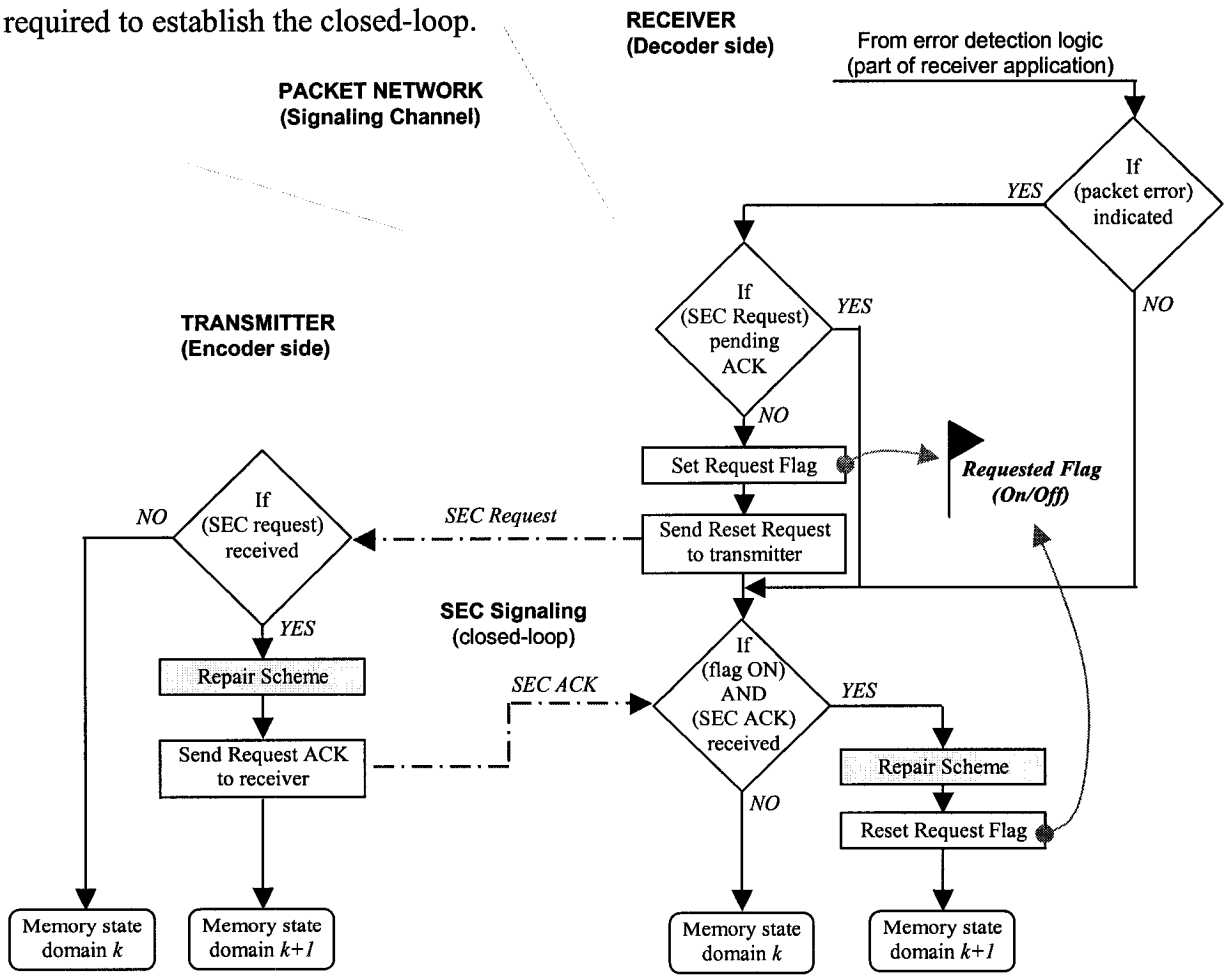


Figure 5.4 – SEC Signaling and logical flow diagram

[15] The additional bandwidth requirement is needed only when packet losses are detected. In other words, no SEC algorithm cycle take place during periods of no errors.

As mentioned earlier, this system is idle until an error is detected. The system is engaged when an error occurs as determined by the error detection mechanism already contained within the ITU-T G.729A algorithm. Figure 5.4 is divided into three sections, namely the encoder logic (left), the closed-loop signaling (center), and the decoder logic (right). The first error (upper right decision of the decoder logic), associated with packet *n*, forces a verification of the *Requested Flag* to be performed (middle decision). In this case, the *Requested Flag* is in its initial state (Cleared/OFF/FALSE) and allows the decoder logic to set the *Requested Flag* before transmitting a *SEC Request* to the encoder (the encoder operation will be covered shortly). At this stage the *Requested Flag* is ON/TRUE and the *SEC Request* is in transit through the network. The third decision from the decoder logic will direct the decoder to use the current memory state (e.g. memory state domain *k*) to conceal or reconstruct speech frame *n* (i.e. the Requested Flag is turned ON/TRUE and the *SEC ACK* is FALSE/not received yet). Assuming the reception of a *SEC ACK* when the flag is set implies a *SEC Request* was previously issued to the encoder. The function of the encoder is really simple and basically involves monitoring the receipt of the *SEC Request*, updating the memory state according to the common scheme, and acknowledging or informing the decoder that the upgrade was performed.

As discussed earlier, we rely on RTP to provide a fast but unreliable delivery model and RTPC to exchange signaling information and statistics. The implementation of the signaling logic is very simple and may require very few bits depending on how it is implemented. If it is included in existing RTP/RTCP messages (assuming a two-way real-time conversation), then the information can be transmitted with the voice data or the statistics report. This requires additional logic to recognize the presence or absence of the signaling information if we do not want to introduce a constant overhead. The drawback is that the packet may get lost in which case it further deteriorates

the memory state, especially in the case where a returned *SEC ACK* from the encoder is lost possible deadlock of the flag). A simple deadlock avoidance mechanism could be implemented, which would reset the *Requested Flag* after a predetermined elapsed period of time. However, a more robust approach would be to use RTPC messages to convey the signaling information in both directions. A stringent handshake approach could be implemented using RTCP messages to provide a reliable signaling delivery method. This proposed concept or method could be a research topic on its own, which is beyond the scope of this thesis. The basic additional transmission requirement for the proposed algorithm will be discussed in the following section as it directly relates to the repair scheme.

5.3.2 DESCRIPTION OF REPAIR SCHEMES

In the preceding sections we described the SEC operation without going into the details of the repair schemes. In this section we describe the Reset to defaults (RST) and the Last Known Good (LKG) initialization algorithms or repair schemes introduced in Section 5.2. We will investigate the implementation feasibility with respect to the memory state elements described in Section 4.5.2. In Section 4.5.4, we analyzed the convergence of each of the 5 memory state elements. We showed that memory state elements m^a, m^d, and m^e converge naturally to a synchronized state after *4*, *2*, and *1* frame respectively and that m^b and m^c convergence is variable but on average are synchronized after approximately 27 and 25 frames respectively. The SEC algorithm will be exposed to varied network delays. Hence, we decided to perform an initial investigation of the performance of memory elements re-synchronization under network delays. The objective of this investigation is to find the most appropriate re-initialization configuration for the memory state elements.

5.3.2.1 Reset to defaults (RST)

Initial Investigation

Figures 5.5 to 5.7 show the result for TS-01 (e_f01s04.wav) for a simulated network delay of 6 frames. For the following discussion, and in order to simplify the text, we will refer to the two graphs in Figure 5.5(a) and Figure 5.5(b), as the signal and the error respectively. In Figure 5.5, all memory state elements are initialized to their default values (see section 4.5.2), six frames after the error is detected. When compared with Figure 4.4 to Figure 5.5, from left to right, we observe a small signal difference between the two graphs. Correspondingly, we see that Figure 5.5 shows a smaller concealment error than that of Figure 4.4. Moving toward the right of the figure, we notice that the leading edge of both state errors is similar. At the state error synchronization point, starting at the 70 ms mark, Figure 5.5 looks fairly distorted when compared with the signal of Figure 4.4 but has less squared error. We would expect that Figure 5.5 will generate artifacts similar to performing receiver-based concealment using silence substitution, specially since the transition into the following frame is quite abrupt. However, the speech segment from the 80 ms mark presents less error than on Figure 4.4, which illustrates a gain in performance.

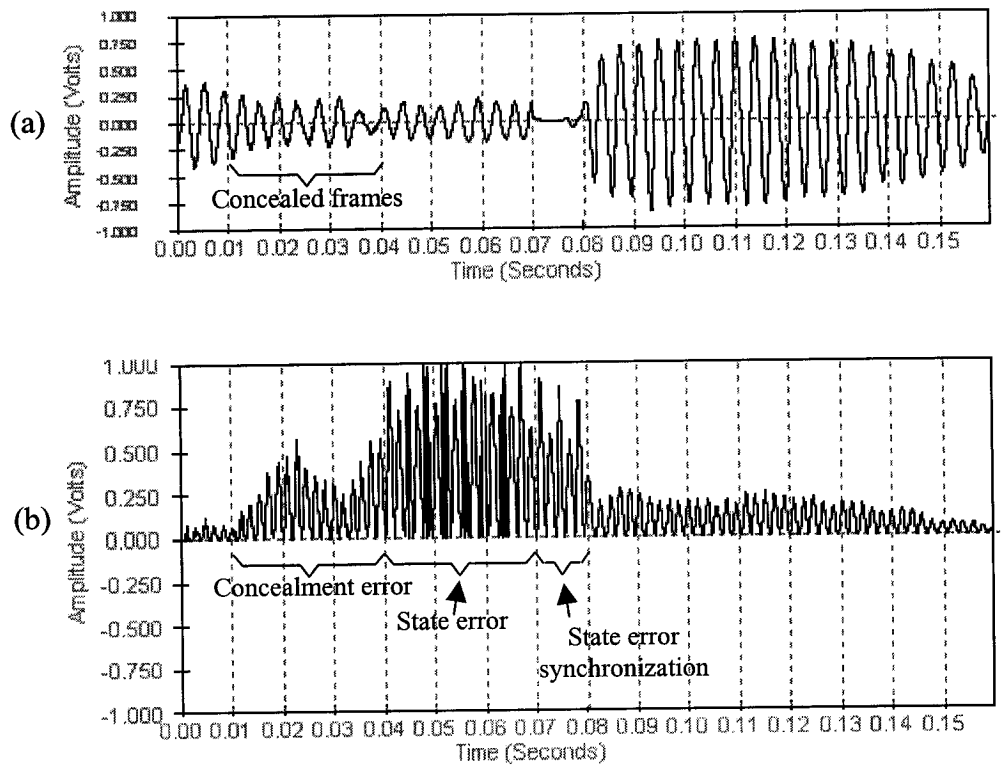


Figure 5.5 – Memory state synchronization using the RST scheme after a network delay of 6 frames and applied to m^a, m^b, m^c ,m^d, and m^e

What is most remarkable is the similarity of both the signal and the error from that point, when compared with Figure 4.3, showing the same signal reconstruction but with no error. Re-initializing all memory state elements is probably not the best configuration, especially if longer network delays are expected. Therefore, assuming the natural convergence of memory state elements m^a, m^d, and m^e occurred before the re-synchronization point, we investigate the effect of synchronizing memory state elements m^b, and m^c. That configuration is shown in Figure 5.6.

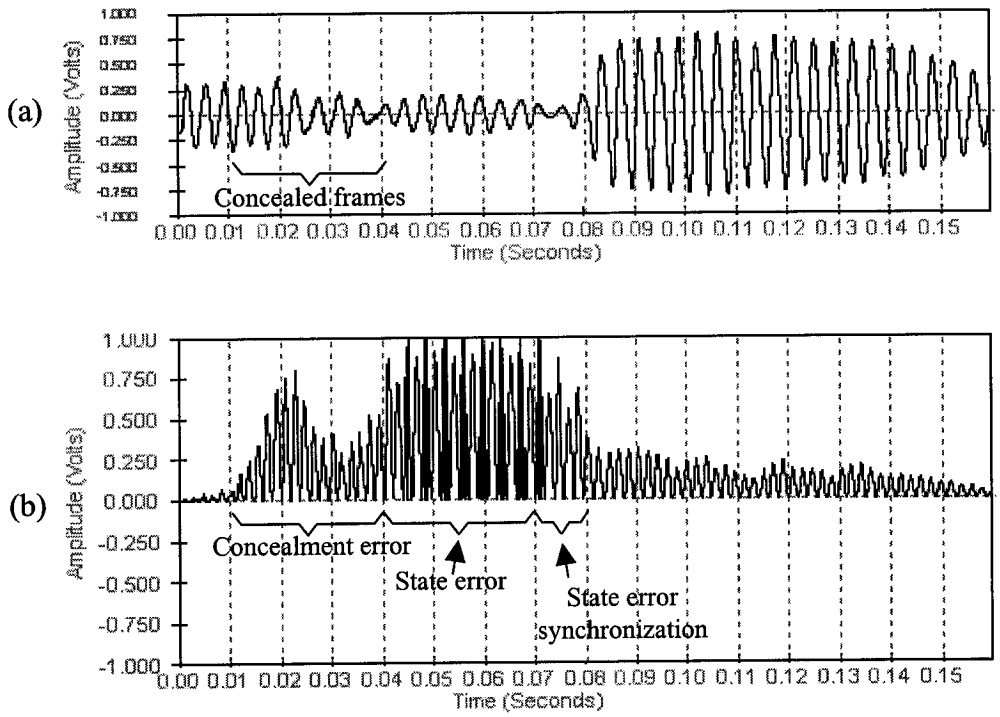


Figure 5.6 – Memory state synchronization using the RST scheme after a network delay of 6 frames and applied to m^b and m^c

Comparing with Figure 4.4, we observe a similar signal pattern but this time the concealment error and state error results are very similar until the point of state error synchronization. The main difference is with respect to the signal distortion during the re-initialization. We find that there is less distortion and state error this time around. Finally, we perform the same simulation but this time with memory state elements m^b only.

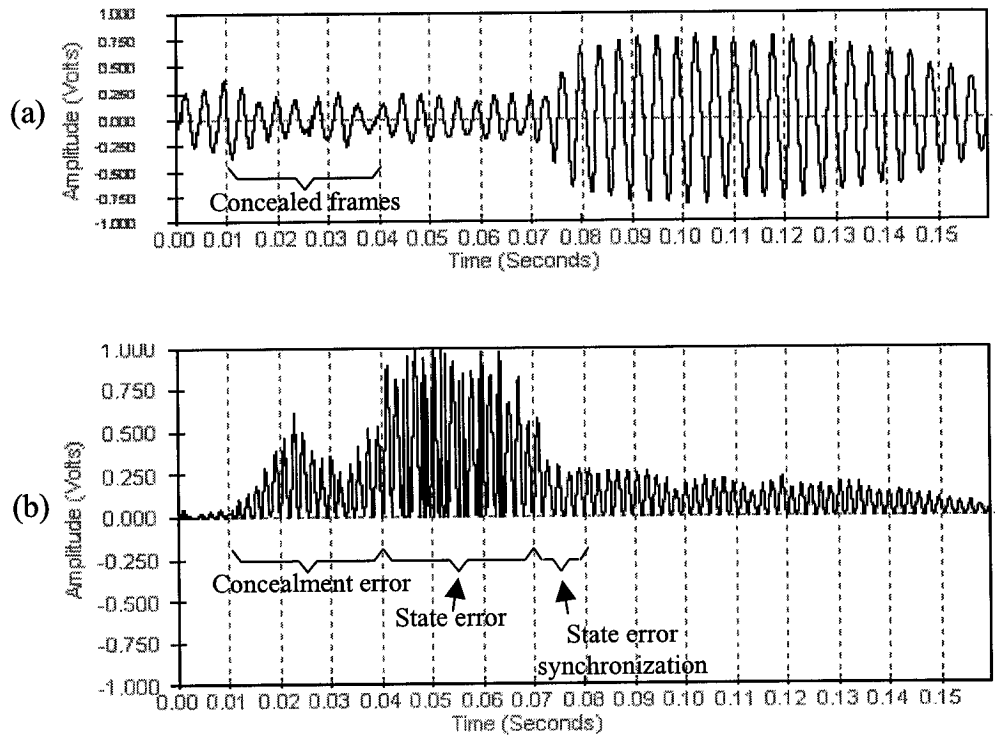


Figure 5.7 – Memory state synchronization using the RST scheme after a network delay of 6 frames and applied to m^b only

Figure 5.7 shows the results of that configuration. This time we notice that the signal resembles that of Figure 4.4 up to the synchronization point. From that point on, the signal has a greatest resemblance to that of Figure 4.3. Looking at the error, we again note less concealment error and less state synchronization error.

Several simulations were executed to test the performance of the three re-initialization configurations presented. Simulations parameters were changed, namely, the test speech stream, the size of the error bursts, the interval between subsequent error insertions and the network delay. Similar results were constantly obtained.

The state error propagation has clearly stopped in Figure 5.5 to Figure 5.7. On the other hand, in the case of the signal shown in Figure 4.4, we believe the observed differences in the concealed signal

and the concealment error are attributable to the propagation of the state error as a consequence of the previous error burst.

The performances of the re-initialization configurations were confirmed through informal subjective testing by the author. The first two re-initialization configurations presented noticeable artifacts that clearly identified the third re-initialization configuration as superior, namely, the m^b only re-initialization configuration.

Operation

This process is very simple and consists in re-initializing memory state elements with their default values in this case, all zeros.

Bit rate

Since the operation consists in re-initializing the memory state element with its default values, a single bit is required for the receiver to request a memory state re-initialization to the transmitter. Then, a single bit is required for the transmitter to notify the receiver of the changed state. Therefore, assuming a 10% loss rate over a stream of 100 frames, it implies that 10 frames will be dropped. The one-way bit rate increases by an additional 10 bits. Since each G.729A speech encoded frame is 80 bits, a total of 8000 bits are normally transmitted. Therefore, adding the SEC-RST scheme to the standard G.729A algorithm would require 8010 bits be transmitted, representing a 0.125% bit rate increase or an operating bit rate of 8.01Kbps.

Complexity

This scheme does not require any additional memory. The added processing associated with the selected re-initialization configuration requires updating 154 memory locations (in this case for fixed-point arithmetic implementation, 16-bit words) is considered negligible.

5.3.2.2 Reset to last known good (LKG)

Initial Investigation

We proceeded with this initial investigation using the same approach as for the RST scheme. As discussed earlier, memory state elements m^a, m^d, and m^e converge naturally to a synchronized state after *4*, *2*, and *1* frame respectively, which we assume is faster than the re-synchronization through the network. This implies that memory state elements m^b and m^c need be investigated in different combinations. As per the case of the RST scheme, we could ear many artifacts when memory state element m^c was initialized in any of the combinations. Consequently, this section highlights the initialization of memory state element m^b only. Results were very similar with the exception that we did not observe as important distortion as for the signal shown in Figure 5.5.

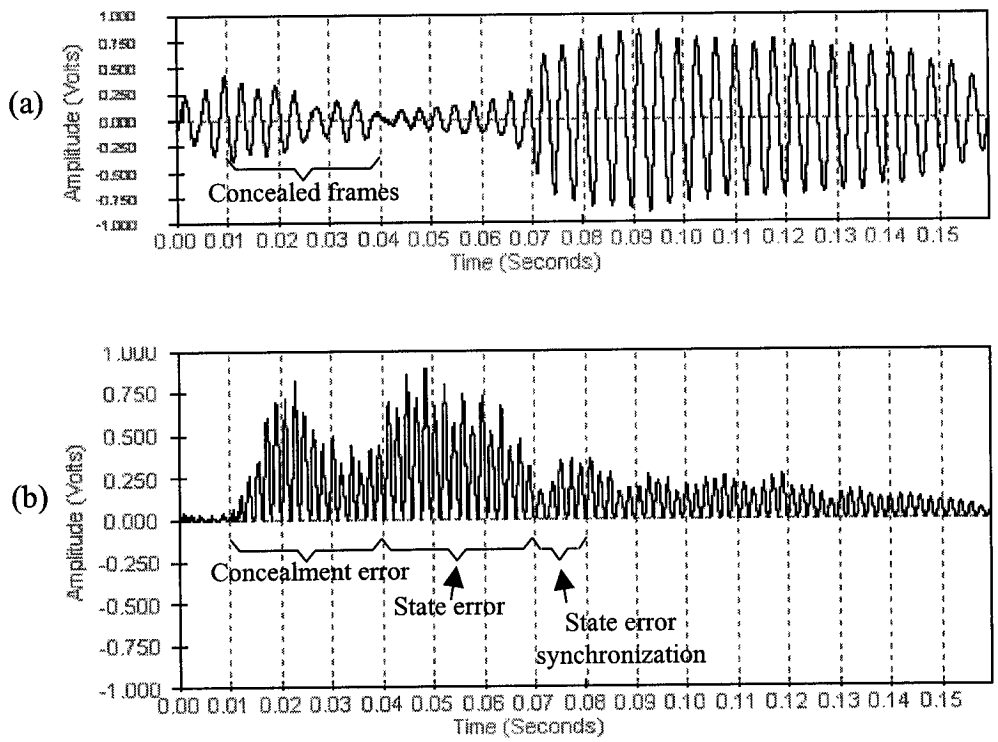


Figure 5.8 – Memory state synchronization using the LKG scheme after a network delay of 6 frames and applied to m^bonly

Figure 5.8 and Figure 5.9 show the results when only considering the memory state element m^b for the re-initialization configuration. Figure 5.9 shows the result obtained for network delays of seven

frames. In this case, we can see that the state error lasts longer. In both cases, the re-initialization configuration (with memory state element m^b only) leads us to believe that the LKG scheme performs better than the equivalent RST scheme.

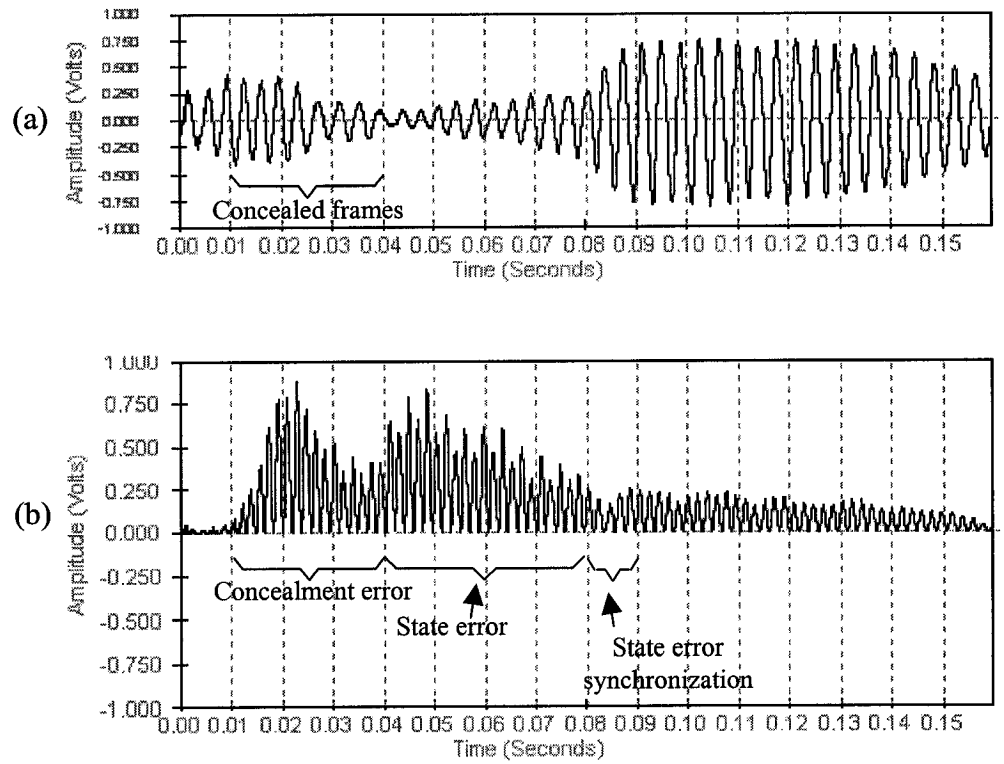


Figure 5.9 – Memory state synchronization using the LKG scheme after a network delay of 7 frames and applied to m^b only

In general, the re-initialization of the memory state elements (under different re-initialization configuration – exception to the memory state m^b only re-initialization configuration) with their last known good state seemed to provide less state error and less signal distortion than those presented in Figure 5.5 and Figure 5.6, but they exhibited discontinuities at frame boundaries following the re-initialization. Several simulations were executed and subsequent informal subjective testing by the author was performed on the produced results. Overall, they often present noticeable artifacts. On the other hand, the re-initialization configuration using memory state element m^b constantly demonstrated *similar or clearer* speech during those tests.

Based on the informal subjective tests performed and the objective evidence as demonstrated in Figure 5.8 and 5.9, we select the re-initialization of m^b only configuration as the SEC-LKG scheme.

Operation

The memory state element m^b resides in the excitation vector after each encoding (decoding), as described in Section 4.5.2. A history buffer (of type First-In-First-Out (FIFO) – often implemented as a circular queue) is required at the encoder to implement this scheme. Associated with this buffer are two basic buffer operations, namely, a store and retrieve function. After each encoding, the m^b vector is stored following the FIFO model. A request to re-initialize the memory state element will require an index to the buffer to be able to retrieve the proper state. This index will be issued from the receiver to the transmitter. A single vector is stored at the decoder (i.e. the last known good m^b vector before the detected error). An index is also stored to serve as a timestamp for the vector.

When the receiver detects a missing packet, it immediately stores the vector (e.g. the last known good state vector) and captures the timestamp index. It then sends a request to re-synchronize and the captured timestamp index is sent to the transmitter. The transmitter receives the request with the timestamp index. It searches the history buffer for the index and retrieves the associated vector. The vector is then restored. The transmitter sends an acknowledgement message to the receiver. Finally, on receipt of the acknowledgement message, the receiver restores the vector. At this point both the transmitter and receiver have the same memory state element m^b.

Bit rate

In this case, the receiver must inform the transmitter which archived state to use in order to re-initialize the memory state element. Then, a single bit is required for the transmitter to notify the receiver of the changed state. Again, assuming a 10% loss rate over a stream of 100 frames, it

implies that 10 frames will be dropped. The one-way bit rate increase is proportional to the size of the index. Therefore, assuming an 8-bit index is used, it would require an additional 80 bits. Since each G.729A speech encoded frame is 80 bits, a total of 8000 bits are normally transmitted. Therefore, adding the SEC-LKG scheme to the standard G.729A algorithm would require 8080 bits to be transmitted, representing an 1% bit rate increase or an operating bit rate of 8.08Kbps.

Complexity

The maximum network delay will determine the size of the history buffer. Assuming we use this scheme to repair the state error for a maximum network delay of 180 ms requires the capability to store 18 vectors. The size of each vector is 154 X 16-bit = 308 bytes per vector, consequently, 5544 bytes are required.

The index requires uniquely identifying 18 vectors, hence, at least 5-bit are required (to simplify the calculation, we assume some index robustness algorithm would be implemented and would use 3 bits, hence, we would use an 8-bit index). An index must be stored for each vector store in the history buffer, consequently, 18 X 8-bit = 18 bytes are required. The encoder side requires a total of 5544 bytes + 18 bytes = 5562 bytes. On the other hand, the decoder requires memory for a single vector and one location to store the associated index.

The performance complexity for this algorithm is assumed to be negligible as CPU operations to store or retrieve a vector are usually performed using fast instructions. Actual performance measurement results will be presented in Table 5.11.

5.4 ENVIRONMENT FOR SEC PERFORMANCE EVALUATION

5.4.1 OVERVIEW

In this section, we investigate, through extensive testing, the performance of the proposed algorithm, in support of the ITU-T G729A standard. The experiments will assist us in identifying the practical limitations of the proposed algorithm, thereof allowing us to determine its viability in the real world.

Initial work must be performed to establish an experimental framework. The most important is the use of a tool to carry out the experiment's simulations. We will first introduce the Voice Workbench (VWB) application that was developed to support our experimentations. Another basic requirement is to set reference points against which we can compare our results. The first few sub-sections of this section will establish the necessary setup. Then, simulations will be conducted in order to assess the performance of the proposed algorithms and its two variants. Results are assessed based on a recent objective measure given by the PESQ standard (as described in chapter 2) that is reported to provide results correlating well to subjective performance assessment results. Informal subjective performance assessment was also performed by a small group of candidates including the author. The method was a much scaled down version than the one suggested in the ITU-T P.830 Recommendation [37], but was sufficiently convincing for this thesis synopsis. Finally, we discuss the results obtained before concluding this chapter.

5.4.2 TESTING ENVIRONMENT: THE VOICE WORKBENCH

In order to properly investigate the methods and techniques to use to recover from packet losses, we must be able to generate frame losses in a simulated[16] environment. For this initial objective, we

[16] A simulated environment is a cost effective means to validate the feasibility of a proposed concept, method, or implementation. Once this initial phase successfully concludes, the logical next phase is to validate the proposed concept in a real environment.

elected to create an application called the Voice WorkBench (VWB) that implements the G.729A recommendation. At the core of the application lies the ANSI C fixed-point implementation source code as provided by the supplement to the recommendation. This supplement to the G.729A recommendation provides a benchmark for implementing and demonstrating the algorithm. Appendix A provides a description of the developed application.

5.4.3 TEST VECTORS

As discussed in Chapter 2, the speech signal exhibits some fundamental characteristics that are associated with the speech production process and are further associated with the spoken dialect that is governed by its conversation rules. Thus, the signal will contain phonemes spread over words, spoken consecutively over short periods, or sporadically over larger periods, intrinsically making the speech stream an overall random process. From an engineering point of view, this implies that a requirement exists to test codecs with a variety of speech streams to validate and record their performance.

A set of 13 Test Speech (TS) files was used to verify the proposed algorithm. The speech files or test vectors originate from the ITU-T Coded-Speech Database [23] and are identified in Table 5.1 above. The test vectors represent speech streams of both female and male genders generated in two different facilities, namely, Nortel laboratories (i.e. speech files starting with the letter 'e') and the AT&T laboratories (i.e. speech files starting with the letter 'o').

The speech streams conforms to the constraints identified in the ITU-T P.830 recommendation such as the level of background noise, controlled acoustical echo, spoken content, etc. Furthermore, the sentences are representative of voiced and unvoiced sounds as well as periods of silences.

TS #	Test Speech File name	Size (KBytes)	Voice Gender	File Format
1	e_f01s04.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
2	e_f02s03.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
3	e_f02s05.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
4	e_m01s04.wav	126	Male	16-bit (mono) Linear PCM at 8000 samples/seconds
5	e_m02s07.wav	126	Male	16-bit (mono) Linear PCM at 8000 samples/seconds
6	o_f01s05.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
7	o_f01s25.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
8	o_f02s17.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
9	o_f02s32.wav	126	Female	16-bit (mono) Linear PCM at 8000 samples/seconds
10	o_m01s14.wav	126	Male	16-bit (mono) Linear PCM at 8000 samples/seconds
11	o_m01s21.wav	126	Male	16-bit (mono) Linear PCM at 8000 samples/seconds
12	o_m02s19.wav	126	Male	16-bit (mono) Linear PCM at 8000 samples/seconds
13	o_m02s37.wav	126	Male	16-bit (mono) Linear PCM at 8000 samples/seconds

Table 5.1 - Test vectors as provided by the ITU-T Coded-Speech Database

5.4.4 REFERENCE RESULTS

Initially, we need to establish reference results to serve as a basis for comparison purposes. This information is in fact taken from exposing the 13 test speech files under periods of no packet errors as well as periods of error bursts using the standard G.729A algorithm with its embedded concealment algorithm. Table 5.2 shows the ideal results for the 13 test speech files when no errors are inserted. Three performance indicators are presented and they are the MSE, the SNR, and the PESQ results as introduced in Section 2.6.3.

ID	Test Speech File Name	MSE	SNR	PESQ
1	e_f01s04.wav	0.011502715	-9.297143922	3.4
2	e_f02s03.wav	0.006697605	-8.910747181	3.488
3	e_f02s05.wav	0.005828728	-11.09282589	3.503
4	e_m01s04.wav	0.005853245	-11.74273827	3.531
5	e_m02s07.wav	0.00560458	-12.25942286	3.862
6	o_f01s05.wav	0.009309931	-5.827200423	3.486
7	o_f01s25.wav	0.0067153	-10.07687492	3.425
8	o_f02s17.wav	0.008783652	-17.7596695	3.315
9	o_f02s32.wav	0.00478045	-18.97410796	3.419
10	o_m01s14.wav	0.009652442	-15.87195016	3.25
11	o_m01s21.wav	0.006900699	-18.47257759	3.477
12	o_m02s19.wav	0.006992423	-16.03675547	3.511
13	o_m02s37.wav	0.011506448	-16.20426567	3.206

Table 5.2 – Expected ITU-T G.729A results under no error condition

These values represent the relative references against which comparisons will be performed to assess the merit of the proposed algorithm. So far, the squared error was used since it was adequate to easily and objectively demonstrate the item under discussion. However, among all the estimators presented in Section 2.6.3, the PESQ value correlates best with subjective performance assessment testing (i.e. MOS scores). Because of this, the focus will be on the PESQ results for the remainder of this thesis.

We now consider the behaviour of the G.729A algorithm under periods of error burst representative of network packet losses. A difficulty with the assessment of the codec under the presence of errors (e.g. packet losses) is that there must be sufficient understanding of the impact of inserting errors within the speech stream. For example, the insertion of errors based on a random process could potentially insert the majority of errors in periods of silence, or conversely, in periods of high signal energy. Therefore, errors will initially be inserted deterministically to ensure that the performance results are tractable and reproducible. This should be sufficient since the source information is inherently random. However, we also include a smaller set of simulations where errors were inserted on a random basis. A final phase in each experiment was to correlate results from deterministic error insertions against random error insertions for a smaller set of simulations.

As discussed in Chapter 4, a temporal correlation exists between lost packets where consecutive packets may be part of the same error, hence, the algorithm will be exposed to single, double, triple, quadruple, quintuple, and sixtuple error bursts (i.e. consecutive errors). In Section 4.5.4 we mentioned that, on average, the state error re-synchronizes unassisted after approximately 30 frames. With this in mind, along with the expertise we developed from performing our sets of experiments, we determined that error bursts will be inserted at intervals of 40 frames. This will ensure we

minimize the possible error propagation that may occur from previous error burst(s) (i.e. isolating the effects to better understand each error and the effect of the proposed algorithm).

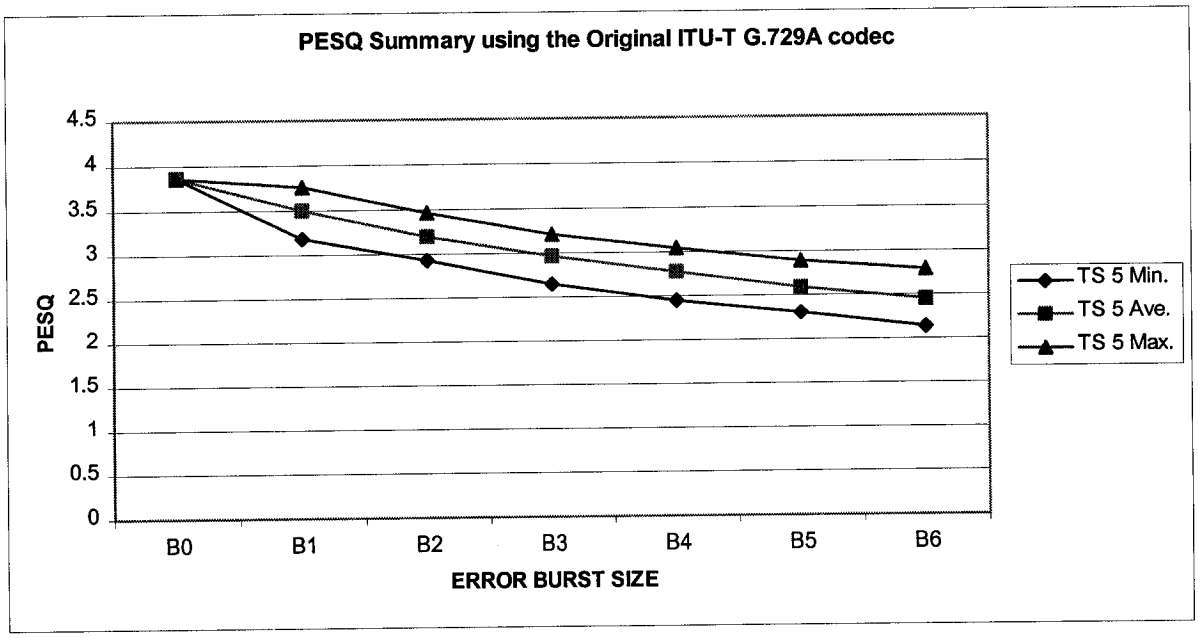


Figure 5.13 – PESQ results for TS 5 using the Standard ITU-T G.729A algorithm under different Error Bursts length

Figure 5.13 correlates with the discussion so far and confirms that missing packets effectively impair the reconstructed signal. The figure presents the average values along with the best and worst values of the set.

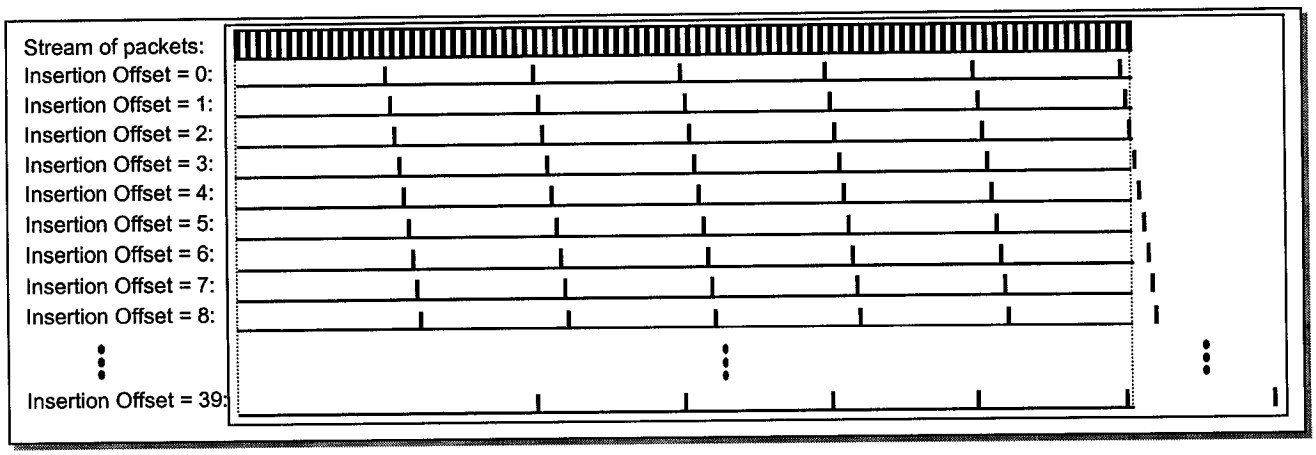


Figure 5.14 – Offsetting the insertion of errors across the test speech streams

The following paragraphs explain how Figure 5.13 was constructed. As will see in the next section, similar graphs will be used to compare the results of our proposed algorithm.
Since the number of files available is limited, we decided to maximize the use of these files by ensuring every possible region of each file has errors inserted in it. We achieved this goal by shifting the error insertion points by an offset value from simulation to simulation. Figure 5.14 illustrates this approach.

Each simulation is composed of several trials where the first trials will have error bursts inserted at the $40^{th}, 80^{th}, 120^{th}, \ldots, frame$, the second trial will have error bursts inserted at the $41^{st}, 81^{st}, 121^{st}$, frame, and so on, allowing us to gather the *expected* results. The calculated loss rate from one simulation to another may vary slightly and is caused by different number of error insertions as depicted in Figure 5.14.

In the following discussion, we use Figure 5.15 to explain how each simulation was performed in order to gather reference data as well as experimental data. Figure 5.15 shows two sets of PESQ results (from two different algorithms, but the results displayed are not important at this time) representing two simulations with different error insertion points along a same speech stream (e.g. TS 10 - o_m01s14.wav). Each data point was obtained from performing a trial (40 trials per simulation in the figure). The figure clearly demonstrates the unpredictable nature of speech, even when using a deterministic approach for the insertion of errors. The first set, using the Standard G.729A algorithm, represents one of the reference results of Table 5.2.

All reference results and experimental results were produced this way, whereas each test speech was subjected to error bursts ranging in length from one to six at a time. In total, six error bursts

conditions, for 40 trials representing different error insertion points, repeated for the 13 test speech files, required 3120 trials to build the reference database.

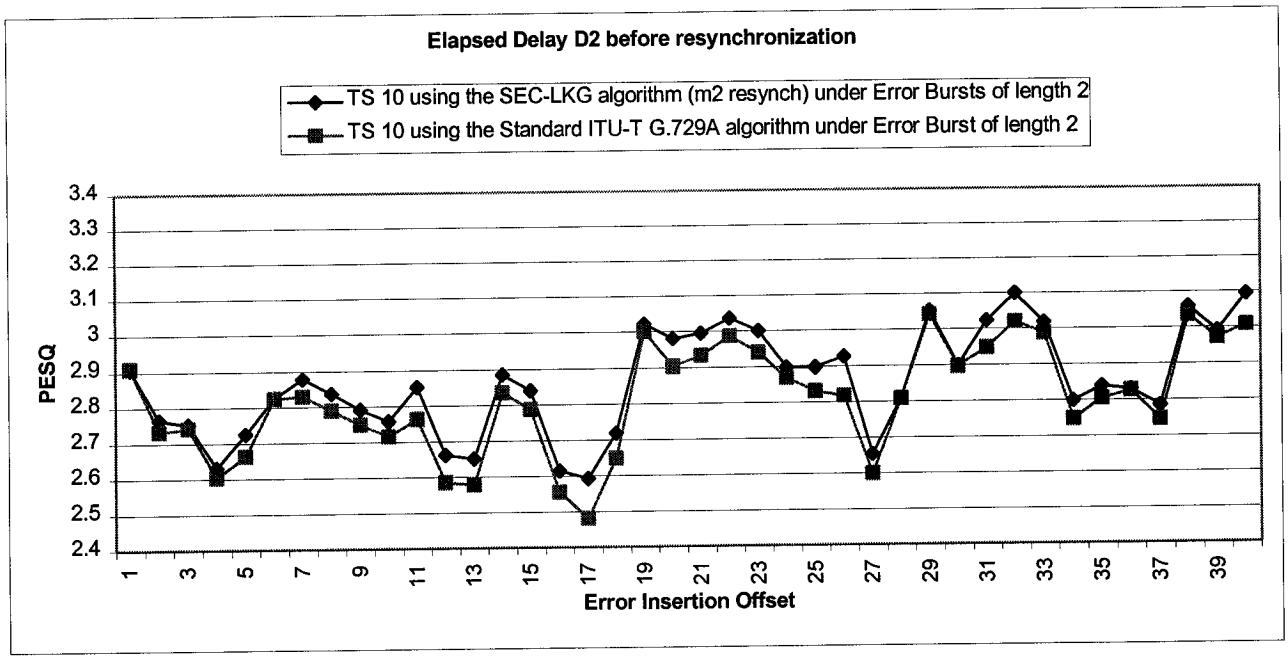


Figure 5.15 – The effects of offsetting the error insertion across a stream (TS 10 in this case) and using the ITU-T G.729A SEC-LKG algorithm with Error Bursts of length 2

Averaging the values in Figure 5.15 under respective error bursts conditions allows us to generate the graph presented in Figure 5.13 summarizing the performance of the standard algorithm for each specific test speech. Figure 5.13 is based on a set of 40 trials for each simulation representing a packet loss condition.

5.5 PERFORMANCE OF THE SEC ALGORITHM

In this section, the performance of the proposed SEC algorithm introduced in Section 5.1 with associated repair schemes, the *Reset To Default (RST)* and the *Last Known Good (LKG)* repair schemes presented in Section 5.3.2, will be thoroughly tested. The proposed algorithm (an extension, or add-on, feature to the Standard ITU-T G.729A algorithm) was designed with the

objective of minimizing the propagation of the state error using a closed-loop approach across the network.

Therefore, this section will expose the algorithm to network delays as well as packet error conditions for all the test speech vectors presented earlier. The previous sections described how data was produced and established a reference data set that we will use extensively in this section. The discussions and demonstrations will use the SEC-RST or SEC-LKG schemes interchangeably as they were both identically tested. However, discussions will highlight the differences between both schemes.

The PESQ method to objectively assess the speech quality performance was introduced in Chapter 2 and used thus far along with the squared error measurements. In this section, PESQ results will be used to demonstrate the performance of the proposed algorithm.

5.5.1 THE EFFECT OF NETWORK DELAYS

Of importance to the assessment of the proposed algorithms is the fact that the closed-loop method is established over an IP network. As discussed in Chapter 4, network latency exists when using IP networks (averaged delays reported to be between 30-100 ms and 31-106 ms including terminal serialization delays as per Figure 4.1). This latency or delay is time variant and presents an important challenge for the successful implementation of such an algorithm. The implementation of the SEC algorithm described in Section 5.3 was incorporated in the VWB application that forms the basis of our simulations. The network dimension is implemented using delays ranging from 10 ms (1 frame delay – represented by *D0* in the graphs) to 180 ms (18 frame delay – represented by *D17* in the graphs). We investigated every delay from 10 ms to 180 ms in increments of 10 ms. Figure

5.15 actually represents the results of the SEC-LKG algorithm after a re-synchronization delay, D2, representing 30 ms. Two sets are presented, the first set representing the reference performance (e.g. the Standard G.729A algorithm) and the second representing the set under evaluation (e.g. Standard G.729A with SEC-LKG), both exposed to the same error bursts. However, only the SEC algorithm is exposed to the network delays. In Figure 5.15, where the network delay is set to 30 ms, the SEC-LKG scheme almost constantly outperforms the performance of the standard algorithm (remember the PESQ results is the MOS – higher scores depict better quality). Recall Section 5.3.2 where repair schemes were investigated. The re-synchronization of the memory states is effectively mitigating the state error propagation by re-initializing the memory states at both the encoder and decoder sides after a short delay. The effect was a reconstructed signal with more similarities to the source signal. Comparing Figure 5.16 against Figure 5.15, where the same test speech (e.g. TS 10), the same error condition (e.g. error bursts of size 2) are used except for a network delay of 120 ms versus 30 ms, we can observe a lower performance, on average, of the SEC-LKG algorithm.

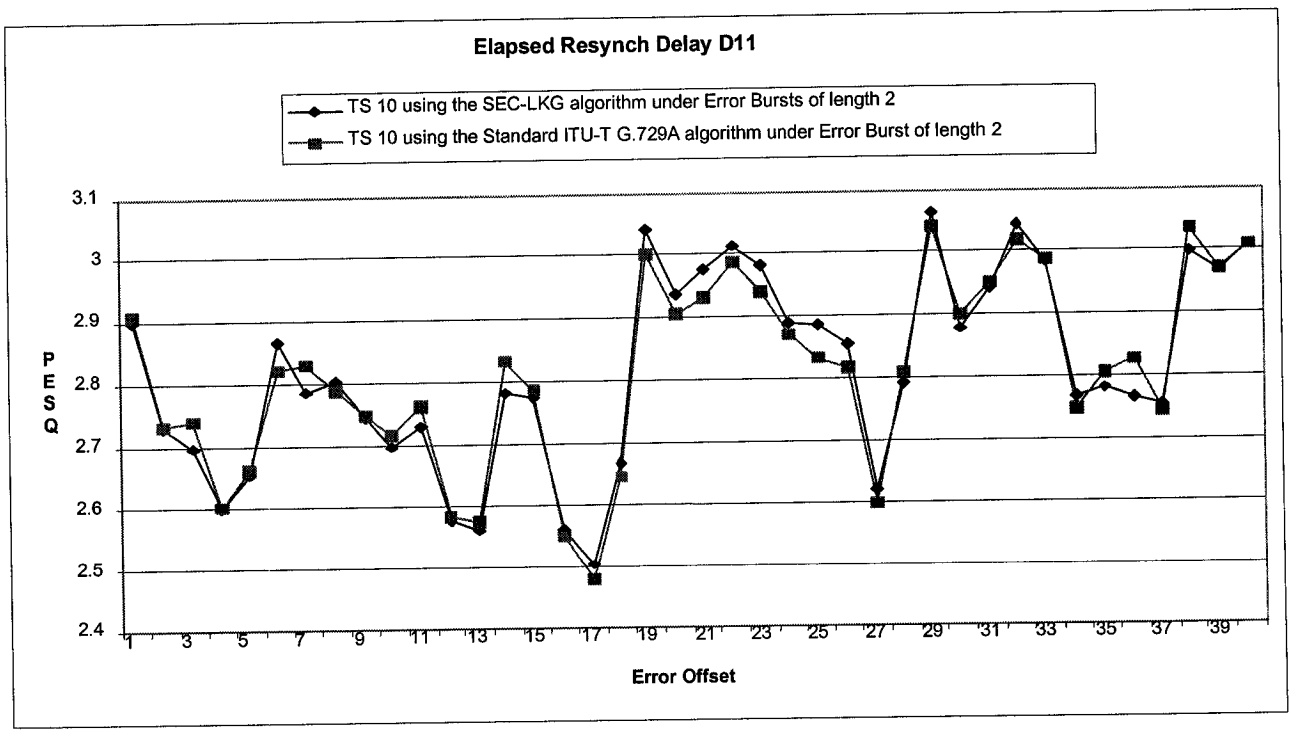


Figure 5.16 – The effects of network delays (delay of 12 frames in this case)

While Figure 5.15 showed an almost steady difference between the standard and the SEC-LKG algorithms, Figure 5.16 shows the SEC-LKG algorithm performing better in some instances and not in others (in contrast to the standard). It is clear at this time that the network delays do not positively contribute to the performance of the SEC-LKG algorithm. However, what is not so clear is how long it takes before the SEC-LKG algorithm under performs the standard algorithm or what is the maximum network delay for which the SEC-LKG algorithm will provide performance gains.

In Section 5.4.4 we described a scenario representing an error condition containing a set of 40 trials. Overall it took 3120 trials to gather the reference data for the 13 test speech files under the 6 error conditions. Similarly, 18 X 3120 = 56160 trials are required to gather 18 different network experiments or representations for the algorithm under evaluation (e.g. for each network delays D0 – D17), and twice that number to evaluate SEC-RST as well. It was done!

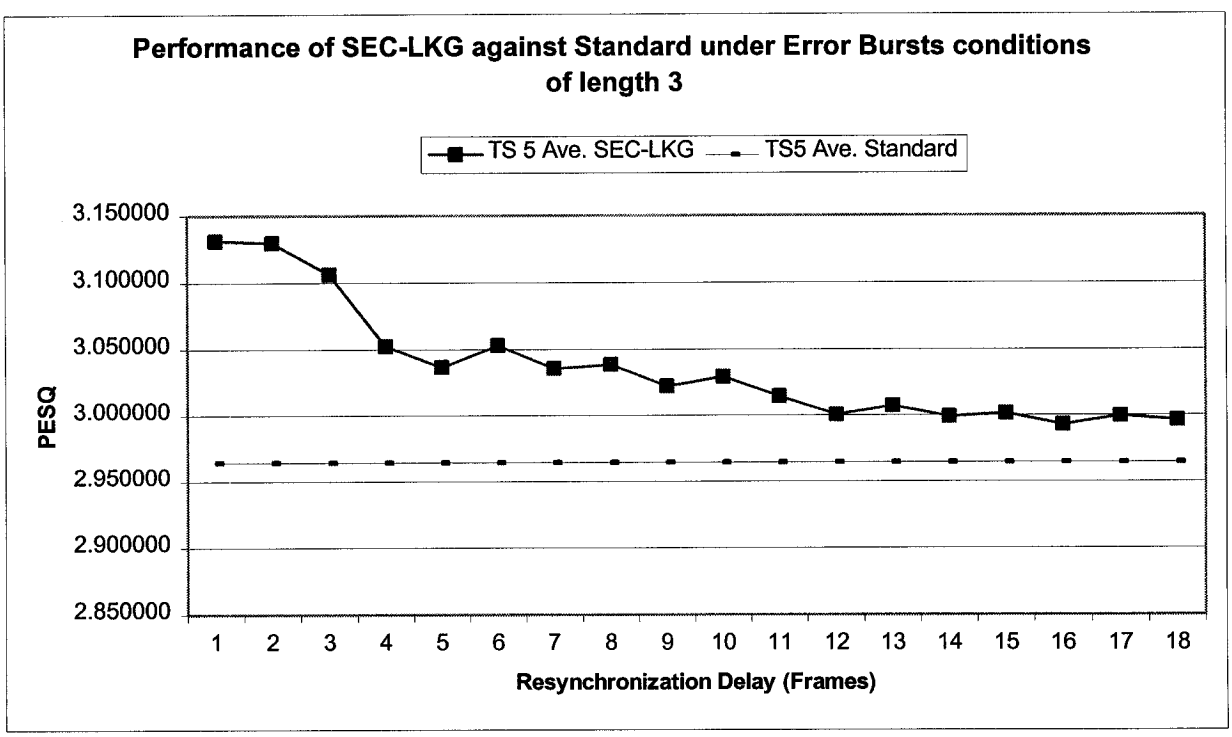


Figure 5.17 – Performance results of the ITU-T G.729A SEC-LKG algorithm under length 3 Error Bursts Conditions (using TS 5)

Figure 5.17 shows the averaged PESQ results at every investigated network delay (i.e. re-synchronization delay) using the SEC-LKG algorithm with TS 5. Each point in the figure represents the average PESQ results for 40 trials (one for each error insertion point/offset). Hence, the leftmost point represents the average PESQ result for re-synchronizing after one frame (D0), then point 2 for re-synchronizing after 2 frames (D1), until point 18 for re-synchronizing after 18 frames (D17). The averaged PESQ result from the associated reference data serves as the watermark or benchmark in Figure 5.17. For example, Figure 5.15 indicates the averaged PESQ result under error burst of length 3, using the standard G.729A algorithm with TS 5, to be 2.96.

In the case of Figure 5.17, we can see that the SEC-LKG algorithm performs well, providing an increase in PESQ of almost 0.2 when exposed to a network delay of 10 ms (e.g. 1 frame delay).

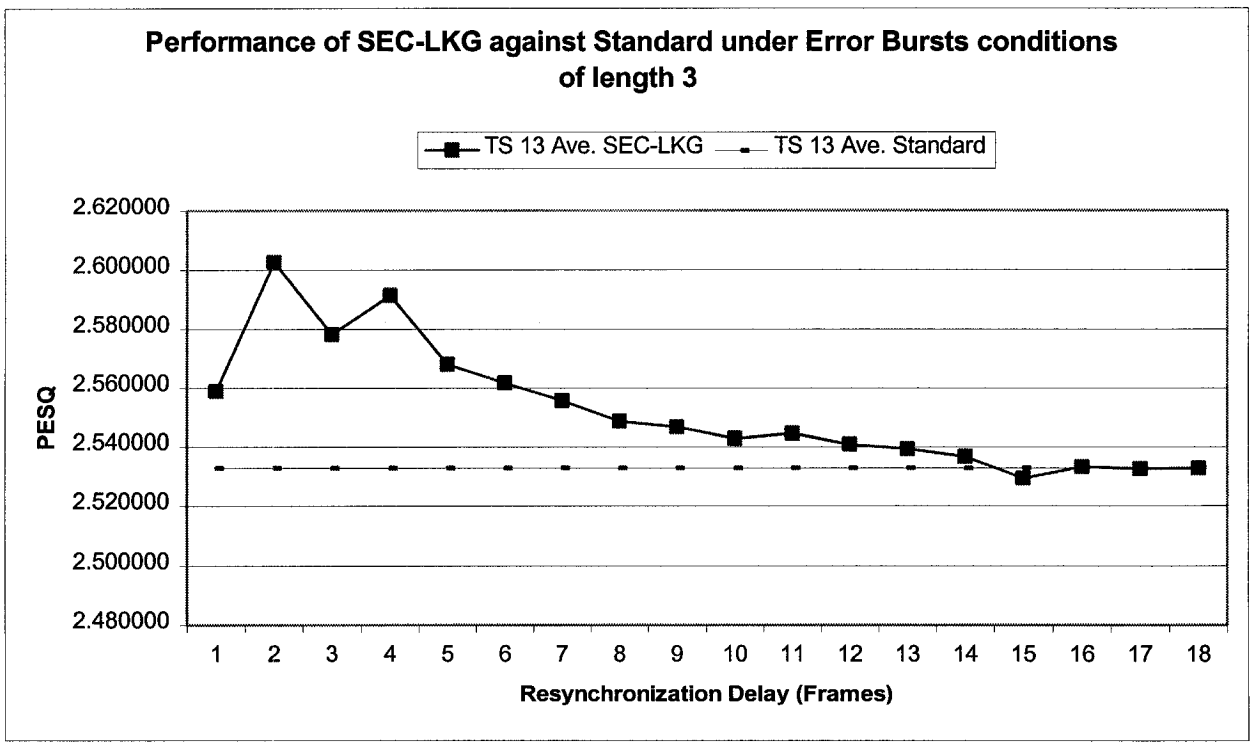


Figure 5.18 – Performance results of the ITU-T G.729A SEC-LKG algorithm under length 3 Error Bursts Conditions (using TS 13)

However, the performance gain decreases rapidly to provide PESQ gains around 0.08 after a network delay of 50 ms (e.g. delay of 5 frames). From that point, PESQ gains around 0.04 are still achieved. Figure 5.18 shows a similar behaviour for a different test speech (e.g. TS 13). However, the performance does not last as long as in the first example. We define the point at which the performance of the G.729A with SEC algorithm crosses the standard G.729A algorithm (our reference) as the *convergence* point. In Section 4.5.4, we subjectively assessed the time it took for the memory state elements to naturally converge to a synchronized state between the encoder and the decoder. We assume that the convergence point is the point where the natural convergence of the memory state occurs in the standard algorithm. The convergence point also tells us, in the case of Figure 5.18, that the SEC-LKG will degrade the speech performance from that point on. In the case of Figure 5.17 we can observe the convergence occurring after a delay of 18 frames (180 ms). We initially believed that a 180 ms delay was optimistic for the algorithm and opted for this limit based on the time it took to run these experiments (e.g. approximately two hours to generate each figure). On the other hand, Figure 5.18 shows convergence just past frame 14 (140 ms).

Similar results to those represented in Figure 5.17 and Figure 5.18 were obtained for all test speech vectors, and for both the SEC-LKG and SEC-RST schemes, under all error burst conditions. When reviewing the results, we observed that some exceptions exist where the convergence occurs early (e.g. after short delays), especially in the case of single missing packets (e.g. error bursts of length 1). Our observation correlates with the statements reported and associated findings in Chapter 4, to the effect that the standard G.729A algorithm is more robust to single packet lost (by being able to synthesize a better estimate of the missing segment) than it is for longer error bursts.

5.5.2 CONVERGENCE DUE TO NETWORK DELAY

In this section, we need to find the average convergence point in order to determine if the proposed algorithm is suited for operation in an IP network environment. The convergence point will define the maximum network delay until which the proposed algorithm will contribute positively to the standard algorithm (from a PESQ point of view).

CONVERGENCE DELAY - SEC-RST						
	Burst sizes (Bx where x is the size or length or the burst)					
Test Speech File No.	B1	B2	B3	B4	B5	B6
1	3	6	5	11	11	15
2	3	4	10	9	12	13
3	5	7	10	11	12	14
4	18	18	12	18	18	16
5	18	18	18	18	18	18
6	5	8	10	15	18	18
7	3	16	18	18	18	18
8	5	10	16	16	18	16
9	3	8	9	11	12	14
10	18	18	18	18	18	18
11	5	5	9	9	10	10
12	4	18	14	12	12	8
13	10	18	18	18	18	19
Average Convergence Time (in frames)	**7.769231**	**11.84615**	**12.84615**	**14.15385**	**15**	**15.15385**

Table 5.3 – Convergence results for the SEC-RST algorithm

CONVERGENCE DELAY - SEC-LKG						
	Burst sizes (Bx where x is the size or length or the burst)					
Test Speech File No.	B1	B2	B3	B4	B5	B6
1	3	4	5	6	8	8
2	3	4	9	10	11	14
3	3	4	5	11	13	14
4	18	13	13	13	12	18
5	16	18	18	18	18	18
6	5	7	10	13	13	18
7	3	11	18	17	17	18
8	5	9	11	11	18	13
9	3	4	10	11	12	13
10	5	18	18	18	18	18
11	4	7	8	10	11	11
12	3	12	11	7	7	4
13	10	12	15	17	17	18
Average Convergence Time (in frames)	**6.230769**	**9.461538**	**10.76923**	**12.46154**	**13.46154**	**14.23077**

Table 5.4 - Convergence results for the SEC-LKG algorithm

The determination of the convergence point was often subjective and Figure 5.18 is a good example. By observing closely points 16, 17, and 18, we can see that they are slightly above the performance of the standard algorithm. In such cases we often elected to declare the convergence to occur at point 15, all considered. Table 5.3 and 5.4 show the convergence results for all test vectors for the SEC-RST and SEC-LKG schemes respectively. Highlighted values represent undetermined convergence point past 18 frames. The averages for Table 5.3 and Table 5.4 are plotted in Figure 5.19.

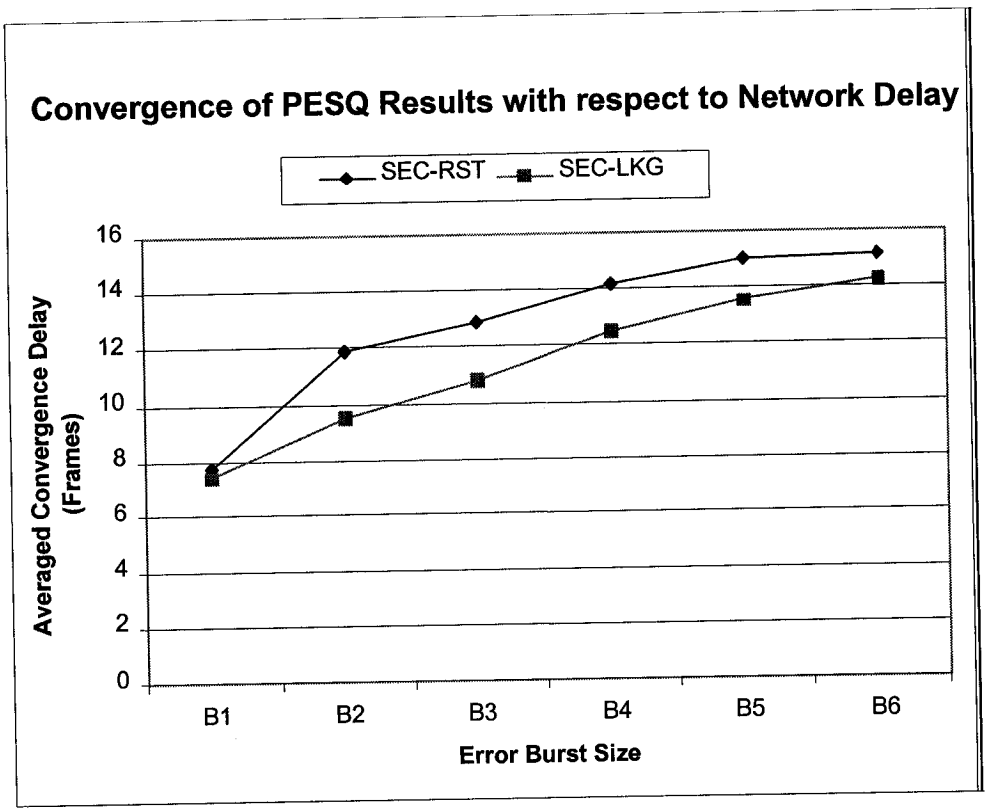


Figure 5.19 – Averaged convergence results for G.729A SEC-RST versus G.729A SEC-LKG

An analysis of Table 5.3 and Table 5.4 (e.g. the highlighted values/cells) leads us to conclude that the averaged results will be slightly higher (e.g. depict later convergence on average) and Figure 5.19 presents a good approximation of the algorithm behavior. Figure 5.19 shows that the SEC-RST constantly exhibit later convergence (e.g. 15 frames for error burst of length 6) compared to the

SEC-LKG. Another interesting observation is that, as the error burst size increases, the time to converge also increases. Since the unidirectional network delay is between 30-100 ms (as discussed in chapter 4) we expect both the SEC-RST and the SEC-LKG schemes to be usable in a network environment especially considering that many convergence point occur later than 180 ms.

5.5.3 EVALUATING THE ALGORITHM PERFORMANCE

So far, looking at Figures 5.17, 5.18, and 5.19, it looks like both the SEC-RST and the SEC-LKG schemes would be beneficial to minimize state error propagation. In this section we describe and demonstrate the evaluation approach used to assess the averaged performance of the proposed algorithm. Considering that the unidirectional network delay is between 30-100 ms, we could assume that a minimum round trip delay (e.g. to go and back) would approximately be in the range 60-200 ms. Consequently, the minimum network delay at which we can expect the algorithm to be used is 60 ms. This implies that the values falling prior to this minimum, when looking at Figure 5.17 and Figure 5.18, are not usable to assess the average algorithm performance. This will have a marked impact on the overall expected value since we constantly observed the early values (e.g. up to 60 ms) in graphs similar to that of Figure 5.17 to provide the highest performance gains. To facilitate the discussions, we define the *utilization window* to represent the re-synchronization delays between 60 ms and 180 ms (e.g. 6 frames = D5 to 18 frames = D17 – see Figure 5-30).

Figure 5.20 shows the averaged performance gains of SEC-RST for TS 10 under all error burst conditions, over the *utilization window*. A first observation from Figure 5.20, is that the SEC algorithm is only used when errors are detected. Consequently, if there is no error, as in the case of the first point (e.g. B0), then the performance is the same as if only using the standard G.729A algorithm. Looking at the remaining error conditions, we note that the SEC-RST performs better, on

average, than the standard algorithm. However, the PESQ performance gains are marginal (i.e. way less than 0.0625 and probably less than 0.0125). The same observations apply to the SEC-LKG performance, as displayed in Figure 5.21.

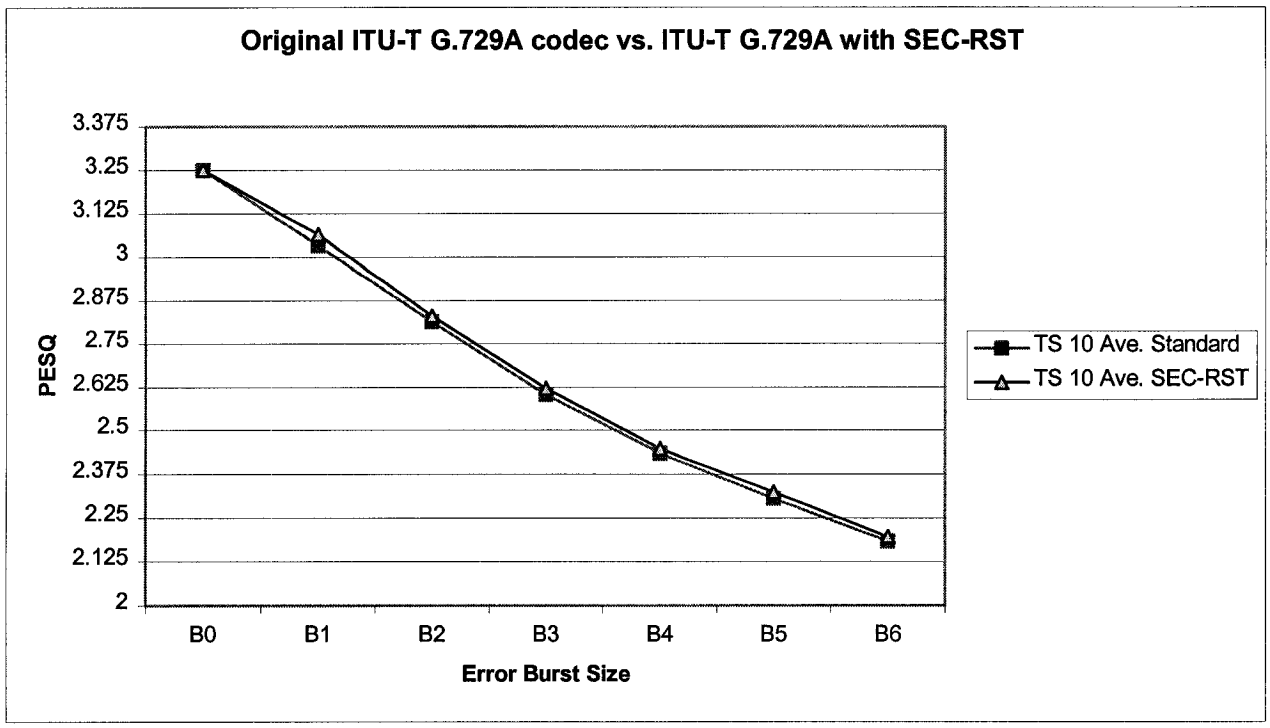


Figure 5.20 – PESQ results for TS 10 using G.729A SEC-RST under different Error Bursts length

The performance data for Figure 5.20 and Figure 5.21 for SEC-RST and SEC-LKG respectively, was gathered in a same fashion for the remainder of the test speech vectors. We were able to observe minor variations in the performance gains where some graphs represented degradations for error bursts of length one, and sometimes but rarely, for length two. Bursts of length 3, 4, 5, and 6 constantly showed both the SEC-RST and the SEC-LKG to perform better than the original algorithm. In general, performance gains where observed but were marginal in all cases.

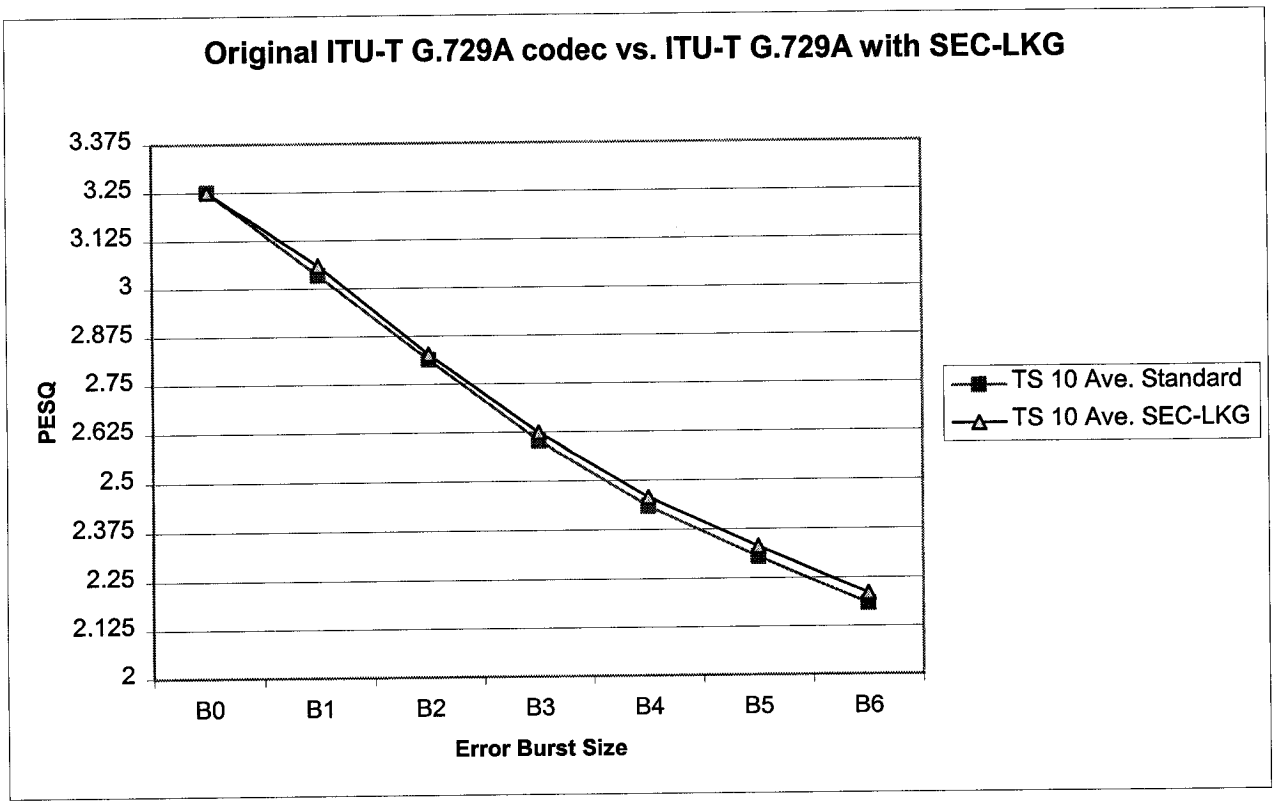


Figure 5.21 – PESQ results for TS 10 using G.729A SEC-LKG under different Error Bursts length

Closely observing Figure 5.21 we are able to see an increase in performance as the burst size increases. This behaviour suggests that the algorithm could provide better performance for error bursts of length 7 and above. This behaviour seems to be the norm but is sometimes difficult to evaluate looking at the graph (e.g. see Figure 5.20) but is obvious when looking at the tabulated results (refer to the corresponding TS entries from Table 5.5 to Table 5.10).

5.5.4 SEC OVERALL PERFORMANCE

Results demonstrating the overall performance or expected performance of the proposed algorithm are presented in a consolidated manner in this section. Figures 5.22 to 5.27 and associated Tables 5.5 to 5.10 show the SEC-RST and SEC-LKG performance for all test speech files and under all error bursts conditions, taken over the *utilization window*. Discussion follows the figures and tables.

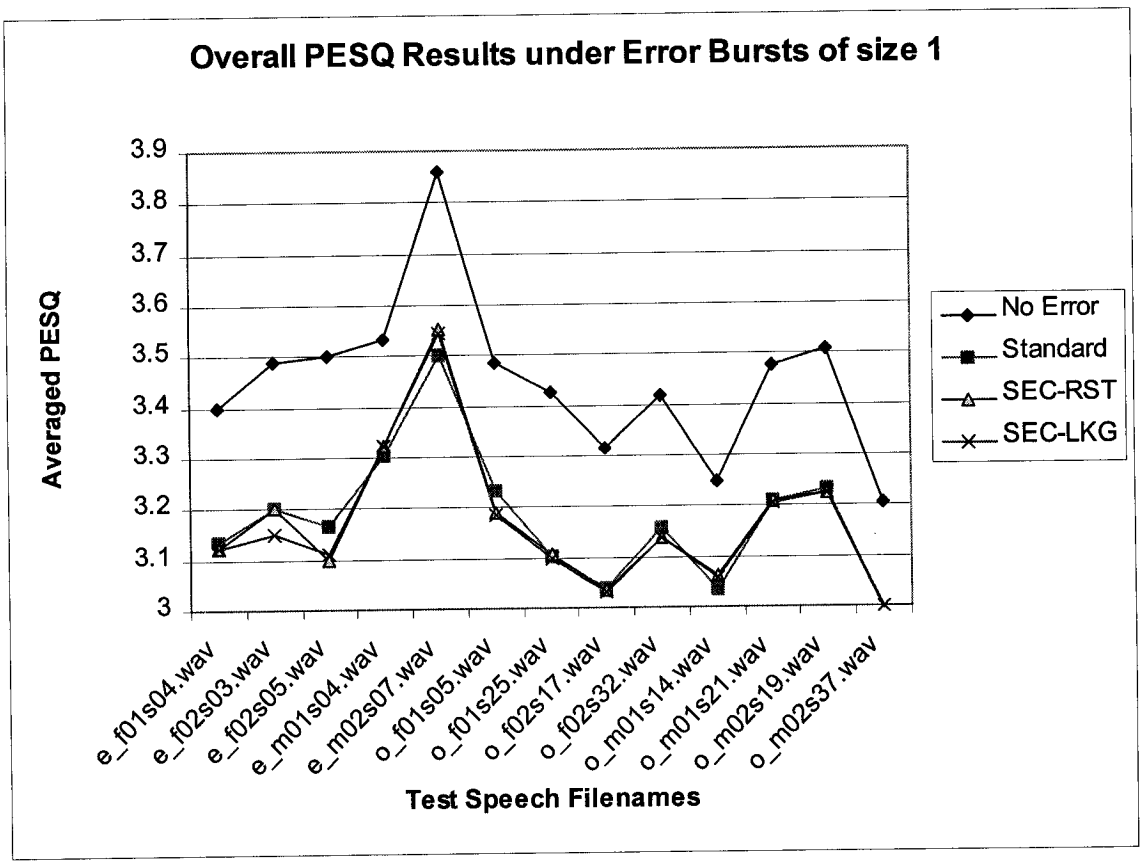


Figure 5.22 – Overall PESQ results under Error Bursts of size 1

PESQ	FILENAME	No Error	Standard	SEC-RST	SEC-LKG
B1	e_f01s04.wav	3.400	3.135	3.123	3.123
	e_f02s03.wav	3.488	3.202	3.154	3.148
	e_f02s05.wav	3.503	3.164	3.100	3.112
	e_m01s04.wav	3.531	3.304	3.325	3.324
	e_m02s07.wav	3.862	3.502	3.552	3.546
	o_f01s05.wav	3.486	3.231	3.190	3.184
	o_f01s25.wav	3.425	3.104	3.107	3.098
	o_f02s17.wav	3.315	3.039	3.035	3.031
	o_f02s32.wav	3.419	3.157	3.139	3.140
	o_m01s14.wav	3.250	3.034	3.064	3.059
	o_m01s21.wav	3.477	3.210	3.209	3.207
	o_m02s19.wav	3.511	3.233	3.226	3.223
	o_m02s37.wav	3.206	2.996	2.996	3.001
OVERALL AVERAGE		**3.452**	**3.178**	**3.171**	**3.169**

Table 5.5 – Overall PESQ results under Error Bursts of size 1

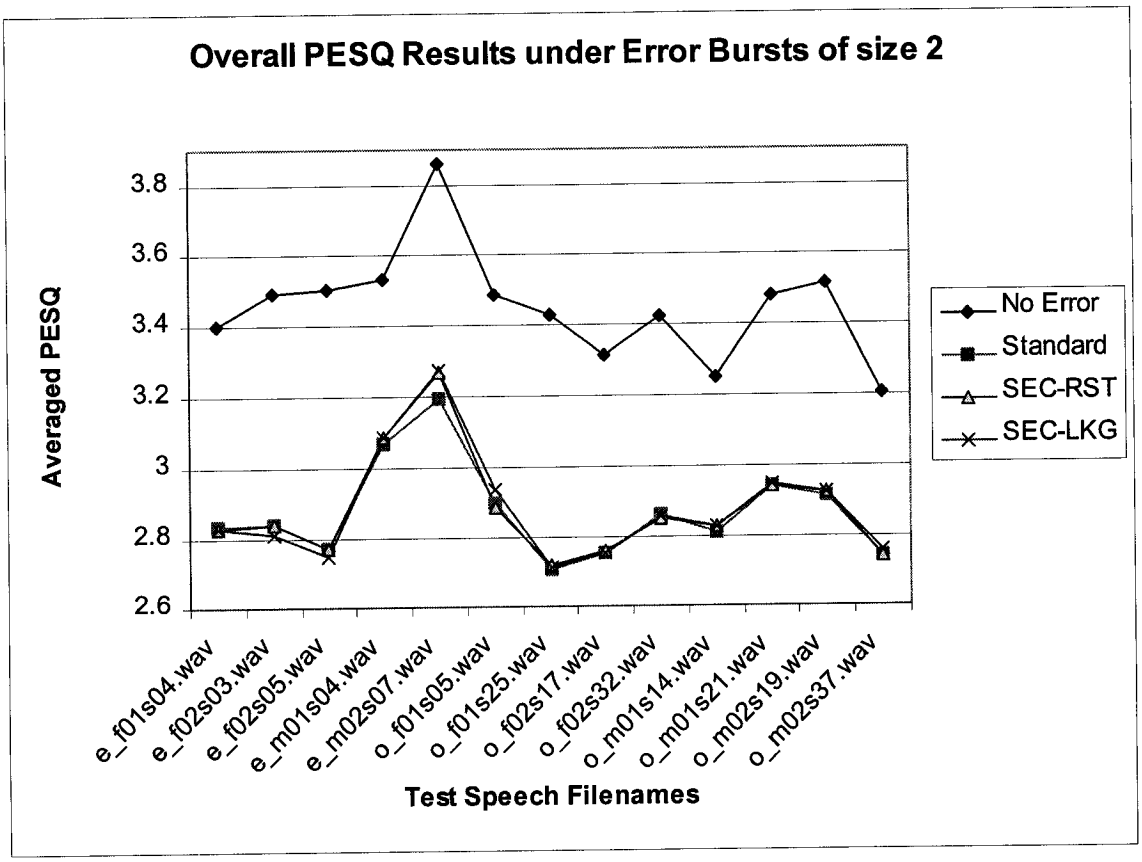


Figure 5.23 – Overall PESQ results under Error Bursts of size 2

PESQ	FILENAME	No Error	Standard	SEC-RST	SEC-LKG
B2	e_f01s04.wav	3.400	2.836	2.827	2.827
	e_f02s03.wav	3.488	2.840	2.815	2.813
	e_f02s05.wav	3.503	2.772	2.772	2.747
	e_m01s04.wav	3.531	3.066	3.083	3.084
	e_m02s07.wav	3.862	3.193	3.268	3.273
	o_f01s05.wav	3.486	2.897	2.888	2.935
	o_f01s25.wav	3.425	2.711	2.718	2.712
	o_f02s17.wav	3.315	2.756	2.757	2.754
	o_f02s32.wav	3.419	2.864	2.852	2.854
	o_m01s14.wav	3.250	2.813	2.830	2.827
	o_m01s21.wav	3.477	2.943	2.944	2.946
	o_m02s19.wav	3.511	2.912	2.925	2.925
	o_m02s37.wav	3.206	2.743	2.743	2.758
OVERALL AVERAGE		**3.452**	**2.873**	**2.879**	**2.881**

Table 5.6 – Overall PESQ results under Error Bursts of size 2

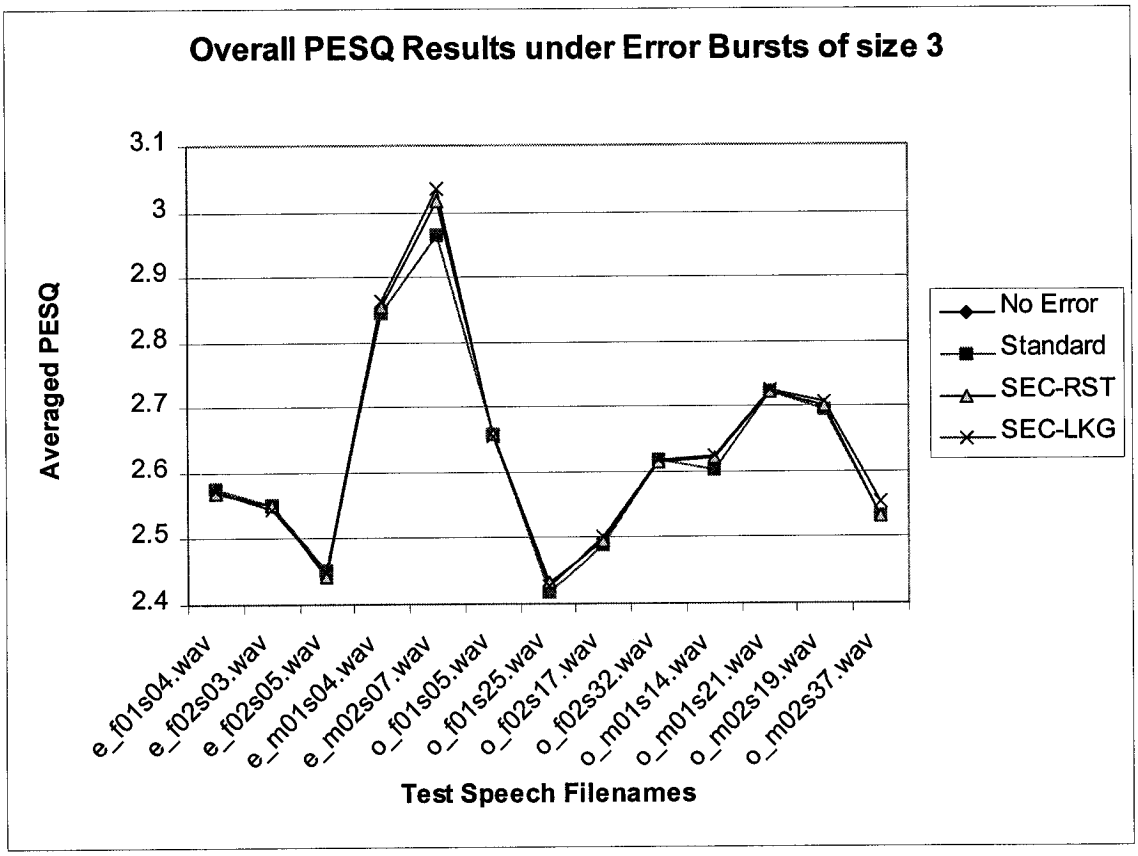


Figure 5.24 – Overall PESQ results under Error Bursts of size 3

PESQ	FILENAME	No Error	Standard	SEC-RST	SEC-LKG
B3	e_f01s04.wav	3.400	2.575	2.569	2.571
	e_f02s03.wav	3.488	2.551	2.549	2.545
	e_f02s05.wav	3.503	2.453	2.444	2.450
	e_m01s04.wav	3.531	2.846	2.855	2.864
	e_m02s07.wav	3.862	2.964	3.018	3.036
	o_f01s05.wav	3.486	2.657	2.657	2.656
	o_f01s25.wav	3.425	2.417	2.433	2.429
	o_f02s17.wav	3.315	2.489	2.496	2.501
	o_f02s32.wav	3.419	2.617	2.616	2.619
	o_m01s14.wav	3.250	2.602	2.622	2.623
	o_m01s21.wav	3.477	2.722	2.723	2.727
	o_m02s19.wav	3.511	2.696	2.701	2.706
	o_m02s37.wav	3.206	2.533	2.533	2.552
OVERALL AVERAGE		**3.452**	**2.625**	**2.632**	**2.637**

Table 5.7 – Overall PESQ results under Error Bursts of size 3

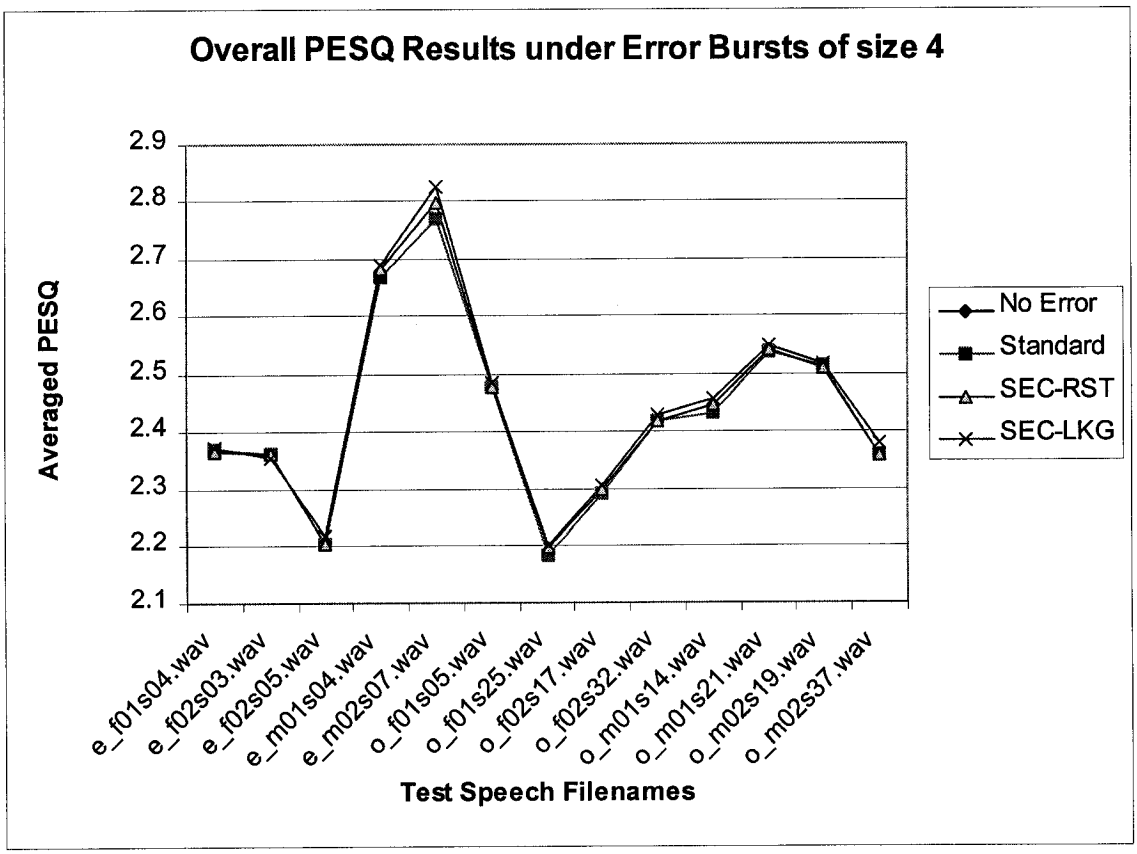


Figure 5.25 – Overall PESQ results under Error Bursts of size 4

PESQ	FILENAME	No Error	Standard	SEC-RST	SEC-LKG
B4	e_f01s04.wav	3.400	2.369	2.368	2.373
	e_f02s03.wav	3.488	2.361	2.351	2.355
	e_f02s05.wav	3.503	2.207	2.207	2.220
	e_m01s04.wav	3.531	2.669	2.681	2.691
	e_m02s07.wav	3.862	2.769	2.800	2.827
	o_f01s05.wav	3.486	2.479	2.480	2.485
	o_f01s25.wav	3.425	2.185	2.198	2.200
	o_f02s17.wav	3.315	2.294	2.299	2.307
	o_f02s32.wav	3.419	2.421	2.419	2.428
	o_m01s14.wav	3.250	2.434	2.448	2.457
	o_m01s21.wav	3.477	2.540	2.542	2.549
	o_m02s19.wav	3.511	2.513	2.509	2.518
	o_m02s37.wav	3.206	2.361	2.361	2.380
OVERALL AVERAGE		**3.452**	**2.431**	**2.436**	**2.445**

Table 5.8 – Overall PESQ results under Error Bursts of size 4

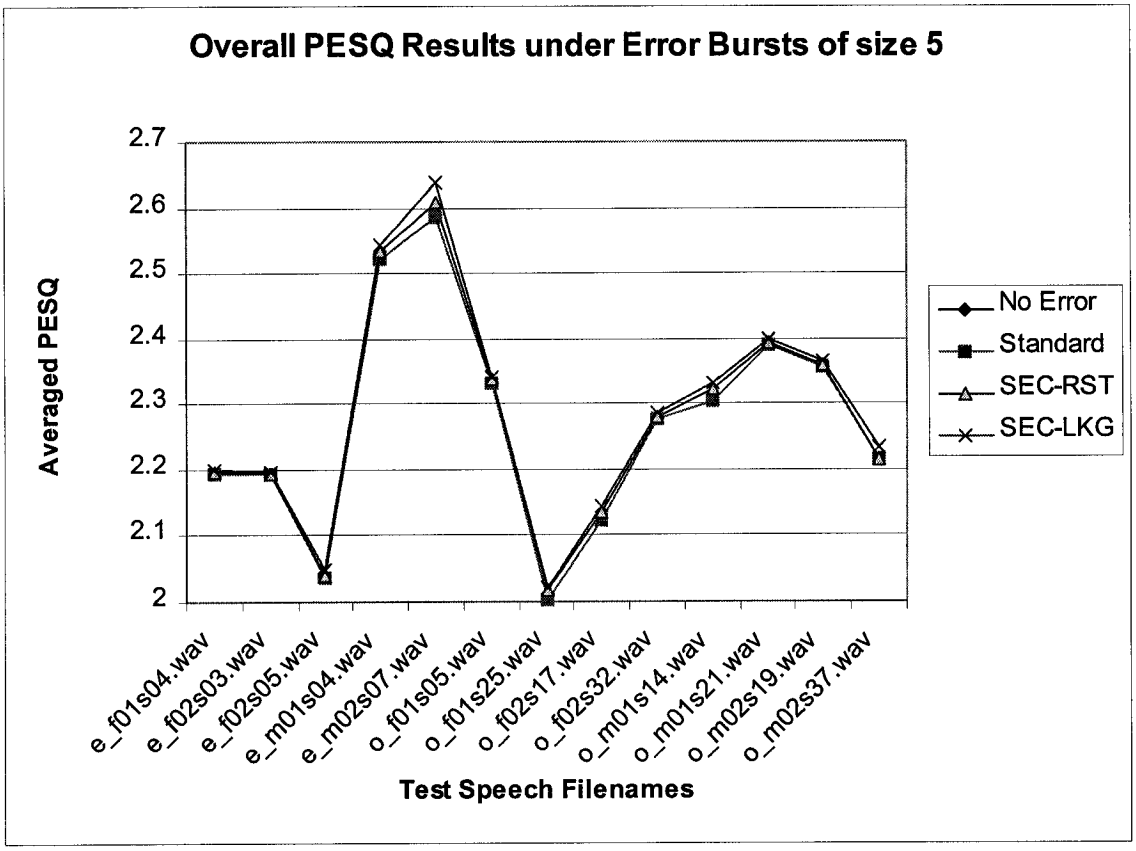


Figure 5.26– Overall PESQ results under Error Bursts of size 5

PESQ	FILENAME	No Error	Standard	SEC-RST	SEC-LKG
B5	e_f01s04.wav	3.400	2.193	2.196	2.201
	e_f02s03.wav	3.488	2.194	2.198	2.196
	e_f02s05.wav	3.503	2.038	2.041	2.050
	e_m01s04.wav	3.531	2.523	2.534	2.543
	e_m02s07.wav	3.862	2.587	2.606	2.638
	o_f01s05.wav	3.486	2.330	2.337	2.341
	o_f01s25.wav	3.425	2.004	2.019	2.022
	o_f02s17.wav	3.315	2.122	2.136	2.144
	o_f02s32.wav	3.419	2.275	2.279	2.284
	o_m01s14.wav	3.250	2.304	2.323	2.330
	o_m01s21.wav	3.477	2.389	2.394	2.400
	o_m02s19.wav	3.511	2.357	2.358	2.365
	o_m02s37.wav	3.206	2.215	2.215	2.234
OVERALL AVERAGE		**3.452**	**2.272**	**2.280**	**2.288**

Table 5.9 – Overall PESQ results under Error Bursts of size 5

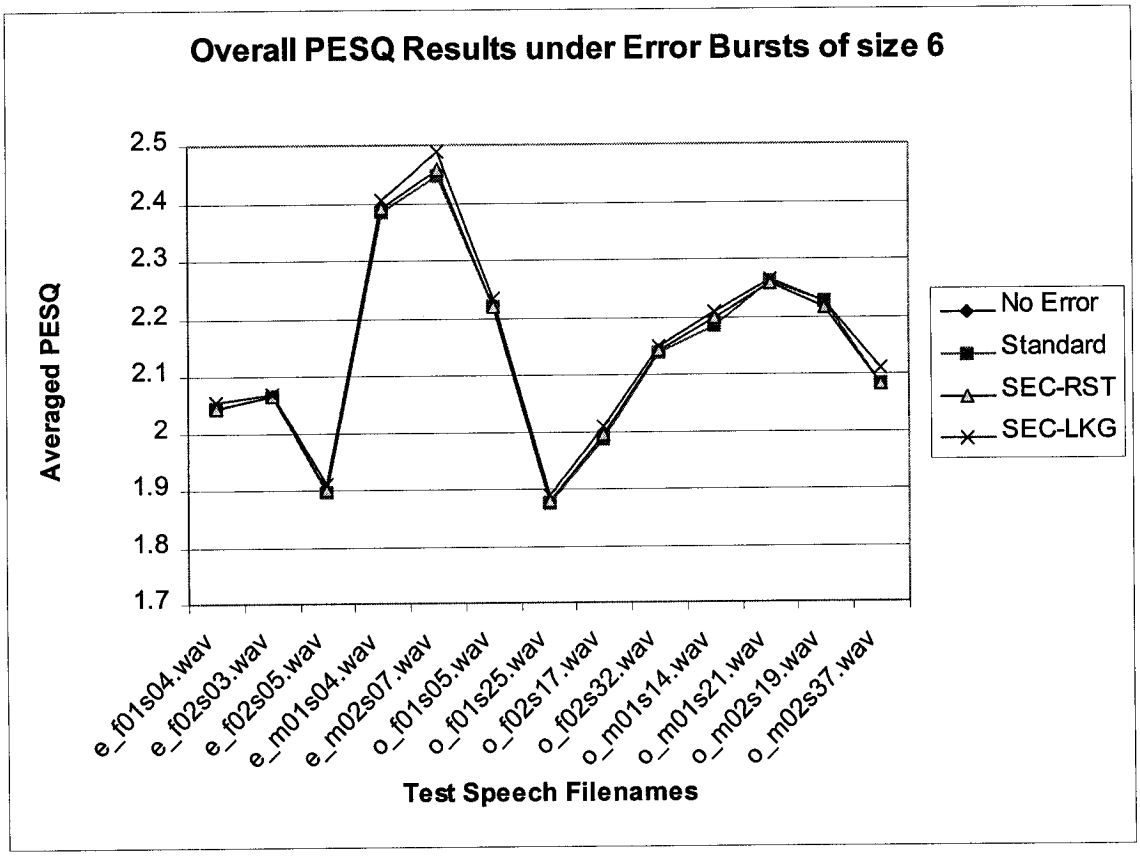


Figure 5.27 – Overall PESQ results under Error Bursts of size 6

PESQ	FILENAME	No Error	Standard	SEC-RST	SEC-LKG
B6	e_f01s04.wav	3.400	2.045	2.045	2.054
	e_f02s03.wav	3.488	2.064	2.066	2.067
	e_f02s05.wav	3.503	1.898	1.901	1.912
	e_m01s04.wav	3.531	2.384	2.393	2.405
	e_m02s07.wav	3.862	2.447	2.457	2.490
	o_f01s05.wav	3.486	2.219	2.220	2.232
	o_f01s25.wav	3.425	1.874	1.880	1.889
	o_f02s17.wav	3.315	1.989	1.994	2.008
	o_f02s32.wav	3.419	2.140	2.141	2.149
	o_m01s14.wav	3.250	2.183	2.197	2.210
	o_m01s21.wav	3.477	2.260	2.257	2.266
	o_m02s19.wav	3.511	2.226	2.215	2.227
	o_m02s37.wav	3.206	2.082	2.082	2.109
OVERALL AVERAGE		**3.452**	**2.139**	**2.142**	**2.155**

Table 5.10 – Overall PESQ results under Error Bursts of size 6

Summary of Results

Figure 5.28 draws the overall expected PESQ result taken from Tables 5.5 to Table 5.10.

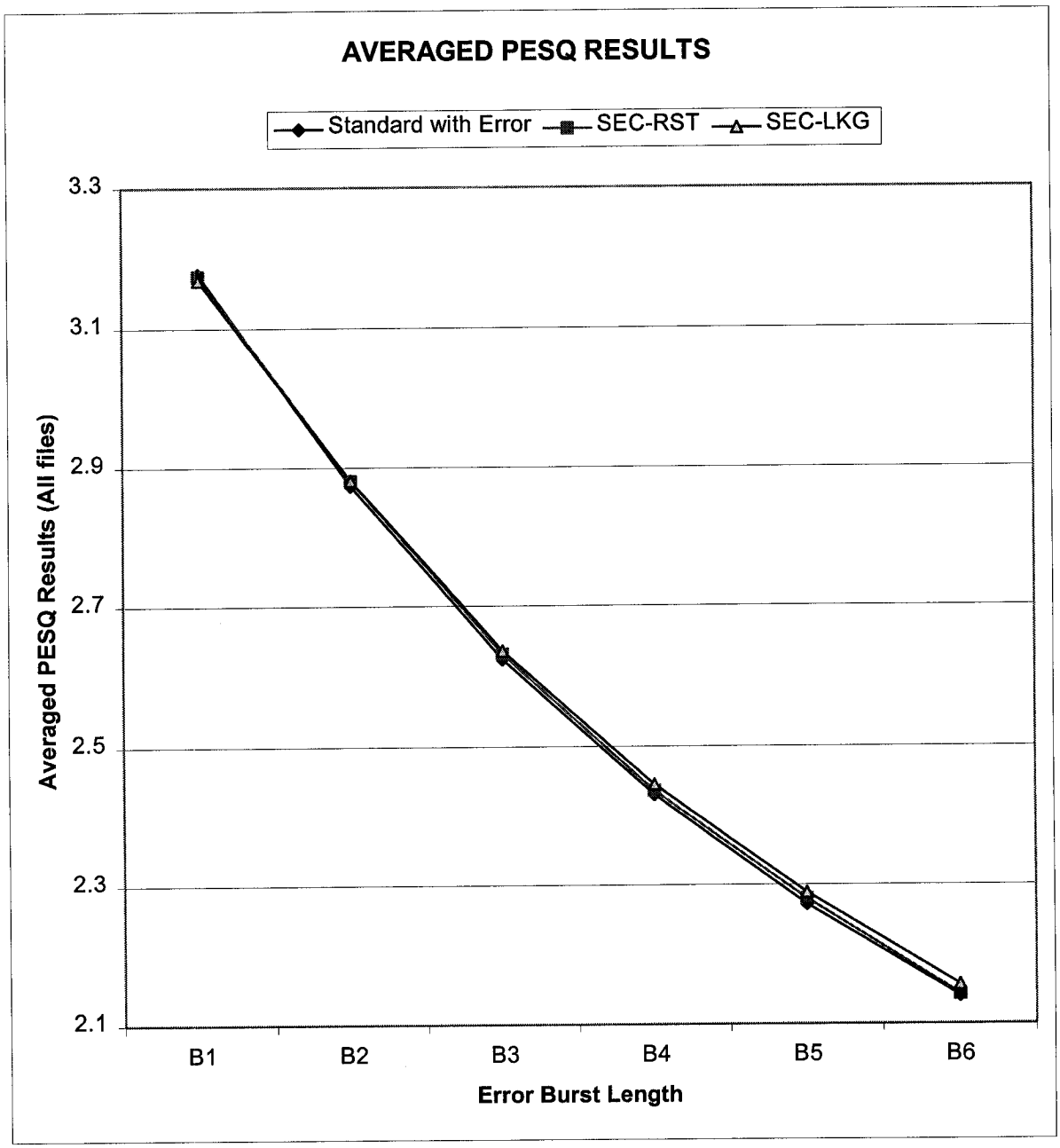


Figure 5.28 – Overall Averaged PESQ Results

Figure 5.29 also shows the overall expected values, but for MSE values as tracked by the VWB application.

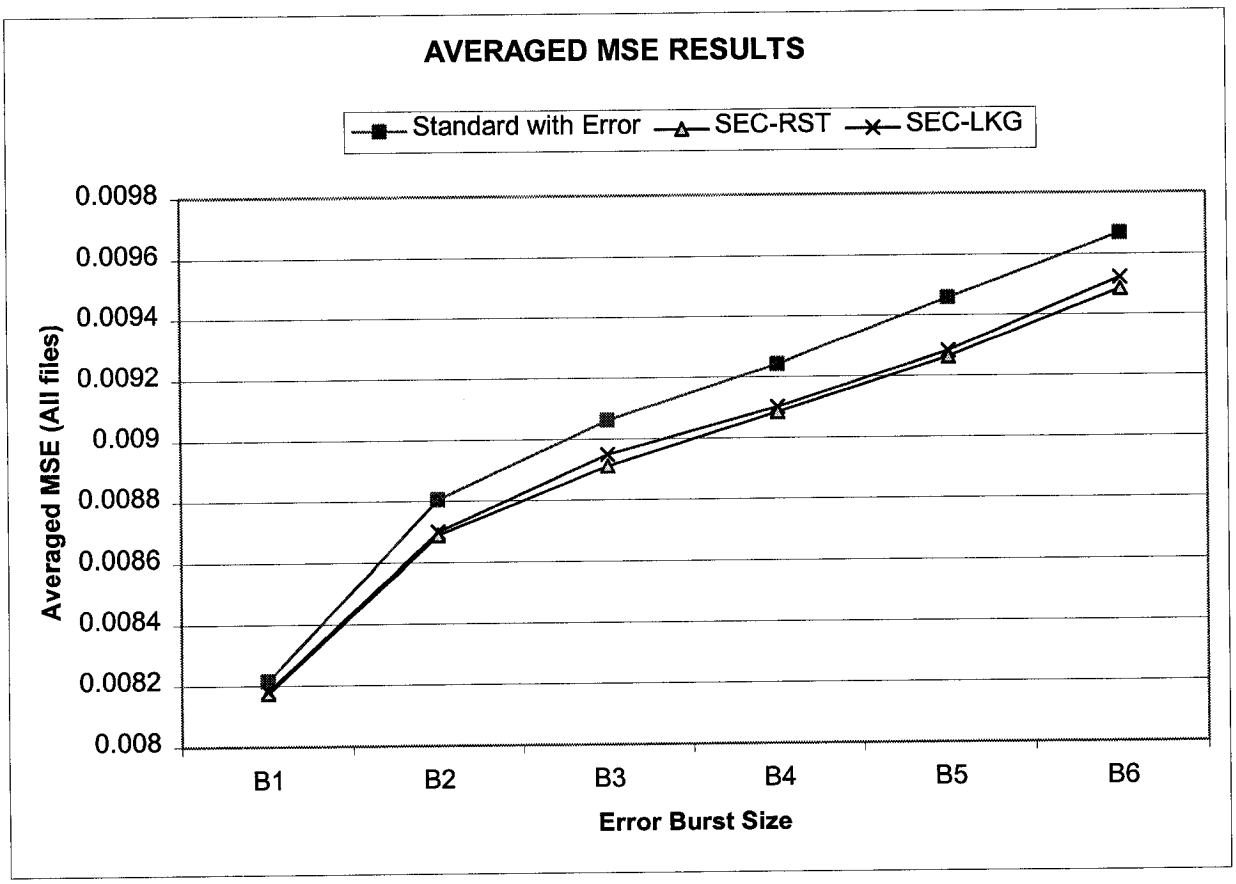


Figure 5.29 – Overall Averaged MSE Results

Overall the SEC-LKG performs best except for error bursts of length 1. Similarly, SEC-RST follows in second place. This means that on average both the SEC-LKG and SEC-RST algorithms perform better than the standard algorithm. What is surprising is that we only get a marginal overall PESQ gain. However, Figure 5.29 shows a better demarcation between the original algorithm and the SEC algorithm. In this case, SEC-RST provides lower MSE values than the LGK. We explain this with the knowledge we acquired when running the trials. Since the memory state is re-initialized with zeros, the reconstructed speech will initially exhibit less energy (observations say it last for half a frame to a frame, for example, see Figure 5.6) at the re-synchronization point. To the opposite, we assume that LKG always re-initializes its memory state with some values, hence would experience a random behavior compared to RST.

Discussions

In this section we initially summarize the approach taken to generate the results and then discuss or rationalize the results obtained. In the course of the initial investigation we realized that we could enhance the speech quality following packet losses if memory state elements were immediately re-initialized following the loss. However attractive this observation was, its implementation is not practically feasible due to the network delay constraint and drove the development of the SEC algorithm. As covered in Section 4.5.4, a manual approach to inspecting memory state provided an estimate of the memory state convergence. The longest convergence delay was attributed to signal dependent memory state elements m^b and m^c which took on average 27 and 25 good frames respectively to converge to a synchronized state (51 and 47 good frames respectively without applying any threshold – in our case when the result was within 10% of the expected result). These observations and knowledge acquired with respect to quantifying network delays (see Section 4.2.2) provided sufficient evidence that a closed-loop algorithm across the network was feasible. However, the investigation performed, in Section 5.3.2, to define the basic repair schemes to use within the SEC algorithm, followed by subjective testing, conducted us to use repair schemes that only re-initialize memory state elements m^b (remember that memory state m^a, m^d, *and* m^e converged rapidly to a resynchronized memory state).

To simulate the effect of the network we elected to apply delays on a frame basis. The reason is simple; the memory state error really changes when the memory state elements are updated and not as a direct relation to time. Reference data using the standard ITU-T G.729A algorithm was generated to create a comparison basis against which we tested the proposed algorithm performance. A large set of simulations using all test speech files, and several scenarios, where every network delay from a minimum network delay of 10 ms (1 frame) to a maximum network delay of 180 ms (18 frames) were applied as well as error bursts conditions. Error bursts conditions consisted of

distributed small error bursts of length 1 to length 6 representing packet loss rates in the neighbourhood of 2.5, 5, 7.5, 10, 12.5, and 15 percent respectively (versus a single long error burst).

The data collected allowed us to plot results for each scenario where Figure 5.15 and Figure 5.16 are samples of the results generated. As explained earlier in this chapter, the figures highlighted that both SEC-RST and SEC-LKG were potentially providing gains on average but a different representation of results was required to really understand the effect of network delays on the memory state re-synchronization. Figure 5.17 and Figure 5.18 are sample plots of such a representation where the plotted curve provides a characteristic pattern. The pattern clearly shows the PESQ performance of the algorithm with respect to the standard G.729A algorithm performance (e.g. the watermark). The first observation of the characteristic pattern is that the gain or PESQ performance of the SEC algorithm is directly related to the network delay. As expected, the SEC algorithm provides best benefits when the network delay is small (e.g. smaller than 60 ms) allowing the memory state re-initialization to occur shortly following the packet loss. When network delays are somewhat longer, some performance gains may be achieved using the SEC algorithm but for even longer delays, the performance gains are very small, hence negligible, and for some test speech files, degrading the speech quality beyond that of the standard algorithm.

Figure 5.30 brings several data representations together, along with a categorization scale based on network delay or performance. The scale attempts to weigh the benefit of using the SEC algorithm with respect to the achieved network performance. A sketch of the characteristic pattern or curve is introduced on the left side of Figure 5.30 (rotated 90°). The centre plot summarizes the results of all characteristic patterns using the *utilization window*, and the scale on the rightmost side of the figure provides a framework that will be used to facilitate discussions.

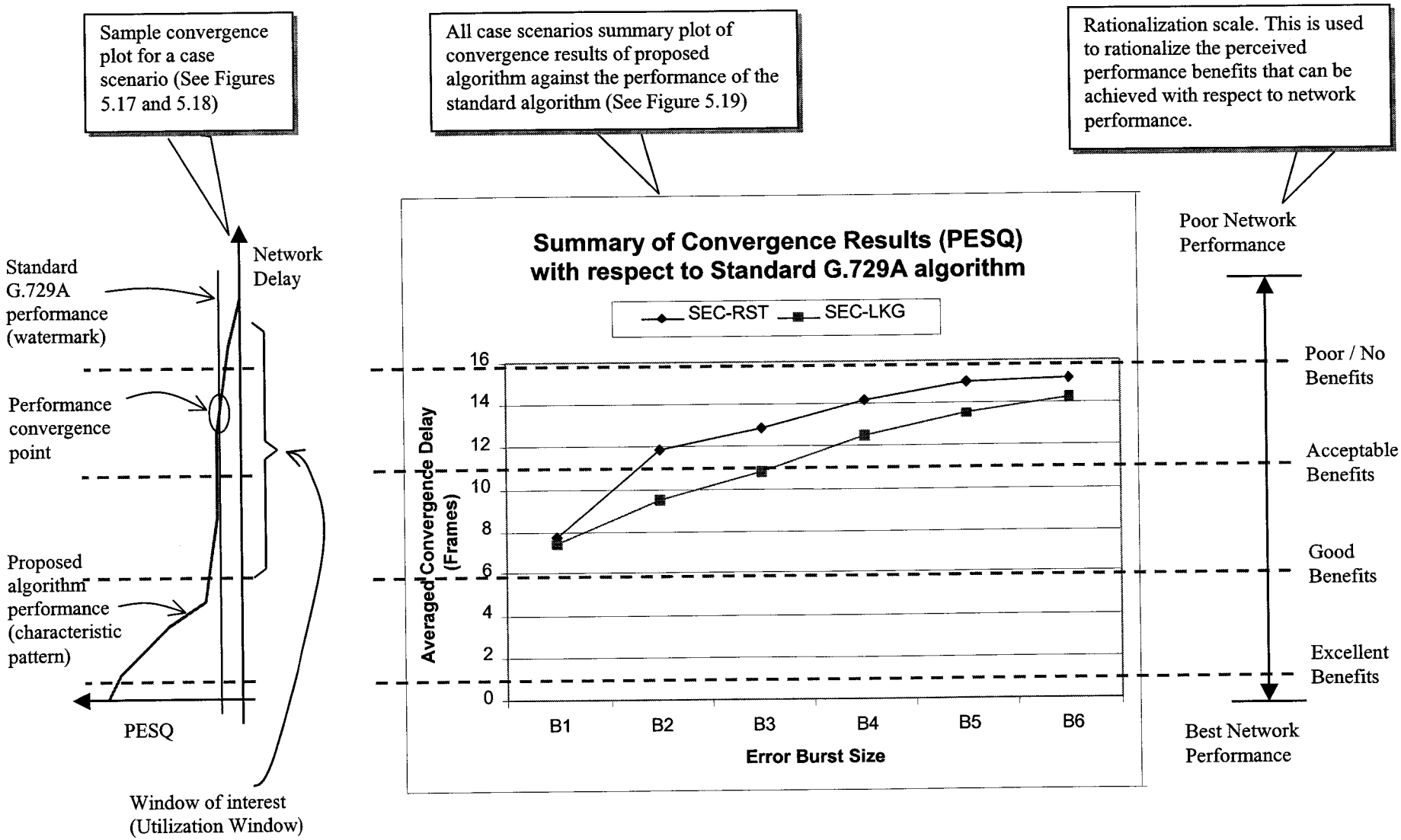


Figure 5.30 – Consolidated evaluation of algorithm performance (Algorithm convergence vs. network delay)

The circle shown on the leftmost plot of Figure 5-30 indicates the PESQ performance convergence point, that is, the intersection of the proposed algorithm performance versus the standard algorithm performance for a specific scenario. In Section 5.5.2, the PESQ performance convergence points were compiled for all scenarios and entered in Table 5.3 and Table 5.4 for the SEC-RST and SEC-LKG algorithms respectively. Table entries were averaged to generate Figure 5.19, which summarizes the convergence points of all characteristic patterns with respect to the error burst sizes.

Studying the summary plot of Figure 5.19 (duplicated in the center plot of Figure 5.30) we can see that the convergence requires from 7-16 good frames versus 25 received good frames to converge to a resynchronized memory state, hence, the convergence time is on average inferior to results obtained from the manual inspection approach presented before. A quick look at the summary plot proposes that the convergence time is dependent on the size of the error burst. Figure 5.19 shows that the average convergence time will be longer if the error bursts are lengthy. In fact, the convergence really begins when the first good frame following the error bursts is received. Consequently, subtracting the error burst size from the obtained convergence value allows us to observe an almost constant convergence time of 8 and 10 frames for the SEC-LKG and SEC-RST algorithms respectively, irrespective of the error burst length. Interestingly, the constant convergence time coincides with the average phonemes length. We attribute the results to the fact that on average, phonemes last 80 ms or 8 frames (as reported in Section 2.2), and that the test speech files used separate phonemes with period of silences that naturally forces a rapid convergence of memory state values.

An important factor that was considered in Section 5.5.3 was the minimum network delay, or RTT, after which the SEC algorithm could practically be used. We estimated this value to be 60 ms (e.g. 6 frames). We elected for an upper limit of 180 ms, which provides a good representation of poor network performance in both directions (and we were limited to the data set initially captured - the PESQ performance gains for the last few frames captured were extremely small in all cases, even if the convergence values estimated in Table 5.3 and Table 5.4 were converging later than 18 frames). We called this windowing the *utilization window* (as shown in Figure 5-30). We used the *utilization window* to effectively discard some of the values to ensure the network delay constraint was well represented in our experiments. This approach allowed us to present a PESQ performance summary for each test speech files as shown in Figure 5.20 and Figure 5.21 for SEC-RST and SEC-LKG respectively (e.g. test speech file TS 10 in this case).

A first look seems to indicate that the gains are marginal. While the overall PESQ performance is difficult to assess from Figure 5.28, the overall MSE performance of Figure 5.29 provides sufficient evidence that the SEC algorithm does mathematically provide a better system match. Looking at the consolidated results for Tables 5.5 to Table 5.10 we can observe the SEC-LKG algorithm constantly outperforming the other methods (excepted for error bursts length 1 – also see Figure 5.20 and Figure 5.21), the SEC-RST comes in second place, and the standard algorithm is last but only by a marginal difference. Overall, the SEC algorithm did not perform as well as we initially hoped. In all cases, PESQ performance gains for data filtered using the *utilization window* (e.g. to represent the network delay constraint) were only marginal. We analysed Figure 5.17 and Figure 5.18 and compared values with Table 5.4 to find that Figure 5.17 and 5.18 are best cases scenarios. We used Figure 5.17 and 5.18 to introduce the characteristic pattern principle that describes the algorithm performance. While Figure 5.17 and 5.18 imply best-case scenarios we observed that the trailing segment of the characteristic pattern generally exhibited a negative PESQ performance contribution

(i.e. speech signal degradation). Since we opted to average the PESQ performance results passing through the *utilization window*, we are likely to believe that the PESQ performance gains achieved in the leading segment of the *utilization window* were quasi-nulled by the negative PESQ performance contribution of the trailing segment. This implies that the differences between the watermark and obtained value must be marginal in almost all other cases.

Figures 5.22 to 5.27 and associated Tables 5.5 to 5.10 present the averaged PESQ performance from all results gathered for both the SEC-RST and the SEC-LKG algorithms. Among these figures, Figure 5.22 displays the worse results but is also the easiest to explain. Recall in Section 4.5.3 when we explained the concealment process of the G.729A. We observed that the concealment of a single packet loss provided a better estimate for the substitution speech frame than for longer losses. This behaviour was attributable to the number of estimated components for the synthesis of the speech frame substitute (e.g. concealment error component alone versus concealment and memory state error component). Figure 5.20 and Figure 5.21 are also creating further evidence to that effect. When studying Table 5.3 and Table 5.4, with the exception of a few speech test files, convergence mainly occurs before 5 frames are processed. In other words, the natural convergence of the standard algorithm occurs before the SEC algorithm can repair the memory state error (e.g. the characteristic pattern, as for Figure 5.17 or Figure 5.18 but with much earlier convergence and potentially bad performance on the trailing side of the utilization window, following the convergence point). Therefore, the action of the SEC algorithm has a disruptive effect due to the unnecessary repair of the memory state beyond the convergence point. Looking at Figure 5.22, we easily see the negative performance, on average, of such memory state re-initialization. Both the SEC-RST and SEC-LKG algorithm constantly under perform the standard algorithm with the exception of those few speech test files that present later convergence. In contrast, any error burst greater than 1 will present a speech frame substitute affected by concealment and memory state error

components (i.e. estimated values become further astray from their intended location). This intuitively translates to longer convergence of the memory state elements compared with the case of an error burst of size 1. Consequently, for error burst size longer than length 1, we note a continuous increase in PESQ results as the convergence point moves toward the trailing side of the characteristic pattern plot. From Figure 5.23 to Figure 5.27, we can easily observe the SEC-LKG curve on top of the standard algorithm. Tables 5.6 to Tables 5.10 show that the overall PESQ gains increase by about 0.01 to 0.02 for each graph (i.e. as the error burst size increases). From a PESQ evaluation point of view it does not mean much as the difference is too small. Since the PESQ provides a MOS value correlating to subjective test, we hardly see how someone can differentiate or evaluate the difference between 0.01 or 0.02 using the MOS listening-quality scale.

A weighting scale based on network performance is proposed in Figure 5.30 to assist rationalize our results. The scale weighs the algorithm benefits (based on the characteristic pattern) in comparison with the network performance. The scale allows us to easily discuss and rationalize the proposed algorithm performance through the use of examples. For example, assuming the network has been designed or was upgraded with sufficient bandwidth to provide network performance that would allow us to resize or shrink the *utilization window* by removing trailing values. This would imply that the averaged PESQ performance values presented earlier would contain less negative PESQ performance values, which would mostly contribute to increase the overall PESQ performance. In addition, if some network QoS traffic prioritization scheme was implemented, to prioritize VoIP streams as well as associated error correction or repair traffic, in that case, it would have the effect of moving the *utilization window* down (see Figure 5.30) and use a region where the proposed algorithm provides more benefits, hence, contributing to increase the overall PESQ performance. Therefore, we assume that any research effort contributing to re-localizing and restricting the

utilization window towards areas on the scale showing "good" or "excellent" benefits will likely increase the usefulness of the proposed algorithm.

In [58], a memory state re-initialization algorithm was briefly introduced at the beginning of Chapter 5. The proposed algorithm with a similar objective in mind, resynchronizes memory state elements without any signaling between the encoder and decoder. It simply re-initialized the encoder and decoder memory state elements at regular intervals. At this point we wish to compare our technique with this one, but with some reservations since the measurement we performed were using the new de-facto ITU-T P.862 standard while they used an independently or in-house developed algorithm to assess the performance of the algorithm. This algorithm re-initialized the memory state elements every 10^{th} frame and provided substantial performance gains. This implies that the longest the memory state could be de-synchronized would be 10 frames. Since the re-initialization is independent of the time when a packet loss occurs, it also implies, that on average, memory state elements would resynchronize after five frames following the error, which correlates well with our results obtained and associated discussion and rationalization (e.g. remember that early re-initialization enables using values providing good PESQ performance gains – the leading segment of the characteristic pattern).

The discussion thus far characterized the proposed algorithm with an ***overall marginal*** PESQ performance gain when operating in a best effort delivery environment (e.g. the IP network) and with no guaranteed quality of service. Some key observations were made during the analysis of the results and are presented before closing the discussion. They are:

- After studying Figure 5.20, Figure 5.21, and Figure 5.29 we came to a first restriction in using the SEC algorithm. We observed that re-initializing memory state for single error burst degrades speech quality rather than assist it. Consequently, the algorithm should be modified accordingly. However, this would imply additional delay before the re-synchronization request can be transmitted for longer error bursts.

- The performance of the proposed algorithm is dependent on the network performance and will improve for networks with lower delay.

- Both repair schemes presented were closely following the behaviour or curve of the standard algorithm but no obvious demarcation or indicators were present to assist us determine a clear preferred scheme for the correction of the state error propagation. We believe other schemes using local interpolation (for example, from information contained in frames discarded because they arrived to late, may still be usable to minimize the amount of divergence in memory state elements, before resynchronization occurs) of the values at the encoder and decoder respectively could potentially reduce or remove some of the artefacts generated when using memory state element m^c, or m^b and m^c together, which could potentially provide substantial performance gains.

- Finally, recall that it was reported that the speech distortion or degradation due to the memory state error is as important (as noticeable by the ear) as the speech degradation caused by the packet loss itself. We still believe it to be true but only for a limited number of frames following the error. The convergence results obtained seem to indicate that memory state error propagates beyond that point but it appears that past a few frames, the ear is in

fact masking the associated distortion (e.g. from that point the speech degradation is considered negligible). This observation would also account for the difficulty to ear the SEC-RST or SEC-LKG performance gains compared with the standard algorithm.

Processing Complexity

Finally, Table 5.11 presents the processing complexity results for the SEC-RST and the SEC-LKG. The results are computed from the execution performance of the algorithms. For each trial, an associated report is generated where encoding and decoding processing time for each frame was cumulated for the stream under trial. Since the complexity measure is in MIPS, which is with respect to time, we can easily correlate the performance of the proposed algorithm. Table 5.11 below shows the averaged results for 100 trials.

PERFORMANCE RESULTS			
	Encoder Original	**Encoder SEC-RST**	**Encoder SEC-LKG**
Averaged Execution Time (micro seconds)	3168.5	3181.5	3190.5
Estimated MIPS	10.5	10.543	10.572
Added delay	0	0.04308	0.07290
	Decoder Original	**Decoder SEC-RST**	**Decoder SEC-LKG**
Averaged Execution Time (micro seconds)	733.5	736	738.5
Estimated MIPS	2.430	2.439	2.447
Added delay	0	0.00828	0.01656

Table 5.11 – Processing complexity results for proposed algorithm

We computed the MIPS relative to the advertised performance of the G.729A (i.e. 10.5 MIPS – see Table 2.1). Therefore, the presented results were obtained by comparing the performance of the proposed algorithm relative to the reported performance of the standard algorithm.It is not surprising to notice that there is only a slight increase in complexity. As shown in Figure 5.4, the SEC

algorithm is simple to implement. The RST scheme is the simplest to implement since it only re-initializes the encoder and decoder values. On the other hand, the LKG scheme requires the storing, searching, and restoring from a history buffer. Still, the processing requirement is kept to a minimum by ensuring that the size of the history buffer stays small.

5.6 SUMMARY

In this chapter we investigated the state error propagation problem in details. We determined that the G.729A codec must maintain the encoder and decoder memory states in synchronization for its optimal operation. We determined that a lost packet inevitably forces an invalid memory state update with values that are astray from their optimal values. As a result, state error appears and degrades the quality of speech. This state error shows for several frames, following the lost packet, until it converges back to its optimal state. Given that telephony applications are constrained by round trip time (RTT) around 300 ms, and that initial investigation determined it takes, on average, some 27 frames for the memory state to return to an optimal state, we elected to propose and algorithm that uses signaling across the network to synchronize the memory state, hence minimize its propagation. A closed-loop algorithm we named State Error Correction (SEC) was designed an implemented to be our test tool. We then thoroughly investigated two repair schemes to be used to update the memory state where informal subjective tests were conducted by three testers, including myself, to confirm which of the memory state elements would required re-initializing to achieve objective as well as subjective performance gains (i.e. speech quality). In conclusion to the subjective testing conducted and to the observations we made using the test tool, memory state m^b, was the only memory state element requiring re-initialization. The repair schemes named Reset to defaults (RST) and Last Known Good (LKG) are used to implement two different re-initialization concepts. Firstly, the RST scheme re-initializes the values with initial state values. LKG on the other

hand, is more complex as it requires a history buffer at the encoder to re-initialize to a past common state as directed by the encoder (e.g. assuming some correlation exists between the stored state and the time it gets restored).

We then established a testing environment with tools (e.g. VWB application) and test speech files. Using the tool, we generated the reference data set against which to compare results from the SEC-RST and SEC-LKG algorithms. The reference data was used in all graphs to qualitatively compare the results obtained. The target data or results were obtained by executing over 100 000 trials. Each test speech file was used several times to test the SEC-RST algorithm. The tests were then repeated for the SEC-LKG algorithm. Experimental data thus generated along with the reference data created the basis data sets necessary for the understanding of the effect of network delays on the resynchronization of the memory state. We then built on this initial data to generate and establish the convergence time of the proposed algorithm. From that point we were able to provide an assessment of algorithm performance associated to a best effort delivery environment and with no guaranteed quality of service representative of a real environment. The proposed algorithm in such an environment could only provides ***overall marginal*** PESQ performance gains. However, we were able to suggest possible enhancements that could potentially increase the algorithm performance. We then closed the discussion of results by making some valuable observations.

The complexity of the algorithm is recorded in Table 5.11, where the standard G.729A algorithm operates at 10.5 MIPS, the SEC-RST at 10.45 MIPS, and the SEC-LKG at 10.57. The additional bandwidth requirement is variable and depending on the error rate. For example, for a 10% error rate (e.g. missing packets) the SEC-RST and the SEC-LKG additional operating bit rate is increased by 0.125% and 1% respectively. The advantage of this algorithm is that the increased bandwidth

does not directly add to the data stream delivery as it is used on the opposite path (i.e. a bi-directional system), which uses a different set of queues/buffers.

The SEC algorithm, using the RST and the LKG schemes, demonstrated overall performance gains. While the complexity with respect to the implementation, CPU speed, and bandwidth is very low, the performance gains are **marginal**, consequently, this implementation is **no longer attractive**.

At this point we want to answer the questions raised in Chapter 1:

- *Can we avoid memory state error propagation?* The answer is yes but not by any known practical means. You have to ensure that the decoder never receives errors (i.e. missing packet) but we all know this will never happen. Hence, we do not believe the state error can be avoided. A single missing packet is always followed by an invalid update of the memory state. On a more practical note, we now firmly believe that better concealment methods will assist minimizing the memory state error and its propagation.

- *Can the network react quickly enough to correct the memory states de-synchronization?* No, the most important gains are achieved when the memory state is re-synchronized shortly after the lost packet. Being restricted by a minimum network delay of 6 frames (e.g. 60 ms) implies that the values that would be most meaningful to the repair are excluded, and consequently, they do not contribute to the repair. However, the implementation of QoS may be beneficial for such an algorithm and may provide better performance results.

6.0 CONCLUSION

This research work has been extremely fruitful for its author as it presented an opportunity to enrich his knowledge and understanding of the problem of voice transmission over IP networks. This final chapter is divided in two sections. Section 6.1 provides a synthesis of the main topics covered in this thesis. A brief overview of each topic is presented and the associated thesis contributions are highlighted. Finally, Section 6.2 suggests a few research areas to further extend the scope of this work.

6.1 CONTRIBUTIONS

The following topics are summarized in Sub-sections 6.1.1 to 6.1.3, inclusively.

- Contributions to this field of research (e.g. voIP);
- Contributions to the School of Information Technology and Engineering (SITE); and,
- Contributions to the author's knowledge and experience.

6.1.1 Contributions to this field of research

Effect of Packet Losses on Memory State

We explained the main problems associated with IP networks and described the best effort delivery mechanism that makes VoIP possible. We rationalized the combined effect of network delays and end devices delays by providing more information on the context that leads to packet loss. This provided us with the capability to investigate the effect of the network on the delivery of a real-time stream. We were then convinced that packet losses are ought to occur and that speech codecs should be robust against these losses. This naturally drove us to study packet repair methods. We learned that there exists many strategies on how to attempt to repair the losses and that they fall under three different classes, namely, Sender-based recovery, Receiver-based Concealment (i.e.

Packet Lost Concealment (PLC)), and mixtures of these two named Sender/Receiver-based methods. From that point, we had a good understanding of the G.729A algorithm and we were able to run numerous simulations, using a specially developed tool (e.g. VWB). It allowed us to listen to and visually inspect the reconstructed speech under impairments conditions. We thoroughly investigated packet losses and consequently were able to demonstrate that the speech quality is a direct result of the ability of the receiver to conceal the packet loss. The speech quality degradation is associated with two causes. On one hand, the error directly associated with the packet loss and on the other hand the memory states error (where encoder and decoder memories are no longer the same) following the first concealment of the packet loss. Using our own new notation, we were able to walk through the process and highlight that the state error starts *following the first concealment* of the packet losses (i.e. within a same burst that is). This is the fundamental reason why the G.729A PLC is robust to isolated losses. The **contribution** is that using the new notation we gained a better understanding of the state error and can say that: the only mean to avoid memory state error propagation is to ensure no packet gets lost in the first place, and, the state error starts *following the first concealment* of the packet losses.

Reducing Memory State Error Propagation

We proposed an algorithm that could be perceived as a sender/receiver-based repair method in the context of repairing the state error versus the signal as per the presented methods earlier. We implemented an algorithm at both the encoder and decoder that exchanged signaling information across the network to re-synchronize their memory states following missing packets. Two memory-state re-initialization schemes were used but provided similar performance. Thorough testing was conducted and a large quantity of data gathered by simulating several runs, over different network delays, and different error conditions. Overall, we were only able to achieve *marginal gains*. By listening to the results, we could occasionally hear a clearer speech. The algorithm was easy to

implement and was aimed to be an add-on or enhancement adaptable to other codecs. **We attributed this lack of significantly improved performance** to the overly long network delays as the analysis of results indicated good performance possibilities but only if re-synchronization occurs after a very short lapse of time (i.e. 20-30 ms when the repair is most meaningful). We also came to the **conclusion** that memory state convergence occurs between 7-16 frames (e.g. 70-160 ms), which make the network closed-loop approach close to inadequate for earlier convergences but somewhat adequate for the longer convergences. A more responsive network infrastructure would be beneficial to attend to the former instances.

6.1.2 Contributions to SITE

Framework for easy testing of speech compression standards

Another important contribution indirectly related to this research topic is the development of an application that assisted the understanding of problems and associated effects. The simulation framework offered all the functionality necessary to run thousands of simulations that allowed finding the expected results of experimentations. Therefore, the Voice WorkBench (VWB) application and associated source code will benefit SITE students and professors by:

- Providing a reusable and expandable framework for similar simulations; and,
- Providing a sample prototype application from which other simulation requirements can be identified.

6.1.3 Contributions to the author's knowledge and experience

Study of Speech Compression Algorithms for VoIP

Speech compression methods were reviewed in a chronological context providing a fluid understanding of approaches, principles, and attributes, associated to the low bit-rate codecs. The

description of the most common attributes associated to speech codecs, namely, the algorithmic delay, the bit-rate, the complexity, and the quality of speech, assisted us in understanding their properties and visualizing how they would fit in a Voice over IP (VoIP) system. We were then in a position to analyze standard codecs and select the one that would be used to investigate and demonstrate the topic of this thesis. The ITU-T G.729A Recommendation, a hybrid speech codec, was selected based on its low bit rate property, good quality of synthesized speech, moderately low delay, and moderately low processing complexity.

Given that this thesis considered the effects of packet losses, we also reviewed subjective and objective evaluation methods. A **first outcome** from that study was the selection of an objective speech quality evaluation tool tailored to test speech codecs: the ITU-T P.862 Recommendation - Perceptual Evaluation of Speech Quality (PESQ). It was used extensively to objectively study the effects of packet losses and the performance of the proposed algorithm. The **second outcome** was that we were able to conduct informal subjective testing during thesis work, which assisted us in rationalizing some of the discussions.

Understanding of problems associated with packet networks

The review of packet-networks started in Chapter 1 where fundamental differences, compared with General Switched Telephone Network (GSTN), highlighting serious challenges to be overcome for the implementation of VoIP architecture were presented. We explained the store-and-forward approach used in IP networks with a focus on routers since they are the main cause of packet losses. We also discussed the main problems associated with IP networks, namely, network delays, jitter, and packet losses, and saw how header compression, packet fragmentation, and Quality of Service techniques could be used to reduce some of the worst impairments.

We then provided a broad vision of multimedia protocols such as RTP/RTCP, H.323, and SIP since they are inherently used to architect VoIP systems. In other words, the review provided more insight with respect to the causes of network impairments, thus **benefiting** the reader and the author by providing a better understanding of the main challenges associated to VoIP implementations.

6.2 FUTURE WORK

Many observations were made during this thesis requiring additional attention. Since the observations alluded to topics that were not related with the objectives of this thesis we kept them to propose as future work.

6.2.1 SPEECH DEPENDENT PLC

Packet Loss Concealment (PLC) schemes are non discriminatory or speech independent. No matter the speech segment lost, the PLC scheme will execute the same algorithm over and over which will show different performance results according to the subjected speech segment.

Concealment alone will not provide a good quality stream and can leak residual errors in the subsequent frames being decoded. Recovery and concealment methods described in this thesis do not have the capability to maintain the stream quality under severe network degradation. Therefore, we propose future work with respect to this topic, where a combined approach would use sender-based recovery as well as receiver-based concealment techniques to maintain adequate speech quality under adverse network conditions.

Important speech quality degradation is observed when the error occurs at the beginning of a speech transition, that is, going from no speech to speech. Concealment methods are typically based on the

concept of generating a similar estimate of the missing frame based on past values. However, if the errors were to occur at the start of the speech transition (e.g. say frame #4, #5, and #6 are in error in Figure 6.1), then, no knowledge would exist and there will be no way for the concealment scheme to know how to estimate the missing frame(s) (i.e. what was said.). Therefore, a combined approach would use sender-based recovery as well as receiver-based concealment techniques to maintain adequate speech quality under adverse network conditions.

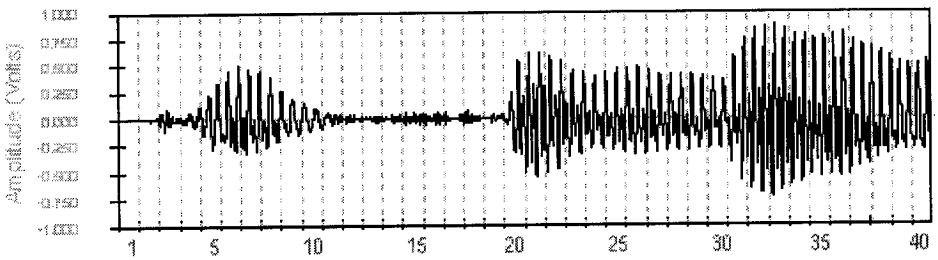


Figure 6.1 – G.729A Reconstructed signal with no packet loss

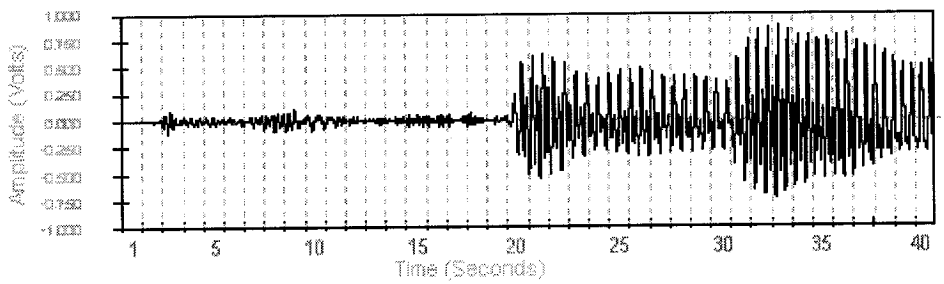


Figure 6.2 – G.729A Reconstructed signal with a burst of 3 packet losses occurring at packets #4, #5, and #6.

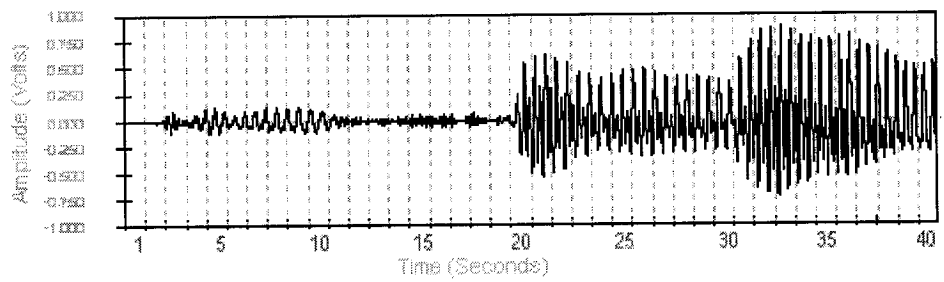


Figure 6.3 – G.729A Reconstructed signal with a burst of 2 packet losses occurring at packets #5, and #6.

When a loss occurs at the beginning of speech activity, the decoder is left with no adequate a priori information to effectively guess a replacement for the lost frame. Figure 6-1 shows a segment of test speech that has been encoded and then decoded using the ITU-T G.729A algorithm. An error burst of 3 consecutive speech frame losses was inserted starting at frame #4 in Figure 6.2. We can observe that the reconstructed signal deviates significantly from the reference signal presented in Figure 6.1. In this instance, there was some a priori information as contained by frame #3 upon which the PLC scheme generated the replacement frames. In Figure 6.3, the error burst size was decreased to two missing frames starting at frame #5 and it had the effect of providing additional a priori information to the concealment scheme. From this figure, we can easily observe that the concealment scheme provided a better approximation to the lost frames. Numerous observations similar to the one above were constantly experienced with the variety of test speech files on hand. The visual evidence was sufficiently convincing for us to firmly suggest that packet concealment methods should be dependent on the location of the lost packet within the speech stream.

6.2.2 IMPROVING THE PERFORMANCE OF SEC USING SILENCE COMPRESSION

Conversation between two parties consumes about 80% of the available bandwidth while the other 20% represents conversational waits or silences that naturally occur during a conversation. Furthermore, we know that the flow of the communication is logically split in two channels, an in coming and an out-going channel. Since one speaks while the other listens, approximately 40% of active information is transiting through a channel at a time. This implies the remaining 60% is idle and considered a bandwidth inefficiency that most likely translates to the transmission of silence periods. Not using silence compression implies that periods of silence as well as periods of speech activity are treated equally by the compression algorithm (e.g. as was used for this thesis).

Obviously, this may not provide the best benefits considering that the bandwidth is usually an expensive resource.

Figure 6.4 represents the case where silence compression would be used. Assuming that 50% of the streams are converted and that the remaining ones are substituted by silence segments, it would effectively reduce the number of packets transiting through the network. This is better explained with the illustration of the end-to-end behaviour of the system shown in Figure 6.4.

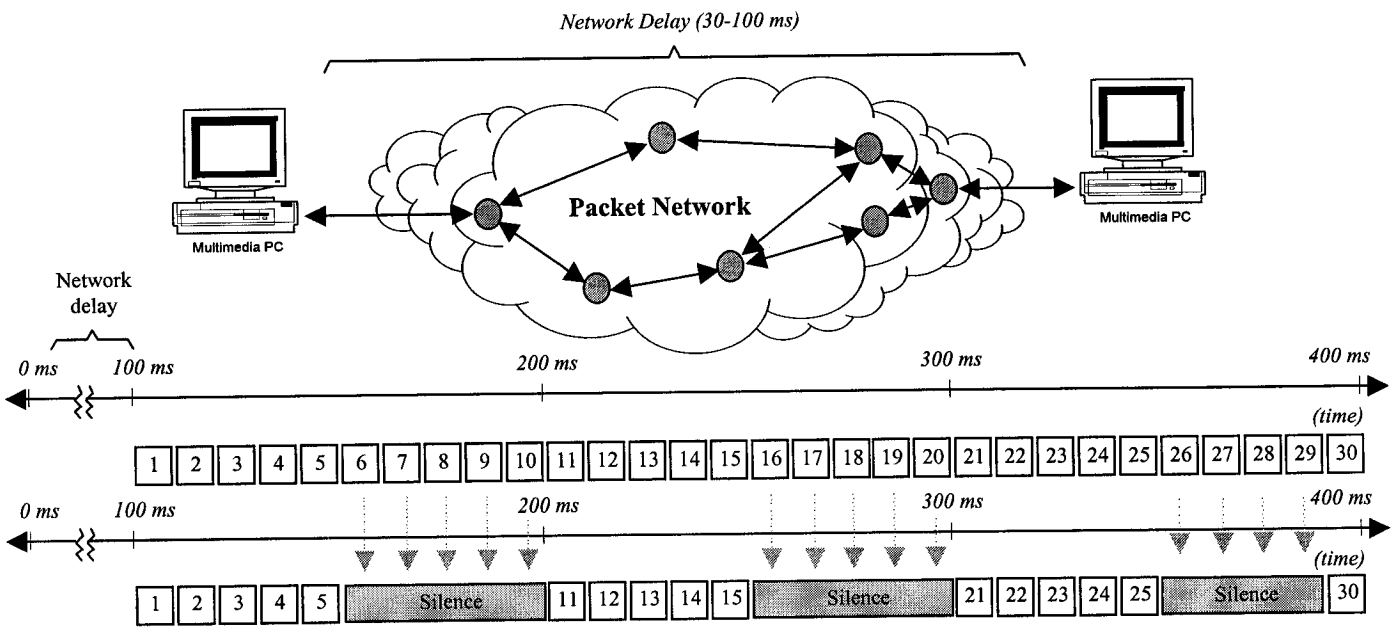


Figure 6.4- The use of the silence compression scheme

It shows that only 5 packets, instead of 10, would be transiting through the network during the first 100 ms. Assuming that the exchange frequency between the two communicating parties is balanced, the second and third hundred milliseconds time slots would be affected in a similar fashion. This implies that the number of speech frames subjected to state memory error would be reduced and would benefit from the SEC algorithm that was proposed in this thesis (see Section 5.3.2). Assuming frame #3 is lost, as discussed before, the memory state error would start propagating from frame #5. Without silence suppression, the memory state would propagate but attempt to converge

through frames #6, #7, #8, #9, #10, ..., and through frames #11, #12, #13, #14, #15, ..., with silence substitution (with respect to the reference frame numbering on the upper graph of the figure). Two factors may assist obtain a better performance:

- The encoded speech segment containing voice information would probably exhibit less power variation than the case where silence occurs thus keeping memory state values within a smaller range (e.g. less opportunity to drift far apart). Hence, convergence should occur faster; and
- Since memory state error propagation can be minimized using the SEC algorithm if the correction occurs shortly following the lost packet, the algorithm could be used in a more effective area (e.g. the same effect as if the network delay is shorten below 60 ms/6 frames, for say), hence, providing better performance.

Another approach could be to integrate re-synchronization signalling within the silence substitution frame to re-initialize the memory state (all memory state elements) during periods of silence following a dropped packet.

6.2.3 SEC IMPROVEMENTS OVER QOS IMPLEMENTATION

To this date, the Internet architecture is still supported through a flat service model offering, that is, it is offering equal transmission opportunities (or best effort delivery) [20][6]. However, marked progress is identified [18][52] where more corporate intranets are being tuned for delivering Quality of Service (QoS). Going back to the results obtained in this thesis, we basically dropped the results representing low network delays since they were not practical.

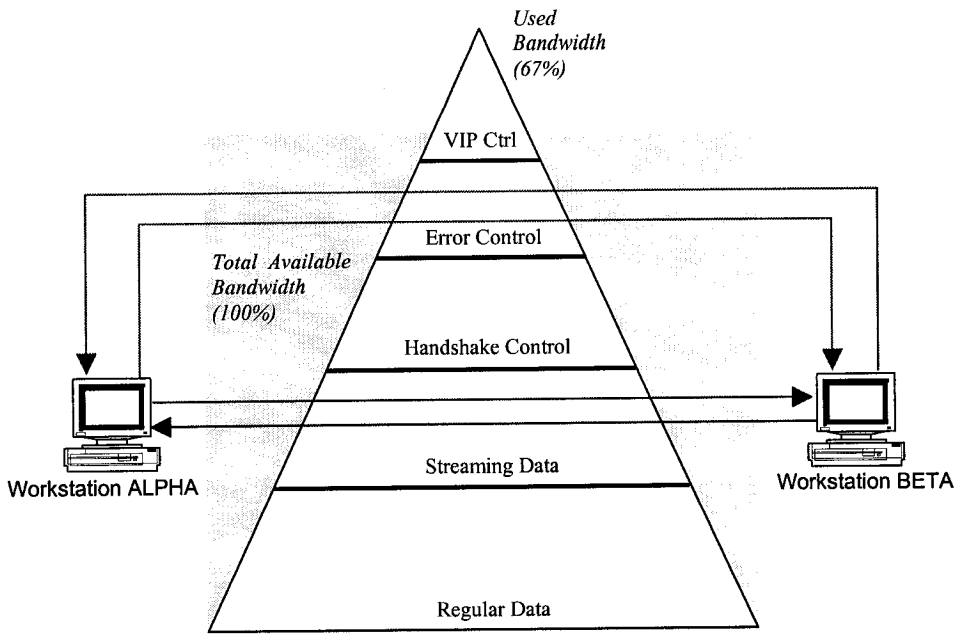


Figure 6.5 – Implementing a QoS Strategy

Implementing a QOS strategy as shown in Figure 6.5, where the top of the pyramid has highest priority, would offer fast delivery of speech frames but would also expedite any error control messages (e.g. *SEC Request* and *SEC Ack*). Not only would it assist the proposed SEC algorithm, it would in fact greatly assist at resolving some of the problems encountered in VoIP systems.

6.2.4 AUTOMATING MEMORY STATE RE-SYNCHRONIZATION/RE-INITIALIZATION

The IP network uses several universally synchronized clocks for the Network Time Protocol (NTP). A last suggestion would be for the packet loss statistics to be maintained at both the transmitter end (encoder) and the receiver end (decoder). The automation algorithm could operate on the encoder and decoder independently but in parallel and offer a minimum exchange through the network to re-synchronize the memory state. The algorithm would be based on network statistics (e.g. number of packet losses, the interval between losses, etcetera) to establish an exact moment in time for the encoder and decoder to simultaneously re-initialize their memory. Of course, the method could also use signalling to exchange or control the state but on a less frequent basis.

APPENDIX A

Voice WorkBench Application Documentation

1.0 APPENDIX A - VOICE WORKBENCH APPLICATION

1.1 APPLICATION NAME

Voice WorkBench (VWB) Version 1.0 Dated Nov 2002.

1.2 OBJECTIVES

The main objective is the design and implementation of the VWB Application to provide an environment for simulating experiments to assess the performance of the proposed algorithm. In support of the thesis, the ITU-T G.729A algorithm was selected as the basic codec algorithm to be implemented and used in experiments.

1.3 SCOPE

The application will provide an organized environment for the easy management of experimental data. It will facilitate re-runs (i.e. repeatability) of simulations under the same conditions while recording these conditions and associated results.

2.0 SYSTEM DESIGN

2.1.1 DEVELOPMENT ENVIRONMENT

The target environment for the simulations is based on an IBM PC Compatible computer architecture running Windows 95, Windows 98, Windows Millennium, Windows 2000, or Windows XP as its Operating System (OS).

2.1.2 PROGRAMMING LANGUAGE

The source code is written and compiled using Microsoft Visual C++ 5.0 Professional compiler. It requires the addition of DirectX 6.0 (Direct Sound component found under ..\Program Files\ DevStudio\Vc\lib\Dsound.lib) library and windows multimedia library (..\Program Files\ DevStudio\Vc\lib\Winmm.lib) within the project for proper compilation of code.

2.2 APPLICATION ARCHITECTURE OVERVIEW

The application kernel is designed to facilitate communication exchange between the application components. To that end, the exchange mechanism will allow for the synchronization of control and signaling information as well as provide a common memory location for the exchange of data structure content. The diagram in the figure below shows this application kernel surrounded by an arrangement of devices and their associated modules.

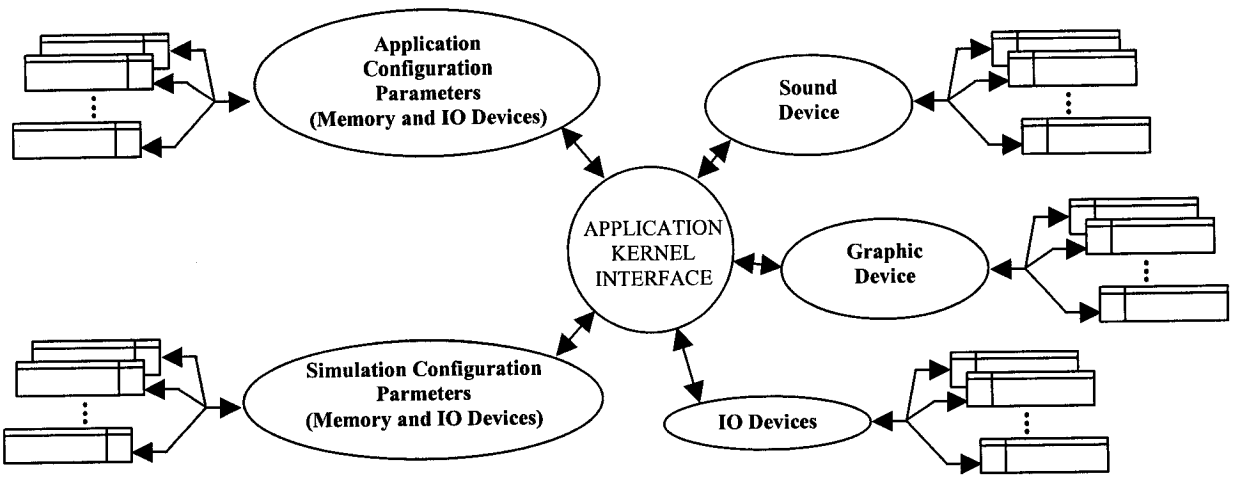


Figure A.1 – Application Kernel Architecture

Each module has a set of private data structures that exchanges data using a common set of interface functions to a globally shared memory area. The shared data structures will be managed within the Application Kernel Interface and effectively provide for code reuse within the application.

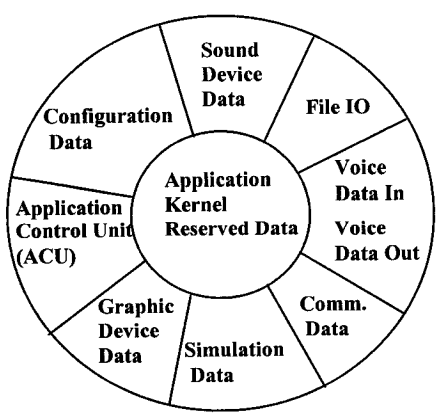


Figure A.2 – Application Kernel Interface Overview

2.3 SIMULATION SYSTEM OVERVIEW

Figure 3 below presents the high-level data manipulation performed for the end-to-end processing of real-time data through the system during simulations. The data stream is initiated from the speaker or speech file and flows through the encoder, the communication system, and the decoder, in a sequential manner, speech frame by speech frame. The source stream will come from a wave file as will the output stream. In the course of each simulation, several text files will be generated to compile and review interim results. The Encode Log files will mostly contain information pertaining to the different stages in the encoding process. The Media Log files will contain information about generated network artifacts such as the insertion rate of packet losses, the network delays, and bit errors (not implemented in the version 1.0). The Decode Log file will contain information pertaining to the different stages in the reconstruction process. Finally, the Statistics Log file will capture key factors or attributes of simulations in an attempt to recognize or identify patterns to be exploited to enhance the system and validate algorithm performance.

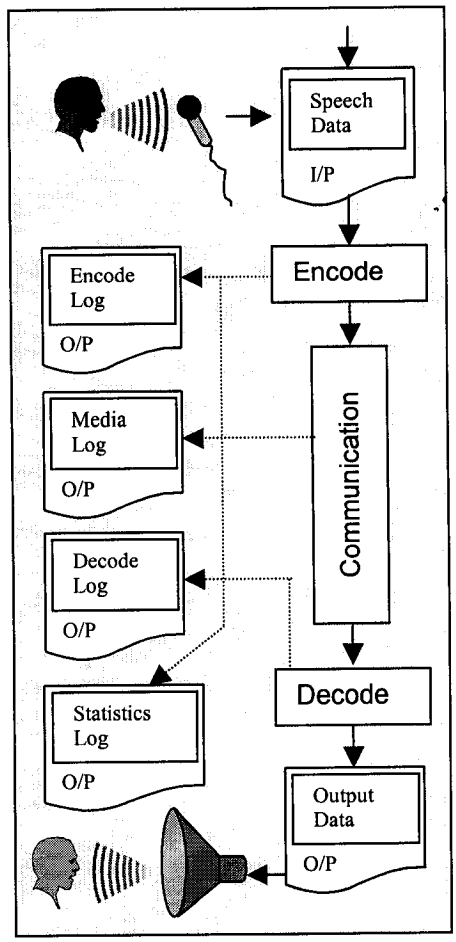


Figure A.3 – The Simulation System

Other files are also used to capture results in a consolidated manner as well as capture simulation parameters that describe the simulation that was performed.

Figure 4 provides a detailed representation of the real-time data flow used in the VWB application. The flow begins from switch *S1* until *S10*. Three flow bypass or direct connection (e.g. *S2* -> *S9*, *S4* -> *S7*, and *S5* -> *S6*, were built to easily fault find any algorithm problems or flow problems).

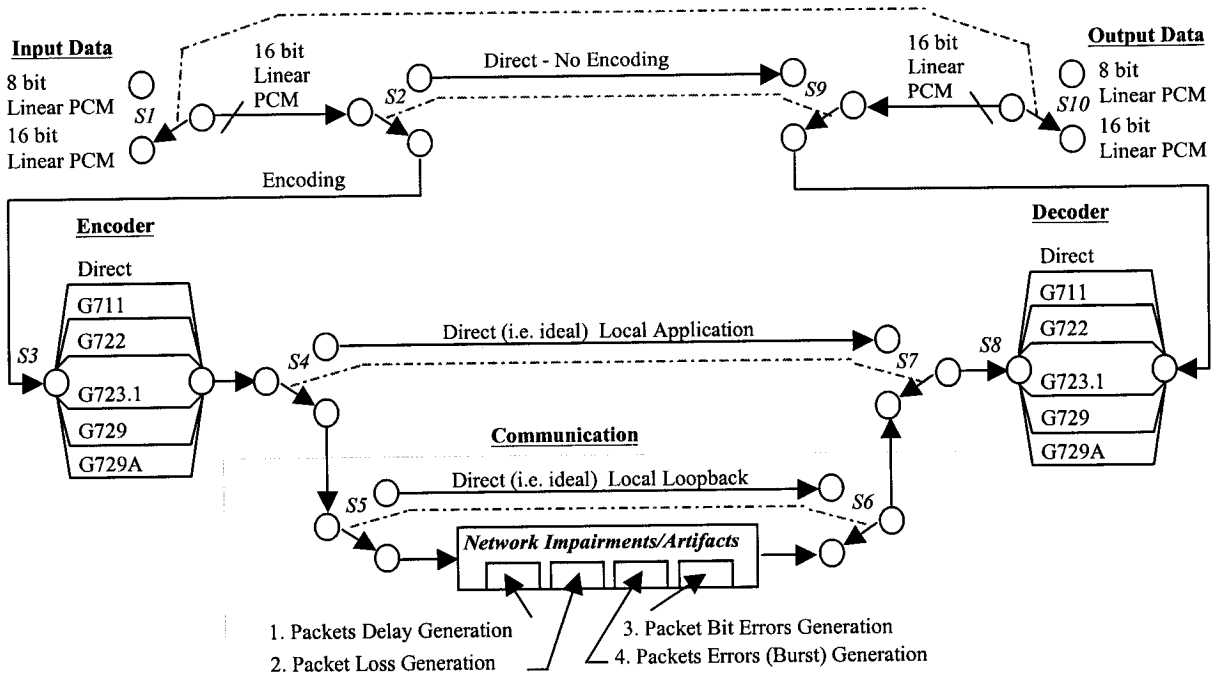


Figure A.4 - The Simulation System data path

Selector switches *S3* and *S8* operate in parallel and allow for the selection of the desired algorithm. Only a few standard algorithms are listed in Figure 4 but any standard and non-standard codec could be added as another selection.

Description / Switch Number	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Enabled 8-bit Linear PCM Source (Local Read/Write test Enabled)	UP	UP	---	---	---	---	---	---	UP	UP
Enabled 16-bit Linear PCM Source (Local Read/Write test Enabled)	DN	UP	---	---	---	---	---	---	UP	DN
Encoded/Decoding Enabled (Local Encoded/Decoding test Enabled)	UP	DN	A-Z	UP	---	---	UP	A-Z	DN	UP
Network Enabled	UP	DN	A-Z	DN	---	---	DN	A-Z	DN	UP
Ideal Net transmission	UP	DN	A-Z	DN	UP	UP	DN	A-Z	DN	UP
In error Net transmission	UP	DN	A-Z	DN	DN	DN	DN	A-Z	DN	UP

Table A.1 – Simulation System selection matrix

Table 1 shows some of the ways the simulation system could be configured and the associated basic functions performed. In this case, a two-way switch can be ON or OFF (i.e. equivalently UP or DOWN) and a multi-pole switch identified by a selection from A to Z corresponding to a specific compression algorithm.

2.3.1 DATA STREAM CHRONOLOGY

First, to minimize the implementation complexity of the data stream, a data block will be read from file to memory. Then data will be streamed from memory within the processes of the system. The left side of Figure 5 shows the different memory blocks used to store the real-time speech stream under some format associated to a system process. Ten processes are shown in Figure 5 and form a unidirectional transmission system. Process P-1.0 performs the initial conversion of data from a wave file to 16-bit Linear PCM data. The encoder, process P-2.0, in the case where the G.729A algorithm is selected, will receive 80 samples at a time (10 ms) from memory block 0. The associated compressed data will then be stored in memory block 1. The compressed data will then be packetized by process P-3.0 and the packetized stream stored in memory block 2 before being processed by process P-4.0, P-5.0, and P-6.0 respectively. Memory block 3 will contain the modified data stream or more precisely, the data stream affected by network impairments (e.g. packet losses). Then, process P-7.0, P-8.0, and P-9.0, perform the speech signal reconstruction under periods of no network degradation as well as during periods of network degradation. Last, the decoded data stream will be saved from the output memory buffer to an output wave file for storage and later replay (e.g. process P-10.0).

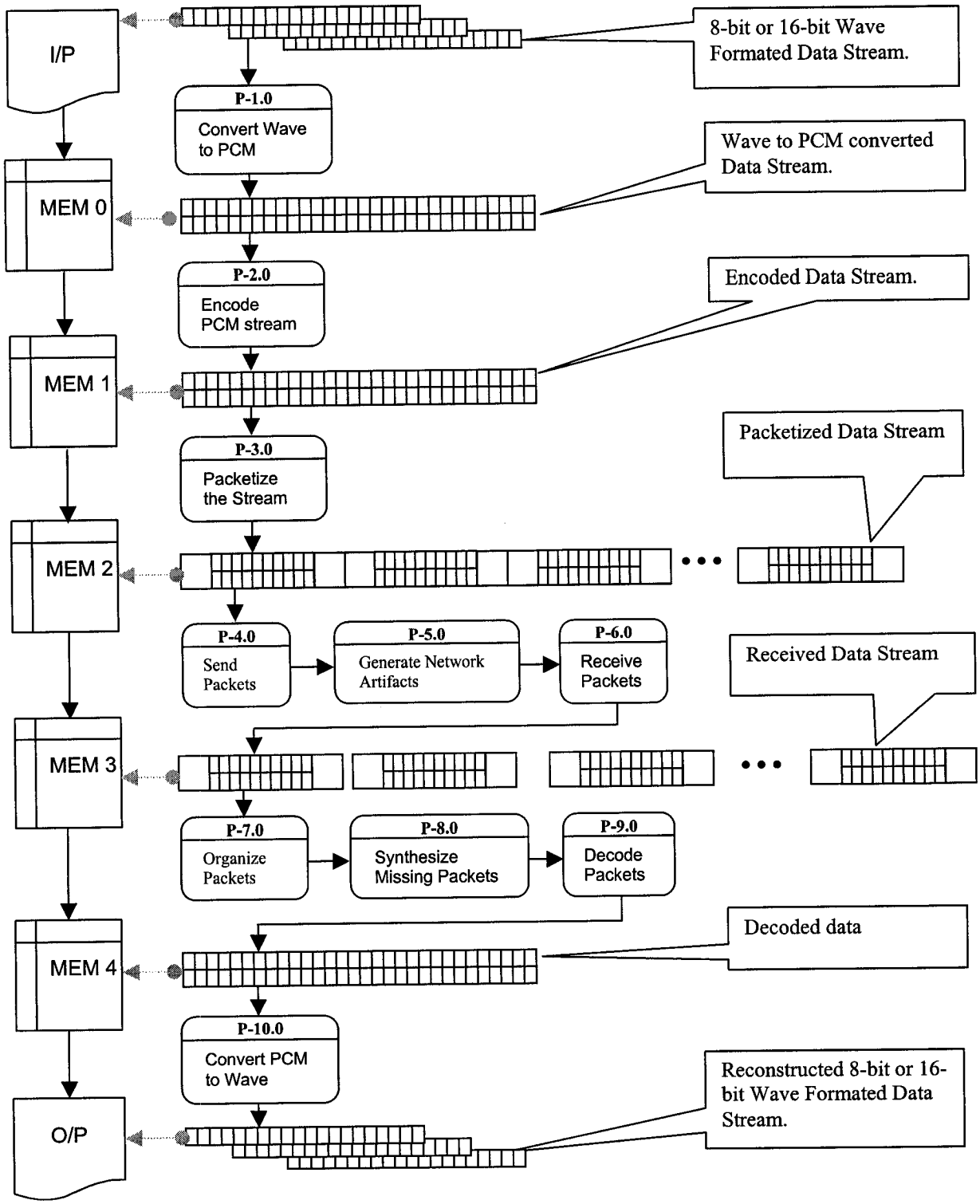


Figure A.5 - The Simulation System speech stream chronology

Several memory blocks are used to ensure we can compare and/or measure the data at different test point in the system (e.g. comparing the output signal against the input signal, performing measurements such as averages, Mean Squared Error (MSE)), etc). The application architecture design allows measuring the execution speed of each different process on a frame-by-frame basis (e.g. modular design). Furthermore, this initial stand-alone simulation application design can readily be extended to become a network based simulation application

Figure 6, below, is a coarse overview of the network application being simulated as it relates to the seven layers of the OSI model presented in Chapter 3 of the thesis. The figure shows the layers of interest for our system. The network layer is represented by the IP protocol, the session/presentation layer is represented by the RTP/RTCP protocol, and the application layer implementing the communication application (e.g. transmitter and receiver embedding one or more codecs and other added features) of the VWB application. Generally, a multimedia protocol such as ITU-T H.323 or SIP is required to establish calls and provide the necessary signalling between the communicating end devices.

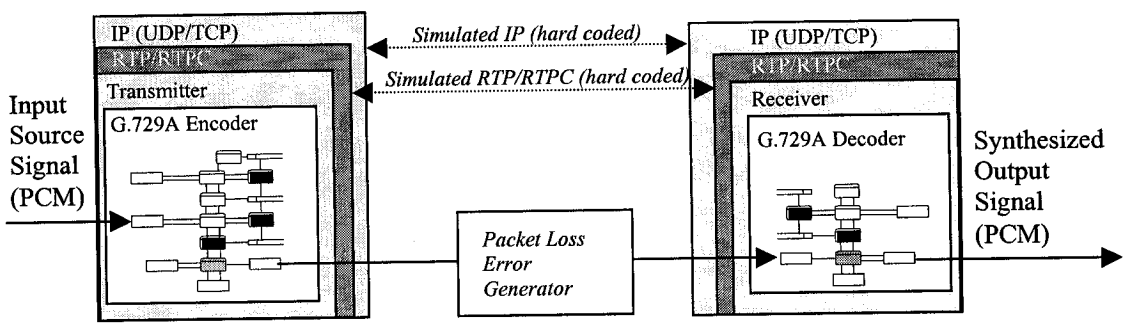


Figure A.6 – Layered block diagram of the simulation environment

In our case, call establishment is not required as the application is a unidirectional point-to-point exchange between an encoder and a decoder and signalling will be performed through the simulated

RTCP channel. VoIP systems use IP/UDP/RTP to transmit speech data on a best effort basis and use IP/UDP/RTCP for the gathering of performance statistics such as the delay variability or jitter. In both cases, the IP and RTP/RTCP protocols are hard coded or simulated in the VWB application. The transmission of network packets using the IP/UDP/RTP stack is simulated using memory buffers that basically transfer speech frames one after the other from the send buffer to the receive buffer. Errors are inserted in the receive buffer which is representative of the real case. The timing relationship between speech frames processed at the encoder and decoder is preserved by a sequential number (e.g. in this case the frame number) as contained in RTP while RTCP signalling is simulated using hard-coded messages.

3.0 SYSTEM IMPLEMENTATION

As mentioned earlier, the application was developed using Microsoft Visual C++ Version 5.0 that provides a Graphical User Interface (GUI) to facilitate the execution of simulations (i.e. experiments) and their assessment.

The application relies on 3 main views to run and investigate simulations, namely, the project view, the simulation view, and the measurement view. In this section, screen captures have been taken from the VWB application. They are presented in Figure 7, 8, and 9. Overlays have been added to the figures to facilitate the description of key areas. A basic description of their implementation is provided in the following sub-sections.

3.1 THE PROJECT VIEW

The Project View has been designed to simplify the management of experiments by organizing and consolidating simulations data. Figure 7 shows the 3 main areas of the project view. As referenced

by label A, the hierarchical organization contains files under simulations at the lowest level. The mid-level contains one or more simulations that belong to a specific project. At the top or root of the tree, projects provide to top organization unit. Projects are located under a root that is a specific directory location on the file system (e.g. the hard drive). The files are described as:

Project directory and file: The project directory is the higher-level organizational unit that should differentiate between the different experiments carried out. Under the project directory, a project file consolidates simulation performance information with respect to each simulation run performed under the experiment. The file (e.g. Prj.txt) provides a summary view of results and also contains the information necessary to find the respective simulation files, hence, enables repeating the simulation to reproduce the results if necessary.

Simulation directory and file: The simulation directory is the second-level organizational unit under which all simulation files specific to the current simulation are stored. A simulation file (e.g. ...simula.txt) is located under each simulation directory and provides simulation parameter settings as well as results obtained.

Files: The files are classified as support files and target files. Support files are configuration or description files describing the project/simulation at hand. Target files are files representing the speech streams in whatever format (i.e. binary files, ASCII files, and/or bit stream files).
Label B highlights the area used to identify this view and would be replaced by more useful functions in future versions of the application. Label C shows 3 group of buttons, from top to bottom, to manage projects, simulations, and files respectively. For example, a project must initially be created to be able to create a simulation and associated files.

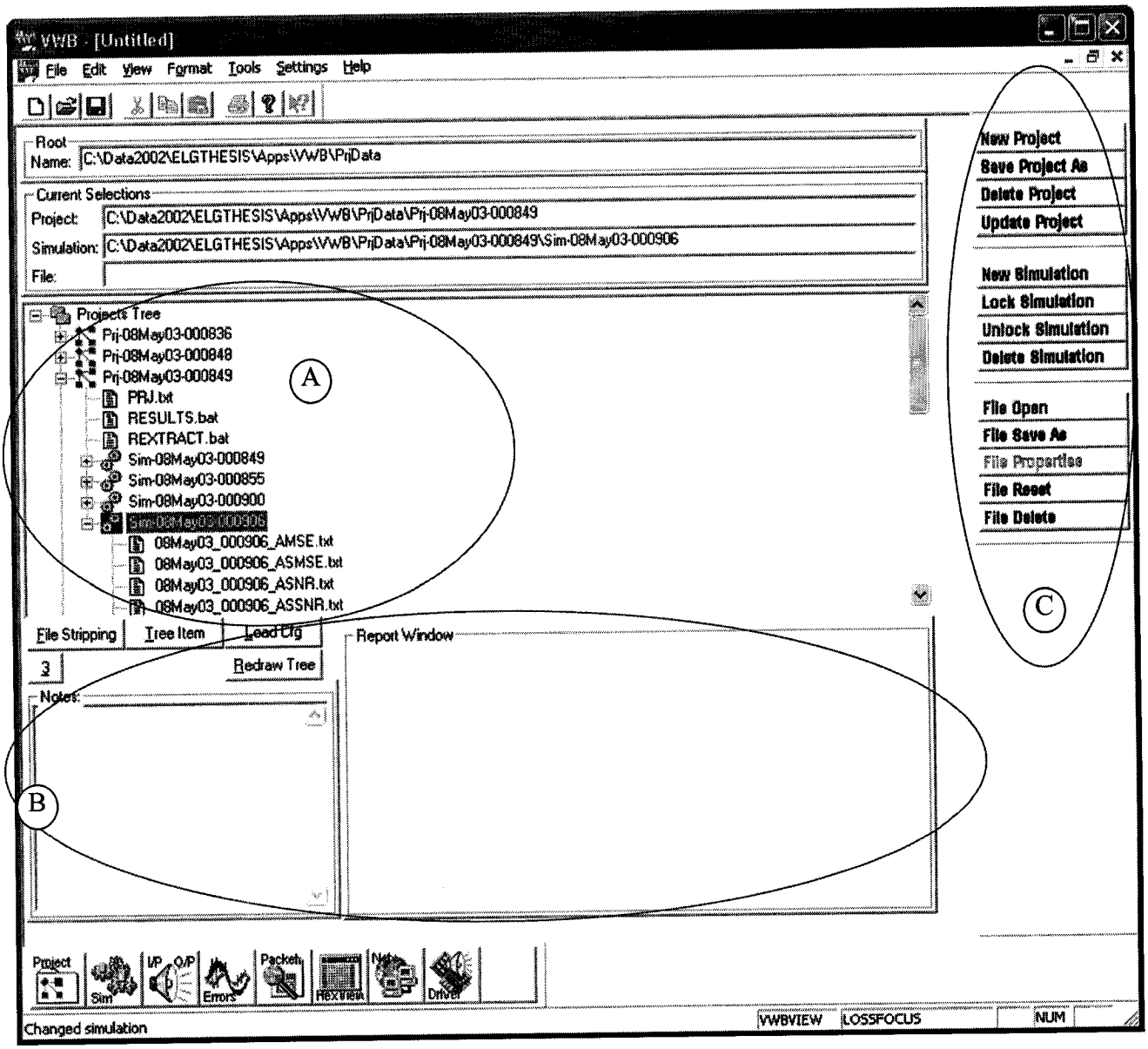


Figure A.7 – Screen capture of the VWB Project View

3.2 THE SIMULATION VIEW

The proposed algorithm is aimed at enhancing the robustness to packet losses of low-bit rate speech codec, more precisely, codec equipped with memory state such as the ITU-T G.723.1 and the ITU-T G.729 algorithms. Earlier in the thesis, we elected using the ITU-T G.729A algorithm to conduct experimentations. The ANSI C fixed-point implementation source code provided by the supplement to the ITU-T G.729A recommendation was used to implement the original ITU-T G.729A standard

that serves as the benchmark, as well as variant versions implementing the proposed algorithm. The other versions are modifications to the recommendation for the purpose of exploring and investigating various scenarios. Figure 8 shows the simulation panel of the VWB application.

The simulation view implements the Simulation System of Figure 3. To that end, the components of the Simulation System data path and the associated components of the Simulation System speech stream of Figure 4 and Figure 5 respectively, form the processing engine under the hood of the simulation view.

Running simulations then became as easy as selecting the desired compression algorithm and parameters for different speech frame loss scenarios. The tool allows the selection of many simulation parameters such as the test speech file, the codec, the error insertion model, algorithmic delays, and other key parameters necessary to conduct a thorough assessment of the proposed algorithm performance. The view offers many feedback to the user while setting parameters as well as when running the simulation. The basic information fed back to the user is the input or source file information as well as the output or reconstructed file information.

The simulation parameters and data are saved under the simulation directory as previously explained in Section 3.1. Then, the files can be accessed through the Project View or the file system using Windows Explorer.

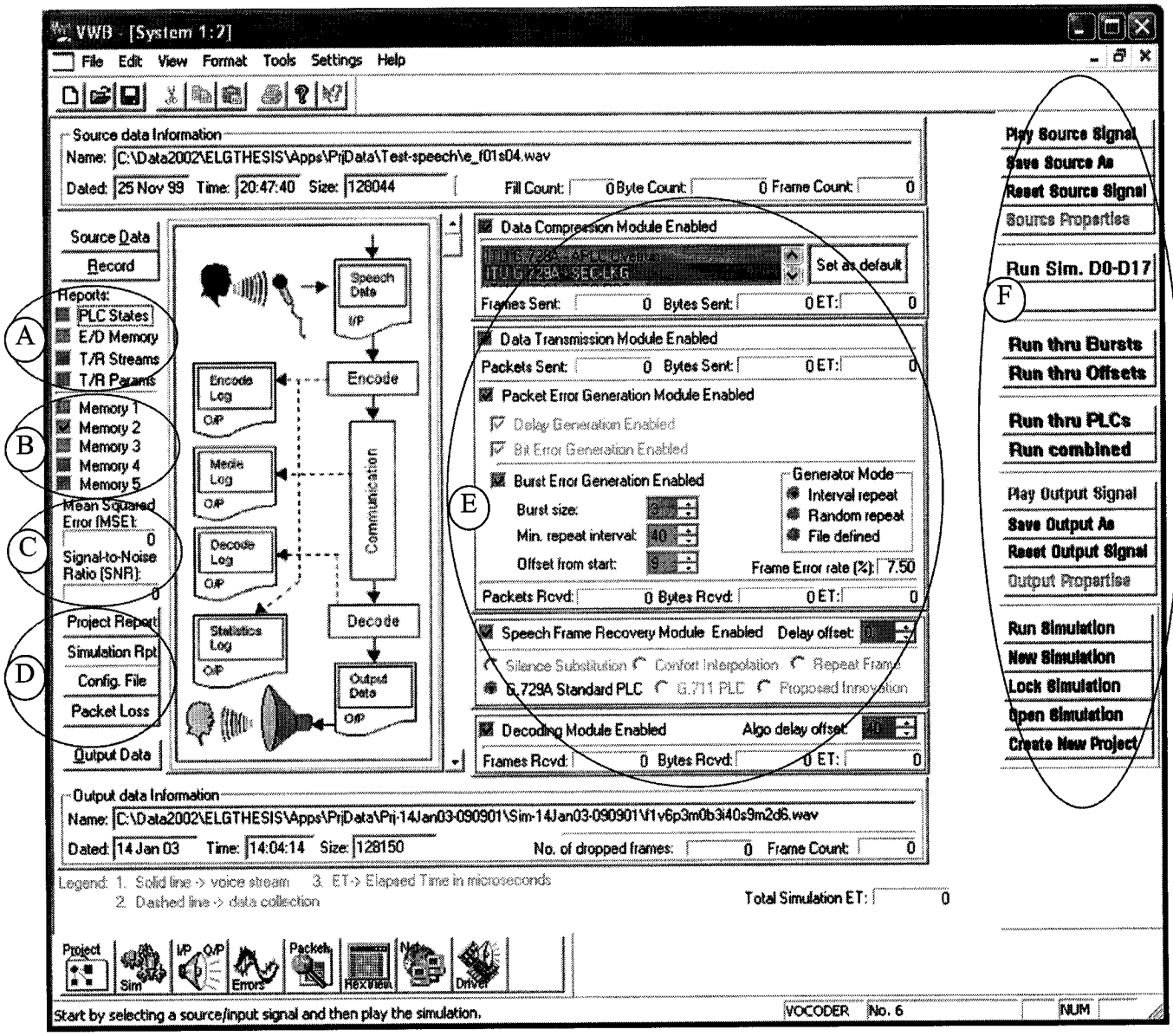


Figure A.8 – Screen capture of the VWB Simulation View

Several advantages provided by the tool are ease of tracking, changing, and reporting simulation parameters (i.e. number of frame losses, memory state description reports, etc.), and the capability to run simulation sets. The following table describes the labelled components of the Simulation View.

LABEL	NAME	DESCRIPTION
A	4 types of stream reports can be selected/generated.	1. PLC state report: provides the sequence of events occurring during and following a packet error. 2. Encoder and Decoder memory report: provides a report to compare encoder memory structures against decoder memory structures. 3. Transmission/Reception stream report: provides browsing through the transmitted source data and the received decoded data. 4. Transmission/Reception parameters: provides browsing through the transmitted/received parameters (e.g. compressed representation)
B	Enabling/disabling memory state elements re-initialization.	Each individual memory state elements forming the memory state can be enabled (i.e. re-initialization of the memory structure) in any combinations. This allows running simulations with the objective of finding the memory state element configuration that delivers best performance.
C	Overall measurements	For each simulation run, the total MSE and SNR values are computed.
D	Configuration and performance reports	1. Project report: provides the consolidation of performance results for simulations contained within this project. 2. Simulation report: provides the simulation parameters and performance results, used to run the simulation. 3. Configuration file: opens the application configuration file describing the application defaults. 4. Packet loss: reports on the frame numbers that were in error during transmission.
E	Selection of many simulation parameters	This circle captures 4 modules directly related to running simulations. 1. Data compression: selection of the compression algorithm (e.g. implemented standard and non-standard codecs). Proposed algorithm will also be an available codec selection. 2. Data transmission: simulates transmission and reception of data. An embedded module is the Packet Error Generation that insert errors into the transmitted stream. 3. Speech recovery: by default, the ITU-T G.729A includes is own PLC algorithm. Research with respect to PLC algorithms can be implemented within this module. 4. Decoding: Decoding is in tandem with encoding. Codec selection is performed in the Data Compression module.
F	Simulation functions	6 groups of buttons to support simulations. The 1st and 5th button groups are used to replay and manage the source and reconstructed speech signal. The 2nd , 3rd , and 4th groups of buttons support the automated running of simulations, such as, running simulations through all error bursts size, running simulations through all network delays, running simulations through all error burst insertion point, etc. Finally, the 6th group of buttons assist managing running single simulation (e.g. manually one at a time)

Table A.2 – Description of the Simulation View components

3.3 THE MEASUREMENT VIEW

The tool allows for the time-domain visualization[17] of the original source signals along side of the reconstructed signal and the error measurements through three collated view scopes (see Figure 9).

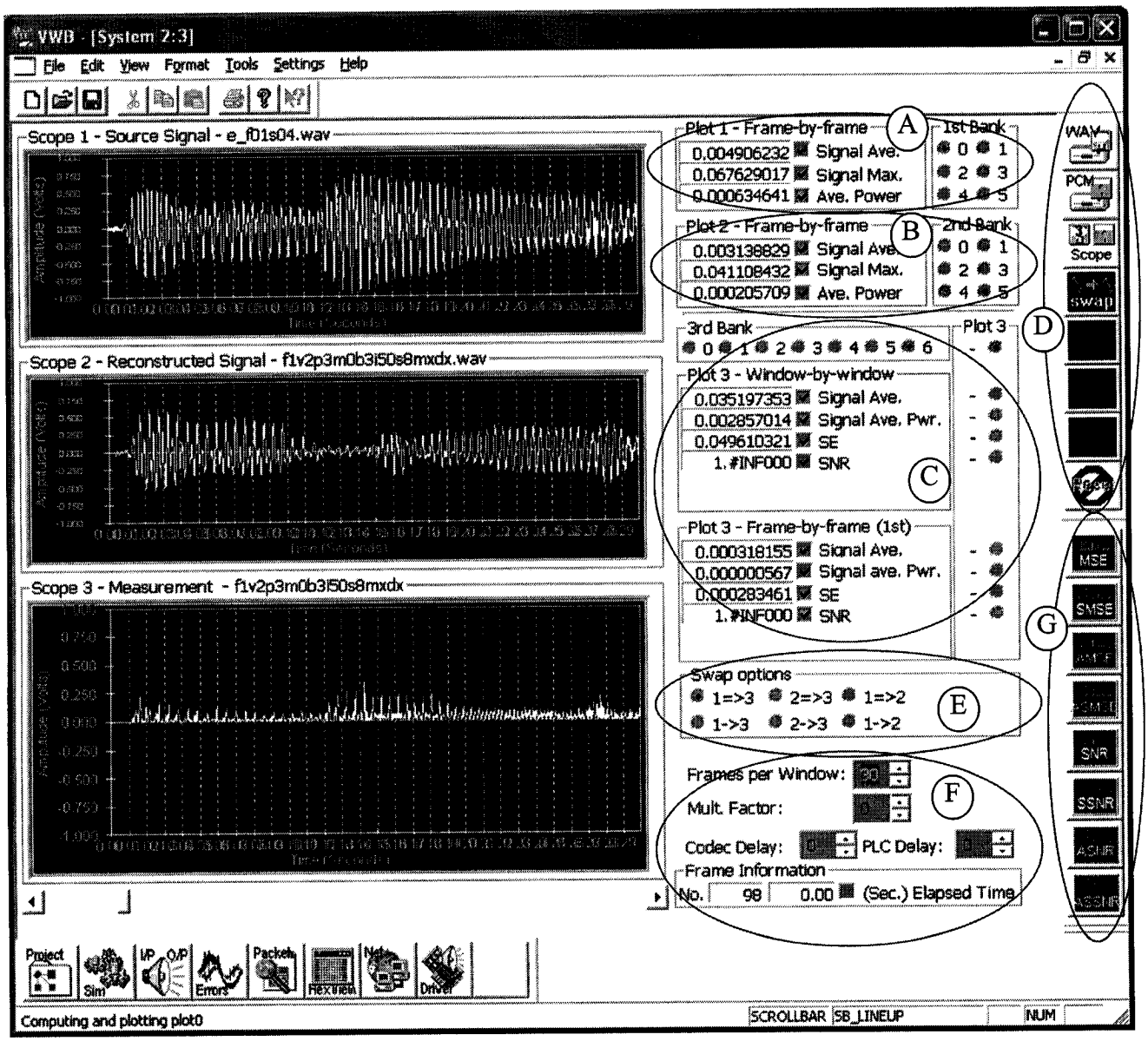


Figure A.9 – Screen capture of the VWB Measurement View

[17] Homer Dudley, a pioneer in the field of vocoders and speech synthesis. He believed that with practice, the study of the speech waveform could be learned, to be cognizant of the sound content of the waveforms, as reported in Gold b. and Morgan N., ***"Speech and audio Signal Processing - Processing and perception of speech and music"***, John Wiley and Son, 2000.

In fact, the tool allows visualizing the original source signals under several format associated with a specific processing stage of the system. This is possible since the application design enables interim storage of the speech frame at each stage as depicted by Figure 5. Figure 9 shows memory block or bank selection that selects the source data for each scope to display. Associated to the data displayed are real-time measurements provided by the application that are computed on a frame-by-frame or window-by-window[18] basis. Some of the basic measurements used are:

Signal mean (or average):
$$x_{ave} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)$$

Squared error:
$$\varepsilon(n) = \sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2$$

Mean Squared Error (MSE):
$$e_d(n) = \frac{1}{N}\sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2$$

Signal Power:
$$Power\ difference = 10 \bullet \log_{10}\left(\frac{P_{source}}{P_{reconstructed}}\right) dB$$

Signal-to-Noise Ratio (SNR):
$$SNR = 10\log_{10}\left\{\frac{\sum_{n=0}^{N-1} s^2(n)}{\sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2}\right\} dB$$

Segmented SNR:
$$SEGSNR = \frac{10}{L}\sum_{i=0}^{L-1} \log_{10}\left\{\frac{\sum_{n=0}^{N-1} s^2(iN+n)}{\sum_{n=0}^{N-1} (s(iN+n) - \hat{s}(iN+n))^2}\right\} dB$$

An absolute version of some of those measurements was also implemented to permit investigating codecs that allows 180 degree phase inversion of the reconstructed signal. Table 3 below describes the Measurement View of Figure 9 using the labels that have been overlaid on top of the screen capture.

---

[18] A window in this case refers to the scope width. For example, Figure 9 shows thirty 10 ms speech frames per window.

LABEL	NAME	DESCRIPTION
A	Plot 1 measurements	Shows the signal average, the signal maximum, and the signal average power for the first frame displayed in the scope.
B	Plot 2 measurements	Shows the signal average, the signal maximum, and the signal average power for the first frame displayed in the scope.
C	Plot 3 measurements	Shows the signal average, the signal average power, the squared error, and the SNR for the window displayed in the scope as well as for the first frame displayed in the scope.
D	Data management and manipulation	The top three buttons allow saving the scope data under several formats (e.g. wave, PCM, or textual numbers). The swap button allows for copying data from one memory block or bank to the other. Three buttons are left unassigned. The reset button resets all memory blocks values to zeros.
E	Swap options	The swap or copy option is triggered by the swap button under label D. The first row displays swap functions applicable to the window length at a time while the second row displays swap functions applicable to the length of the first frame of the window.
F	Scope settings	The most practical setting allows setting the number of frames per window. The Mult. Factor setting changes the gain of the signal displayed by the scope.
G	Computations	These buttons enable the computing of the total stream length using currently selected scope source signals.

Table A.3 – Description of the Measurement View components

3.4 OTHER VIEWS

The remainder of the views implemented in the application were used to verify the correct operation of devices, data structures, data conversion, data sequencing, and other such functionality or capability.

4.0 CONCLUSION

This annex described the VWB application in some detail. The level of details is deemed sufficient to understand the operation of the application as well as the high-level organization of the source code. This version 1.0 of the application is considered a prototype with a sound application architecture designed to readily evolve into a real-time network application, in this way contributing to the implementation of a network tool to

Bibliography

[1] Black U., "***Advance Internet Technologies***", Prentice-Hall, Inc. 1999.

[2] Bolot JC, "***Characterizing End-to-End Packet Delay and Loss in the Internet",*** Journal of High-Speed Networks, vol. 2, no. 3, pg. 305-323, December 1993.

[3] Carne B., "***Telecommunication Primer***" Prentice-Hall,1995

[4] Cidon I., Khamisy A., and Sidi M., "***Analysis of packet loss Processes in High-Speed Networks***", IEEE Transactions on Information Theory, pg 39-98 108, January 1993.

[5] Cox V., Hassle B., Lacuna A., Shahraray B., and Rabiner L., ***" On the Applications of Multimedia Processing to Communications***", Proceeedings of the IEEE, 86(5), May 1998.

[6] Cox V., Kamm C., Rabiner L., Schroeter J., and Wilpon J., ***"Speech and Language Processing for Next-Millennium Communication Services***", Proceedings of the IEEE, Vol 88, No. 8, August 2000.

[7] Dong H. and Gibson J., ***" Universal successive refinement of CELP speech coders"***, Department of Electrical Engineering Southern Methodist University, Dallas, TX 75275, International Conference on Acoustics, Speech, and Signal Processing, May 7-11, 2001, Salt Lake City, Utah.

[8] Fingscheidt T. and Vary P., ***"Softbit Speech Decoding: A New Approach to Error Concealment"***, IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 3, March 2001.

[9] Fink D.G., Christiansen D., and Al., " ***Electronics Engineer's Handbook***", second Edition, McGraw-Hill, Inc 1982.

[10] Freeman R.L., "***Telecommunication System Engineering***", Third Edition, John Wiley & Sons, Inc. 1996.

[11] Galand C. and al., "***Adaptive Code Excited Predictive Coding***", IEEE Transaction on Signal Processing, Vol. 40, No. 6. June 1992.

[12] Gao Y. et al., ***"EX-Celp: A speech coding paradigm"***, Conexant Systems Inc, International Conference on Acoustics, Speech, and Signal Processing, May 7-11, 2001, Salt Lake City, Utah.

[13] Gold b. and Morgan N., ***"Speech and audio Signal Processing - Processing and perception of speech and music"***, John Wiley and Son, 2000.

[14] Goodman D. J., Lockhart G. B., Wasem O. J., and Wong W. "***Waveform Substitution Techniques for Recovering Missing Speech Segments in Packet Voice Communications***", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 6, December 1986.

[15] Gunduzhan E. and Momtahan K., ***"A Linear prediction Based Packet Loss Concealment Algorithm for PCM Coded Speech"***, IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 8, November 2001.

[16] Halsall F., "***Data Communication, Computer Networks and Open Systems***", Third Edition, Addison-Wesley Publishing Company, Inc 1992.

[17] Hardman V. et al., "***Reliable Audio for use over the Internet***", Proceeding INET 95, Hawaii, 1995.

[18] Hassan M., Nayandoro A., and Atiquzzaman M., ***"Internet telephony: Services, Technical Challenges, and Products***" IEEE Communications Magazine, Vol. 38, No. 4, pp. 96-103, April 2000.

[19] Haykin S., " ***Adaptive Filter Theory***", 3rd edition, Prentice-Hall, Inc. 1996.

[20] Hersent H. and Al., "***IP Telephony - Packet-based multimedia communications systems***", Addison-Wesley, 2000.

[21] Hu J., Xu S., and Chen J., ***"A Modified Pitch Detection Algorithm"***, IEEE Communications Letters, Vol. 5, No. 2, February 2001.

[22] ISoon I.Y. and S.N. Koh, "***Low distortion speech enhancement***", IEEE Proceedings, Vision, Image, and Signal Processing, Vol. 147, No. 3, June 2000.

[23] ITU-P supplement 23: ***"ITU-T Coded-Speech Database"***, International Telecommunication Union, February 1998.

[24] ITU-T G.711 Appendix I. ***"A high quality Low-Complexity Algorithm for Packet Loss Concealment with G.711"***, International Telecommunication Union, September 1999.

[25] ITU-T G.711. ***"Pulse Code Modulation (PCM) of voice frequencies"***, International Telecommunication Union, Extract from Blue Book, 1993.

[26] ITU-T G.722. ***"7 Khz Audio Coding within 64 kbps"***, International Telecommunication Union, Extract from Blue Book, 1993.

[27] ITU-T G.723.1. ***"Dual Rate Speech Coder for Multimedia Communication Transmitting at 5.3 and 6.3 kbps"***, International Telecommunication Union, March 1996.

[28] ITU-T G.726. ***"40, 32, 24, 16 kbps Adaptive Differential Pulse Code Modulation (ADPCM)"***, International Telecommunication Union, Geneva 1990.

[29] ITU-T G.728. ***"Coding of Speech at 16 kbps using Low-Delay Code Excited Linear Prediction"***, International Telecommunication Union, September 1992.

[30] ITU-T G.729. ***"Coding of speech at 8 Kbit/s using Conjugate Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)"***, International Telecommunication Union, March 1996.

[31] ITU-T G.729A. ***"Reduced complexity 8 Kbit/s CS-ACELP speech codec"***, International Telecommunication Union, November 1996.

[32] ITU-T G.729B. ***"Annex B: A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70"***, International Telecommunication Union, November 1996.

[33] ITU-T G.729D. ***"Annex D: 6.4 kbit/s CS-ACELP speech coding algorithm"***, International Telecommunication Union, November 1996.

[34] ITU-T G.729E. ***"Annex E: 11.8 kbit/s CS-ACELP speech coding algorithm"***, International Telecommunication Union, November 1996.

[35] ITU-T H.323. ***"Packet-based multimedia communications systems"***, International Telecommunication Union, September 1999.

[36] ITU-T P.800. "***Methods for subjective determination of transmission quality***", International Telecommunication Union, August 1996.

[37] ITU-T P.830. "***Subjective performance assessment of telephone-band and wideband digital codecs***", International Telecommunication Union, February 1996.

[38] ITU-T P.861. "***Objective quality measure of telephone band (300-3400 Hz) speech codecs***", International Telecommunication Union, February 1998.

[39] ITU-T P.862. "***Perceptual Evaluation of Speech Quality for speech codecs***", International Telecommunication Union, February 1998.

[40] Jain A.K., "***Fundamentals of Digital Image Processing***", Prentice-Hall, Inc. 1989.

[41] Jayant N., "***Signal compression: Technology targets and research directions***", IEEE Journal on Selected Areas of Communication, 10(5): 796-818, June 1992.

[42] Jiang W. and Schulzrinne H., "***Perceived Quality of Packet Audio under Bursty Losses***", IEEE Infocom 2002.

[43] Johnston J. and Brandenburg K., "***Wideband Coding: Perceptual considerations for speech and music***", In S. Furui and M. M. Sondhi, editors, Advances in Speech Signal Processing, pages 109-140. Marcel-Dekker Inc, New York, 1992.

[44] Kabal P. and Ramachandran R. P., "***The Computation of Line Spectral frequencies Using Chebyshev Polynomials***", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 6, December 1986.

[45] Kang S. and Fisher T., "***Trellis Excitation Speech Coding at Low Bit Rates***", IEEE Transaction on Communications, Vol. 42, No. 2/3/4, February/March/April 1994.

[46] Kessler G.C., and Southwick P., "***ISDN Concepts, Facilities and Services***", McGraw-Hill Series on Computer Communications, 1997, 3rd edition.

[47] Kostas T., Borella M., Sidhu I., Schuster G., Grabiec J., and Mahler J., "***Real-time Voice Over Packet-Switched Networks***", IEEE networks, January/February 1998.

[48] Lee W.C.Y., "***Mobile Cellular Telecommunications – Analog and Digital Systems***", Second Edition, McGraw-Hill, Inc. 1995.

[49] Leon-Garcia A., "***Probability and Random Processes for Electrical Engineering***", Second Edition, Addison-Wesley Publishing Company, Inc. 1994.

[50] Li B., et al., "***Qos-Enabled Voice Support in the Next Generation Internet: Issues, Existing Approaches and Challenges***", IEEE Communications Magazine Vol. 38, No. 4, pp 54-61, April 2000.

[51] Lida K. and Kawahara K., "***Performance Evaluation of the Architecture for End-to-End Quality-of-Service Provisioning***", IEEE Communications Magazine, Vol. 38, No. 4, pp. 76-81, April 2000.

[52] Liu H. and Mouchtaris P., "***Voice over IP Signaling: H.323 and beyond***" IEEE Communications Magazine, Vol. 38, No. 10, pp. 142-148, April 2000.

[53] Lynn P. and Fuerst W., "***Introduction to Digital Signal Processing with Computer Applications***", John Wiley & Sons, Second edition 1997.

[54] Lyons R., "***Understanding Digital Signal Processing***", Addison Wesley, 1997.

[55] Makhoul J., "***Linear Prediction: A Tutorial Review***", Proceedings of the IEEE, Vol. 63, No. 4, April 1975.

[56] Mier E. et Al., "***Voice-over-IP Gateways: Sounding Good***", Business Communications, February 1998, pp. 23-29.

[57] Minoli D. and Minoli E.,"***Delivering Voice over IP Networks***", Wiley Computer Publishing, Toronto, 1998.

[58] Montminy C, "***A study of speech compression algorithms for Voice Over IP***", M.A.Sc Thesis, Ottawa-Carleton Institute for Electrical and Computer Engineering, School of Information Technology and Engineering, Ottawa, Ontario, Canada.

[59] Painter T. and Spanias A., "***Perceptual Coding of Digital Audio***", Proceedings of the IEEE, Vol. 88, No. 4, April 2000.

[60] Perkins C. and Hodson O., ***"Options for Repair of Streaming Media"***, Request For Comment (RFC) 2354, Network Working Group, June 1998.

[61] Perkins C., Hodson O. and Hardman V., ***"A survey of Packet Loss Recovery Techniques for Streaming Audio"***, IEEE Networks, Vol. 12, No. 5, pg. 40-48, Sept/Oct 1998.

[62] Proakis J. G. and Salehi M., "***Communication Systems Engineering***", Prentice-Hall Inc., Second edition 2002.

[63] Proakis J., "***Digital communication***", McGraw Hill, Inc. 1995.

[64] Proakis J.G. and Manolakis D., ***"Digital Signal Processing – Principles, Algorithms, and Applications"***, Second Edition, Macmillan Publishing Company, Inc. 1992.

[65] Pullen .M, "***Understanding Internet Protocols - Through Hands-On Programming***", John Wiley and Son, 2000.

[66] Quatieri T, ***"Discrete-Time Speech Signal Processing, Principles and Practice"***, Prentice-Hall, Inc. 2002.

[67] Ramamohan R.K., "***Techniques and Standards for Image, Video, and Audio coding***", Prentice-Hall, inc. 1996.

[68] RFC – RTP/RTPC , Request For Comment (RFC) 1889, Network Working Group, June 1998.

[69] Roden M., "***Digital Communication Systems Design***", Prentice-Hall, Inc. 1988.

[70] Rosenberg J.D., "***G.729 Error Recovery for Internet Telephony***", Lucent Technologies, Bell Laboratories & Columbia University.

[71] Ruggeri G., Beritelli F., and Casale S., ***" Hybrid multi-mode/multi-rate CS-ACELP speech coding for adaptive Voice Over IP"***, University of Catania, International Conference on Acoustics, Speech, and Signal Processing, May 7-11, 2001, Salt Lake City, Utah.

[72] Salomon D., ***"Data Compression: The Complete Reference"***, Springer-Verlag New York, Inc., Second Edition 2000.

[73] Sanneck H. and Le N., "***Speech property-based FEC for Internet Telephony Applications***", Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference (MMCN), January 2000.

[74] Sayood K., ***"Introduction to Data Compression"***, Academic Press, Second Edition 2000.

[75] Schafer R. and Rabiner L., ***"Digital Representations of Speech Signals"***, Proceedings of the IEEE, Vol. 63, No. 4, April 1975.

[76] Schulzrinne H. and Rosenberg J., ***"The Session Initiation Protocol: Internet-Centric Signaling"*** IEEE Communications Magazine, Vol. 38, No. 10, pp. 134-141, April 2000.

[77] Schulzrinne H. et al., "***A transport Protocol for Real-time Applications***", Request For Comments (RFC) 1889, Network Working Group, January 1996.

[78] Shlomot E., Cuperman V., and Gersho A., ***"Hybrid Coding: Combined Harmonic and Waveform Coding of Speech at 4 kbps"***, IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 6, September 2001.

[79] Spanias A., ***" Speech Coding: A tutorial review"***, Proceedings of the IEEE, Vol. 82, No. 10, October 1994.

[80] Stallings W., "***Data and Computer Communications***", Fourth Edition, Prentice-Hall, Inc.1994.

[81] Steinmetz R. and Nahrstedt K., "***Multimedia: computing, communications, and applications***", Prentice-Hall, Inc. 1995.

[82] Stevens, R.W., "***UNIX Network Programming Volume 1 – Networking APIs: Sockets and XTI***", Prentice-Hall Inc. , Second Edition, 1998.

[83] T1.521a-2000 – Annex B, "***Supplement to T1.521-1999, Packet Loss Concealment for Use with ITU-T Recommendation G.711***", Approved June 7, 2000, American National Standards Institute, Inc.

[84] Vetterli M. and Kovacevic J., "***Wavelets and Subband Coding***", Prentice-Hall, Inc. 1995.

[85] Voran S., ***"Objective Estimation of Perceived Speech Quality - Part I: Development of the Measuring Normalizing Block Technique"***, IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 4, July 1999.

[86] Voran S., ***"Objective Estimation of Perceived Speech Quality - Part II: Evaluation of the Measuring Normalizing Block Technique"***, IEEE Transactions on Speech and Audio Processing, Vol. 7, No. 4, July 1999.

[87] Wah B., Su X., and Lin D., "***A survey of Error-Concealment Schemes for Real-Time Audio and Video Transmissions over the Internet***", Proceedings IEEE International Symposium on Multimedia Software Engineering, December 2000.

[88] Wang J. and Gibson J., ***" Parameter Interpolation to enhance the frame Erasure Robustness of CELP Coders in packet neworks",*** Department of Electrical Engineering Southern Methodist University, Dallas, TX 75275, International Conference on Acoustics, Speech, and Signal Processing, May 7-11, 2001, Salt Lake City, Utah.

[89] Westwater R., ***"Digital Audio Presentation and Compression"***,CRC Press LLC, 1999.

[90] Yensen T. ,"***Structures and Interfaces for Voice Over IP***", M.Eng. Thesis, Ottawa-Carleton Institute for Electrical and Computer Engineering, Ottawa, Ontario, Canada, September 17, 1998.

[91] Yong M., "***A New LPC Interpolation Technique for CELP Coders***", IEEE Transaction on Communications, Vol. 42, No. 1, January 1994.