Assessing and Improving the VVoIP Call Quality

by

Hitham Assem, BSc.

Master of Science (MSc.) Thesis

Hamilton Institute

National University of Ireland Maynooth

Maynooth

Co. Kildare

August 2013

Research Supervisor: Dr. David Malone

Head of Department: Prof. Douglas Leith

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Master of Science is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not violate any law of copyright, and has not been taken from work of others save and to the extent such work has been cited and acknowledged within the text of my work.

Signed: Hitham Assem

(Student) ID No.: 11252097

Date: 01/08/2013

Abstract

Voice and Video over Internet Protocol (VVoIP) is a real time application that allows transmitting voice and video through the Internet network. Recently, there has been progress in this field due to continuous effort in developing new voice and video codecs that react appropriately under different network conditions. In addition, there are other factors that indirectly benefited VoIP. Today, computer networks are faster due to advances in hardware and breakthroughs in algorithms. Thus, the quality of VVoIP calls has improved considerably but still a lot of effort is needed to improve it under variable network conditions especially given that users are accustomed to the quality of service (QoS) they have enjoyed for years with the public switched telephone network (PSTN) and they will not accept lower call quality. We have observed from our collaboration with our industrial partner IBM that besides the need of improving the VoIP call quality, there is a need also for introducing more accurate and automated testing methods for estimating the VVoIP call quality.

The scope of the thesis and consequently its main contribution focuses on two main issues. First, assessing and monitoring the voice and video call quality. Second, improving and enhancing the voice call quality. In order to meet the first objective, we propose an improved simplified computational model to better predict the call quality which can be used further in monitoring purposes. In addition, we propose a VVoIP QoE automated framework to monitor and predict the voice and video call quality using different codecs under different network conditions. For the second objective and since the quality of VoIP calls under extreme conditions of packet loss still remains a major problem that needs to be addressed for the next generation of VoIP services, we have proposed two new adaptive techniques to improve the VoIP call quality: a generic switching codec algorithm and an adaptive redundant control algorithm. We have shown that both techniques will improve significantly the call quality if used in any VoIP application.

Dedication

To my Lord (Allah): “Who created me, and who guides me, (78) And who feeds me and gives me drink, (79) And when I become sick, He heals me, (80) And who will make me die, then will give me life, (81) And Who, I hope, will forgive my fault on the Day of Retribution. (82) O my lord, gives me wisdom, and makes me join the righteous (83)”

Quran: Chapter 26, Verses 78-83

Acknowledgment

One of the joys of completion is to look over the journey past and remember all the family and friends who have helped and supported me along this fulfilling road.

First of all, I am deeply grateful to Allah, my Lord, who gave me the power and confidence to pursue this work and complete my MSc studies.

I would like to express my sincere gratitude to my supervisor Dr. David Malone for his constant support and encouragement. He gave limitless portions of his time that made me feel that I am his only student. He always guided me with his advice while giving me the freedom and flexibility to pursue my research in various directions. Thanks to you David.

I am very grateful to my colleagues in Hamilton Institute, who provided me with help whenever needed. I would also especially thank the administrative stuff of Hamilton Institute, comprising of Rosemary Hunt and Kate Moriarty. In addition, I would like particularly to thank Mohamed Adel from WIT, Rami Ghorab and Ahmed Selim from TCD, Waleed Abo Hamad, Amr Arisha and Wael Rateb DIT and Ahmed Shosha from UCD whom I consider true brothers, for their unconditional support in everything since I came to Ireland. I would like also to thank all of the members of my research group for their continuous support and cooperation: Dr. Brendan Jennings from WIT, Dr. Pat O'Sullivan and Jonathan Dunne from IBM Software Lab, Yi Han from UCD and Himanshu Dadheech from WIT.

I would not have contemplated this road if not for my parents, Ahmed Assem and Azza Enany, who instilled within me a love of creative pursuits and science, all of which finds a place in this thesis. In particular, the patience, understanding and continuous help shown by my father is greatly appreciated. I am sure that without his continuous guidance, I can do nothing. I want also to thank my sister and brother, Dina and Mohamed for their encouragement and continuous support.

Last but not least, I would like to thank the person who this thesis would never have been possible without her support, understanding and patience. I would like to express my sincere gratitude to my soul-mate, my beloved fiancé Rana Maher for her understanding and love during this fulfilling road. She has been always my source of power and motivation. Thanks to you Rana. I love you.

Publications, Patents & Responsibilities

Publications

The following Conferences/Workshops publications were prepared in this course of this Masters. My contribution to these papers leads, respectively, to Chapter 4, Chapter 5, Chapter 6 and Chapter 7 respectively.

1. *Haytham Assem, David Malone, Jonathan Dunne and Pat O'Sullivan, "Monitoring VoIP Call Quality Using Improved Simplified E-model", IEEE International Conference on Computing, Networking and Communications (ICNC 2013), San Diego, USA, pp. 927-931. IEEE, January 2013.*

2. *Haytham Assem, Mohamed Adel, David Malone, Brendan Jennings, Jonathan Dunne and Pat O'Sullivan, "Online Estimation of VVoIP Quality-of-Experience via Network Emulation", Irish Signals and Systems Conference (ISSC 2013), Letterkenny, Ireland, June 2013.*

3. *Haytham Assem, Mohamed Adel, David Malone, Brendan Jennings, Jonathan Dunne and Pat O'Sullivan, "A Generic Algorithm for Mid-Call Audio Codec Switching", IFIP/IEEE International Workshop in Quality of Experience Centric Management (QCMAN 2013), Ghent, Belgium, pp. 1276-1281. IEEE, May 2013.*

4. *Haytham Assem, David Malone, Jonathan Dunne and Pat O'Sullivan. "A New Adaptive Redundancy Control Algorithm For VoIP Applications", IEEE Global Communications Conference (GLOBECOM 2013), Atlanta, USA. IEEE, December 2013.*

Patents

The following patents have been approved for filing sponsored by IBM Dublin and were extracted from my work in this Masters.

1. *Application for US patent, Haytham Assem, David Malone, Jonathan Dunne, Pat O'Sullivan, Daniel B. Kehn and James Galvin, "System and Method for enhanced unified call quality using FEC".*

2. *Application for US patent, Haytham Assem, David Malone, Jonathan Dunne, Pat O'Sullivan, Paul B.French and Pat Galvin, "System and Method to measure audio quality using non-intrusive method".*

Responsibilities

- The work represented in Chapter 4 and 7 of the thesis has mainly been carried by myself. David Malone reviewed the work and guided me in the correct direction. Finally, Pat O'Sullivan and Jonathan Dunne resolved any difficulties faced during testing in the IBM environment and provided industrial insight into the issues discussed.

- The work represented in Chapter 5 and 6 of the thesis has mainly been carried jointly between myself and Mohamed Mahmoud. David Malone and Brendan Jennings reviewed the work, guided us in the correct direction and shared in the conference paper writing. Finally, Pat O'Sullivan and Jonathan Dunne resolved any difficulties faced during testing in the IBM environment and provided industrial insight into the issues discussed.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we discuss the motivations behind the work of this thesis and provide an overview of the material presented in the following chapters.

1.1 Scope of the thesis

Recently and unlike the previous decades, during which almost all telecommunications network traffic was voice, we find today that the telecommunications market is driven largely by IP oriented applications and technologies. Voice and Video over Internet Protocol (VVoIP) applications have become an important application and are expected to carry more and more voice traffic over TCP/IP networks. VVoIP allows the integration of voice or video or both over the same channel. This leads to a new generation of applications, for instance, voice mail can now be easily integrated into the E-mail, virtual conference rooms are being placed around the world and services such as caller ID and call forwarding can be easily implemented in a packet switched network instead of the traditional circuit switched network. In the near future, we are expecting to see an overhaul of new services. In fact, VVoIP can be considered one of the most remarkable aspects in shaping the future of communications. As a consequence, VVoIP technology slowly replaces traditional telephony. This remarkable change has its impact on the users that are accustomed to the quality of service (QoS) they have enjoyed for years with the public switched telephone network (PSTN). On the other hand, VVoIP is based on IP networks that do not provide perfect network conditions and cannot guarantee stable and high call quality [1].

In real-time voice/video applications, the speech/video quality is impaired by packet loss, jitter, high delay and insufficient bandwidth. Consequently, VVoIP applications require low delay, low packet loss rates, low jitter and sufficient bandwidth in order not to affect the interaction between call participants. Thus, the prediction of the voice/video quality in different network environments and traffic loads may be of the same importance as monitoring the network to measure the voice/video quality as to prevent critical and potential deterioration before being raised. VVoIP has also benefited from improvements in digital signal processing as chips are being specifically designed to run certain type of voice/video codec algorithms. At the same time, there are also other efforts to provide a better QoS to VVoIP, such as class based queuing (CBQ) that differentiates traffic based on IP source addresses, and multi protocol layer switching (MPLS) that provides fast forwarding of packets at the router level, in addition to the switches that transmit at speeds of 1 to 40 Gbps, and hence the bandwidth will be plentiful.

Over the last few years there has been a remarkable progress in the field of VVoIP. The Telecom industry and top software companies like IBM (SUT), Google (G-talk) and Microsoft (Skype) have invested more to enhance the quality of the VVoIP of their applications in order to keep their potential users satisfied who continuously demand better perceived call quality. We have observed from our collaboration with our industrial partner IBM that such huge challenge requires improving the testing procedures for measuring the VVoIP call quality which is considered a time consuming process along with the industry needs for new methods to improve and provide better VVoIP call quality under different network conditions.

The main contribution of this thesis is to assess the VVoIP call quality by deriving an improved simplified model to better predict the VoIP call quality which can be used further for monitoring purposes. Also, we have proposed a new complete automated testing framework for measuring the voice and video call quality for different codecs under different network conditions without any audio/video sequences and without end user involvement for quality rankings. In addition, we focused in this thesis on improving the VoIP call quality by developing new method for packet loss recovery mechanism and proposing a codec switching algorithm illustrating how this proposed algorithms will improve the VoIP call quality if introduced to any VoIP application.

1.2 *Structure of the thesis*

The remainder of the thesis is organized as follows: Chapter 2 and 3 give a background of different topics that the thesis will be based on like the different types of audio and video codecs, protocols involved in the VVoIP system and the different methods used for measuring and estimating the voice and video call quality. Chapter 4 and 5 present our contributions regarding assessing the voice and video call quality. Chapter 6 and 7 deals with our contributions in improving the voice call quality. Finally, Chapter 8 summarizes and concludes the thesis. This structure can be described more as follows:

Chapter 2 introduces an overview of the codecs and protocols used in the Voice and Video over IP system. We discuss the different types of audio and video codecs and provide a brief summary about the protocols used in the VVoIP applications particularly the 'Real Time Protocol and Session Initiation Protocol' used in our research.

Chapter 3 presents the assessment of the VVoIP quality: the Quality of Experience (QoE) concept and its relation with the Quality of Service (QoS), the MOS score used for assessing the VVoIP call quality, commonly used methods for assessing the VVoIP quality, and finally the factors affecting the VVoIP quality perceived by the end user.

Chapter 4 introduces our first contribution in the thesis. It proposes an improved simplified E-model to better predict call quality. We demonstrate its results by implementing the derived model in a complete monitoring system where we show how each parameter is measured in runtime. The main advantage of our work in this chapter is the derivation of such model which is less complex than the original E-model and it is more accurate than the simplified versions.

Chapter 5 introduces our second contribution in the thesis. It suggests a framework for measuring the voice and video call quality to make use of the different derived non-intrusive testing methods like those proposed in Chapter 2. The suggested framework deals with the problem of the time consuming and intensive computation of the processing of audio/video sequences. We present in this chapter our framework which measures the voice and video call quality in advance and predicts the most appropriate codec according to the conditions of the current network without any audio/video sequences and without the involvement of the end user for quality rankings.

Chapter 6 introduces our third contribution in the thesis. It is presented in developing a technique to improve the voice call quality in the VoIP applications. The proposed technique in this chapter is based on codec switching during the call based on the network conditions. We have reviewed the literature in this area and our contribution in this aspect is presented by performing a detailed analysis of codec switching on voice quality for a wide range of codecs, deriving some heuristics for when and how often codec switching should be done. These heuristics are incorporated into our codec switching algorithm proposed in the chapter.

Chapter 7 introduces our fourth contribution in the thesis. It introduces a new loss recovery mechanism to be integrated with any VoIP application, named the APU Algorithm, addressing the drawbacks of the reviewed work. We show that our proposed algorithm will outperform the QoE when compared to the use of pure Reed-Solomon codes.

Chapter 8 is concerned with the conclusions of the thesis. It highlights the main contribution and achievements of the thesis. It also addresses the possible future work for further assessing and improving the VVoIP call quality.

The work described in this thesis has been accepted in four international venues. In addition two US patents are extracted from the work in this thesis have been approved for filing. The full list of the publications and patents are listed on page 6.

Chapter 2

Codecs & Protocols

In this chapter, we give a background of the audio and video codecs. We give also a brief description of the protocols that are involved in the VVoIP system especially those used in our particular research: Real Time Protocols and Session Initiation Protocol.

2.1 Codecs

The basic process of VVoIP includes the following steps as shown in Figure 2-1. This process starts by converting the voice/video into digital packets that will be converted into Internet protocol packets and then transmitting these packets over IP based network. Finally and at the receiver side, the conversion of digital packets into analogue voice/video occurs. The process of compression is carried out by a voice/video encoding algorithm called codec, which allows the call to be transmitted over the IP network. In general codecs vary in voice quality, the bandwidth required, sample period, frame size, frames per packet, computational requirements, etc.

Codecs are used to convert an analog voice/video signal to a digitally encoded version. Codecs vary in the sound/video quality, the bandwidth required, the computational requirements, etc. Each service, program, gateway, etc typically supports several different codecs, and when talking to each other, negotiate which codec they will use. The codecs also introduce a digitizing delay as each algorithm requires a certain amount of data to be buffered before it is processed. If the codec is very complex to implement, more CPU resources would be required and hence this too affects the VVoIP call quality.

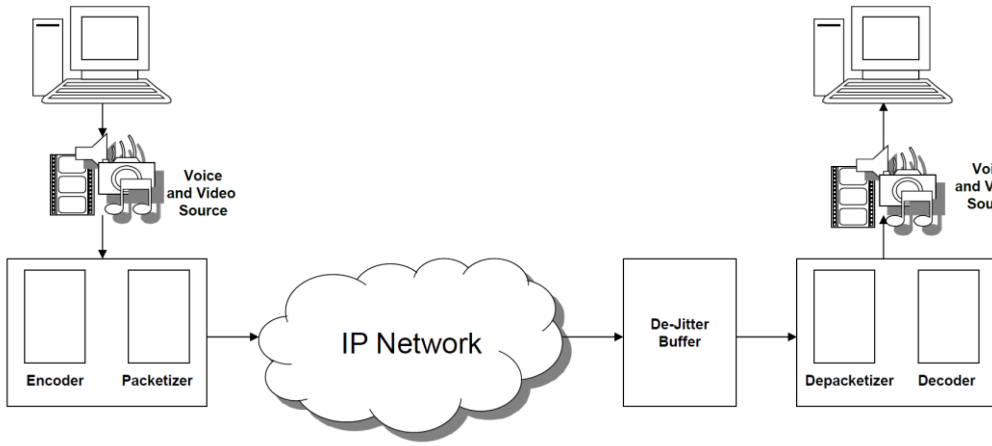


Figure 2-1 VVoIP System

2.1.1 *Audio codecs*

Every system implementing VoIP/IP Telephony uses an audio codec to encode the audio signals at one end and de-compress the same at the other end. Although most of them are standardized, VoIP vendors implement proprietary codecs too. Some examples of popular standardized codecs include G.723 [91], G.729 [92] etc. The type of codec used is an important factor that affects the VoIP call quality, as higher the compression, the lower the size of data to be transmitted over the other side over the IP network. But there is an opposite side too as the voice quality generally suffers with higher compression rates. Most codecs can accommodate different target compression rates like 8 Kbps, 6.4 Kbps, 5.3 Kbps etc (Standard 64 Kbps required to transmit voice over T1 lines – Single channel, PCM). The bit rates mentioned are for audio only, and protocol overheads must be added over that, hence the actual bit rate realized is quite high due to the effect of such overheads.

Some of the most commonly used codecs are under the ITU standards such as G.711 [93], G.729 [92] and G.723 [91] while others are not such as SILK [94], SPEEX [95] and ILBC [96]. So in this thesis, we usually use two terms to refer to these codecs as ITU codecs and non-ITU codecs. A summary of these codecs is shown in Table 2-1 and 2-2.

Table 2-1 ITU-T codecs

codec	Coding Method	Sample Rate	Bit Rate	Latency
G.711	A-law or μ-law, PCM, Lossy	8 kHz	64 kbit/s	125μs (typical)
G.711.1	MDCT, A-law, μ-law, Lossy	8 or 16 kHz	64, 80, 96 kbit/s	11.875 ms
G.721	ADPCM, Lossy	8 kHz	32 kbit/s	
G.722	sub-band ADPCM, Lossy	16 kHz	64 kbit/s	4 ms
G.722.1	Modulated Lapped Transform, (based on Siren codec), Lossy	16 kHz	24, 32 kbit/s	40 ms
G.722.1C	Modulated Lapped Transform, (based on Siren codec), Lossy	32 kHz	24, 32, 48 kbit/s	40 ms
G.722.2 (AMR-WB)	multi-rate wideband ACELP, Lossy	16 kHz	6.60-23.85 kbit/s	25 ms
G.723	ADPCM, Lossy	8 kHz	24 or 40 kbit/s	
G.723.1	MP-MLQ, ACELP, Lossy	8 kHz	5.3, 6.3 kbit/s	37.5 ms
G.726	ADPCM, Lossy	8 kHz	16, 24, 32, 40 kbit/s	125us
G.727	ADPCM, Lossy	8 kHz	16, 24, 32, 40 kbit/s	
G.728	low-delay CELP, Lossy	8 kHz	16 kbit/s	0.625 ms
G.729	CS-ACELP, Lossy	8 kHz	8 kbit/s	15 ms
G.729D	CS-ACELP, Lossy	8 kHz	6.4 kbit/s	
G.729E	CS-ACELP, Lossy	8 kHz	11.8 kbit/s	15 ms
G.729.1	CELP, TDBWE, TDAC, Lossy	8 or 16 kHz	8-32 kbit/s	48.9375 ms

Table 2-2 Non ITU-T codecs

codec	Coding Method	Sample Rate	Bit Rate	Latency
AMR-WB+	ACELP	8, 11.025, 16, 22.05, 32, 44.1, 48 kHz	6 kbit/s to 36 kbit/s (mono) 7 kbit/s to 48 kbit/s (stereo)	60–90 ms
GSM-HR	VSELP	8 kHz	5.6 kbit/s	25ms
GSM-FR	RPE-LTP	8 kHz	13 kbit/s	20-30ms
GSM-EFR	ACELP	8 kHz	12.2 kbit/s	20-30ms
iLBC	Block Independent LPC	8 kHz	13.33, 15.20 kbit/s	30, 20ms
iSAC	Transform coding	16 kHz or 32 kHz	10 to 52 kbit/s	33 to 63ms
SILK	LTP	8, 12, 16, 24 kHz	6 to 40 kbit/s	25ms
Speex	CELP	8, 16, 32, (48) kHz	2.15 to 24.6 kbit/s (NB) 4 to 44.2 kbit/s (WB)	30ms (NB) 34ms (WB)

2.1.2 Video codecs

In telecommunications and in particular in VVoIP products, the video codec is software that is used to enable compression at the sender side or decompression at the receiver side of the digital video. There is a tradeoff between the video quality perceived by the end user, the bit rate indicated by the quantity of the data needed to represent it, complexity of the encoding and decoding algorithms, robustness to data/packet losses, overhead in the one-way delay and some other factors [97]. Video codecs seek to represent the analog data in a digital format to be sent over the IP network in telephony applications. Due to the nature of the analog video signals, which represent luma and color information separately, a common initial step in image compression is storing the image in YcbCr color space. The conversion to such a space provides two benefits: first, it works on the separation between the luma signal from the chroma signal, this helps to achieve more efficient data compression. Second, it works on decorrelating the color of the signal to provide higher compressibility. The most commonly used video codecs are shown in Table 2-3. Refer to [97] for more information about the video codec design.

Table 2-3 Most Commonly used Video codecs

codec	**Compression format**	**Method of Compression (Lossy/Lossless)**	**Inventor/Creator**
X264	MPEG-4 AVC/H.264	Lossy/Lossless	X264 team
Xvid	MPEG-4 ASP	Lossy	Xvid team
FFmpeg	MPEG-1, MPEG-2, MPEG-4 ASP, H.261, H.263, etc.	Lossy/Lossless	FFmpeg team
DivX	MEG-4 ASP, H.264	Lossy	DivX Inc.
Nero Digital	MPEG-4 ASP, H264	Lossy	Nero AG
VP7, VP8	VP7, VP8	Lossy	Google
Windows Media Encoder	MPEG-4 version 2	Lossy	Microsoft

2.2 Real Time Protocols

Real time protocols cover specific needs of applications with real-time characteristics. Real-time applications such as Voice over IP (VoIP), videoconferencing applications, video on demand, continuous data applications and control and measurement applications have specific requirements from the lower layers, mainly in terms of packet loss, delay and jitter.

The TCP protocol is not appropriate for VVoIP because there is much delay associated with retransmissions. Accordingly, UDP protocol is used at the transport layer in VVoIP applications. However, UDP has some drawbacks that it does not ensure that packets will be delivered in the order they were transmitted. Figure 2-2 describes the header of the user datagram protocol (UDP) [4]. The UDP packet header consists of 4 fields only, and none of these fields contains a sequence number as in TCP. This can lead to some problems when transmitting audio packets over IP networks, because, if packets arrive out of its order at the destination, there is no mechanism to re-order packets. The lack of a sequence number can also lead to other major problems because a receiver is not be able to determine if an audio packet got lost in the network

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             Data                              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2-2 UDP Packet [2]

In particular, real-time protocols have to be able to deliver high throughput, manage multicast, handle the transmission quality and be friendly to the rest of the traffic and more importantly to the congestion-sensitive TCP traffic.

2.2.1 RTP – (Real Time Protocol)

RTP is designed for end-to-end, real-time, transfer of stream data. The protocol provides a facility for jitter detection of out of sequence arrival of data and jitter compensation, which is common during transmissions on an IP network. RTP supports data transfer to multiple destinations through IP multicast. RTP is considered as the primary standard for audio/video transport in IP networks. Usually, RTP is used in conjunction with a signaling protocol which assists in setting up connections across the network. RTP was developed by the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETF) and first published in 1996 as RFC 1889 [3], superseded by RFC 3550 [4] in 2003. The RTP protocol was designed to work in conjunction with UDP at the transport layer to ensure proper delivery of packets with real-time characteristics. Figure 2-3 describes the headers of the RTP protocol according to RFC 3550 [4]. Note that with RTP, every packet has a sequence number and a timestamp. The sequence number allows the receiver to order packets. The timestamp helps the receiver to identify the time at which voice packets were generated at the sender side. The headers of the RTP protocol are described below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2-3 RTP Header [3]

The RTP protocol is suitable for audio and video streaming. In the case of video streaming, two RTP sessions are established, each with different SSRC identifiers; one is used for audio transmission and the other for video. This separation between audio and video is in order to have the ability in a conference call to control which medium the participants would like to receive [4]. The RTP protocol is suitable for unicast and multicast sessions. On the other hand, RTP does not guarantee either the delivery of packets or Quality of Service (QoS) [4]. Figure 2-4 shows the RTP of the Ethernet or wireless communication over the TCP/IP protocol stack. The RTP can be used too with SCTP as the transport protocol; we focus on UDP as the transport protocol because it is widely used.

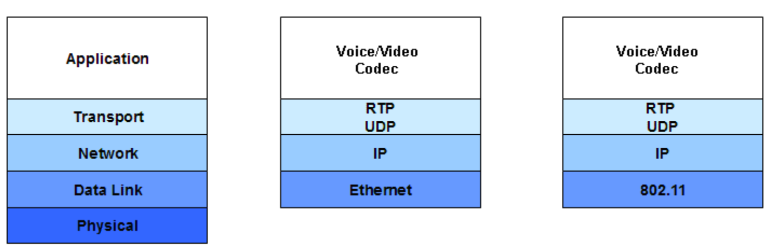


Figure 2-4 RTP Over different Network Conditions

2.2.2 RTCP – (Real Time Control Protocol)

RTCP stands for Real Time Control Transport Protocol and is defined in RFC 3550 [4]. RTCP works with RTP as RTP does the delivery of the actual data, where as RTCP is used to send control packets to participants in a call. The main function is to provide feedback on the quality of service (QoS) being provided by RTP. The most important factors that can monitored by the RTCP reports are the packet loss, fraction lost, jitter and one way delay. The Fraction lost is the number of packets lost divided by the number of packets expected. If duplicate packets are present of a so-called "negative loss", then the fraction lost is automatically set to 0 [4]. The RTCP packets or reports are received at equal periodic intervals. Usually, the recommended value for a minimum fixed interval for sending RTCP packets is 5 seconds [4].

One of the major drawbacks in the RTCP is that it reports packets which arrived, even if they arrived late. This drawback has been solved in an enhanced version of RTCP-XR which stands for Real Time Control Protocol Extended Reports. It reports to the sender the lost of packets due to jitter buffer discards [4].

2.3 Session Initiation Protocol (SIP)

SIP, the session initiation protocol, is the IETF protocol for VVoIP and other multimedia sessions, like instant messaging, online games and other services. It has been standardized by the RFC 3261 [6] - SIP: Session Initiation Protocol. SIP is like the web protocol, HTTP, as messages consist of headers and a message body. The SIP messages bodies are defined in the session description protocol (SDP) which has been standardized by the RFC 4566 [7]. SIP may use any port but it usually uses port 5060 as its default protocol for either UDP or TCP. In general and especially in VVoIP, SIP offers a some features such as media transfer, conference call and even call hold. Also, SIP is very flexible and can easily add extra features and keep downward interoperability.

SIP can be regarded as the enabler protocol for telephony, voice and video over IP (VVoIP) services. The following features in SIP as provided in [8] have played a major role in the enablement of IP telephony:

- Name Translation and User Location: This ensures the successful receipt of the call to the called party wherever they are located carrying out any mapping of descriptive information to location information. Ensuring that details of the nature of the call (Session) are supported.
- Feature Negotiation: This allows the persons or group involved in the call (multi-party call) to agree on the features supported in order to check that all the involved parties can support the same features. For instance, video may be is not supported for certain participants in the call.

- Call Participant Management: During a call, one of the participants can invite and bring other users onto the call or cancel connections to other users. In addition, users can be transferred or placed on hold.
- Call feature changes: A user should be able to change the call features during the call. For instance, a call may have been set up as voice only, but during the ongoing call, the users may need to enable a video function. A new user joining a call may need different features to be enabled in order to participate in the call.

2.3.1 Network Elements

SIP defines server network elements. Although two SIP endpoints can communicate without any intervening SIP infrastructure, which is why the protocol is described as peer-to-peer, this approach is often impractical for a general service. RFC 3261 [6] defines these server elements. The main network elements involved in the SIP communication can be illustrated as follows:

User Agent

The User Agent (UA) is the end point in the communication process which used in creating or receiving SIP messages to manage a SIP message. The UA can perform the role of the User Agent Client (UAC) to send the SIP messages while the receiver will act as a User Agent Server (UAS). These roles of UAC and UAS only last for the duration of a SIP transaction. The User-Agent field is sent in request messages, which means that the receiving SIP server can see this information. SIP network elements sometimes store this information, and it can be useful in diagnosing SIP compatibility problems.

Proxy server

An intermediary entity that acts as both a server (UAS) and a client (UAC) for the purpose of making requests on behalf of other clients. It plays the role of the routing to send the job requested to another entity closer to the targeted user.

Registrar

It is the server that is responsible for accepting the register requests and places the information it received in these requests into a location service in order to register one or more IP addresses to a certain SIP URI. More than one user agent can register at the same URI, with the result that all registered user agents will receive a call to the SIP URI. The SIP registrars are usually located with the SIP proxies.

Redirect server

A user agent server that generates redirection responses to the requests it receives, directing the client to contact another set of URIs. The redirect server allows proxy servers to direct SIP session invitations to external domains.

Session border controller

Session border controllers serve as *middle boxes* between UA and SIP server for various types of functions, including network topology hiding, and assistance in *NAT traversal* [98].

Gateway

Gateways are used in connecting the SIP network with different type of networks like the public switched telephone network (PSTN) which use different protocols and technologies.

SIP Messages

Recall the definition of SIP and that it is like HTTP; therefore there are two different types of SIP messages: requests and responses (Refer to [6] for more information about different types of requests and responses).

2.3.2 SIP VoIP Example

Let us briefly consider an example SIP call from Alice to Bob:

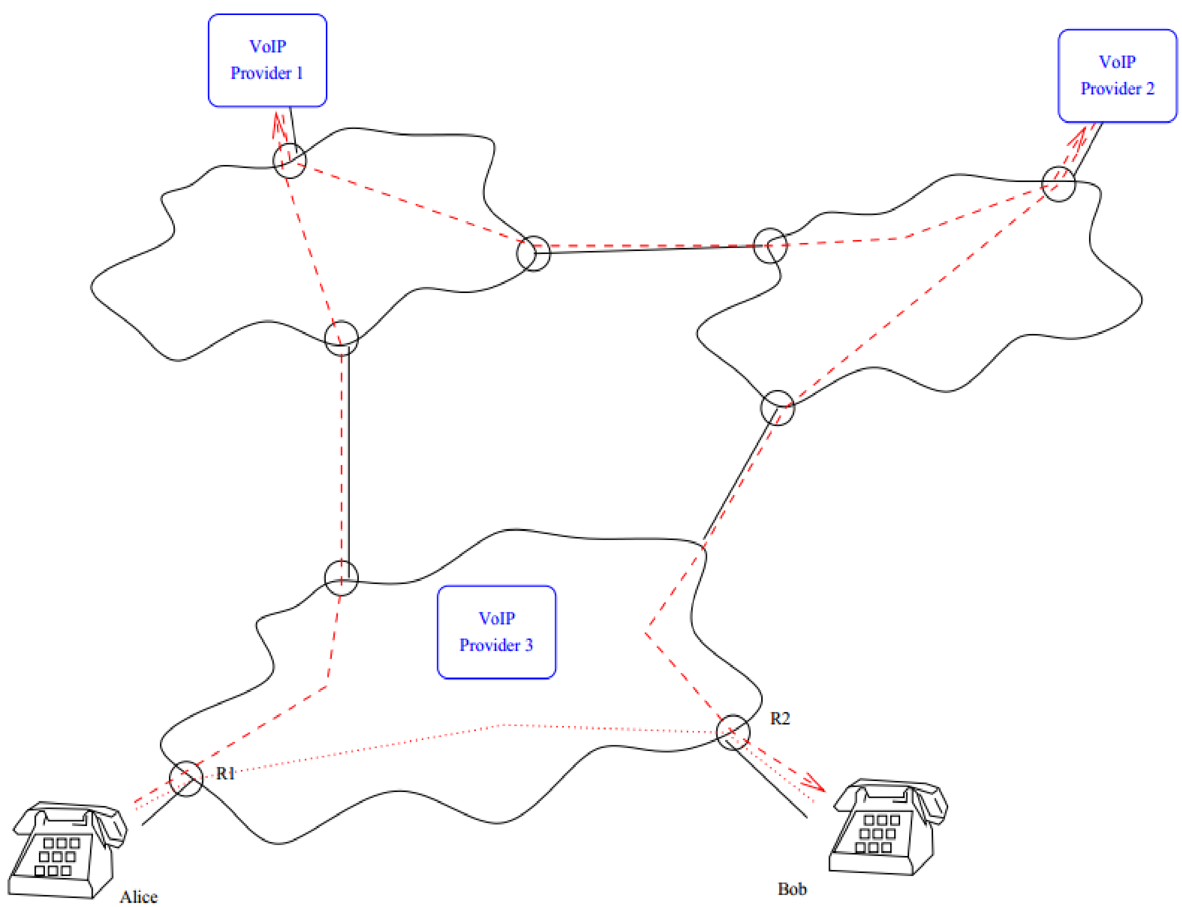


Figure 2-5 SIP Example

The example in Figure 2-5 assumes that Alice is using VoIP Provider 1 (VP1) as her proxy while Bob is using VoIP Provider 2 (VP2) as his proxy. So if Alice wants to call Bob. First, Alice sends a SIP URI to VP1 via TCP then the VP1 determines that the URI points to VP2, so the call's setup request is relayed there via TCP. Afterwards, VP2 tells Bob about the call via TCP; if he wants to, he can accept it, notification is sent back to Alice via VP1. A notification is sent back to Alice via VP1. Finally, Alice establishes a direct UDP data connection to Bob for the voice traffic.

2.4 Summary

In this Chapter, different types of audio and video codecs were described. We highlighted the protocols involved in the VVoIP System concentrating on the used ones in this thesis: Real Time Protocols and Session Initiation Protocol. The different types of the Real Time Protocols were reviewed including the RTP and RTCP protocols. We have also reviewed some literature related to the SIP protocol including its features and network elements. The main objective of this Chapter is to give a brief background on the codecs and protocols used in our work that is presented in next Chapters.

Chapter 3

Assessing VVoIP call Quality

In this chapter, we focus on assessing the VVoIP call quality, we introduce the term 'QoE' and a quantifiable metric (MOS) for measuring the quality of VVoIP. Finally, we described the different methods for measuring and assessing the VVoIP call quality and the different factors that affects the VVoIP quality.

3.1 Quality of Experience

In this section, a brief description about the Quality of Experience is given (see [9] for more details). The term 'Quality of Experience' was introduced in the late 90's; it has gained momentum and has been extended to different application contexts. Before the use of this term, the term 'Quality of Service' was used for many years. In the field of telephony, quality of service was defined by the ITU in 1994 [11]. Quality of service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, cross-talk, echo, interrupts, frequency response, loudness levels, and so on. Recently, the most well-known term to indicate a user's perception is QoE. QoE is also referred to as users perceived quality of service [12] and captures the experience of the users more clearly than end-to-end QoS [13].

The concept of QoE has emerged in this field mainly with the basic motivation that QoS is not powerful enough to fully express everything nowadays involved in a communication service. The increase of multimedia streaming over the Internet over the last years has created an interest in VVoIP quality assessment. Recently, QoE focuses on the human perception of quality at the end user rather than the network centered approach that has been used for many years before the classic approach. Human special factors are the main new thing that is embedded in QoE for assessing the VVoIP call quality [9]. So QoE is user-dependent because some users are easier to please than others. The number of users and subscribers in the quality testing assessment are highly related to the accuracy of the QoE evaluations obtained [41]. In general, major factors that affect QoE include cost, reliability, efficiency, privacy, security, interface user-friendliness and user confidence. Environmental variables that can influence QoE include the user's terminal hardware (for instance, hard-wired or cordless telephone set), the working environment (for instance, fixed or mobile) and the importance of the application (for example, casual texting versus critical videoconference communications). Although, the QoE is not always numerically quantifiable, it is considered the most significant single metric in the real world to evaluate the end-user experience [9]. It is in the best interest of any enterprise especially in the VVoIP applications to maximize its user QoE.

3.1.1 Definition of Quality and Definition of Experience

In order to give a formal understanding of the terms 'Quality' and 'Experience', we will first define the concept of an event as in [9]:

Event: An observable occurrence. An event is determined in space (i.e. where it occurs), time (i.e. when it occurs), and character (i.e. what can be observed).

Now we can define the term 'Experience' as [9]:

Experience: An experience is an individual's stream of perception and interpretation of one or multiple events.

After we defined the term Experience, we define the 'quality' as [9]:

Quality: Is the outcome of an individual's comparison and judgment process. It includes perception, reflection about the perception, and the description of the outcome. In contrast to definitions which see quality as "qualitas", i.e. a set of inherent characteristics, we consider quality in terms of the evaluated excellence or goodness, of the degree of need fulfillment, and in terms of a "quality event".

3.1.2 Definition of Quality of Experience

In the telecommunications context and especially in the VVoIP services, QoE is influenced by network conditions, device factors, service and content. The definition that is usually referred to in the literature is defined by ITU-T Rec. P.10 (Amendment 2, 2008) [10] as:

QoE: "The overall acceptability of an application or service, as perceived subjectively by the end user." which includes the complete end-to-end system effects and it may also be influenced by user expectations and context.

Some problems have arisen from the forgoing definition. Thus, in [9] the following definition of QoE was developed to overcome such problems related to the ITU-T Rec. P.10 definition [10]. The definition is as follows:

QoE: "Degree of delight of the user of a service. In the context of communication services, it is influenced by content, network, device, application, user expectations and goals, and context of use."

3.1.3 Factors influencing Quality of Experience

We categorized the factors affecting the quality of experience and the end user perceived call quality as follows:

Human Factors

This factor indicates that the individual perception of the audio call quality (qmos) is different from person to person according to the experience of using IP telephony products and the perception of the user.

Device Factors

Device factors include the impact of all of the devices involved in the communication process, because each device (MCUs, Routers, Firewalls, NATs, Modems, Operating system, Processor, memory … etc) might cause propagation delay or packet loss or even jitter which will affect at the end the user perception.

Network Factors

The network factors indicate the current network conditions during the call. These network conditions are often summarized by considering four quantities (Bandwidth, Delay, Jitter and Loss). We will describe each of these factors later in this chapter.

3.1.4 Relation between Quality of Service and Quality of Experience

Quality of Service has been defined by the ITU (ITU-T Rec. E.800, 2008) [11] as: "[The] Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service".

Due to the growth in the recent years in multimedia and VVoIP applications, users are expecting a very good quality of service compared to the PSTN call quality they have enjoyed for years, but best effort IP networks may not provide this. This is because best effort IP networks don't guarantee high call quality due to the nature of the IP network itself. The probability of this instability and unreliability increases from wired through wireless to mobile networks.

VVoIP applications are sensitive to network performance and conditions [14, 15]. For instance in VoIP call quality, the one-way delay can affect the user perception in the conversation even if the delay variation might be more critical from the end user perception point of view. The delay variation can be the cause of the packet-reordering and degradation of the perceptual quality. A jitter buffer is used to reduce the impact of packet delay variation. Moreover, the packet loss is another QoS factor that should be low and under a certain threshold level to sustain high perceived call quality. While in video streaming, if a key frame gets lost; video cannot be watched until the next key frame arrives. Several studies have been conducted to try to quantify the impacts of the network on the call quality [16, 17, 18, 19].

In a competitive environment, the assessment of user experience is one of the most important factors for the service provider. Therefore, it is highly desirable to build a model that can relate the QoE to QoS in order to understand, expect, predict and monitor the human perception from the QoS factors. Moreover, QoS measurements help in maintaining the requirement of the VVoIP applications and they help in selecting the parameter of VVoIP application with respect to available network resources. As result of this, efficiency of network and user experience can be improved.

In summary, many QoS factors of the network have a complex impact on QoE. We can see the QoE as an end-to-end measure of user satisfaction, regardless of the network technology underneath, while QoS measures the network performance.

3.2 Mean Opinion Score (MOS)

In voice and video communication, quality usually dictates whether the experience is a good or bad one. Besides the qualitative description we hear, like 'quite good' or 'very bad', there is a numerical method of expressing voice/video QoE called Mean Opinion Score (MOS). MOS gives a numerical indication of the perceived quality of the media received after being transmitted and eventually compressed using codecs. MOS is expressed in one number, from 1 to 5, 1 indicates the worst quality and 5 the best. MOS is quite subjective, as it results from what is perceived by people during tests. However, there are objective methods that measure MOS as will be discussed later in this chapter. The satisfaction level at the end user corresponding to the MOS score is shown below in Table 3-1.

Table 3-1 MOS vs. Satisfaction Level

MOS Score	Satisfaction level
5	Perfect (Like face-to-face conversation or radio reception)
4	Fair. Imperfections can be perceived, but sound still clear. This is (supposedly) the range for cell phones.
3	Annoying.
2	Very annoying. Nearly impossible to communicate.
1	Impossible to communicate.

The values do not need to be whole numbers. Certain thresholds and limits are often expressed in decimal values from this MOS spectrum shown in Figure 3-1. For instance, a value of 4.0 to 4.5 indicates complete satisfaction [103]. This is the normal value of PSTN and many VVoIP services are targeting such quality, often with success. Values dropping below 3.5 are considered unacceptable by many users. MOS can simply be used to compare between VVoIP services and providers. But more importantly, they are used to assess the performance of codecs under different network conditions. Testing accurately the quality of VVoIP is still considered a challenge, however services have greatly improved over the last few years as both the providers became more reliable and the ISPs offer better connections. Having a metric to measure changes or degradation in the quality of the VVoIP connection after testing can help in identifying problems. Accordingly, VVoIP calls often are in the 3.5 to 4.2 MOS range [13].

3.2.1 Types of MOS rating

In this section, we briefly review different types of MOS rating: Listening MOS, Network MOS and Conversational MOS. In addition, we show the factors affecting each type of MOS (see [99] for more details about the types of MOS rating.

Listening MOS

Listening MOS is a rating of the Listening Quality (MOS-LQ) of the audio stream that is played to the user. This value takes into consideration the audio fidelity and distortion and speech and noise levels, and from this data predicts how a large group of users would rate the quality of the audio they hear. This value takes into consideration the speech and noise levels of the user along with any external distortions, and from this data predicts how a large group of users would rate the audio quality they hear.

The Listening MOS varies depending on:

- The type of codec used (Narrowband or Wideband codec).
- Audio capture device characteristics.
- Occurrence of transcoding.
- Background noise at the sender side.
- Percentage of Packet loss (either random or burst losses).
- Speech level.

Since this type of MOS rating is a function of a large number of factors, it is preferred to measure the listening MOS statistically rather than using a single call.

Network MOS

Network MOS is a type of rating of the Quality of audio/video that is played to the user indicating the effect of the network QoS factors on the QoE at the end user perceived. This value takes into consideration only network factors such as: codec, random packet loss, burst losses and jitter. The difference between Network MOS and Listening MOS is that the Network MOS considers only the impact of the network on the QoE, whereas Listening MOS also considers the payload (speech level, noise level, etc). This makes Network MOS useful for identifying network conditions impacting the audio quality being delivered and providing solutions for the impairments of the network that impact the call quality. For each codec, there is a maximum possible Network MOS that represents the best possible Quality (MOS) under perfect network conditions. Because the maximum Network MOS varies depending on the scenario (because different codecs are used), it is usually more interesting to look at the average degradation of the Network MOS during the call. The average degradation can be broken down into how much is due to network jitter and how much is due to packet loss. For very small degradations, the cause of the degradation may not be available.

Conversational MOS

Conversational MOS is a rating of the audio or multimedia stream played to the user that takes into consideration the listening or seen quality of the audio/video played and sent across the network, the speech, noise levels for audio streams, echoes, and lip synchronization which is considered an important factor in multimedia call quality assessment. Such a MOS value represents how a large group of people would rate the voice or multimedia quality of the connection for holding a VVoIP call.

The Conversational MOS varies based on the same factors as Listening MOS, as well as the following:

- Echo.
- Network delay.
- Delay due to jitter buffering.
- Delay due to devices and codecs.

Similarly to the listening MOS, it is better to calculate the conversational MOS statistically rather than by using a single call due to the large factors that influence such type of MOS [99]. Throughout our research, we will use mainly the Network MOS and Conversational MOS. For instance, we use the Network MOS in monitoring the network conditions that influence the call quality and taking some decisions based on the expected Network MOS in order to improve the call quality. In addition, we use the conversational MOS in order to understand the real perception of end users under different network conditions.

3.3 Factors affecting VVoIP call quality

In this section, we describe the various factors that affect the call quality while transporting Voice and Video over IP (VVoIP) Networks. We take a look at factors like the type of audio/video codec used, delay, jitter and jitter buffer, packet loss, packet size, silence suppression, echo and other network parameters that affect the call quality for VVoIP applications.

3.3.1 Audio/Video codec

The type of codec used is an important factor that affects either the voice or video call quality as the higher the compression, the lesser the size of data to be transmitted over the IP network. But on the contrary, the voice/video call quality suffers from higher compression rates. Recently, codecs can support different compression rates such as 5.3 kbps, 6.4kbps and 8kbps etc. The bit rates mentioned are for audio only. There is an overhead result from the protocol added and hence the actual bit rate is higher. In addition, there is a digitizing delay introduced by the codec as each codec requires a certain amount of data to be buffered before being processed. If the codec is very complex to be implemented, more CPU resources would be required. All the mentioned factors affect the VVoIP call quality.

3.3.2 Delay

Delay is the amount of time that a packet takes to travel from the sender's side to reach the receiver's end caused by codecs, router queuing delays, etc. Different types of delays through the VVoIP system are shown in Figure 3-1. Refer to [100] for more description of different types of delay involved in the VVoIP system.

There are two distinct types of delay called fixed and variable.

- Fixed delay components add directly to the overall delay on the connection.
- Variable delays arise from queuing delays result in a jitter across the network. Variable delays are handled through the de-jitter buffer at the receiving router/gateway.

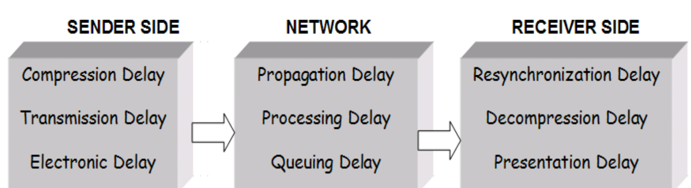


Figure 3-1 Types of Delay

Coder processing delay

Coder delay is the time spent in the digital signal processor to compress a block of PCM samples. Another naming is used for it: processing delay. This delay varies according to the type of codec used and the processor speed too. For instance, algebraic code excited linear prediction (ACELP) algorithms analyze a 10 ms block of PCM samples, and then compress them.

Algorithmic delay

There is usually look ahead introduced by the compression algorithm which is considered an additional delay. This is due to the fact that the compression algorithm relies on known voice characteristics to correctly process sample block N. This delay is a result of the fact that the algorithm must have some knowledge of what is in block N+1 in order to accurately reproduce block N.

Packetization Delay

Packetization delay is the time taken to fill a packet payload with encoded/compressed speech before sending over the IP network. This delay is dependent on the sample block size required by the coder and the number of blocks placed in one frame.

De-Jitter Delay

The variable delay of the arrival of the packets called jitter arises from queuing delays; the jitter must be removed before the signal leaves the network. In most of the current routers/gateways this is done with a de-jitter buffer at either the receiving router or gateway. The de-jitter buffer works on transforming the variable delay into a fixed delay. This is done by holding the first sample arrived for certain period of time before it plays it out. This holding time is called De-jitter Delay.

3.3.3 Queuing/Buffering Delay

After the compressed voice/video payload is built, a header is added and the frame is queued for transmission over the IP network. Voice/Video needs to have a priority in the router/gateway [104]. Therefore, a voice/video frame must only wait for either a data frame that already plays out, or for other voice/video frames ahead of it. So, such waiting time is called Queuing or buffering delay.

3.3.4 Jitter

Jitter is the variation in delay of the packets arriving at the receiving end caused by congestion, insufficient bandwidth, varying packet sizes in the network and out of order packets. Excessive jitter may cause packet loss in the receiver jitter buffers thus affecting the playback of the audio and video streams. When the packets are sent from the codec after compression, they are sent at a fixed or constant rate with equal spacing between them. But when they are received at the other side, the decompression algorithm also expects the packets to arrive with equal spacing between them and in the same order as they were sent. But since network introduces delays at packet level, the packets may arrive at different time intervals and they may not arrive in the same order, as they were sent.

3.3.5 Packet loss

There is always some risk of packets being lost in an IP Network. It may be for many reasons such as excessive collisions, physical media errors, overloaded links etc. Some of the current protocols such as TCP allow for recovery of lost packets, but others protocols such as UDP, which is commonly used in VVoIP applications, doesn't allow recovery of lost packets. In general, packet losses up to 5 % may not cause a noticeable degradation in voice quality. But more than 5 %, loss might lower the call quality.

3.3.6 Packet Size

The packet size poses an interesting tradeoff. If the RTP packet size is bigger, the overall bandwidth is reduced as more information is packed in to a single packet and there is a smaller amount of overhead control packets and header information which needs to be added to every packet sent over the IP network. From this point of view, it is better that the packet size is bigger, but if the packet size is too big then more packetization delay is introduced which is induced as the sender needs to wait for more time to fill the payload. Generally, it is better to send bigger sized packets as to reduce the overall bandwidth required. But this is done by increasing the inter-arrival timing so it is better to check whether there is a delay budget that allows for it. In certain person to person calls in VVoIP applications, cRTP (Compressed RTP) is preferred as it compresses the header information required to send control signals across. Generally, cRTP reduces the size of each packet by almost half, but processing overload on routers is generated.

3.4 Methods for assessing VVoIP call quality

In this section different parametric methods for testing the voice and video over IP call quality are presented. We present in this section the most commonly used methods for measuring the VVoIP call quality which are relevant to this thesis. These methods have been proposed in recent years. Each method is briefly described, and the parametric formula is detailed. As measuring voice and video quality is important to the service providers and end users, ITU-T provides two test methods subjective and objective testing that can be described as follows:

3.4.1 Subjective Testing

Conversational quality testing is complex, and hence, used much less frequently. In a conversational test, a pool of listeners is typically placed into interactive communication scenarios and asked to rank a voice or video quality. Testers introduce different effects such as packet loss and delay, and the test subjects are asked for their opinion on the quality of the connection. The loss has a direct effect on either the voice or video call quality. But the situation is different in the effect of one-way delay as described below.

3.4.1.1 Voice Testing

In non-interactive tasks, one-way delays of several hundred milliseconds can be tolerated; for highly interactive tasks, even short delays can introduce conversational difficulty and might irritate the user. The task dependency of delay introduces some question over the interpretation of conversational call quality metrics. For example, two identical VoIP system connections have 250 milliseconds of one way delay; however, one supports a highly interactive business negotiation, while the other supports an informal chat between friends. In the first example, users may say that call quality was bad: in the second case, the users probably would not even notice the delay. Hence, the effect of delay is very task dependent.

3.4.1.2 Video Testing

On the contrary to voice testing, in video quality assessment, the effect of delay will not be included in the conversational subjective video testing but should be included in the multimedia call quality assessment. This can be explained because when an assessment is placed into an interactive video communication service only, and a pool of users is asked to assess the video call quality, users will not observe the delay factor as they are assessing the quality of the video pictures only but delay will be included in assessing the multimedia call quality as the synchronization between the audio and video will be taken into consideration from the users' perception point of view.

3.4.2 Objective Testing

In recent years new methods were developed for measuring MOS scores in an objective way (without human perception) in order to save time. So, in general the objective testing methods are classified as Intrusive and Non-intrusive testing methods. The main difference between both methods is that Intrusive testing methods require the original and degraded files to compare and get a MOS score indicating the call quality, unlike the non-intrusive methods, that focus on converting the network QoS factors to a MOS score indicating the QoE without the requirement of recording or comparing files.

In an effort to supplement subjective listening quality testing with lower cost objective methods, the ITU developed P.861 (PSQM) [21], the newer P.862/P.862.2 (PESQ) [22, 23] and the G.107 [24] (E-model) for voice quality assessment. PSNR and G.1070 [25] were introduced for obtaining video quality. These measurement techniques determine the distortion introduced by the IP network or codec by comparing an original reference signal sent into the system with the degraded signal that came out. Although these techniques were developed for lab testing of codecs, they are widely used for VVoIP network testing. In audio testing, the P.861 [21] and P.862 [22, 23] algorithms divide the reference and impaired signals into short overlapping blocks of samples, calculate Fourier Transform coefficients for each of these blocks and finally compare the sets of coefficients. P.862 [22, 23] produces a PESQ score that has a similar range to MOS (1-5); however, it is not an exact mapping. The new PESQ-LQ score is more correlated with listening quality MOS. These algorithms both require access to both the original file and the degraded file in order to measure the PESQ MOS score. In 2004, the ITU standardized P.563 [26], a single ended objective measurement algorithm that is able to operate on the received audio stream only. The MOS scores produced by P.563 are more widely spread compared to PESQ scores produced by P.862 but it is necessary to average the results of multiple tests in order to achieve a stable quality metric. Consequently, this method is not suited for measuring single calls but it should be used over many calls to produce reliable results. As this type of algorithm requires significant computation for every sample (processing for each of 8,000 samples per second for narrowband voice and 16,000 samples per second for wideband voice), the processing load and memory requirements are quite significant [101]. In video testing, PSNR works in quite similar way to PESQ. This method assesses the performance of video transmission systems by calculating PSNR (Peak Signal to Noise Ratio) between the original and the received (degraded) video. PSNR is a differential metric which is computed using images. It is quite close to the widely-known SNR (Signal to Noise Ratio), but the difference is that PSNR gives a better predictionn to the Quality of Experience of users. In this section, we focus on the most commonly used methods, either intrusive or non-intrusive, which will be used in our research.

3.4.2.1 Intrusive Testing

Intrusive objective methods compare either the original speech or video signal to the corresponding degraded speech or video signal to obtain a MOS rating. Intrusive objective methods developed for objective voice quality measurement include PAMS [27], PSQM [21], MNB [21], PSQM+ [28], and PESQ [22, 23]. The PEVQ [29], SNR and PSNR are the most widely intrusive techniques used for assessing the video quality. Recently, the new intrusive methods are reliable, but normally are unsuitable for monitoring live traffic because of the need for a reference audio sequence. The ITU-T P.862 [22, 23] PESQ (Perceptual Evaluation of Speech Quality) and PSNR (Peak-Signal-to-Noise-Ratio) are the most widely used methods for measuring the voice and video quality respectively.

Voice Testing - Perceptual Evaluation Speech Quality (PESQ)

PESQ algorithm consists mainly of two stages. The first stage is computing the series delays between original and degraded signals which is called "time alignment". In the second stage, the original and degraded signals are transformed to a psychophysical representation in the human auditory system. The output from the PESQ algorithm is a MOS score range from 1 to 5 [23]. PESQ is a powerful tool for measuring one-way speech transmission voice quality and its results are highly correlated with the subjective measure of the voice quality. However, PESQ cannot be used for measuring the conversational call quality as it does not take into account the delay impairments. Refer to [23, 24] for more details about P.862 PESQ. Figure 3-2 shows the PESQ testing operation.

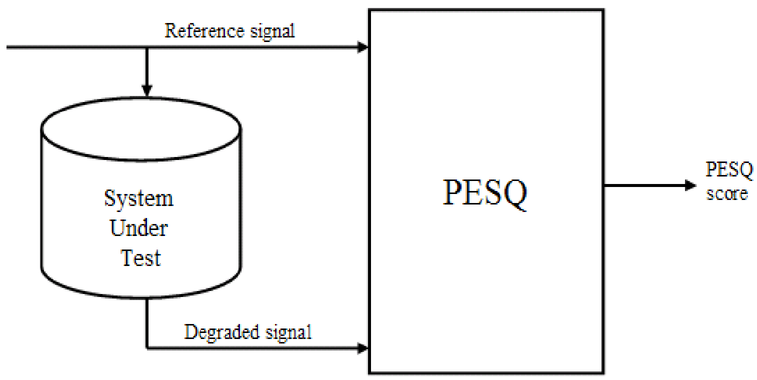


Figure 3-2 PESQ Testing

Video Testing - Peak-Signal-to-Noise-Ratio (PSNR)

The PSNR is considered an intrusive testing method for video quality assessment that compares the original video signal with the degraded video signal through the IP network. Signal-to-Noise-Ratio (SNR) and PSNR are considered the most traditional ways of evaluating quality of video processing system. PSNR formula is derived using the Mean Squared Error (MSE) with the maximum possible value for the luminance with 8-bit value (255) as shown in (3.1, 3.2):

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[(f(i,j)-F(i,j)\right]^2}{M \times N} \qquad (3.1)$$

$$PSNR = 20 \cdot \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \qquad (3.2)$$

Where *f(i, j)* is the original signal at pixel *(i, j), F(i, j)* is the formed or reconstructed signal, and *M x N* is the number of pixels in the picture. The result finally will be in decibels ranging from 30 to 40 for medium to high quality video. However, PSNR values do not perfectly correlate with a perceived visual quality due to the non-linear behavior of the human visual system. For this, a great effort has been exerted to develop several objective video quality models but PSNR continues to be the most popular evaluation of the video quality.

3.4.2.2 Non-Intrusive Testing

Non-intrusive methods do not need a reference signal and are used for monitoring live traffic. The non-intrusive methods that have been developed include methods like in [30] [31] based on the ITU-T G.107 E-model [24], methods based on speech recognition [32] [33], output based method such as 3SQMTM [34], and methods based on artificial neural networks [35] [36] [37], etc. On the other hand, VQM introduced by G.1070 [25] provides an online technique to calculate video quality based on various factors including current network conditions, codec used and properties of the transmitted sequence.

Voice Testing - E-Model

The E-model is considered an objective model proposed by ITU-T G.107 [24]. It takes into account various degradations that affect the speech quality and the end user level of satisfaction. Unlike the PESQ approach, the E-model can monitor the real time call quality by mapping the QoS factors to a QoE MOS score. It calculates finally a rating factor called R that range from 0 to 100. R=0 indicates the worst quality while R=100 indicates the best quality. The R factor value is expressed as in (3.3).

$$R = R_0 - I_s - I_d - I_{e-eff} + A \qquad (3.3)$$

R_0 represents the basic signal-to-noise ratio, including noise sources such as circuit noise and room noise. The factor I_s is a combination of all impairments which occur more or less simultaneously with the voice signal. Factor I_d represents the impairments caused by delay and the effective equipment impairment factor I_{e-eff} represents impairments caused by low bit-rate codecs. It also includes impairment due to randomly-distributed packet losses. The advantage factor *A* can be used for compensation when there are other advantages of access to the user. *R* can be transformed into a MOS scale using equations in (3.4):

For R<0: MOS=1

For $0 \le R \le 100$: $MOS = 1 + 0.035 \times R + R(R-60)(100-R) \times 7 \times 10^{-6}$

For R>100: MOS=4.5 (3.4)

Figure 3-3 shows the components of the connection of the E-model. The transmission parameters used as an input to the computation model are shown in Figure 3-3. Refer to [24] for the method of calculation of such parameters and more understanding about the E-model.

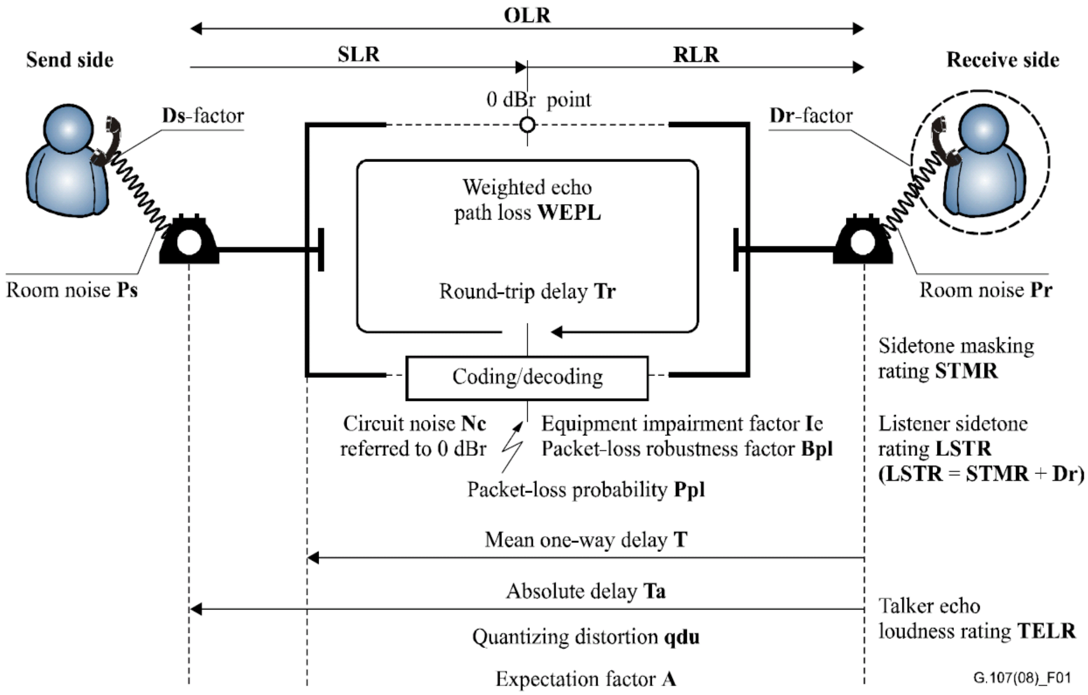


Figure 3-3 Reference Connection of the E-model

Video Testing - ITU-T G.1070

Recently, ITU-T has published a model for predicting video quality in video telephony applications based on the measured parameters of the IP network that will be translated to a score indicating the video quality. ITU-T G.0170 [25] introduced a computational model for point-to-point interactive videophone applications over IP networks. The model is considered a non-intrusive testing method that can be applied in live video quality monitoring purposes. It is similar to the E-Model and is mainly based on the work done by K. Yamagishi et al. [38] [39]. Figure 3-4 shows the whole model which is mainly composed of three functions, one for video quality (Vq), other for audio quality estimation (Sq) and the last one is for the overall multimedia quality estimation (MMq). Particularly, we will use in our research the Vq function to give an estimate for the video quality. Refer to [25] for the method of calculating the different functions.

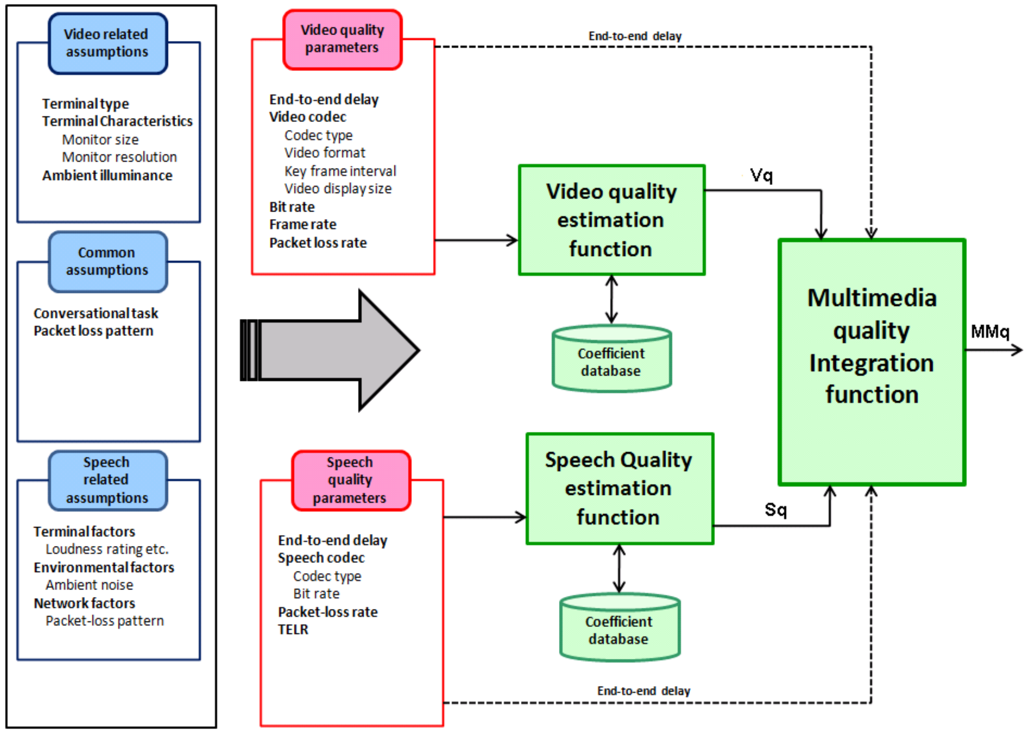


Figure 3-4 ITU-T G.1070 different functions and components [25]

3.5 Summary

After giving a background in the previous Chapter on the codecs and Protocols that are used in the work proposed in this thesis. In this Chapter, we focused on reviewing the literature of assessing the VVoIP call quality. First, we introduced the term QoE and the quantifiable metric MOS used for measuring the audio and video call quality for VVoIP systems. Also, the factors affecting VVoIP call quality have been discussed including the codec used in the call, different types of delays, jitter, packet loss and packet size. The different methods used in assessing the VVoIP call quality have been reviewed including the subjective and objective testing methods. A special focus has been given to the methods used in the work proposed in this thesis. From the next Chapter, we will introduce our contributions in the thesis.

Chapter 4

Monitoring VoIP call quality using Improved Simplified E-model

In this chapter, we introduce the simplified E-model used for monitoring VoIP call quality. We improve this model to better predict the call quality. We demonstrate its results by implementing it in a complete monitoring system and describing the method of measurement of each parameter in runtime. The main advantage in the work in this chapter is the derivation of an improved simplified E-model which is less complex than the original E-model and it is more accurate than the simplified versions commonly used.

4.1 Introduction

The evaluation of data networks depends on several factors. Thus, it is argued that it is not appropriate to use a single metric to evaluate the quality of data networks. Yet in the telephony world, a single number is typically given to rate call quality. Such a value is used as a basis of monitoring and tuning the network. Voice over Internet Protocol (VoIP) is an example of such data network application [40].

In previous years, VoIP has become an important application and is expected to carry more and more voice traffic over TCP/IP networks. In real-time voice applications, the speech quality is impaired by the packet loss, jitter, high delay and insufficient bandwidth. Consequently, VoIP applications require low delay, low packet loss rates, low jitter and sufficient bandwidth in order not to affect the interaction between call participants.

VoIP is based on IP networks; however IP networks frequently provide best effort services, and may not provide perfect network conditions [1]. So, the prediction of voice quality in different environments and traffic loads may be an important part of network monitoring in order to measure voice quality and prevent critical problems before they occur.

As measuring voice quality is important to the service providers and end users, recall from the third chapter that ITU-T provides two test methods subjective and objective testing. Subjective testing was the earliest attempts addressing this issue to evaluate the speech quality by giving Mean Opinion Scores (MOS). The MOS test is one of the widely known accepted tests that give a speech quality rating. ITU-T Rec. P.800 [41] presents the MOS test procedures as users can rate the speech quality on a scale from 1(Poor) to 5 (Excellent). Of course, the numbers of the listeners are considered an important factor in estimating accurate scores. Thus, subjective testing using MOS is time consuming, expensive and does not allow real time measurement. Consequently, in recent years new methods were developed for measuring MOS scores in an objective way (without human perception): PESQ [22], E-model [25] and several others.

PESQ, Perceptual Evaluation of Speech Quality, is considered an objective method for predicting the speech quality. It is an intrusive testing method which takes into account two signals; one is the reference signal while the other one is the actual degraded signal. Both signals are sent through the test that uses a PESQ algorithm and the result is a PESQ score (see Chapter 3 for more information). Consequently, this approach cannot be used to monitor real time calls. Nowadays, a new objective method proposed by TU-T G.107 [25] defines the E-model, a mathematical model that combines all the impairment factors that affect the voice quality in a single metric called the R value that is mapped to the MOS scale. The E-model was designed to provide estimated network quality and has shown to be reasonably accurate for this purpose. It has not been accepted as a valid measurement tool for live networks. The ITU-T G.107 Recommendation [25] states at the beginning of the document that "*it is considered only estimates for the transmission planning purposes and not for actual customer opinion prediction*", unlike the PESQ [22] which is developed to model subjective tests commonly used in telecommunications to assess the voice quality by human beings. Increasingly and against ITU recommendations, the E-model is being used nowadays by industry and research as a live voice quality measurement tool. Thus, simple versions of the E-model [40, 42] have been proposed to simplify the complexity of the original E-model [25] and focus on most important parts that affect the VoIP call quality.

The objective of our work in this chapter is to provide a monitoring system using a simplified version of the E-model corrected for four common codecs to better predict PESQ MOS scores as PESQ is generally considered to provide more accurate predictions of user experience than the E-model. Our objective in this part is to propose a non intrusive testing method that will be able to monitor the VoIP call quality more accurately compared to the current methods used for monitoring the call quality especially that the current methods used are not developed for such purpose but they were only for transmission planning purposes. We implement the derived model in a complete monitoring system in order to be able to use it in the VoIP applications. So after proposing our improved developed mathematical model, we make use of it by implementing it in a full monitoring system.

4.2 Simplified E-model

The original E-model is very complex [25] and involved with many factors. Moreover, the voice processing is not related significantly to the instantaneous judgment of QoS. Thus, a simplified version of the E-model [42] has been introduced to focus on the most important parts and afterwards it was used in a monitoring system [40]. This model takes into account the codec and the present network conditions which are the main two factors that affect the voice quality. The simplified E-model is expressed by equation (4.1) by calculating the evaluation value R.

$$R = R0 - Icodec - Ipacketloss - Idelay \qquad (4.1)$$

Where *R0* represents the basic signal to noise ratio, *Idelay* represents the delays introduced from end to end, *Icodec* is the codec factor and the *Ipacketloss* is the packet loss rate within a particular time. Finally, the *R* value is mapped to the MOS score.

4.3 Improved Simplified E-model

In this section and after describing the simplified E-model used nowadays by research and industry, we will first correct this simplified E-model by the one of the world wide applied industry standard for assessing the speech quality as experienced by the end user of the telephony system (PESQ). Our objective in the first part of this section is to derive the correction coefficients to better predict the PESQ MOS scores. In the second part of this section we show how to measure the different network parameters required for our improved simplified E-model in order to be applicable to monitoring call quality.

4.3.1 Correction of the Simplified E-model

In our experiment shown in Figure 4-1 we have developed a java program that streams RTP packets using 4 main audio codecs (G.711, G.726, G.723.1 and G.729A). We recorded the voice at both ends and measured the PESQ scores under different random packet loss rate range from 0-20%. For each packet loss rate, we repeated the experiment 10 times taking the average MOS PESQ score in order to increase the accuracy of our results as much as possible.

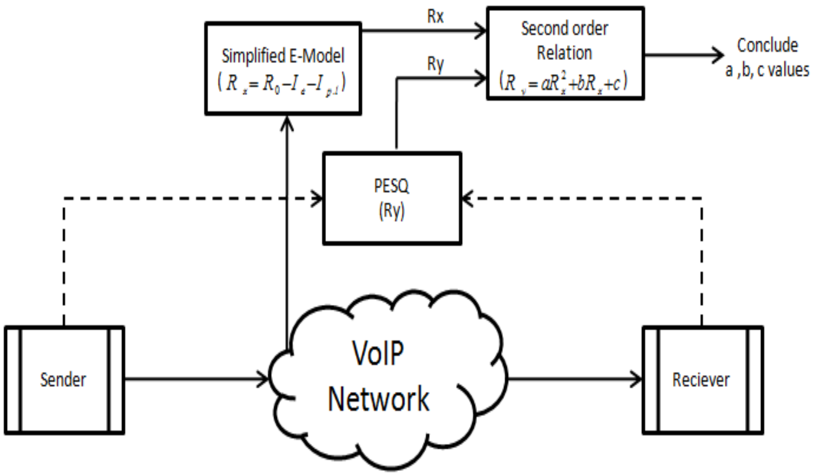


Figure 4-1 Deriving codecs' coefficients a, b and c

The PESQ scores are converted from MOS to R value and this can be conducted by a complicated Candono's Formula as in [43] or by the simplified 3rd-order polynomial fitting [44] as shown in (4.2).

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.060MOS - 57.336 \qquad (4.2)$$

The converted PESQ scores from (4.2) will be the R_y values shown in Figures 4-2 – 4-5 on the y axis. PESQ does not take the delay factor in its account, so we correct the model which named R_x (see equation 4.5) represented on x axis.

We found that the relationship between R_x and R_y is well matched to quadratic relation function as the rate of the increase increases. Likewise a cubic relation function would be a reasonable choice as well, as both are computationally efficient. However, we felt the errors were already small enough using a quadratic. We derived the codec's coefficients (a, b and c) as in Table 4-1 using the least-squares fitting method. The graphs below (Figures 4-2 – 4-5) show the correlation between the converted values from PESQ and the R values from the simplified E-model for individual codecs in different loss range.

Table 4-1 codecs Special Coefficients

codec	a	b	c
G.711	0.18	-27.90	1126.62
G.723.1 5.3k	0.039	-4.2	166.61
G.726 24k	0.046	-4.53	168.09
G.729A	0.063	-8.08	311.72

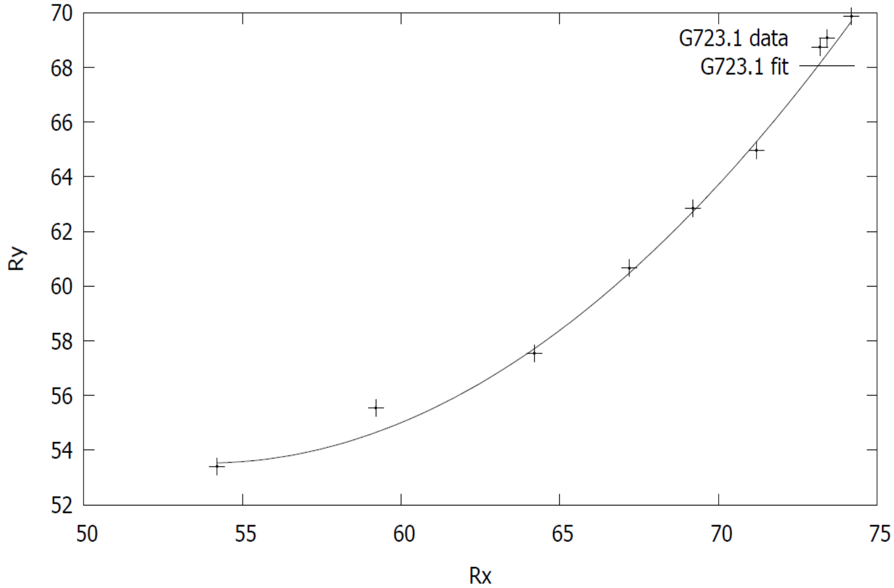


Figure 4-2 Relationship between R_x and R_y for G.723.1 codec

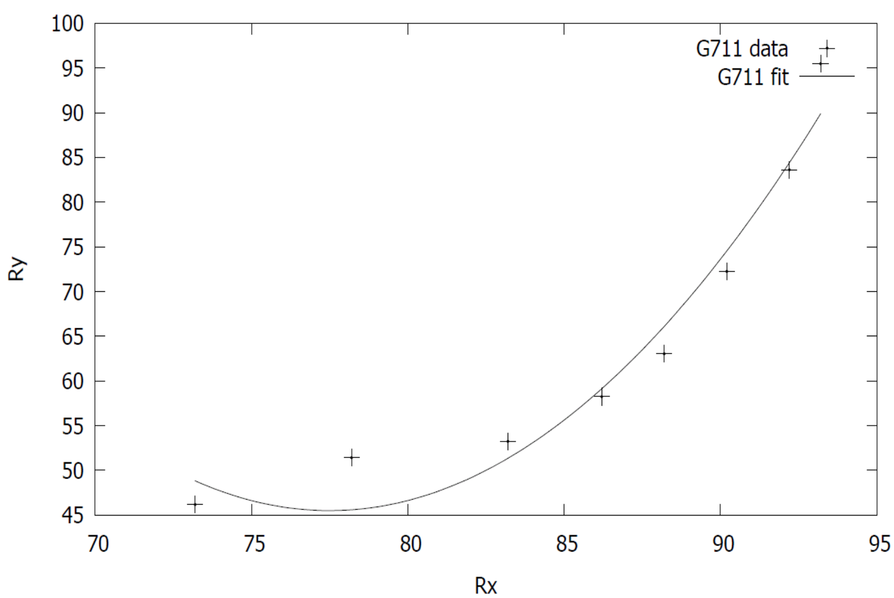


Figure 4-3 Relationship between R_x and R_y for G.711 codec

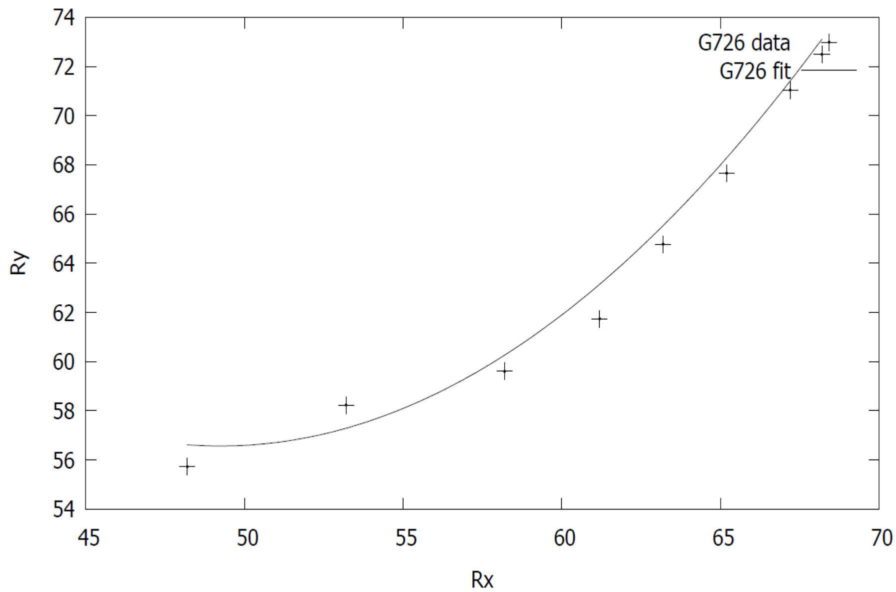


Figure 4-4 Relationship between R_x and R_y for G.726 codec

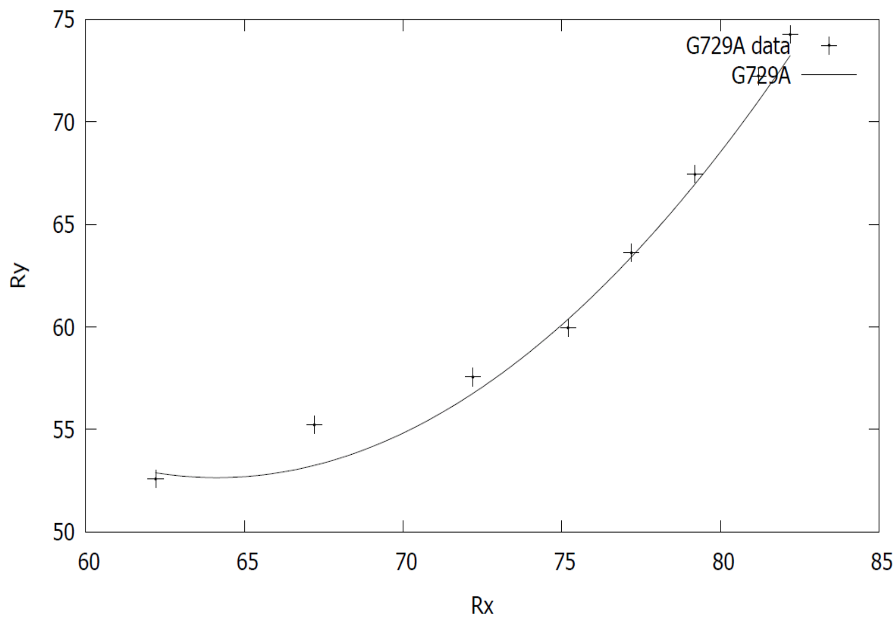


Figure 4-5 Relationship between R_x and R_y for G.729 codec

4.3.2 Network parameters of proposed Improved Simplified E-model

The objective of this model is to determine the voice quality MOS rating by a simplified modified version of the previous E-model described above. The computational model consists of a mathematical function of parameters of the transmission system. The computation itself can be split into several elements and can be expressed by the following equation (4.3). In this section we provide the method for measuring the different network characteristics used in our model in order to be able to monitor the call quality:

$$R = R_y - I_d + A \qquad (4.3)$$

Where R_y is a second order function corrected using curves fitted to PESQ scores which is the standard objective method defined by ITU-T recommendation P.862 [22], I_d is the average delay time within specified period and A is the expectation factor as a result of the communication system. The description and method of calculating the previous parameters (R_y , I_d and A) in (4.3) are as follows:

1) R_y :

R_y as mentioned above is a second order function model corrected with PESQ scores to obtain more accurate results in our monitoring system. R_y can be expressed by the following equation (4.4).

$$R_y = aR_x^2 + bR_x + c \qquad (4.4)$$

Where R_x is a part of the simplified E-model (4.1) which is corrected with PESQ scores, R_x can be obtained by the following expression (4.5) and a, b, c are codecs' coefficients as shown in Table 4-1 and derived in section 4.3.1.

$$R_x = R_0 - I_e - I_{p.l} \qquad (4.5)$$

1.1) R_0

R_0 is the basic signal to noise ratio, including noise sources such as circuit and room noise. However, currently it is difficult to calculate R_0 directly. Thus, ITU-T G.113 [45] provides the common value of R_0 . The inherent degradation that occurs when converting actual spoken conversation to a network signal and back reduces the theoretical maximum R-value (94.2) with no impairments to 93.2 [25]. So, we set the R_0 value to 93.2.

1.2) I_e

I_e is the equipment impairment (codec quality) factors as defined in [45] and [46]. It represents the codec distortion which leads to voice distortion and impairments arising because of signal conversions. Nowadays, its value is determined by looking up the codec in the ITU-T Recommendation G.113 [45] shown in Table 4-2 while Table 4-3 is for the codecs used in our experiment.

Table 4-2 codecs Impairments factors

Audio codec	**Bandwidth (Kb/s)**	**Sample period**	**Impairment factor Ie**
G.711	64	20	0
G.723.1	5.3	30	19
G.723.1	6.4	30	15
G.726	16	20	25
G.726	32	20	12
G.726	40	20	7
G.729	8	10	10
GSM-FR	13.2	10	26
G.729A	8	20	11

Table 4-3 codecs used in derivation

Encoder Type	**References**	**Bit Rate (Kbit/s)**	**Ie value**
PCM	G.711	64	0
ACELP	G.723.1	5.3	19
ADPCM	G.726	24	25
CS-ACELP	G.729A	8	11

1.3) $I_{p.l}$

$I_{p.l}$ is the packet loss percentage within a particular period measured by a certain number of packets. The percentage measured is the loss of packets that occurred when the sender's packets are not received by the receiver. It can be expressed by the following formula (4.6).

$$I_{p.l} = \left(1 - \frac{N}{DS}\right) \times 100\% \qquad (4.6)$$

Where DS is the difference between the largest and smallest sequence number of *N* packets. Statistics and calculation of the Real-time Transport Protocol (RTP) packets can be used to calculate this percentage by the following expression (4.7).

$$DS = LS - SS + 1 \qquad (4.7)$$

Where *LS* and *SS* are the largest and smallest sequence numbers respectively. They are extracted from the RTP header of the sequence number field from the packets received.

2) I_d :

The delay components contributing to I_d provided in ITU-T G.107 [25] are T_a, the average absolute one way mouth to ear delay. T, the average one way delay from the receive side to the point in the end to end path where a signal coupling occurs as a source of echo. T_r, the average trip delay in the 4 wire loop. G.107 [25] gives a fully analytical expression for the function I_d, in terms of T_a, T, T_r and parameters associated with a general reference connection describing various circuit switched and packet switch inter-working scenarios. Assuming perfect echo cancellation, all the factors in I_d can be collapsed in a single term as shown in (4.8) and I_d (d) is now function only of the one way delay d. I_d (d) can be calculated by a series of complex equations in ITU-T G.107 [25] as shown with the plotted curve of I_d vs one way delay in Figure 4-6 (labeled “G.107”).

$$d = T_a = T = T_r / 2 \qquad (4.8)$$

The one way delay (*d*) is the time it takes to get data across the network. The one-way delay measured from one end of the network to the other end is mainly composed of four components that can be expressed in equation (4.9).

$$d = t0 + t1 + t2 + t3 \qquad (4.9)$$

Where *t0* is the network propagation delay, *t1* is the transport delay, *t2* is the packetization delay and *t3* is the jitter buffer delay.

In our work, we approximate these four components by measuring the response time (round-trip delay) as in most modern devices t1 and, t2 shall be small. Thus, ping should be a reasonable choice for estimating d, as it will estimate t0+t3.

In our model we used a simplified version of (4.10) as provided in [47]. This model shows accuracy for one way delay less than 400ms as shown in Figure 4-6 (labeled "AT&T simplified model"). We found this model reasonable as ITU-T recommend that one-way delay should not be more than 150 ms for good speech quality [48].

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3)$$

$$\text{Where} \quad \begin{cases} H(x) = 0, \text{ if } x<0 \\ H(x) = 1, \text{ if } x>=0 \end{cases} \tag{4.10}$$

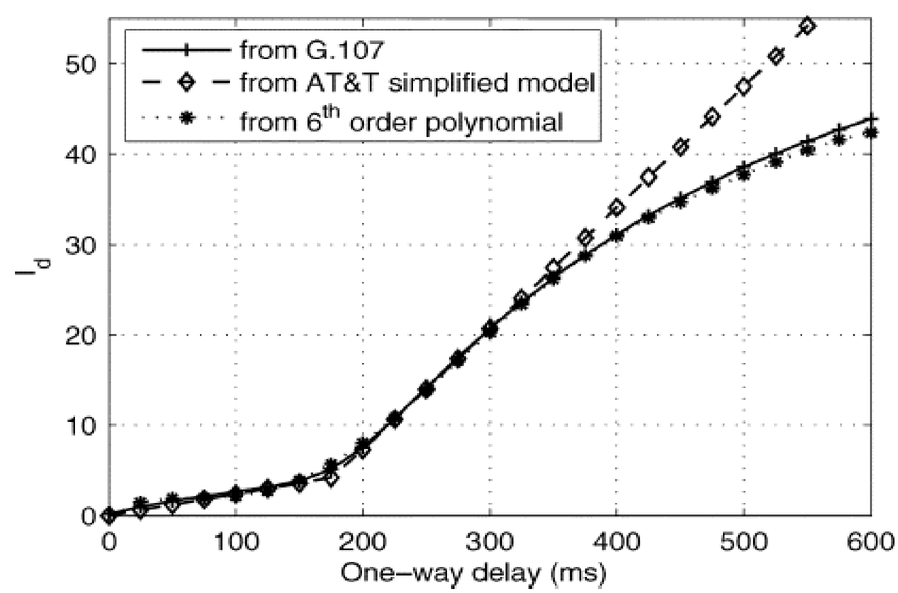


Figure 4-6 I_d versus one-way delay

3) A:

The advantage factor, A represents an "advantage of access", introduced into transmission planning for the E-model (ITU-T G.107) [25]. This value can be used directly as an input parameter to the E-model.

Provisional A values are listed in [25] as shown in Table 4-4. Assuming our communication system is conventional then we neglect A value.

Table 4-4 Examples of the Advantage factor A

Communication System	**Maximum value of A**
Conventional (wire bound)	0
Mobility by cellular networks in a building	5
Mobility in a geographical area or moving in a vehicle	10
Access to hard-to-reach locations, e.g., via multi-hop satellite connections	20

Eventually, the R value of the E-model is finally transformed to a MOS score that will reflect the user level of satisfaction as shown in Table 4-5, with a theoretical range of transmission performance rating factor R from 0 to 100. R=0 represents of the worst quality and R=100 represents the best quality. The R factor value for estimated average score of MOS can be expressed by equation (4.11).

For R<0: MOS=1

For $0 \le R \le 100$:

$$MOS = 1 + 0.035 \times R + R(R-60)(100-R) \times 7 \times 10^{-6}$$

For R>100: MOS=4.5 (4.11)

Table 4-5 Relationship between R-Value and User's Satisfaction

R-Value	**Satisfaction Level**	**MOS**
90-100	Very satisfied	4.3+
80-90	Satisfied	4.0-4.3
70-80	Some users dissatisfied	3.6-4.0
60-70	Many users dissatisfied	3.1-3.6
50-60	Nearly all users dissatisfied	2.6-3.1
0-50	Not recommended	1.0-2.6

4.4 *Monitoring System Design and Results*

The monitoring system could target a specific number of RTP packets to capture and perform an effective MOS value calculation. The system will use a coefficient database for the codec used in the call. This monitoring system is developed for monitoring VoIP quality at the network terminals, and the environment could be a personal or family network with voice quality monitoring.

The whole system works as follows: The system uses a network capture module to capture a certain number of packets passed to a specific IP and port. Non RTP packets will be filtered. When this process completed the packet capture, the system will analyze the data, with delay and packet loss rate as described previously. The MOS score is calculated to assess voice call quality in this period of the call. We took our results online with introducing random packet loss rate in the network using Dummynet [49].

We compared our monitoring system using MOS scores based on the codec's coefficients (see Table 4-3) derived for 4 main codecs with the simplified version of the E-model that is used in monitoring purposes [40, 42] and the PESQ scores. The graphs (Figured 4-7 – 4-10) show our results for the 4 codecs. It can be observed that the MOS scores of our improved simplified E-model based on the coefficient database (Table 4-1) are very close to the PESQ scores unlike the simplified E-model which gives an advantage for the corrected model in monitoring purposes for the VoIP call quality.

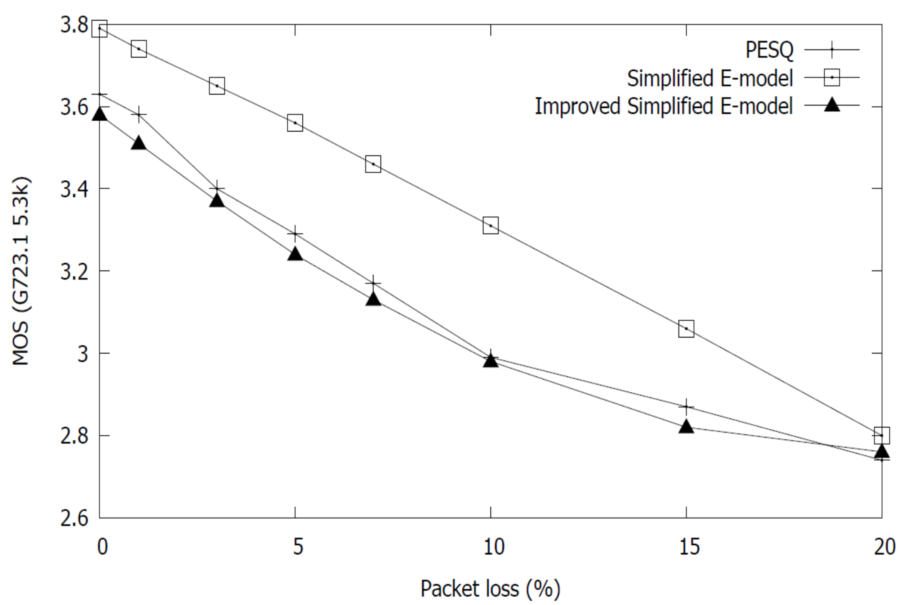


Figure 4-7 Comparative Analysis (G723.1)

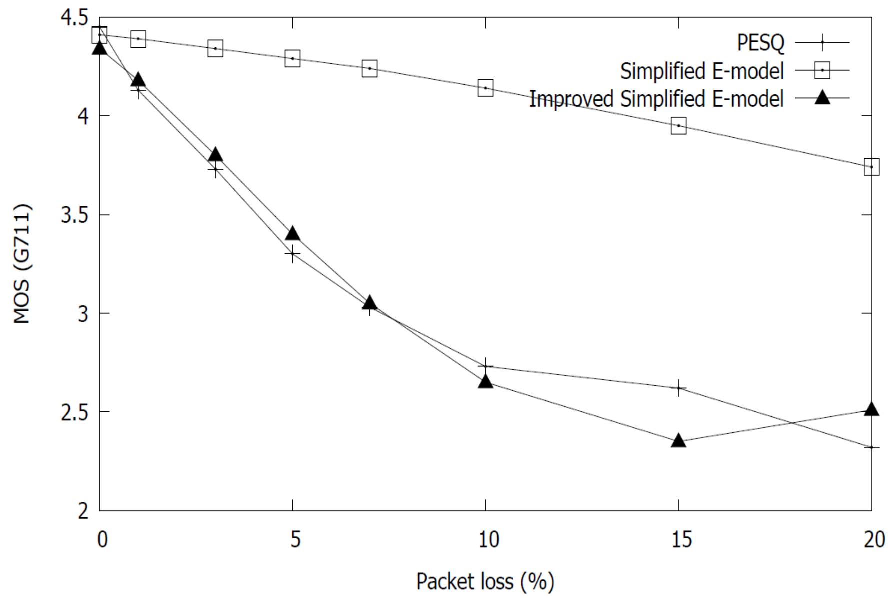


Figure 4-8 Comparative Analysis (G.711)

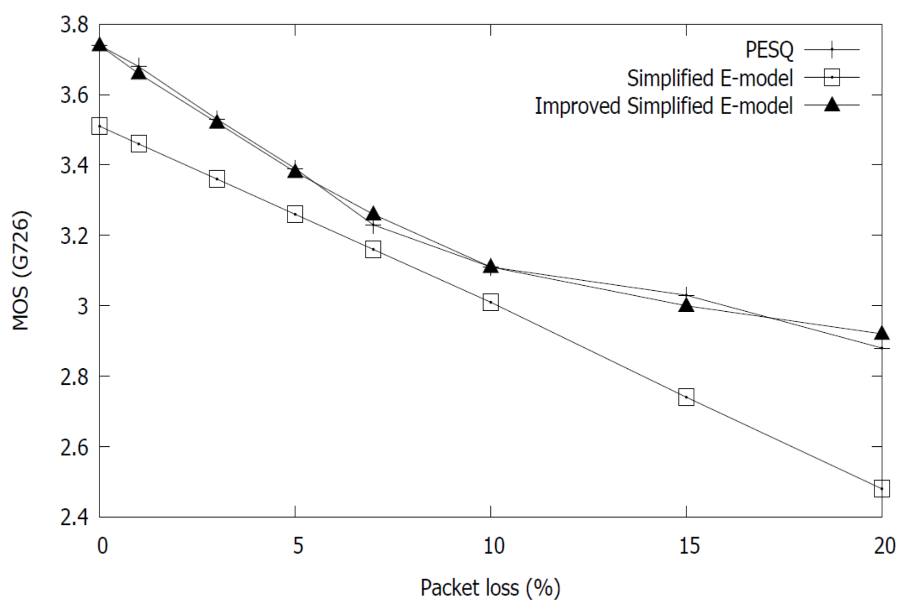


Figure 4-9 Comparative Analysis (G.726)

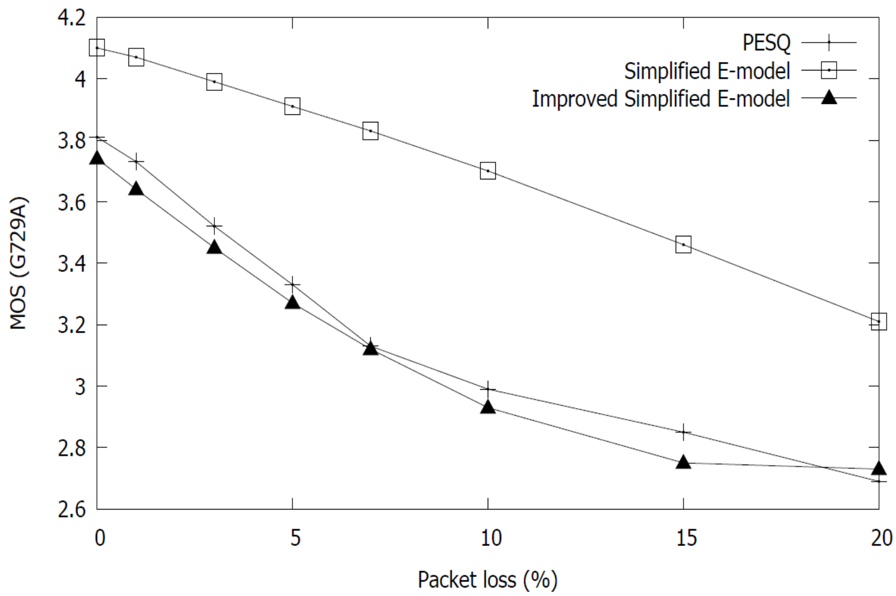


Figure 4-10 Comparative Analysis (G.729A)

4.5 Summary

The E-model brings a new approach to the computation of estimated voice quality. The main advantage of using the E-model that it is classified as an objective non intrusive method that can be applied in real time. Contrary to the ITU-Recommendation, simplified versions of the E-model have been introduced by researchers and industry to be used for monitoring purposes and predicting the VoIP call quality which is not an accurate method for that purpose.

Consequently, we have proposed an improved simplified E-model which better captures a call's PESQ score and we show the method of deriving the coefficients used in the model for four common used codecs (G.711, G.723.1, G.726 and G.729A). We demonstrate its results by implementing it in a monitoring system; our system analyzes the impact of voice quality encoding factors under various network conditions and uses our simplified improved E-model to assess voice quality. The main advantage of our improved simplified version that, it is less complex than the original E-model model and it is more accurate than the simplified versions used.

We stress three benefits of our work proposed in this chapter. The first as confirmed by the experiment, the simplified version of E-model does not provide accurate results compared to PESQ scores. The second, the correction coefficients derived enhance the simplified E-model to monitor/predict the call quality. The third, proposing a complete design of a monitoring system using our improved simplified E-model for 4 common codecs. Another output of our work is a java application that streams RTP packets using a number of codecs. We will explore a more complete testing framework in the next chapter.

Chapter 5

VVoIP Quality of Experience Measurement Framework

In this chapter, we introduce our automated testing framework that can provide an online estimate for both audio and video call quality on network paths without end-user involvement and without requiring any audio/video sequences.

5.1 Introduction

The demand has increased for interactive audio and video calls over the internet; it has become more and more popular in the last decade as top software companies like IBM, Google and Microsoft have invested in the enhancement of their voice and video over IP services. The moving towards the IP telephony has its problems as users are accustomed to the quality of service (QoS) they have enjoyed for years with the public switched telephone network (PSTN). On the other hand, Video and Voice over IP (VVoIP) is based on IP networks that may not provide perfect network conditions [1]. Consequently, monitoring and predicting the VVoIP call quality in different conditions is essential so as to prevent critical and potential problems from arising. Recall from Chapter 3, ITU-T provides two test methods: subjective testing and objective testing. Subjective testing was the earliest attempt to evaluate the quality by giving Mean Opinion Scores (MOS) and ITU-T Rec. P.800 [41] presents the MOS subjective test procedures for audio quality testing. It usually involves 12-24 participants who individually listen to an audio stream for several seconds and rate the audio quality on the scale of 1(Poor) to 5(Excellent). Similarly, BT.500 [77] presents a methodology to obtain MOS values for video quality. Subjective testing using MOS is time consuming, expensive and does not allow real time measurements. We have seen that several techniques were developed for measuring MOS in an objective way (without human perception), the most important techniques we focused on in our work were: PESQ [22, 23] and E-model that represents audio quality, while PSNR (Peak Signal to Noise Ratio) and General Model is used for obtaining video quality. PESQ automatically maps its score to the subjective MOS score. It takes into account two signals; one is the reference signal while the other one is the actual degraded signal. Both the signals are processed by the PESQ algorithm and the result is a MOS score. The major drawbacks of the PESQ approach was that it does not take into account impairments such as acoustic echo and transmission delay and can't be used in real time monitoring purposes. In contrast, the E-model technique, specified in ITU-T Rec. G.107 [25], is a non-intrusive method that uses network metrics locally monitored at the sender to estimate call quality, so it can be used for live call monitoring. One drawback with the E-model is that it requires knowledge of the so-called "impairment factor" of the codec, which ITU-T provides for codecs they specify, but which is not specified for a range of other commonly used codecs (refer to Chapter 3).

For video, PSNR works in a similar way to PESQ. This method assesses the performance of video transmission systems by calculating PSNR (Peak Signal to Noise Ratio) between the original and the received (degraded) video. PSNR is a differential metric which is computed using images. It is quite close to the widely-known SNR (Signal to Noise Ratio), but the difference is that PSNR gives a better indication to the Quality of Experience of users. Transmission of video is subject to a lot of losses, as the original video is encoded first before transmission to lower the bit rate. This results in the distortion of the original video as the commonly used codecs like MPEG-4 or H.264 are usually lossy. Further distortion occurs due to loss of packets which will cause errors when decoding the video at the receiver's side. Delay also can cause unwanted pauses in the received signal, as the receiver might need to pause processing, while the buffer refills. Both packet loss and delay will cause degradation in the interactive video call quality between the end users.

Since processing audio/video sequences is time consuming and computationally intensive, existing objective techniques are not ideal for online VVoIP QoE and since audio/video codecs have different characteristics and usually it is impossible to define in advance the most appropriate codec to use. Given this, we focus in this chapter on the use of a novel testing framework to measure/predict the voice/video call quality in advance and predict the most appropriate codec for that call in advance without any audio/video sequences and without end user involvement for quality rankings. In the future, we are looking for the influence of the live performance and system stability on the resulted MOS from the proposed testing framework.

5.2 Reviewing Related Work

Real time voice applications as VVoIP are QoS sensitive. Although IP networks provide best effort services, they do not perfect network conditions. Given this, monitoring and predicting the VVoIP call quality is important. The existing literature is extensive; we review some of the most relevant previous work. The main difference between our work and those reviewed below is that we present a testing framework that emulates traffic patterns for different audio/video codecs to measure/predict the call quality under current or emulated network conditions without any audio/video sequences and without end user involvement for quality rankings.

Jiang et al. in [42] introduce a voice quality monitoring system based on the SIP protocol that uses RTP statistics to get MOS score using the simplified E-model. In [78] Kim et al. propose a network performance monitoring method using the RTCP statistics to monitor multimedia services like VoIP and IPTV. J.M. da Silva et al. in [79] analyze QoS provided by SIP for voice as they measured the delay, jitter and packet losses. Carvalho et al. [80] propose three visible corrections needed for the E-model in order to give more accurate results indicating the QoE expected at the end user. They proposed a measurement tool based on these corrections. Gong et al. [81] have proposed a pentagram model to measure the QoE based on service integrality, service retainability, service availability, service instantaneousness and service usability. Due to the lack of QoE monitoring systems, Hershey et al. [82] had came with a new approach that aggregates observations from real time applications running on net-centric enterprise systems. They show their results on several VoIP scenarios including a Denial of Service (DoS) event that causes noticeable applications delay. GAP-Model has been introduced by Calyam et al. in [83] to get the QoE of voice and video over IP (VVoIP) applications, The GAP-model is based on an offline model of QoE expressed as function of bandwidth, delay, jitter, and packet loss.

5.3 Measuring Call Quality

In this section, we present the audio and video quality models used in our framework. Moreover, we use in our framework the estimated impairment factor used in the audio quality model (E-model) for a number of non ITU-T codecs derived later in Chapter 6 to support a wide range of codecs.

5.3.1 Extended E-model

Recall the mathematical model of the E-model that is used in our framework. The E-Model is considered a new objective model proposed by ITU-T G.107 [25]. It takes into account various factors that affect the speech quality and it calculates a rating factor ranging from 0 to 100. R=0 represents the worst quality and *R=100* represents the best quality. The R factor value is calculated as in (5.1).

$$R = R_0 - I_s - I_d - I_e + A \qquad (5.1)$$

R_0 is the Signal to Noise ratio(S/N) at 0 dBR point, I_s represents the speech voice impairments, I_d is the impairments due to the delay, I_e is the impairments due to the equipment (e.g.: codecs and packet loss) and A is the advantage factor (e.g.: A=0 for wireline).

From [84, 85, 86] the E-model can be utilized to be used in the speech quality evaluation over VoIP-Based Communication Systems and the R factor expression can be reduced as expressed in (5.2).

$$R = 93.2 - I_d(d) - I_{e,eff} \qquad (5.2)$$

I_d is a function of the one way delay only and can be calculated by the approximated formula expressed previously in Chapter 4 in 4.3.2. $I_{e,eff}$ is function of the codec used type and the packet loss rate and can be expressed by (5.3).

$$I_{e,eff} = I_e + (95 - I_e).\frac{P_{pl}}{P_{pl} + B_{pl}} \qquad (5.3)$$

I_e represents the impairment factor given by codec compression, B_{pl} represents the codec robustness against random losses and P_{pl} represents measured network packet loss in %.

The values of I_e and B_{pl} are given only for ITU codecs in ITU-T G.113 appendix [45] as neither the impairment factors of all the codecs factors are not provided nor can they be calculated easily. ITU-T recommendation G.113 [45] does not provide codec I_e and B_{pl} values for the most well known used codecs like ILBC, SILK, GSM and SPEEX. To establish these values we, for each of these codecs, estimate MOS using the PESQ method by directly comparing reference and degraded voice signals. Later in Chapter 6, we derived a non linear regression model for some of the most commonly used non ITU-codecs by the least squares method and curve fitting. The derived I_e model has the following form (5.4).

$$I_e = a\log(1 + b \times Ppl) + c \tag{5.4}$$

The P_{pl} in (5.5) is the packet loss rate in percentage and the parameters (*a, b* and *c*) will be derived later in chapter 6 (Table 6-2) for the different codecs. Finally, the R-Factor is converted to Mean Opinion Score (MOS) using the formula stated previously in Chapter 4 in 4.3.2.

5.3.2 Video Quality Model (VQM)

ITU-T G.1070 [25] developed video quality model for telephony services. The video quality V_q is dependent on the codec used and transmission parameters, *v1, v2 ... v12*. Video quality V_q is defined as in (5.5):

$$V_q = 1 + I_{coding}\exp\left(-\frac{P_{plv}}{D_{Pplv}}\right) \tag{5.5}$$

V_q represent the MOS value ranging from 1 to 5. Coding losses due to combinations of video bit rate (Br_v[kbit/s]) and video frame rate (Fr_v [fps]) are represented by I_{coding}. D_{Pplv} is the measure of robustness for the video quality against packet loss where the percentage of packet loss is defined by P_{plv} [%]. I_{coding} and D_{Pplv} are further defined in [25] by the following set of equations (5.6) and (5.10) respectively.

$$I_{coding} = I_{ofr}\exp\left(-\frac{(\ln(Fr_v) - \ln(O_{fr}))^2}{2D_{FrV}{}^2}\right) \tag{5.6}$$

O_{fr} is the optimal frame rate where video quality is the maximum. I_{ofr} is the maximum quality at each video bit rate (Br_v), They can expressed as shown in (5.7) and (5.8) respectively.

$$O_{fr} = v1 + v2Br_v \tag{5.7}$$

$$I_{ofr} = v3 - \frac{v3}{1 + \left(\frac{Br_v}{v4}\right)^{v5}} \tag{5.8}$$

D_{FrV} defines the robustness of video quality due to frame rate (Fr_v) and the degree of video quality robustness against packet loss is defined by D_{PplV}. They can be expressed as shown in (5.9) and (5.10) respectively.

$$D_{FrV} = v6 + v7Br_v \tag{5.9}$$

$$D_{PplV} = v10 + v11\exp\left(-\frac{Fr_v}{v8}\right) + v12\exp\left(-\frac{Br_v}{v9}\right) \tag{5.10}$$

Finally, coefficients *v1, v2…v12* are defined according to codec type, key frame interval, video display size and video format.

5.4 Framework Components

In this section, we introduce the components used in our framework; we give a brief description about each of the components and then we show the purpose of using each component in our framework.

5.4.1 Iperf

Iperf [75] was developed by National Laboratory for Applied Network Research (NLANR), Distributed Applications Support Team (DAST) as a modern alternative for measuring maximum TCP and UDP bandwidth performance. Iperf is considered a commonly used network testing tool that can create TCP and UDP traffic. Iperf is a tool for network performance measurement written in C++.

Iperf allows the user to set various parameters that can be used for testing a network, or alternately for optimizing or tuning a network. Iperf has a client and server functionality, and can measure the throughput between the two ends, either unidirectional or bi-directionally. It is open source software and runs on various platforms including Linux, Unix and Windows.

- ***UDP***: When used for testing UDP capacity, Iperf allows the user to specify the datagram size and provides results for the datagram throughput, jitter and packet loss.
- ***TCP***: When used for testing TCP capacity, Iperf measures the throughput of the payload. One thing to note is that Iperf uses 1024*1024 for megabytes and 1000*1000 for megabits.

One of the main advantages of Iperf is that it is considered a cross-platform tool (Figure 5-1) that can be run over any network and output standardized performance measurements. So, it can be installed on any UNIX/Linux or windows operating system. One host must be set as client, the other one as server. Consequently, it can be used to compare the wired and wireless networking equipment and technologies in an unbiased way depending on implementation constraints. Moreover, it is also an open source tool.

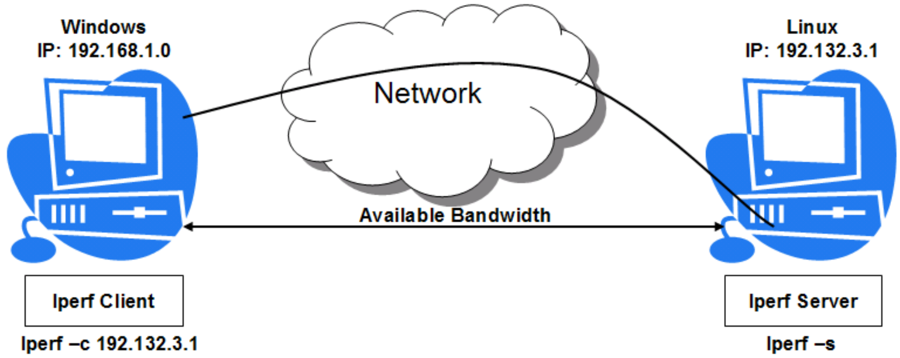


Figure 5-1 Iperf (Cross platform utility)

In a real world scenario, we may need to fill or nearly fill a call path with traffic in order to test its behavior under heavy loads. By tweaking the amount of load placed on the call path, we can figure out failure thresholds for it.

In our framework, we used Iperf to simulate a fixed load of the codec used and continually increase, recording the call quality and completion rate each time we do. Iperf has been recommended before for its accuracy and realism for such purpose [107]. Once, we hit the performance ceiling, we should be able to state that a particular link will carry no more than X G.711 calls or that a certain call path through the network will carry no more than Y G.729A calls or not even more than Z H.264 video calls. In addition, we used Iperf to simulate the traffic pattern of different audio/video codecs to measure the packet loss and jitter.

5.4.2 Dummynet

Dummynet [49] is a live network emulation tool, originally designed for testing networking protocols, and since then used for a variety of applications including bandwidth management. It simulates/enforces queue and bandwidth limitations, delays, packet losses, and multipath effects. It also implements various scheduling algorithms. Dummynet can be used on the machine running the user's application, or on external boxes acting as routers or bridges. Dummynet runs within your operating system (FreeBSD, OSX, Linux, Windows) and works by intercepting selected traffic on its way through the network stack and passing packets to objects called pipes which implement a set of queues, a scheduler, and a link, all with configurable features (bandwidth, delay, loss rate, queue size, scheduling policy...) as in Figure 5-2. Traffic selection is done using the Ipfw firewall, which is the main user interface for Dummynet. Ipfw lets you select precisely the traffic and direction you want to work on, making configuration and use incredibly simple. You can create multiple pipes, send traffic to different pipes, and even build cascades of pipes.

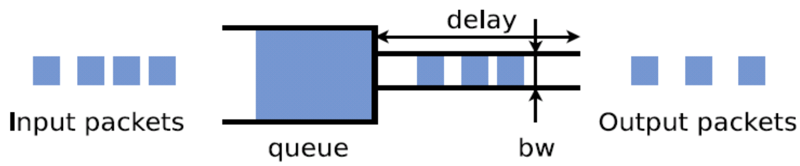


Figure 5-2 Dummynet Pipe [49]

Below, we will see some examples for using the Ipfw rule. The following examples can be illustrated as follows:

Example 1

The first command line creates a pipe which limits traffic into 500Kbit/s, delays packets for 100 ms and drops 10% of the packets. The second command line adds the pipe to the kernel and specifies filter conditions.

```
ipfw pipe 1 config bw 500Kbit/s delay 100ms plr 0.1
ipfw add pipe 1 ip from any to any
```

Example 2

Dummynet can also be used to model multiple paths using a classifier option that matches packets with a given probability; this allows traffic to be randomly directed to one of multiple links. As an example, the rules:

```
ipfw add 1000 prob 0.2 pipe 10 src-port 80 in
ipfw add 1010 prob 0.7 pipe 20 src-port 80 in
ipfw add 1020 pipe 30 src-port 80 in
```

send 20% of incoming HTTP traffic to pipe 10, another 56% (0.7 of the remaining 80%) to pipe 20, and the remaining part to pipe 30. If pipes have different bandwidth or delays, or they are subject to other interfering traffic, one can cause a wide range of effects, from selective packet loss to jitter and reordering.

In our framework, we use Dummynet for two purposes. First, we use Dummynet to change the network conditions (delay, packet loss, queue and bandwidth) to be able to test the QoS and QoE under different network conditions. Second, we use it to set the bandwidth with the Ethernet bandwidth according to the codec emulated in order to measure accurate delay results with the current browsing sessions if any on the computer.

5.4.3 Ping (network utility)

Ping is a computer network utility used to measure the Round-trip time for messages sent from certain source to a destination computer. Ping works by sending Internet Control Message Protocol (ICMP) echo request to the destination and waiting for the ICMP response. During such process, the time is measured from the source to destination and back to the source again (Round trip time) and also any packet loss is measured as well. At the end, the results are shown at the sender side, including minimum, maximum, and the mean round-trip times. The ICMP packet is shown in table 5-3.

	Bit (0-7)	Bit(8-15)	Bit (16-23)	Bit (24-31)
IP Header (20 Bytes)	Version/IHL	Type of service	Length	
	Identification		Flags and offset	
	Time To Live (TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
ICMP Payload (8+ bytes)	Type of Message	Code	Checksum	
	Quench			
	Data (optional)			

Figure 5-3 ICMP packet [87]

We use the ping utility in our framework to estimate the one-way delay required from the caller to the callee, we specify the packet size option in the ping command according to the codec used in order to emulate the codec's traffic patterns.

5.5 Development of the Framework

Our framework measures the QoS of the network based on the codec used and maps it to a QoE MOS score indicating the end user satisfaction level expected during the call accordingly.

Recall that the Packet size (*Ps*) and the Ethernet bandwidth (*Eb*) vary from codec to another. In our framework we calculated them as shown in (5.11) and (5.12) respectively.

$$Ps = Fs \times framesPerPacket + ipHeader + eOverHead \qquad (5.11)$$

$$Eb = Ps \times \left(\frac{bw}{Fs}\right)_{Codec} \qquad (5.12)$$

Ps is the total packet size, Fs is the frame size according to the codec, *framesPerPacket* is the number of frames per packet, *ipHeader* equals 40 bytes composed of the IP, UDP and RTP headers, *eOverHead* equals 38 bytes composed of the preamble, Ethernet header, CRC and Ethernet Inter-Frame Gap, *bw* is the bandwidth required by the codec.

For video transmission, H.264 is not transmitted using fixed packet length, but the packet length changes dynamically according to the available bandwidth in order to attain an acceptable video quality and to minimize the effect of distortion. Based on IBM statistics collected from the subjective quality testing, for transmission of low quality video 300Kbit/s of available bandwidth is needed, whilst for high quality 500Kbit/s would be required. HD video requires a minimum of 1.5Mbit/s bandwidth to be available at both ends of the call. We investigated the variation of packets length under the previous bandwidths in an interval of 60 seconds then took the mean packet length in order to reach an approximation for the packet length at different bandwidths for emulating the video traffic; the results are in Table 5-1.

Table 5-1 Mean Packet Length Estimates for H.264

Bandwidth	**Mean Packet Length (Bytes)**
300 Kbps (Low Quality)	316
500 Kbps (High Quality)	637
1.5 Mbps (HD)	885

Before measuring the QoS of the network and the QoE expected at the end user, the network conditions can be emulated for testing the robustness of different codecs under different network conditions. The IP destination address, port number, codec used and frames/packet are the main inputs before running the testing framework. Dummynet will then emulate the network conditions if any. The delay is measured using the ping command taking into account the packet size and the sending bit rate of the codec used as calculated in (5.11) and (5.12) respectively. Iperf is called to measure the packet loss percentage, throughput and jitter by specifying Datagram size (5.11) and Ethernet Bandwidth (5.12) for audio, or by using Table 5-1 for video to create appropriate data stream according to the codec that will be used during the call. By measuring the throughput which is considered the performance ceiling, we are able to calculate the number of calls that a certain link can carry safely. We can state that a particular link will carry no more than X G.711 calls or Y G.729A calls or Z H.264 calls, this can be expressed by (5.13).

$$nOfCalls = \frac{throughput}{Eb} \quad (5.13)$$

nOfCalls is the number of calls that can be carried through a particular link safely, *throughput* is the average rate of successful message delivery over a communication channel and *Eb* is the Ethernet bandwidth required according to the codec used. In order to increase the accuracy, average QoS network factors are measured by repeating the previous procedures 5 times and taking the average, this is done automatically by the framework. At the end the QoS parameters measured are mapped to QoE MOS score using E-model and VQM described in section 5.3. The pseudo code for our framework is shown in Figure 5-4.

Input: Destination IP, Destination Port No, codec used,
Video format, video frame rate and video bit rate (For Video testing only).

Output: QoS factors of the current network conditions.
QoE MOS ranking and user satisfaction level.

Begin Procedure
1. Step-1: Emulate Network
2. Initialize Dummynet emulator by loading the kernel module.
3. Emulate one or more from one of the following network conditions:
Line Bandwidth, Delay, Random Packet loss, Burst Ratio, Queue length.
4. Step-2: Initialize Test
5. Check the codec selected for the call.
6. Specify the packet size, inter-packet time and sending bit rate.
7. Step-3: Begin Test
8. Counter = 0.
9. **Loop** for Counter less than 5
10. Start Packet trains emulation from the source to the destination.
11. Measure the one-way delay using the ICMP request.
12. Measure the packet loss, throughput and jitter using modified Iperf.
14. Increment Counter.
13. **End Loop**
14. Calculate the average results for one way delay, packet loss, throughput and jitter.
15. Calculate link capability (No of Calls).
16. Step-4: Display Measured QoS factors.
17. Display the previous extracted data to the assigned text boxes.
18. Calculate QoE MOS score using E-model for audio and VQM for video.
19. Display MOS score and user satisfaction level in the assigned text boxes.
20. Step-5: End Test.
21. Flush all inbound/outbound pipes of Dummynet.
End Procedure

Figure 5-4 Framework Algorithm applied at the sender side

5.6 Results and Discussion

In this section, we provide the results of our framework for voice and video testing compared to the most commonly applied industry standard for objective voice and video quality testing: PESQ and PSNR. In order to measure the accuracy of our framework's results, we used a beta version of IBM SUT [88] product; we have measured the audio/video call quality under different packet loss rate using Dummynet. We have compared these real time offline audio and video testing using PESQ and PSNR respectively with the corresponding results using our framework. Each MOS score resulted from our offline testing under certain packet loss rate was repeated 5 times.

5.6.1 Audio testing

Tests are carried out on several codecs: G.711, G.723.1 5.3k, G.723.1 6.4k, G.726, G.728 and G.729A. We show our results in Figures 5-8 – 5-10. The x axis represents the packet loss rate range from 0-20% and the y axis indicates the MOS from the framework and PESQ algorithm. Our results match well the PESQ scores indicating the accuracy of our approach. We observe in our results that our framework's a slight underestimate compared to scores produced from PESQ. This can be explained as we take into our account the delay impairment factor (conversational call quality) while the intrusive testing methods PESQ do not take it into consideration causing a slight difference in some cases.

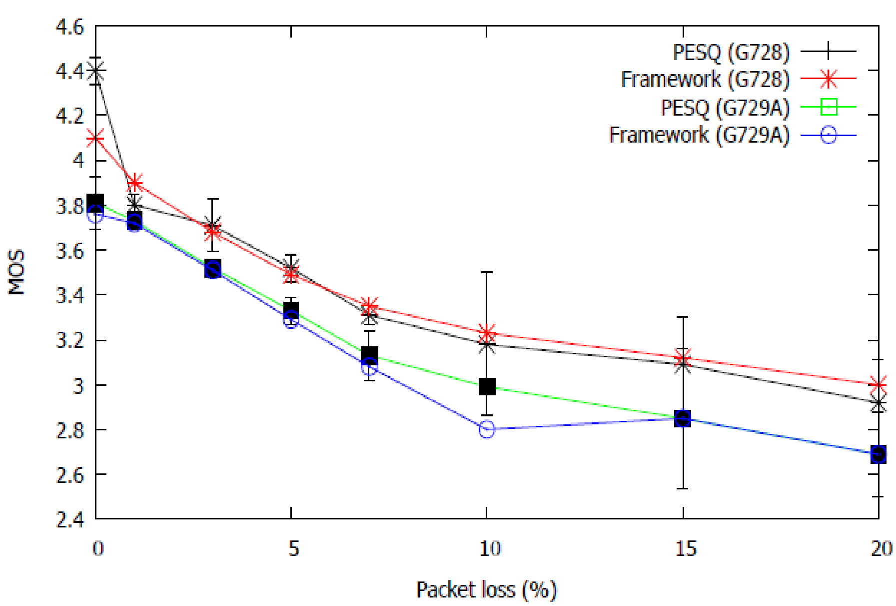


Figure 5-5 Audio testing of G.728 and G.729A codecs

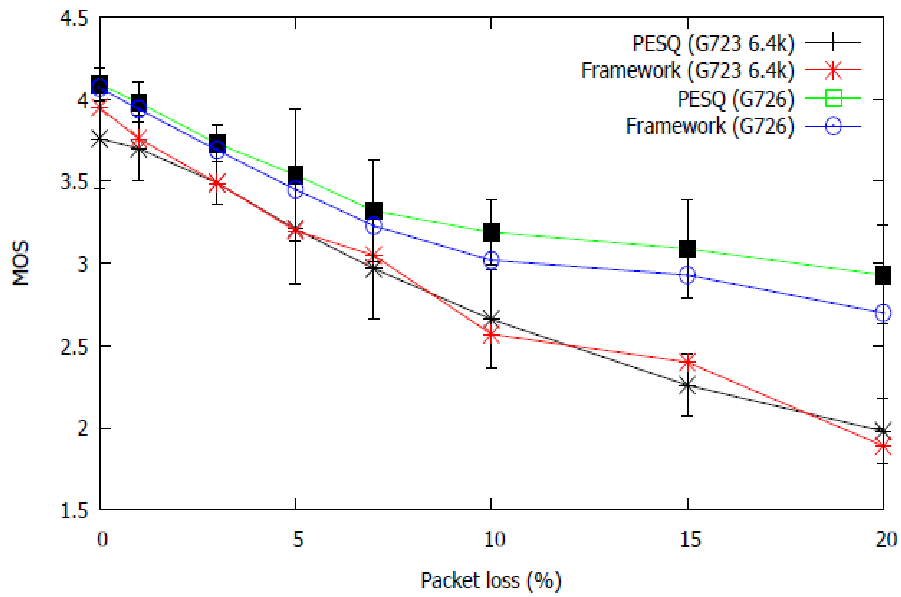


Figure 5-6 Audio testing of G.723 6.4k and G.726 codecs

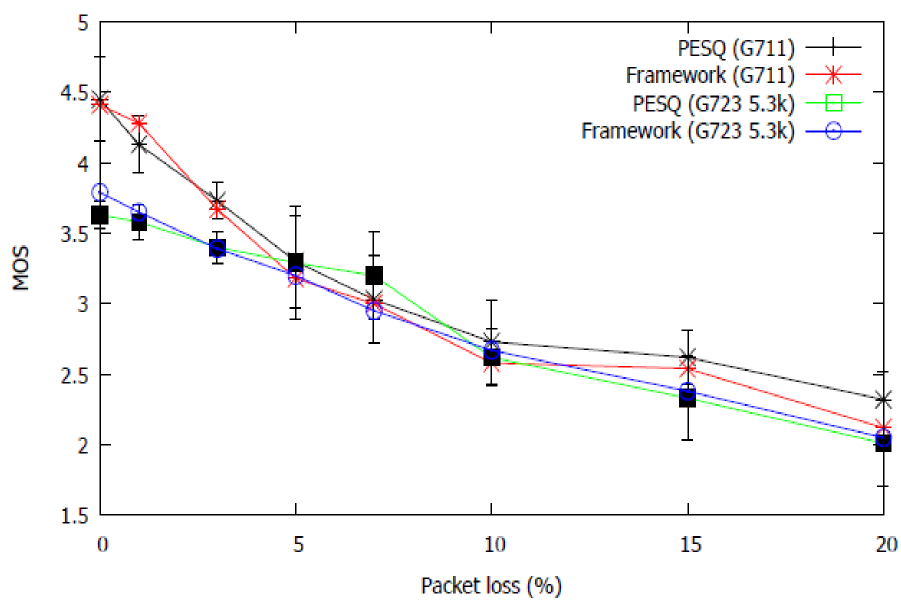


Figure 5-7 Audio testing of G.711 and G.723 5.3k codecs

5.6.2 Video testing

We compared our frameworks results to real time PSNR values of H.264 codec after converting them to MOS values. Table 5-2 which was developed by Ohm [89] is used to map the PSNR to MOS values that can be used to estimate perceived quality. Our results match well the PESQ scores indicating the accuracy of our approach. We interpolate between the values in Table 5-2 by assuming that the relation between MOS and PSNR inside these regions is linear.

$$\text{MOS} = \begin{cases} 5 & PSNR > 37 \\ 0.15 \times PSNR - 0.65 & 31 \leq PSNR \leq 37 \\ 0.153 \times PSNR - 0.813 & 25 \leq PSNR \leq 31 \\ 0.184 \times PSNR - 1.673 & 20 \leq PSNR \leq 25 \\ 1 & PSNR < 20 \end{cases} \tag{5.14}$$

We show our results on two resolutions QQVGA (160x120) and QVGA (320x240) with frame rates of 15 fps and 25 fps respectively. The comparison is presented in the Figures 5-11 – 5-12. The x axis represents the packet loss rate range from 0-6% and the y axis represent the MOS score of the framework and equivalent PSNR values.

Table 5-2 PSNR and MOS Mapping

PSNR [dB]	**MOS**
> 37	>5 (Excellent)
31 – 37	4 (Good)
25 – 31	3 (Fair)
20 – 25	2 (Poor)
< 20	<1 (Bad)

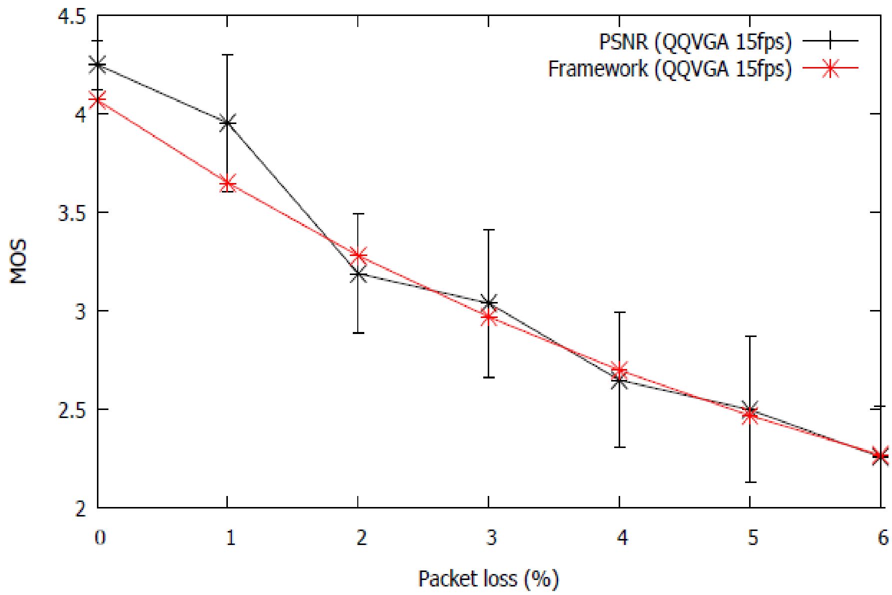


Figure 5-8 QQVGA at 15fps and bitrate of 300Kbit/s

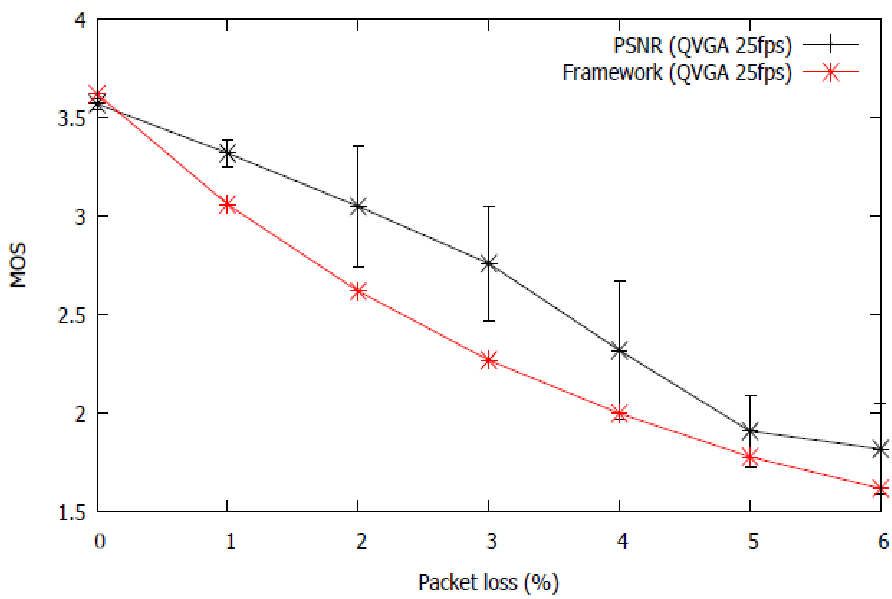


Figure 5-9 QVGA at 25fps and bitrate of 500Kbit/s

5.7 Summary

Processing audio/video sequences is time consuming and computationally intensive. Existing objective techniques cannot provide online VVoIP QoE measurement and since audio/video codecs have different characteristics and is considered impossible to define in advance the appropriate codec according to the current network conditions. Given this, we have proposed in this chapter a VVoIP QoE measurement framework to measure the audio and video conversational call quality without any audio/video sequences and without end user involvement for quality rankings. We have seen acceptable results compared to the quality of a real time voice and video calls using the most commonly used industry standard for objective voice and video quality testing: PESQ and PSNR respectively. In the future, we are intending to support wideband audio codecs and more video codecs. This tool has proven useful to IBM's testing teams in predicting the performance of particular VVoIP configurations under certain network conditions. In the future, we are looking forward to improve our framework by providing online graphical representation for the metrics measured and including more audio and video codecs.

Chapter 6

Improving the VoIP call quality using codec Switching

In this chapter, we introduce a generic switching codec algorithm that is used for improving the VoIP call quality. We empirically studied the impact of codec switching on call quality and derived an algorithm that takes these impacts into account. Our experiments showed that our codec switching algorithm can be applied to a range of different codec packages and it produces a significant improvement in the voice quality.

6.1 Introduction

VoIP applications use UDP as the transport layer protocol. Thus, VoIP applications use datagram sockets to establish client to client communication. Moreover, the Real-time Transport Protocol (RTCP) is used as the application layer for delivering audio over IP networks. The Real-time control protocol (RTCP) will be used in this chapter in monitoring the transmission statistics and QoS parameters. RTCP reports are sent every 5 seconds by the receiver to the sender and consume a small amount of bandwidth.

The network loss in the IP network is considered one of the most important factors that cause degradation in the overall voice call quality. It is very difficult to tolerate with a packet loss greater than 5% as it will be harmful to voice quality [50]. There are some main factors that depend on the amount of packet loss that can be tolerated like the encoding algorithm and the sampling rate of the voice stream. The maximum quality that can be achieved differs from one codec to another under different packet loss rates. Thus, our main concern in this chapter is to introduce a generic switching codec algorithm to be applied on different type of codecs in order to achieve higher call quality compared when using single codec with some conditions to guarantee an increase in the overall call quality under highly variable lossy network. We assume that we need to send RE-INVITE messages, as while changing codecs on the fly is in principle possible without it, in practice RE-INVITE messages are sent by the systems we tested.

Our objective is to improve the call quality using an adaptive codec switching scheme to make use of the different performance of various types of codecs under different packet loss rates. Our proposed algorithm dynamically switches between the codecs of a SIP session communication system in order to improve the speech quality. We developed a generic switching codec algorithm that takes into account many of the drawbacks of switching codecs. We have developed our generic switching codec algorithm based on our study in this chapter of the performance of different codecs and the drawbacks of switching codecs.

Our contribution focuses in this chapter on 3 main points; first, deriving the non-ITU codecs' coefficients in order to be able to monitor the call quality using these codecs and take a decision of switching to another codec based on the monitored quality. We propose our experiment to derive such coefficients in the first part. Second, we studied the drawbacks of switching codecs to guarantee a higher call quality expected from such a switch especially under high or highly lossy network. Third and based on our findings, we propose our generic adaptive codec switching algorithm, we show that our proposed algorithm can be applied to a wide range of codecs and it produces a significant improvement in the performance listening quality compared to a static codec choice at the beginning of the call. Figure 6-1 shows the problem of keeping the use of the same codec even if there are better available codecs for the current network conditions. Figure 6-2 shows the solution of the prior problem, switching codec according to the current network conditions during the call solves this problem.

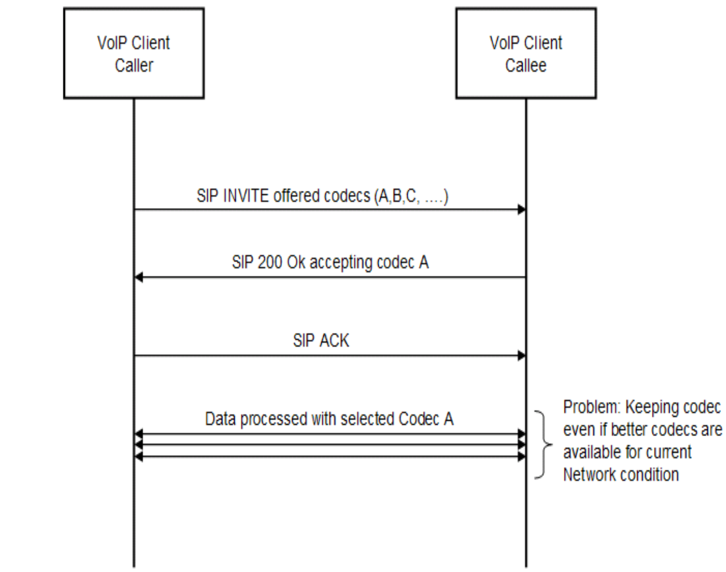


Figure 6-1 A static codec Selection scheme and its inherent problems

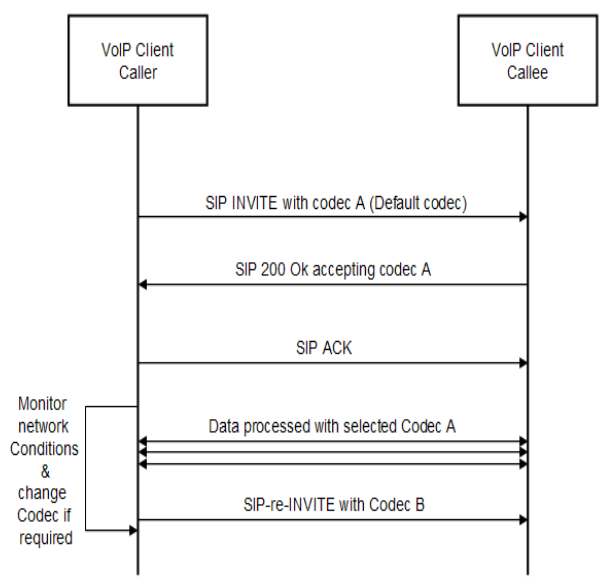


Figure 6-2 Proposed codec Selection Scheme

6.2 Reviewing Related Work

In the current VoIP applications codec switching is typically achieved via Session Initiation and Session Description Protocols (SIP/SDP). The initial session negotiation is achieved by a straightforward handshake protocol interaction wherein each peer exchanges an offer including the list of codecs it supports and a codec is selected. If one peer wishes to switch the code mid-session it initiates a similar handshake procedure to select a new codec; in this scenario it is important that both peers synchronize with each other in order to avoid data misinterpretation.

Ismet et al. [51] compare some of the speech quality of a set of standard codecs under different network conditions, then they propose an adaptive end-to-end based codec switching scheme that fully conforms to the SIP standard. The proposed adaptive mechanism was based on the available bandwidth; it changes to a low bandwidth codec when the packet loss increases, and it changes to a high bandwidth consuming high quality codec when the bandwidth increases. Their adaptive scheme was based mainly on 2 codecs: PCMU and SPEEX. Moreover, they concluded that there is no obvious enhancement in the call quality when switching between any of the following codecs: PCMU, PCMA, SPEEX and GSM under different packet loss rates. A. L. Robustelli et al. [52] describe a voice coder that makes use of an adaptive algorithm which performs an automatic coding switch, according to the packet loss. In their adaptive model, they used GSM and PCMU codecs. They have tested their adaptive technique under three different network conditions: short congestion intervals, congestion intervals equal in duration to intervals with traffic generator and with long congestions intervals. They show that their adaptive coding technique increases the voice quality of VoIP communication in the 3 previous test cases compared to the pure codecs if used. Nelson et al. [53] propose an adaptive codec switching technique by developing the NCVoIP application. The NCVoIP application starts to monitor and analyze the quality of the voice in order to assure the VoIP quality. The NCVoIP application changes to a lower or higher codec transmitted rate according to the predefined threshold values of each codec. They showed in their work that switching the voice codec when the bandwidth is below the transmission rate of the used codec and using TCP to encapsulate the RTP packets, when a congestion network exists, corresponds to a significant voice quality improvement. Interestingly, Marcel et al. [54] describe a technique for seamless VoIP codec switching in the Next Generation Networks (NGN) based on SIP/SDP session re-negotiation by establishing a parallel media stream and RTP packet filtering. They show that their proposed approach does not cause any annoyance or interruption of the audio stream in 90 % of the test cases. In [55] Maja et al. gave a proposal on algorithm for adaptive adjustment of VoIP sources transmission rate based on voice quality estimated at the receiver, they switched between 3 codecs in their algorithm: G.711, G.729A and G.723.1 5.3k. At the end they showed that their algorithm maintains a high level of MOS value in cases of network congestion.

The main difference between our work and those reviewed above is that we perform a detailed analysis of the impact of codec switching on voice quality for a wide range of codecs, deriving some heuristics for when and how often codec switching should be done. These heuristics are incorporated into our codec switching algorithm. In addition and unlike other reviewed approaches, we show that switching codecs based on the packet loss improves call quality but a special care should be taken to avoid the negative impact of switching.

6.3 *SIP session negotiation for codec switching*

codec switching is done through Session Initiation and Session Description Protocols (SIP/SDP) [6, 7]. The process of initiating and re-negotiating a media session is shown in Figure 6-3. SIP is responsible for media sessions establishment, update and tear down. SDP is responsible for codec negotiation. SDP itself is the way media sessions are described. The negotiation process is quite simple. At any time, an entity generates an offer, with all supported codecs. This offer is sent to another entity by an **INVITE** message, carrying an initial SDP offer. If this message is answered with **200 OK**, the initiator confirms by sending an acknowledgment (**ACK**). Accordingly, SDP answer is conveyed through **200 OK** messages. This handshake procedure is used in order to agree on a common codec and other session parameters when establishing a call.

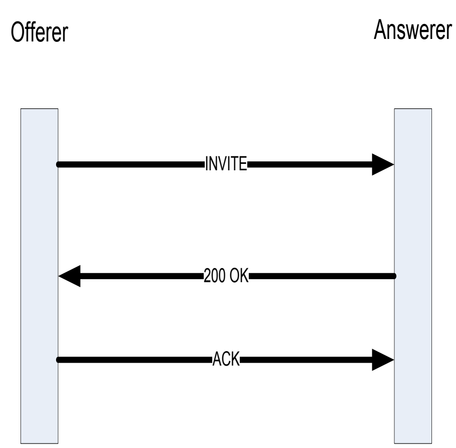


Figure 6-3 SIP Session Re-negotiations

When it is intended to switch codecs, the same offer-answer model is used. As a result, the entity who wants to modify the existing session will create a new offer that contains this media stream, and send that in an **INVITE** request to the other entity (here it is called **RE-INVITE**). It is important to note that the full description of the session, not just the change is sent. The receiver entity must be able to determine if that **INVITE** message is an initial **INVITE** or a subsequent **INVITE** (**RE-INVITE**) by looking at the To Tag parameter in the header of the message. If this parameter is defined, a dialog has already been created and thus, the **INVITE** request is within the dialog and no need to make a new dialog.

Once the negotiation of session parameters completes, both endpoints should be prepared to receive the media data format they agreed on. For codec switching within the call, it is very important that both endpoints synchronize with each other to avoid data misinterpretation.

6.4 Measuring the call quality

In this section, the E-model is introduced in its first part that is a non-intrusive testing method that can be applied at the receiver side to monitor the call quality during the call. We show also the behavior of the ITU codecs among different packet loss rates. One of the main challenges in this mathematical computational E-model provided by ITU-T G.107 is that it needs the codec impairment factors for each codec and ITU provides such factors for the ITU codecs only. Consequenlty, in the second part of this section we derived the codecs' coefficients for some of the commonly used non-ITU codecs in order to be able to monitor their call quality and include them in our codec switching algorithm.

6.4.1 The E-model

In this Chapter, we use another version of E-model different to the simplified version proposed in 4.3.2.
The E-model is an objective model proposed by ITU-T G.107 [25]. It takes into account various degradations that affect the speech quality and the end user level of satisfaction. Unlike the PESQ approach, the E-model can monitor the real time call quality by mapping the QoS factors to a QoE MOS score. It calculates finally a rating factor called R that range from 0 to 100. R=0 indicates the worst quality while R=100 indicates the best quality. The R factor value is expressed in (6.1).

$$R = R_0 - I_s - I_d - I_{e-eff} + A \qquad (6.1)$$

Where R_0 is the signal to noise ratio at 0 dBR point, I_e is the speech voice impairments, I_d indicates the impairments due to the delay, I_{e-eff} is the impairments caused by low bit rate codecs, and A is the advantage factor (e.g.: A=0 for wireline). ITU-T G.107 [25] defines the value of R_0 to 94.77 and I_s to 1.41. Consequently, expression (6.1) can be reduced to (6.2).

$$R = 93.2 - I_d - I_{e-eff} \qquad (6.2)$$

I_d is function in one way delay only and it can be calculated using a 6th order polynomial curve [56] as shown in (6.3) that is derived for a delay less than 600ms. I_{e-eff} is the packet loss dependent effective equipment impairment factor and can be expressed as in (6.4).

$$I_d = -2.468 \cdot 10^{-14} d^6 + 5.062 \cdot 10^{-11} d^5 - 3.903 \cdot 10^{-8} d^4$$
$$+1.344 \cdot 10^{-5} d^3 - 0.001802 d^2 + 0.103 d - 0.1698 \quad (6.3)$$

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}} \quad (6.4)$$

I_{e-eff} is derived using codec-specific value (I_e) which represents the impairment factor given by codec compression and packet loss robustness factor (B_{pl}) that represents the codec robustness against random losses. The values of I_e and B_{pl} for several codecs are provided by ITU in G.113 recommendation [45], which are deduced depending on subjective mean opinion scores tests and network experience. P_{pl} represents the percentage of packet loss and BurstR is the burst ratio when packet loss is bursty (BurstR>1) but it will be equal to 1 if the packet loss is random. Table 6-1 shows a summary of I_e and B_{pl} parameters for the ITU-T codecs.

Table 6-1 Some Coding Information

Audio codec	**Bandwidth (Kb/s)**	**Sample period**	**Impairment factor**	
			I_e	B_{pl}
G.711	64	20	0	10
G.723.1	5.3	30	19	24
G.723.1	6.4	30	15	20
G.726	16	20	25	38
G.726	32	20	12	24
G.726	40	20	7	24
G.729	8	10	10	18
G.729A	8	20	11	17

Finally, the R value is transformed to the MOS score that will indicate the end user level of satisfaction as shown in Table 6-2, the theoretical range of the transmission rating factor R is from 0 to 100. R=0 represents the worst quality while R=100 represents the best quality. The R factor value for estimated average score of MOS can be expressed as shown before in Chapter 4 in 4.3.2. The different MOS values that can obtained by the ITU codecs under different packet loss rates can be shown in Figure 6-4 where the x axis indicates random packet loss range from 0-20 % while the y axis is the MOS value indicating the end user perceived voice call quality.

Table 6-2 Relationship between R-value and user's satisfaction

R-Value	Satisfaction Level	MOS
90-100	Very satisfied	4.3+
80-90	Satisfied	4.0-4.3
70-80	Some users dissatisfied	3.6-4.0
60-70	Many users dissatisfied	3.1-3.6
50-60	Nearly all users dissatisfied	2.6-3.1
0-50	Not recommended	1.0-2.6

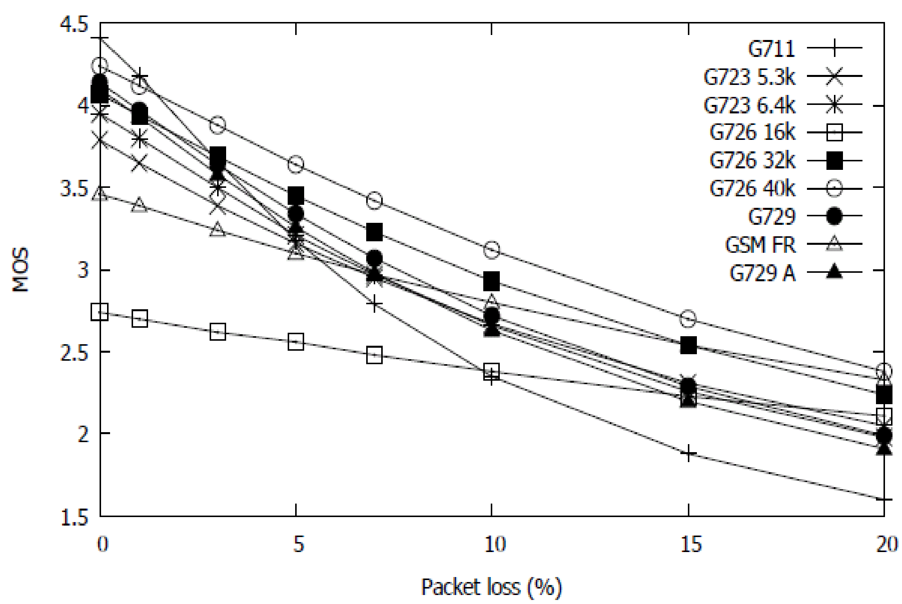


Figure 6-4 ITU codec's performance under different packet loss rate

6.4.2 Deriving non-ITU codec coefficients

Although the new objective E-model has been introduced by ITU-T in order to take in its account all the drawbacks of PESQ, it is still restricted to be used only with the codecs provided by ITU-T as neither the impairment factors of all the codecs factors are not provided nor they can be calculated easily. ITU-T G.113 [45] does not provide the codec impairments factors (I_e, B_{pl}) for non ITU codecs such as ILBC, SILK, GSM and SPEEX. Thus, in this section, we focus to model this codecs to be able to use them within the E-model. The MOS (PESQ) score is obtained directly by comparing the reference and the degraded voice as shown in Figure 6-5. The MOS (PESQ) score measured is converted to a rating factor R of the E-model. The conversion from the MOS to R factor can be calculated by a complicated Candono's formula as in [43] or by the simplified 3rd order polynomial fitting [44] used in this chapter as shown in (6.5).

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.060MOS - 57.336 \qquad (6.5)$$

The MOS (PESQ) factor converted to rating factor *R* does not consider delay impairments (I_d value). Hence, we will consider only the equipment impairment (I_e value which the impairment resulted from the codec and packet loss). Therefore, *R* can be converted to I_e as in (6.6) where R_0 =93.2 [25].

$$I_e = R_0 - R \qquad (6.6)$$

We have measured the PESQ of several codecs under different packet loss rate range from 0-20%, we measured the performance of the most commonly used non ITU codecs: G711, ILBC, SILK, GSM and SPEEX. Each MOS (PESQ) score of the previous stated codecs at each percentage of packet loss is measured 5 times and we took the average in order to increase the accuracy of our results. Thus, we measured more than 170 PESQ scores for all the codecs. We use Dummynet [49] to embed random packet loss rates online. Our results are shown in Figure 6-5 with the packet loss on the x axis and the PESQ MOS score on the y axis.

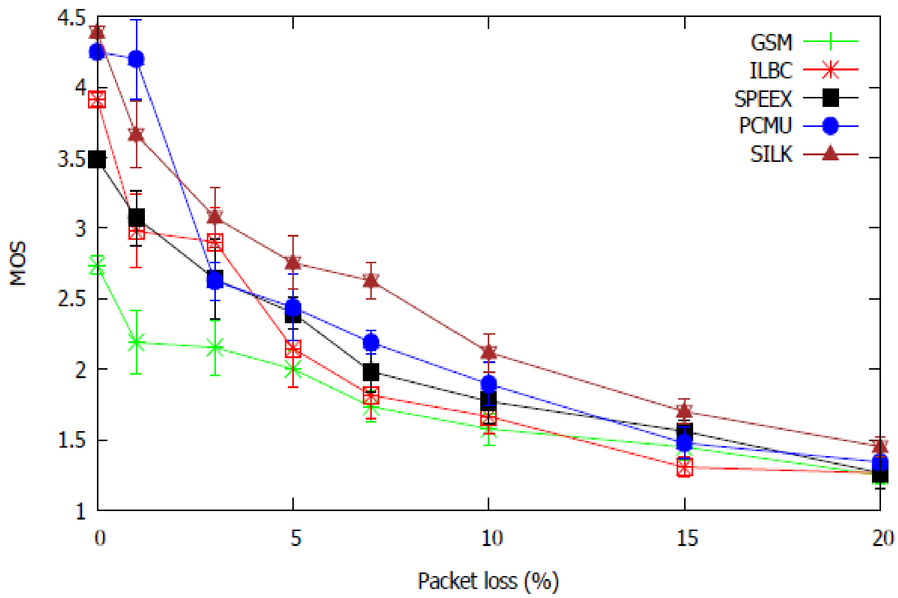


Figure 6-5 Non-ITU codecs performance under different packet loss rate

From the previous figure, we can notice that the performance of the codecs is different under packet loss rates. For example, It can be observed that the SILK codec outperforms at 0% packet loss rate compared to all the other 4 codecs. The PCMU turned to outperform starting nearly from 1% packet loss and until 3% percent packet loss. Afterwards and starting nearly from 4% packet loss, we found that SILK performs best until 20% packet loss. Hence we are proposing in this chapter a generic dynamic codec switching algorithm according to the present codecs in a VoIP application to make use of more than one codec in order to achiever higher call quality under unstable network conditions.

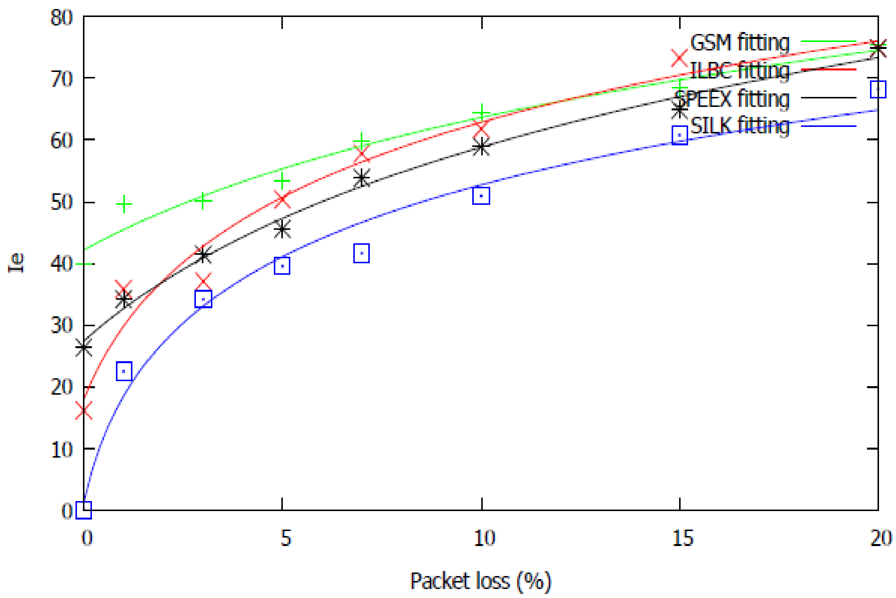


Figure 6-6 Deriving I_e factor for four of the non-ITU codecs

In Figure 6-6, a non linear regression model (similar to the logarithmic function in [56]) can be derived for each codec based on PESQ-LQO by the least squares method and curve fitting. The derived I_e model has the following form (6.7).

$$I_e = a \log(1 + b.\, Bpl) + c \qquad (6.7)$$

The *Bpl* in (6.7) is the packet loss rate in percentage and the parameters (*a, b* and *c*) are shown in Table 6-3 for the different codecs.

Table 6-3 Derived codecs' coefficients

Parameters	**GSM**	**ILBC**	**SPEEX**	**SILK**
a	22.931	20.836	28.244	18.3442
b	0.1555	0.762	0.2043	1.54894
c	42.175	18.013	27.423	1.31953

6.5 Impact of codec Switching

In this section, we focus on studying the negative impact of switching codecs; we study the factors that degrade the overall call quality if special care has not been taken when switching between codecs that may result in degrading the overall call quality under certain network conditions.

6.5.1 Switch-Over Gaps

The switching of codecs during the communication causes a switch-over gap. We define the term switch-over gap as the time taken between sending the RE-INVITE message from the sender side and receiving the ACK from the receiver side indicating the start of transmission with the new codec. In another words, switch-over gap indicates the response time to switch to another codec. Special care should be taken to avoid a large switch-over gap which leads to decrease the responsiveness time to switch to another codec. Our results show that at high packet loss rates, the RE-INVITE message will be at a higher probability of being lost, which will cause multiple retransmissions until the message reaches the intended receiver, and the same also will happen for the 200 OK and ACK messages, as a result, the switch-over gap will increase more.

For guiding the design of a quick responsive codec switching algorithm, we need to minimize the response time as much as possible to make use of the appropriate codec and attain higher call quality. Since the switch-over gap is codec independent, we have measured the switch-over gap between G.711 and ILBC with a packet loss rate range from 0-40%. At each packet loss rate, we have repeated the experiment 10 times for measuring an estimate for the switch-over gap measured in msec. Figure 6-7 shows our results indicating the packet loss percentage on the x axis and the switch-over gap on the y axis.

We identify three distinct regions. The first region which is between 0-10% packet losses corresponds to the minimum switch-over gap with an average of 0.5s—this is the most appropriate range to switch codec. In the second region, the packet loss range from 10-30% will result in an average of 2s—in this region special care should be taken when switching because this may affect the responsiveness of the switching algorithm. In the third region between 20-40% packet loss, it is not recommended to switch as the switch-over gap will dramatically increase, to the extent that might lead to the change of network conditions leading to a false switching decision. Given these observations, we focus our algorithm on switching codecs in the first region (0 -10% packet loss) in order to minimize the switch-over gap and increase the responsiveness of our algorithm.

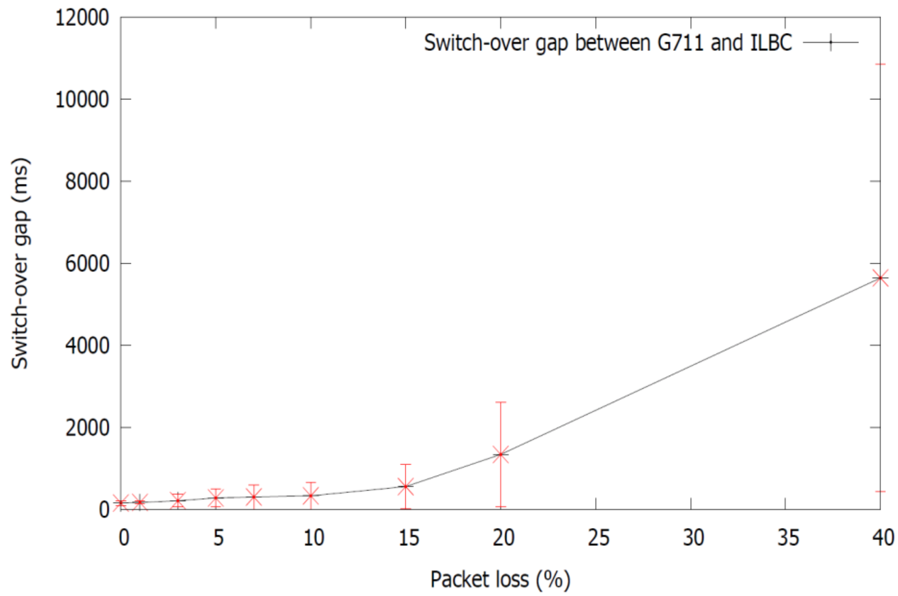


Figure 6-7 Switch-over Gap effect

6.5.2 Number of codec Switches and Silent Gap

Frequent switching of codecs during a session could cause degradation in the overall call quality; in this section we seek to quantify this effect. Restricting ourselves to 0-10% packet loss rate region for minimum switch-over gap, we once again apply the PESQ algorithm to calculate MOS. We use it to quantify the degradation in MOS due to a number of 0-12 codec switches during a 60s period. Codec switching is done at most every 5s, which is the RTCP reporting period.

In order to measure only degradation in the call quality as a result of increasing the number of switches, we selected pairs of codecs which have nearly the same performance. From Figure 6-5, we observe that at 0% percent packet loss, the performance of PCMU and SILK are nearly the same, at 1% percent packet loss the performance of ILBC and SPEEX are nearly the same and at 3%, as well as at 5% packet loss, the performance of PCMU and SPEEX are nearly identical. Additionally, at 7% and 10% iLBC and GSM provide close performance. Thus, we switched several times between these pairs of the stated codecs. The results are shown in Figure 6-8: we see that the relation between the number of switches and the MOS score is well matched by a first order function. Moreover, the slopes of all the lines are nearly the same which means that the rate of degradation is nearly equal under different random packet loss rates that range from 0 - 10%. As shown in Figure 6-8, at 1% packet loss, MOS measured between ILBC and SPEEX are not very accurate as there are slight difference between the MOS scores between them under 1% packet loss rate (see Figure 6-5). A summary table, showing switches during our experiment is shown in Table 6-4. We can therefore conclude that, in this packet loss range, the degradation is approximately 0.1 in the MOS score for the effect of a single switch.

The switching of the codec during the communication could cause a silent gap in the conversation, due to buffer re-initialization. We define the term silent gap as the length of the non-audible gap that results during codec switching. This can be illustrated as shown in Figure 6-8 from the degradation in the MOS when there is no switching compared to one switch.

Table 6-4 Number of Switches between different codecs

Number of Switches/60 sec	Packet loss (%)	codecs Used
0	0	PCMU - ILBC
1	1	ILBC - SPEEX
2	3	PCMU - SPEEX
3	5	PCMU - SPEEX
4	7	ILBC - GSM
5	10	ILBC - GSM

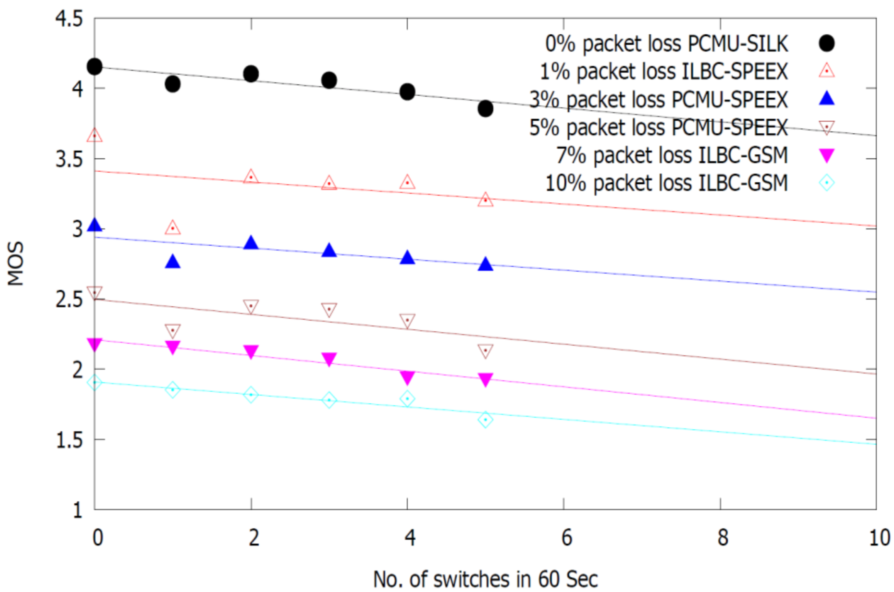


Figure 6-8 Effect of Number of switching on MOS score

6.6 *Proposed codec Switching algorithm*

In this section, we introduce our generic switching codec algorithm. Our objective from this algorithm is to apply it with the supported codecs in a VoIP application. We apply the E-model in real time taking into consideration the negative impact of codec switching to guarantee higher call quality as compared to when fixed codec is used during the call. Our algorithm supports most of the codecs currently used by VoIP applications as ITU-T provides its codecs' coefficients (see Table 6-3). Additionally, in this chapter, we derived the codecs' coefficients for the four popular non ITU codecs: SILK, GSM, SPEEX and ILBC. Thus we believe that we covered a lot of codecs that can be an input to our generic algorithm. Our algorithm runs in a parallel thread at the start of the call. The algorithm assumes the start of the call with its default codec set in the VoIP application. It waits for every 2 successive RTCP reports (10 seconds) to be received and calculates the current average packet loss rate. Based on the percentage of packet loss, the algorithm either proceeds or stops. So we focus only on packet loss less than 10% to guarantee minimum switch-over gap as discussed before. Assuming a packet loss less than 10%, the algorithm measures the predicted call quality using the E-model of all of the other codecs existing in the VoIP application taking into consideration the degradation resulting from the expected switch. All of the MOS measured scores will be saved in a list with the corresponding names of the codecs. After testing the expected call quality of all of the unused codecs, we sort the codecs in a descending order with the MOS score; comparing between the highest MOS score in the list and the MOS score corresponding to the current used codec, we take our decision either to switch to another codec or not.

We have developed our codec switching algorithm in an open source application called Jitsi [57]; Jitsi is an audio/video internet phone and instant messenger written in java. It supports the most important telephony protocols (e.g.: SIP, Jabber/XMPP). The summary of our codec switching algorithm can be summarized in Figure 6-9.

```
begin
    Start the call using the default codec in the VoIP application package;
    while Call is not ended do
        if (! 2 RTCP reports are received) then
            Wait until first 2 RTCP reports are received;
        end
        else
            Calculate average packet loss rate (avgPacketLossRate);
            if (avgPacketLossRate ≤ 10%) then
                Create List codecs <codec, R (codec i)> ;
                R_Current = 93.2 - Ieff; /* Calculate the rating
                factor R (current codec used) */
                MOS_Current= (MOS) R_Current;    /* Convert R
                score to MOS */
                Append the codec name used and its R values in the List;
                Calculate the number of codecs available (nOfCodecs);
                Calculate the number of switches in the previous 60
                seconds (nOfSwitches);
                for (i=0 ; i < nOfCodecs; i++) do
                    switchingEffect = nOfSwitches*0.1;
                    R (codec i) = 93.2 - Ieff;
                    MOS (codec i) = (MOS) R (codec i) -
                    switchingEffect;
                    Append in List<codecName [i], MOS (i)>;
                end
            end
            else
                Do nothing
            end
            SortByDesending (List codecs<codec, MOS (codec i)>);
            highestCodecScore = codecs [0, 1]; /* get the value of
            the codec at the top of the list */
            if (highestCodecScore > MOS_Current ) then
                codecName = codecs [0, 0];
                Switch to codecName;
                nOfSwitches++;
            end
            else
                Do nothing
            end
        end
    end
end
```

Figure 6-9 Proposed codec switching algorithm applied at the sender side

6.7 *Results and Discussion*

Simulation of a simple network containing two machines has been created in order to verify our proposed generic algorithm for codec switching. In real VoIP application, there is always a package that contains more than one codec which can be used in the VoIP application. such example of this packages are Polycom which is used in IBM SUT [105] VoIP product and GIPS which is used in both Skype and G-talk VoIP products [106]. Our goal is to create different packages with different codecs and execute our algorithm based on these codecs. The testing packages used are shown in Table 6-5.

Table 6-5 codec packages used in simulation tests

Package	**codecs present**	**Default codec**
First	• SILK • PCMU • GSM	GSM
Second	• SPEEX • ILBC • GSM	GSM
Third	• ILBC • GSM	ILBC

We applied our generic adaptive codec switching algorithm on the previous packages as VoIP applications support different codecs. The default codec is the codec set by default in the VoIP application to start the call using it. Figure 6-10 shows the testing setup; we tested our algorithm on 3 different packages of codecs as shown in Table 6-5. We show our results based on the different packages.

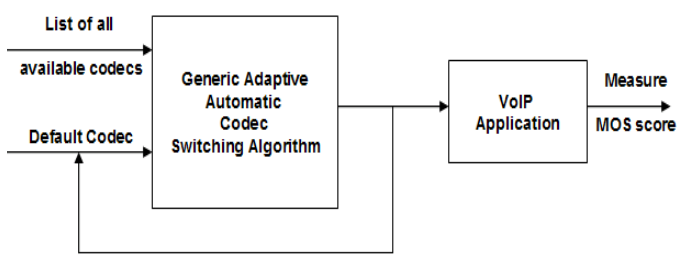


Figure 6-10 Testing Environment Setup for proposed algorithm

We played a sample audio file for 3 minutes; this period is divided later into 10 second chunks. We chose 10 seconds time slices so that it would be synchronized with the RTCP report received. The advantage of this approach is to monitor the MOS score over the whole duration of the experiment, instead of calculating an aggregated result. The results obtained have confirmed what was expected; the MOS score of the adaptive automatic codec switching algorithm was observed to be greater than the MOS relative to the other user using a fixed coding. The results of our algorithm when applied to the 3 packages shown in Table 6-5 are described in sections 6.7.1 and 6.7.2.

6.7.1 First Package

In this experiment and as shown in Figure 6-11, we started the call using GSM codec at 0 % packet loss; it took the algorithm 10 seconds to switch to SILK which has the highest R. Then for the next 60 seconds the MOS for all codecs is degraded by 0.1 as a result of switching. After the end of the previous 60 seconds, the MOS recovered from the negative effect of the switching and returned back to its value as shown in the time slice between 1:20 – 1:30. At 1:20 we applied 1% packet loss, so the switching occurred at 1:30 to PCMU. After 40 seconds, and although the packet loss was increased to 5%, switching didn't occur at the 2:20 as 1 switching was already done in the previous 60 seconds (-0.1 MOS) and the gain from such switch between PCMU and SILK (+0.1 MOS) won't improve the overall call quality. At 2:30 switching happened to SILK, as the counter of *nOfSwitches* (see Figure 6-9) was reset after the minute had already expired.

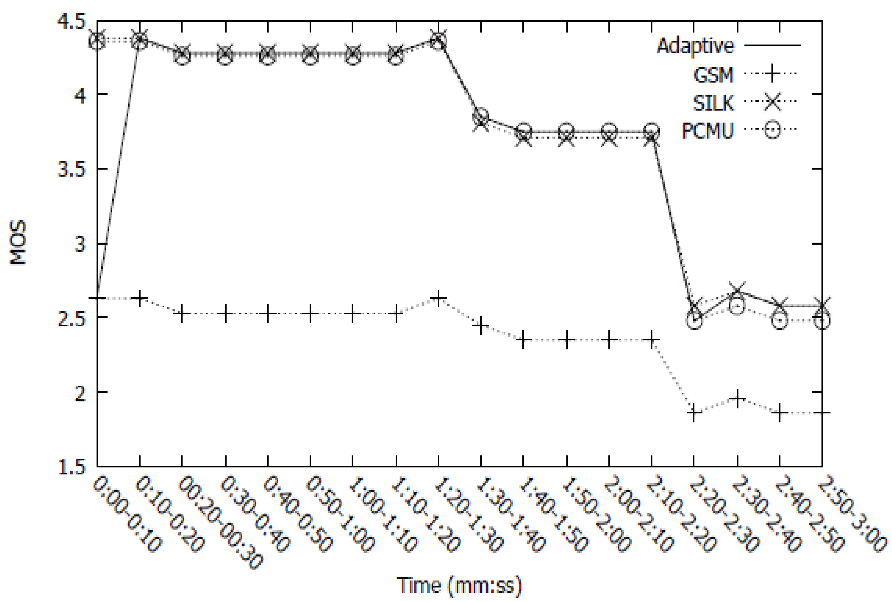


Figure 6-11 Results using GSM, SILK and PCMU codecs

6.7.2 Second Package

In this experiment as shown in Figure 6-12, we started the call using GSM codec at 0 % packet loss; it took the algorithm 10 seconds to switch to ILBC which has the highest R.

At 1:30, we applied a packet loss of 6 %. Thus, the codec was switched to SPEEX in the next slice. At 2:10, the packet loss was decreased to 0%, thus the codec was switched back to ILBC. Although 1 switch occurred before in the previous 60 seconds (-0.1MOS), it is worth switching as the total gain expected from such switch will be +0.33 MOS. In the proceeding slices from 2:20-2:50, the MOS was dropped by 0.2 due to the effect of 2 switches. Consequently, at 2:50 and after the end of 60 seconds from the first switch, MOS will return back to its normal value at current packet loss rate. The call will then continue with the normal value, unless the network has suffered any loss rates again.

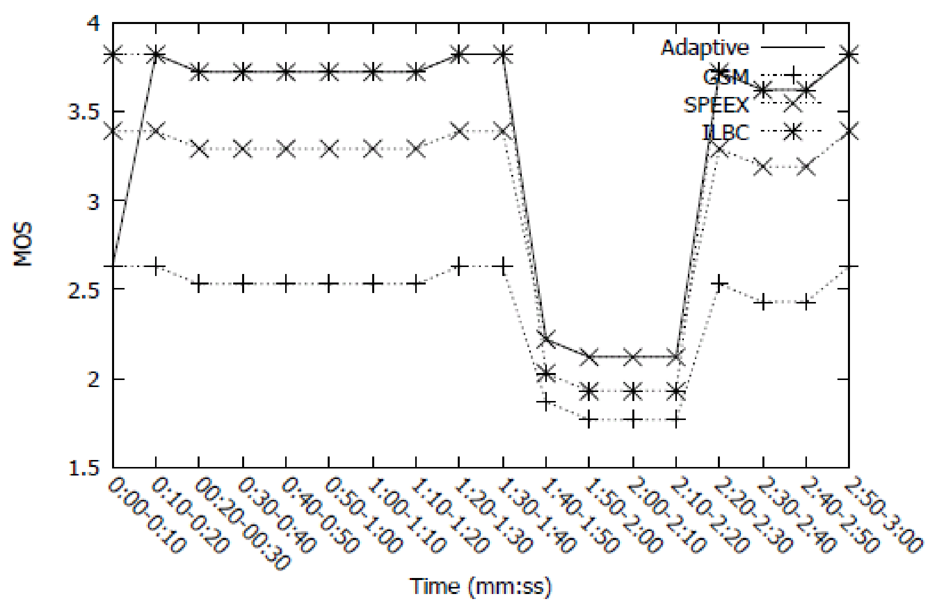


Figure 6-12 Results using GSM, SPEEX and ILBC codecs

6.7.3 Third Package

In this experiment, we started with ILBC at 0 % packet loss. Later we applied packet loss rates of 3% and 6% at 1:00 and 2:00 respectively. But as seen in Figure 6.13 no switching occurred at all as GSM has always a lower R value, so it would not be worth at any point to switch to another codec in a package having GSM and ILBC codecs.

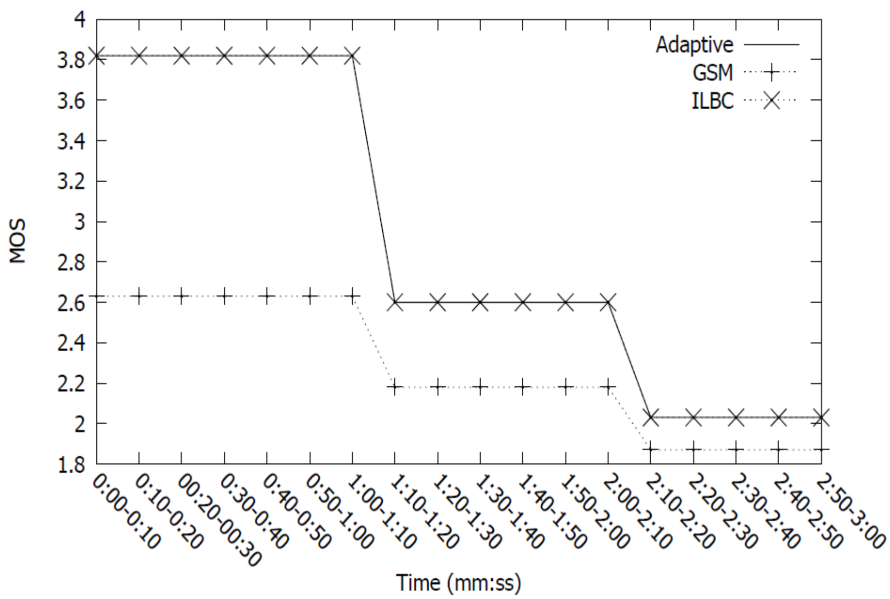


Figure 6-13 Results using GSM and ILBC codecs

6.8 Summary

Switching codecs during an ongoing voice session can improve user's perceived quality-of-experience (QoE) due to the fact that different codecs behave differently under different packet loss conditions in the network. In this chapter, we empirically studied the impact of codec switching on call quality and specified a codec switching algorithm that takes these impacts into account. We found that switching codecs will result in silent gaps and switch-over gaps of different lengths depending on the prevailing packet loss rate. We also found that the number of codec switches within a time interval should be limited so as not to contribute towards degradation in the call quality experienced by users. Our experiments showed that our codec switching algorithm can be applied to a range of different codec packages and that it can produce a significant improvement in voice call quality as compared to the use of a codec selected at the start of a call and maintained for the call duration. We also found that a combination of the PCMU and SILK codecs provides a solution that is more robust to moderate packet loss rates than other commonly used codecs.

For the future work, we are intending to extend our switching algorithm to support wideband audio codecs by applying the newly developed POLQA [17] objective testing method to derive the codecs' coefficients required in monitoring the call quality. Furthermore, we are planning to improve our algorithm by studying loss patterns to assess whether the frequency and distribution of losses affect the codecs' quality differently to include it if it has an impact on the call quality. Another potential in the future also is to include more RS schemes to the algorithm to support more packet loss recovery levels by switching between under different network conditions.

Chapter 7

Improving the VoIP call quality using new adaptive FEC technique

In this chapter, we introduce a new adaptive redundancy control algorithm (APU algorithm) for VoIP applications to improve the call quality under different packet loss rate. We used subjective testing to derive the algorithm. We show that our algorithm improves the conversational call quality compared to the use fixed codes.

7.1 Introduction

In the previous Chapter, a generic switching codec algorithm was proposed. It was shown through performed experiments that our proposed codec switching technique produces significant improvement in the voice quality. In this Chapter, another algorithm for improving the voice call quality is derived and proposed; such algorithm is based on certain packet loss recovery mechanism called FEC.

Forward Error Correction (FEC) is considered one of the common powerful techniques for transmitting audio streams over the Internet to decrease the effect of packet loss. Although these methods reduce the effect of packet loss, they increase the amount of bandwidth and delay in order to recover from the lost packets. In this chapter our objective is to propose a new adaptive FEC mechanism based on the generated codewords from a Reed- Solomon (RS) encoder. This mechanism chooses/switches between 3 different RS codes according to the current Quality of Service (QoS) based on the codec used, random packet loss rate, burst ratio and delay. We have derived the proposed algorithm by performing subjective mean opinion score (MOS) testing based on an interactive assessment tests. We finally show that our adaptive algorithm will result in an overall higher quality compared to the use of fixed RS codes under unstable network conditions.

7.2 Different Methods to recover lost packets

One of the main challenges in VoIP transmission is to guarantee the delivery of the packets at the Receiver side because RTP and UDP used in VoIP applications do not provide error recovery mechanisms at the transport layer. Nowadays, a lot of effort is exerted to provide mechanisms of lost payload at the codec level. However, these mechanisms only recover lost information in the event of small packet loss. Consequently, lost packet recovery algorithms are considered one of the most interesting research areas.

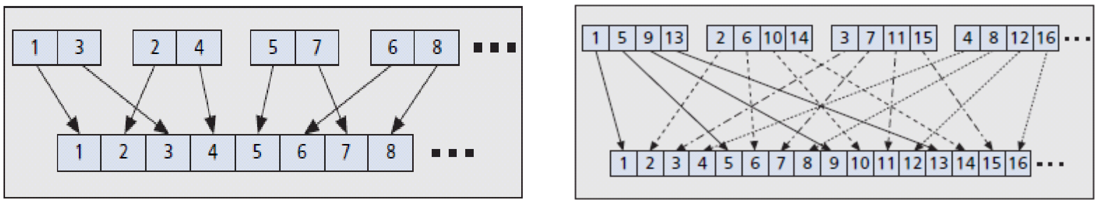
Sender-based loss-recovery techniques typically introduce added end-to-end delay into the media stream [58]. Generally, humans cannot even notice a one-way delay of less than 100 ms, and most users can tolerate a one-way delay of up to 250 ms. If the one-way delay exceeds 250 ms, however, the delay can result in a serious talker-overlap effect that is intolerable for most users [58]. Therefore, we must consider added delays as well as other factors such as bandwidth consumption when evaluating the feasibility and effectiveness of loss-recovery techniques. There are several delivery techniques that are described in sections 7.2.1 - 7.2.6.

7.2.1 Plain delivery

Plain delivery [59] is more prevalent than any other delivery technique in VoIP solutions. Plain delivery does not provide any sender-based effort to improve audio quality when packet loss occurs as it does nothing with the packets. In this technique, each block of the audio data is packaged and sent over the IP network. For instance, if we use G.729A codec, we package 20 bytes of the encoded audio data for a time interval of 20 ms into an IP packet for transmission.

7.2.2 Interleaving

In this approach [60], the main objective is to try to reduce the degradation of perceptual audio quality by scattering more lost frames into several small gaps instead of having one large gap of lost data. This idea came because recently, there was a lot of a research work claiming that listeners can mentally deal easily with a loss if it is distributed into several parts. This approach does not send any additional information in the IP packet; it requires the same bandwidth utilization of the plain delivery. The Interleaving approach is considered only feasible if we already transmit multiple frames of audio in each IP packet. For instance, if we packaged two frames of audio into one IP packet, we would transmit packets of interleaved audio frames, as Figure 7-1(a) illustrates. We could further scatter more lost frames by interleaving more frames into each IP packet before transmission, as Figure 7-1(b) illustrates. It illustrates interleaving four audio frames in each IP packet. The main drawback of interleaving is the impose of large delays into the media stream. Large delay can be irritating to users and can cause serious problems (i.e.: Talker overlap).



(a) Single scattered packet

(b) Four audio scattered packets

Figure 7-1 Interleaving Example

7.2.3 Forward Error Correction (FEC)

Forward error correction (FEC) is a sender-based technique for decreasing the undesired effects of packet loss [61]. FEC works by transmitting redundant packets for error correction. There are many different methods of the FEC technique. The Reed Solomon codes are one of the most used FEC techniques in VoIP applications. The Reed–Solomon encoding scheme works by generating parity bits and sending the parity bits along with the data values. If data values are lost at the receiver side, the Reed–Solomon decoder can reconstruct the original data by using the redundant information sent along with the previous packet. The amount of redundancy added in the block determines the amount of error or packet loss that can be recovered [59].

7.2.4 Redundant Data Transmission

Redundant data transmission (RDT) works by sending audio data more than once [62]. In this technique, the previous transmitted packet is sent along with the new packet. This means that each IP packet will have redundant audio data beside the new data. In this technique, when a packet is lost, we can still recover the lost packet from another IP packet that has the lost information. The amount of redundant data added to each IP packet can differ to provide different degrees of recovery and effectiveness. Some recent techniques introduced are working to send the original data with certain encoding scheme while using a different scheme for sending the redundant data. Jiang *et al*. [63] conducted a study to compare FEC and low bit-rate redundancy (LBR) and showed that FEC is more effective than LBR in terms of delivering voice quality in the face of packet loss. Generally, this approach may be considered a simple type of FEC.

7.2.5 Duplicate packet transmission

This loss recovery technique decreases the packet loss by sending redundant packets but this approach is different from the RDT as described in [59]. It transmits the redundant data in a split or separate IP packets. The main drawback of such a technique is the increase of the bandwidth consumption by requiring every time the addition of different protocols which is an overhead. Hence, using this technique will double the bandwidth of the plain delivery method. For instance, ILBC and G.729A will require 92.8 kbps and 78.4 kbps respectively. In other words, we would need to double the bandwidth of our data network or reduce the number of supported channels by half to use this technique in place of the plain-delivery technique.

7.2.6 Retransmission

The Retransmission technique works by re-sending the lost packets only upon request by the recipient. At the receiver side, there is an implementation to detect the lost packets. Once the receiver detects any missing packet, the receiver will send a request to the sender for re-transmitting the lost data. When the receiver receives the re-send request, it will retransmit the lost packet [59]. Some advanced techniques work on detecting even the lost re-transmission packet. H.P. Sze *et al*. [64] proposed an effective retransmission technique that combines the gap detection and timeout-detection mechanisms. The retransmission technique has variable additional bandwidth requirements; it consumes more bandwidth when there are more lost packets. A number of researchers have studied this and they recommended various retransmission techniques that are more bandwidth efficient than simple retransmission. Nonnenmacher *et al*. [65] proposed an approach that combines FEC and retransmission by using parity FEC packets to repair multiple losses with a single retransmission, thus achieving substantial bandwidth savings.

Among the previous techniques and the different packet loss recovery mechanisms described above, FEC is one of the most powerful techniques that was discussed over the past years and most of the current used VoIP applications focus on providing new FEC techniques to provide a high perceived call quality at the end user as in Skype [66, 68]. Consequently, our main focus in this chapter will be on FEC and proposing a new technique that will put us one step closer to getting optimum conversational call quality over a connection with unstable network conditions.

7.3 *Forward Error Correction (FEC)*

Forward error correction (FEC) is a mechanism that allows reliable transmissions by sending redundant data known as parity. FEC has proven to be highly efficient in the cases that the retransmission is impossible and this applies to the VoIP application. The main drawback in using FEC is that it increases the delay in the communication process because the receiver should wait until the receiving parity data has been successfully received . In addition, the FEC requires additional bandwidth when used. Therefore, the level of the FEC that needs to be applied to a stream of RTP packet has to be addressed carefully.

There are many different FEC codes used for different type of applications. Throughout this chapter, we are particularly interested in systematic forward error correction codes where the original payload of RTP packets appears in the encoded output. The maximum fraction of RTP payload packets that can be recovered with a FEC scheme is determined in advance by the design of the codeword [67]. Hence, in this section we will give a brief summary on the FEC piggybacking scheme and then we present our study on the FEC using pure Reed-Solomon (RS) codes that is used in VoIP applications. In addition, we show our analysis when using different pure RS codes.

7.3.1 *FEC with a piggybacking scheme*

A common way to decrease the packet overhead in FEC is to attach the redundant packets onto the information packet. This technique is called *piggybacking* (for example, nominal stream PCM at 64 Kbps and redundant stream GSM at 13 Kbps). Whenever there is non-consecutive loss, the receiver can conceal the loss, using the lower quality stream. If this happens infrequently, we are in pretty good shape.
In this scheme the (n -1)st and (n -2)nd low bit rate chunks and so on might get imposed in the packets sent. Thus, we get more redundancy, but increase bandwidth and playout time. Figure 7-2 shows FEC piggybacking scheme with a redundancy level 2. In this case if packet n is lost in the network it can be recovered by extracting information in packet n+1. Note that with this scheme the additional bandwidth consumed is equal to 2 times the data rate of the codec plus additional headers of the TCP/IP stack (RTP + UDP + IP headers). Moreover, there is an additional delay that will be added in the communication process in case of a lost packet because if packet n is lost, it is necessary to wait for packet n+1. When packet n+1 arrives it is safe to deliver the RTP data to the nominal jitter buffer space.

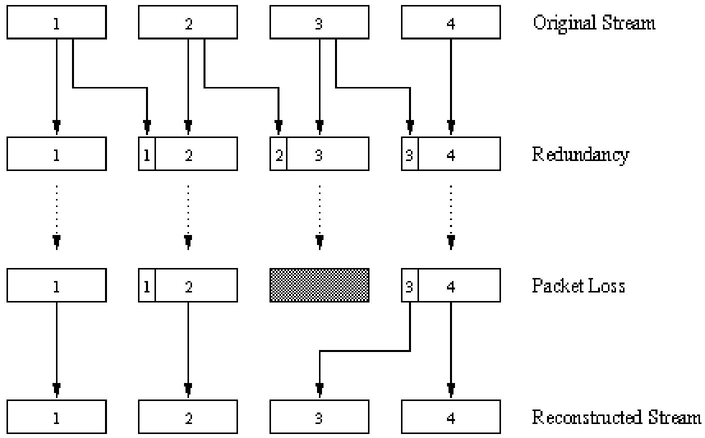


Figure 7-2 FEC with a piggybacking scheme

7.3.2 FEC with Reed Solomon codes

FEC is one of the error control schemes that is used to decrease the effect of packet loss by sending redundant information which is often known as "*parity*". There are two main components of any FEC mechanism. First, a *redundancy control algorithm* which indicates the amount of redundant information to be added to the RTP packets of the voice stream. Second, a *redundancy coding scheme* which describes the method of multiplexing and merging the redundant information added. A commonly used FEC code is the Reed-Solomon (RS) code. The RS code is expressed in the form of RS (n, k) notation, where n is the total number of packets and n-k indicates the amount of parity added as shown in Figure 7-3.

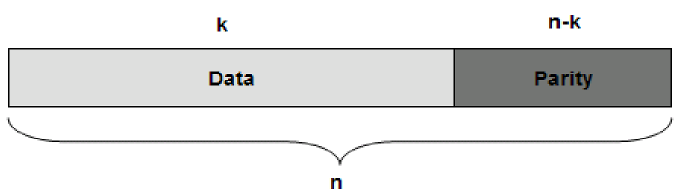


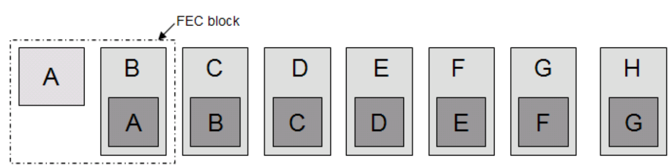
Figure 7-3 RS codeword

RS(n, k) code can recover all losses in the same FEC block if and only if k out of n packets are received successfully. In this particular research, we will use 3 main codes of Reed-Solomon: RS(2,1), RS(3,2) and RS(4,3). Under RS(2,1) coding as shown in Figure 7-4-a, the voice packet is lost during transmission if and only if the next packet that carries information about it is dropped as well. While, under RS(3,2) and RS(4,3) shown in Figure 7-4-a and 7-4-b, the voice packet is lost if and only if the next 2 and 3 packets are lost respectively. It is straightforward now to show that the loss rate after applying different RS codes can be expressed as in Table 7-1, based on the Gilbert model [102] usually used to describe the loss process of audio packets. We define p as the probability of going from state 0 to 1 and q as the probability to go from state 1 to 0, where 1 represents a packet loss while 0 represents a packet reached the destination successfully.

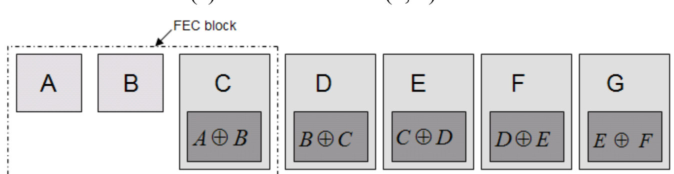
Table 7-1 Loss rates after reconstruction

FEC Mode	Packet loss after reconstruction
No FEC	$\frac{p}{p+q}$
RS(2,1)	$\frac{p(1-q)}{p+q}$
RS(3,2)	$\frac{p(1-q)^2}{p+q}$
RS(4,2)	$\frac{p(1-q)^3}{p+q}$

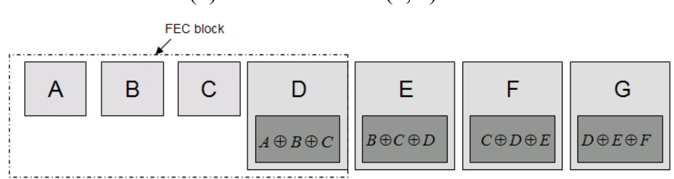
Under RS(2,1) or RS(3,2) or RS(4,3) coding, packets will have double size payloads as observed in the Skype traces [66, 68] due to multiplexing. In RS(3,2) the packet carries the multiplex of its two previous packets while in RS(4,3) it carries the multiplex of the previous 3 packets.



(a) Reed Solomon - (2, 1) code



(b) Reed Solomon - (3, 2) code



(c) Reed Solomon - (4, 3) code

Figure 7-4 Different types of Reed Solomon codes

In RS (2, 1), as an example if packet B is lost and C is delivered successfully then B can be recovered safely, While if B and C are lost then B can never be recovered. On the other hand, in RS(3,2) if B and C are lost while D and E are received successfully then first C can be recovered from D and the parity in E as $D \oplus C \oplus D$ (D XOR C XOR D), then B is recovered from C and the parity in D as $C \oplus B \oplus C$ (C XOR B XOR C). Similarly, for RS(4,3) as it will be capable to recover from 3 consecutive packet losses. This shows that RS(4,3) will be the most robust against burst losses while RS(2,1) will be the least code that can recover from bursts. But on the other hand, when using RS (4, 3) a higher delay will be embedded as it is necessary to wait for the entire block in order to recover the packet. This can be shown from the next graphs in Figure 7-5 - 7-8 that shows the packet loss percentage after reconstruction under different burst ratios.

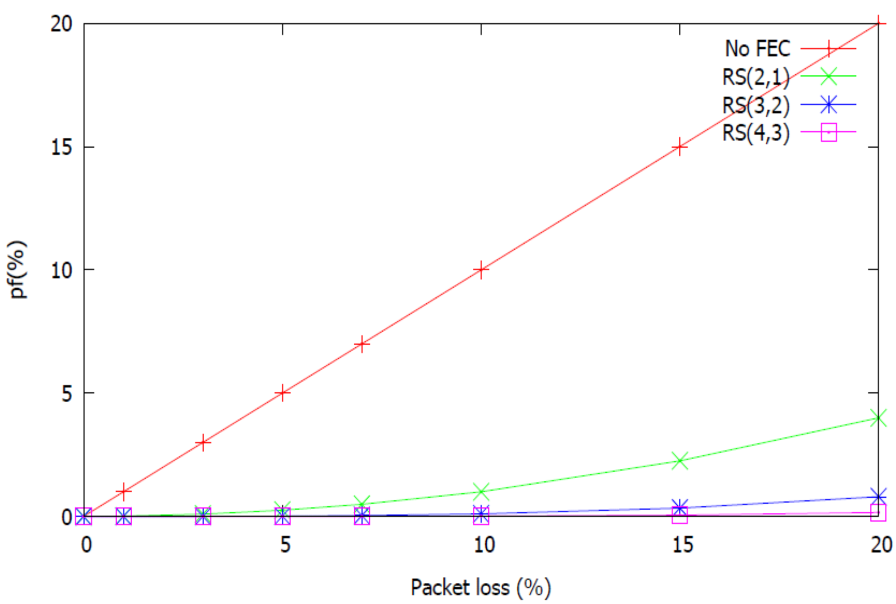


Figure 7-5 Effect of FEC on the packet loss under Burst ratio equals 1

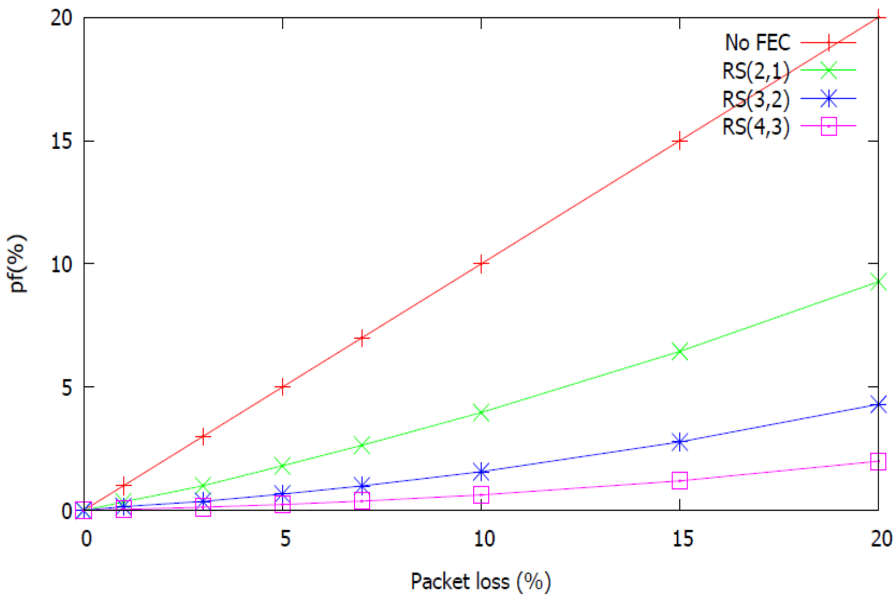


Figure 7-6 Effect of FEC on the packet loss under Burst ratio equals 1.5

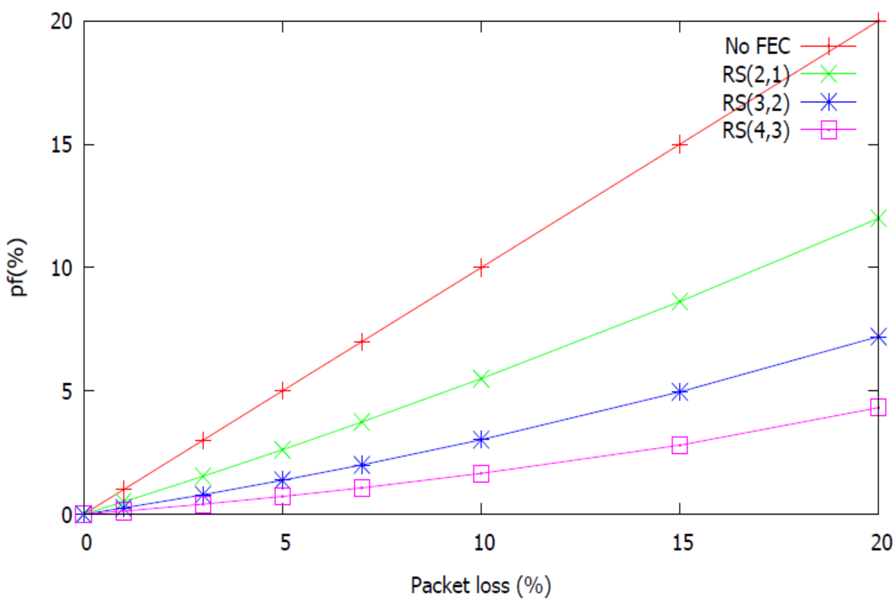


Figure 7-7 Effect of FEC on the packet loss under Burst ratio equals 2

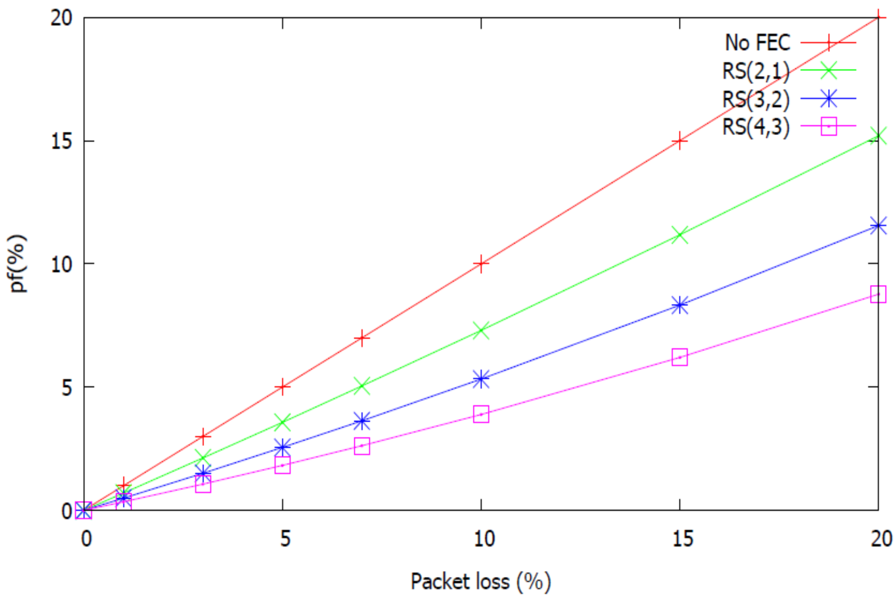


Figure 7-8 Effect of FEC on the packet loss under Burst ratio equals 3

The previous graphs show the percentage of packet loss after FEC (denoted as *pf* on the y axis) and the average packet loss rate (denoted as Packet loss on the x axis) range from 0-20% under different burst ratio range from 1-3. We want to stress two points from prior graphs: first, FEC is more effective with a small burst ratio. Second, choosing the appropriate RS code is highly dependent on the burst ratio. However, when the burst ratio decreases, we can see that some of the behavior of RS techniques might not much differ. This can be illustrated from Figure 7-5 under 15% packet loss as the *pf* (packet loss after reconstruction) value of RS(2,1) and RS(3,2) are nearly equal. The gap between different RS codes increases with the increase of the burst ratio. Although, It is obvious from the graphs that RS(4,3) will be the best selection for getting minimum packet loss but maybe it is not the best choice for the overall call quality. Consequently, the major drawback of using FEC mechanism is increasing the delay and bandwidth required for recovering the packets. The overhead in delay and bandwidth as a result of using FEC is function in the codec used. This can be illustrated in the graphs from Figures 7-9 – 7-10.

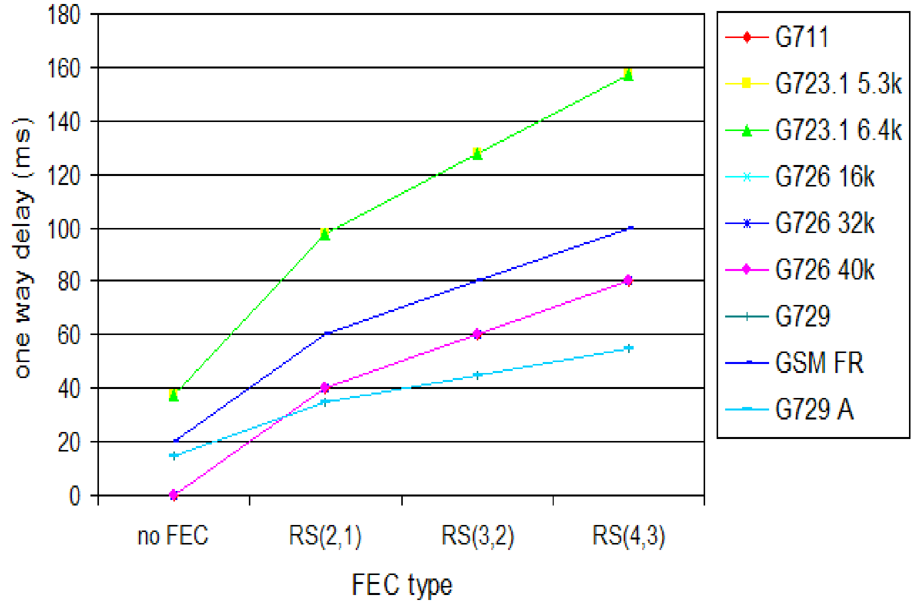


Figure 7-9 Effect of using different RS codes on the delay

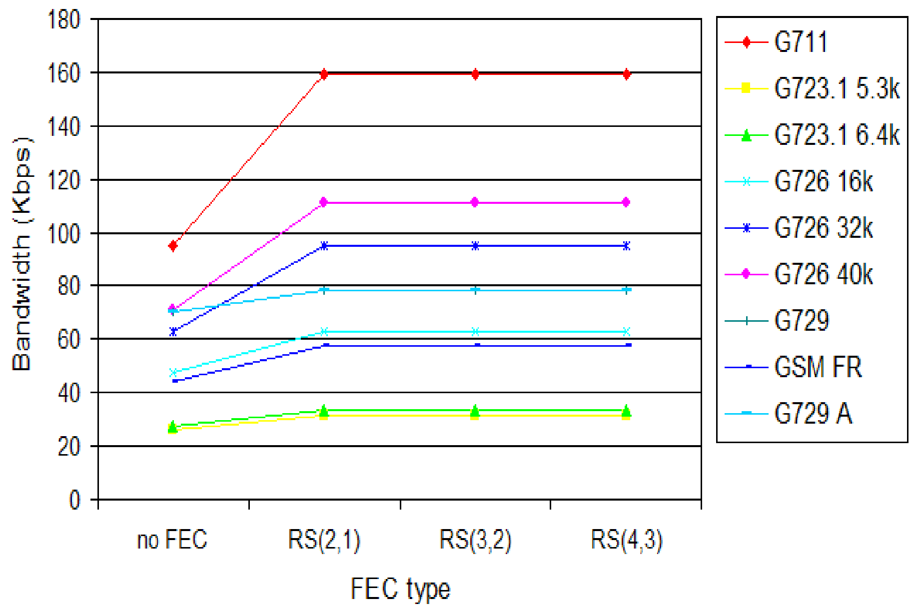


Figure 7-10 Effect of using different RS codes on the bandwidth

In Figure 7-9, the relation of the overhead in the overall one way delay of different codecs is shown. The delay overhead is function on two factors: packet interval (T) and framing delay. Framing delay is the time taken in DSP for digitizing the analogue signal and building frames and doing the opposite at the other side. Measuring this delay is very difficult as it occurs in the DSP. Thus, rfc3550 [4] already provides estimates of the framing delay for different types of codecs. The delay overhead depends also on the packet interval as a result of using Reed Solomon codes which will be necessary to wait for nT time (worst case scenario) in order to recover packet losses.

The delay overhead is added to the average network delay which can be estimated by the RTT/2 and the jitter buffer size that results in a delay in the buffer zone used to overcome jitter that matches the DSP output rate. Thus the delay overhead and the overall one-way delay can be expressed in (7.1, 7.2) respectively.

$$delay\_overhead = nT + framing\_delay \qquad (7.1)$$

$$overall\_delay = (RTT/2) + jitter\_buff\_size + delay\_overhead \qquad (7.2)$$

In Figure 7-9, the delay overhead of G.711 is the same as G726 with its different rates. The delay overhead of both types of G.723.1 is the same. Similarly, G.729 and G.729 A has the same delay overhead. We can consider that the delay overhead of G.729 is considered the least compared to other different codecs.

The additional bandwidth added due to using one of the Reed Solomon codes can be shown in Figure 7-10. It is now clear that RS(2, 1) uses double size payloads but other types of RS codes used in our research (RS(3, 2) and RS (4,3)) can use double size payloads as well by using logical operation between the payloads as an example XOR operation. We define the bandwidth overhead as the difference between the bandwidth with no FEC and after applying RS codes. The highest bandwidth overhead results from G.711 while the least overhead is from G.723.1 and G.729 A; this gives an advantage to using these codecs compared to others. Thus, it is preferable to apply FEC mechanism on low bit rate codecs to preserve bandwidth.

In order to show the effect of the codec used, the packet loss rate, and the burst ratio on the overall quality when using pure RS codes, we use the E-model to deduce the MOS score using two commonly used codecs (G.711 and G.723.1) as an example under different burst ratios (1, 1.5, 2, 2.5 and 3) and different packet loss rate range from 0-20%. Figures 7-11 – 7-15 show the relation between the packet loss rate represented on the x axis and the MOS score represented on the y axis of the G.723.1 codec while Figures 7-16 – 7-20 show the same relation for G.711 codec. We need to stress two points from Figures 7-11 – 7-20. First, the performance of the RS code is highly dependent on the codec used, packet loss rate and burst ratio; thus the previous QoS factors must be taken into account when developing a redundant control algorithm. Second, although the MOS score which indicates the QoE of the end user shows that RS(4,3) prevails compared to other RS codes, this may be changed when adding the delay overhead of the different codes when testing the conversational MOS score which is studied separately in further sections in the chapter; this means that the trade-off between packet loss recovery and over-head delay should be taken into account.

a) FEC applied to G.723.1 codec:

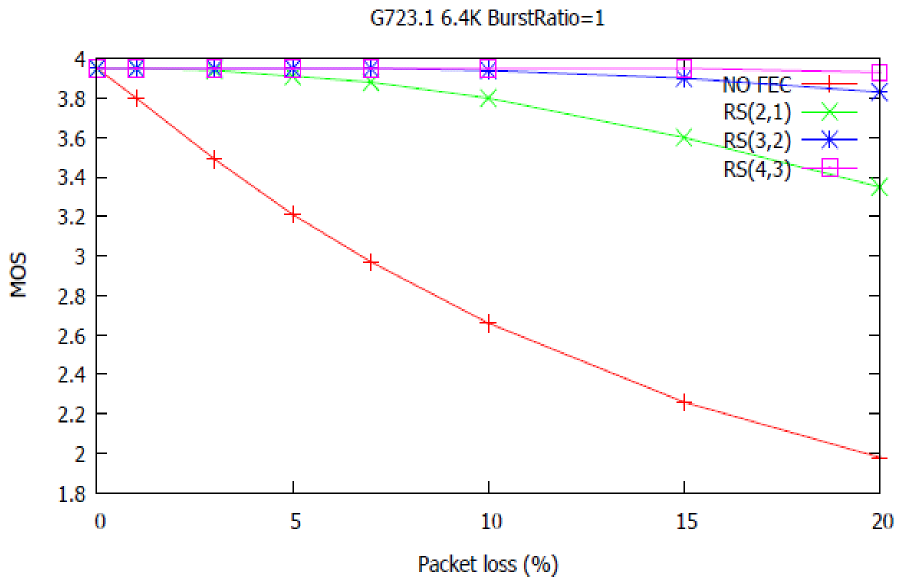


Figure 7-11 MOS score of G.723.1 codec under Burst ratio equals 1

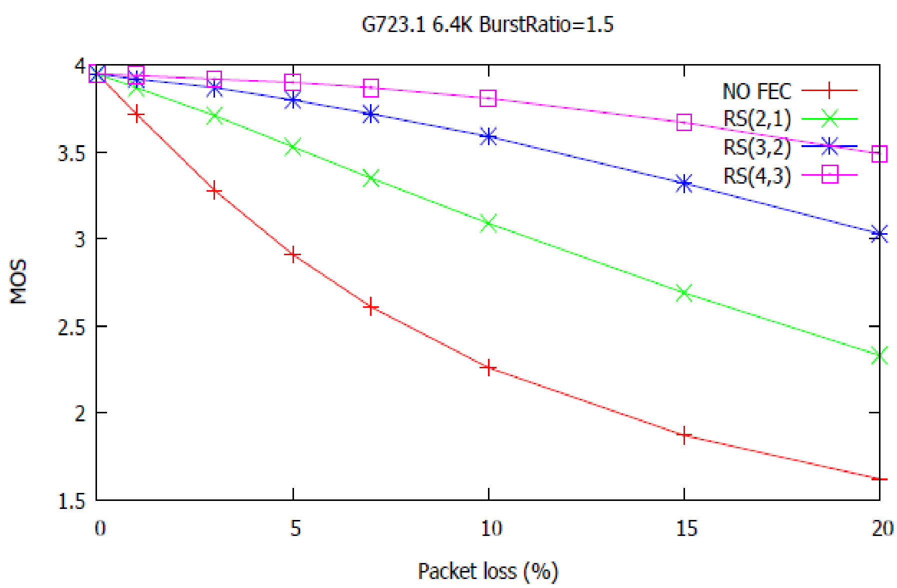


Figure 7-12 MOS score of G.723.1 codec under Burst ratio equals 1.5

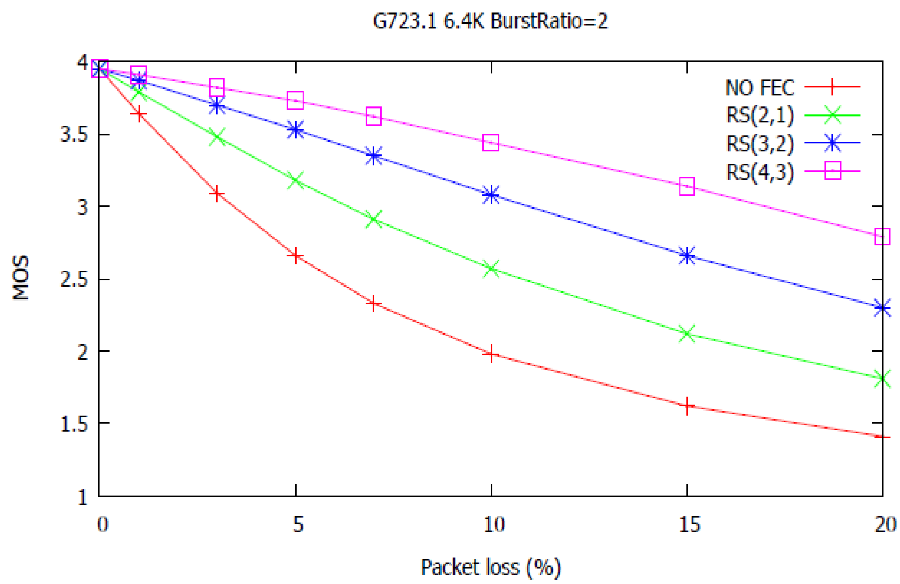


Figure 7-13 MOS score of G.723.1 codec under Burst ratio equals 2

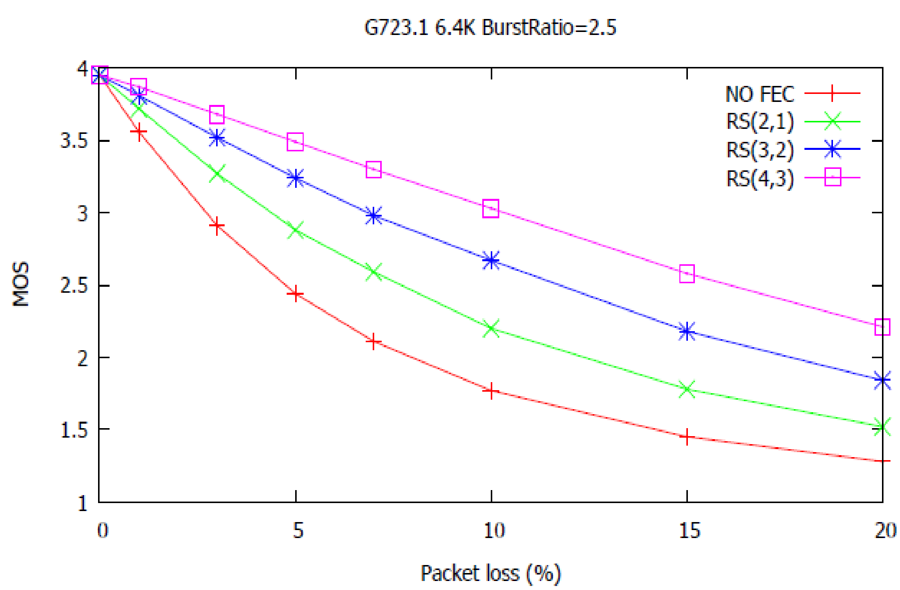


Figure 7-14 MOS score of G.723.1 codec under Burst ratio equals 2.5

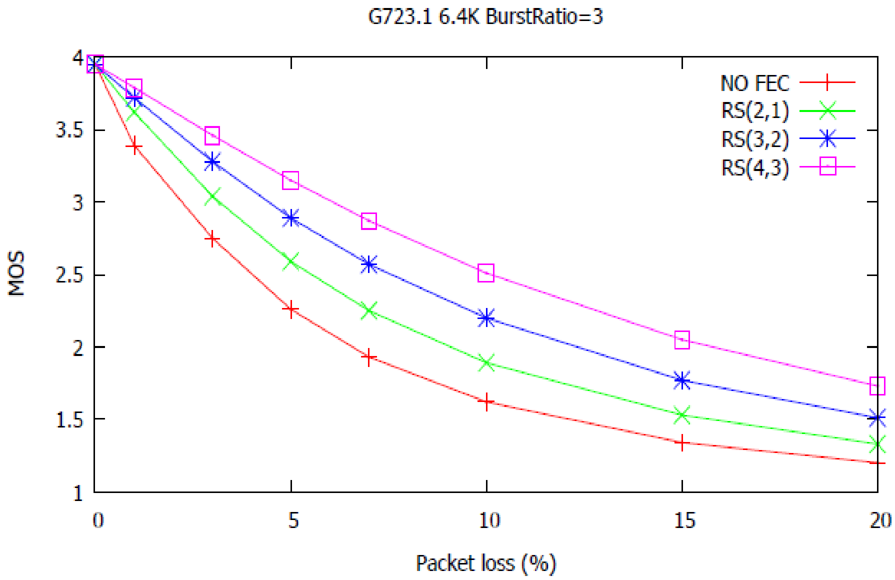


Figure 7-15 MOS score of G.723.1 codec under Burst ratio equals 3

b) FEC applied to G.711 codec:

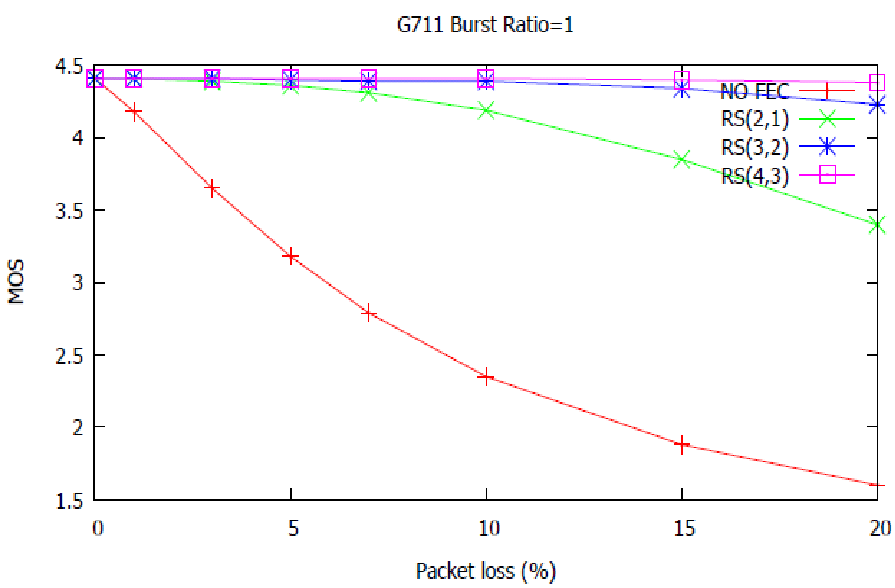


Figure 7-16 MOS score of G.711 codec under Burst ratio equals 1

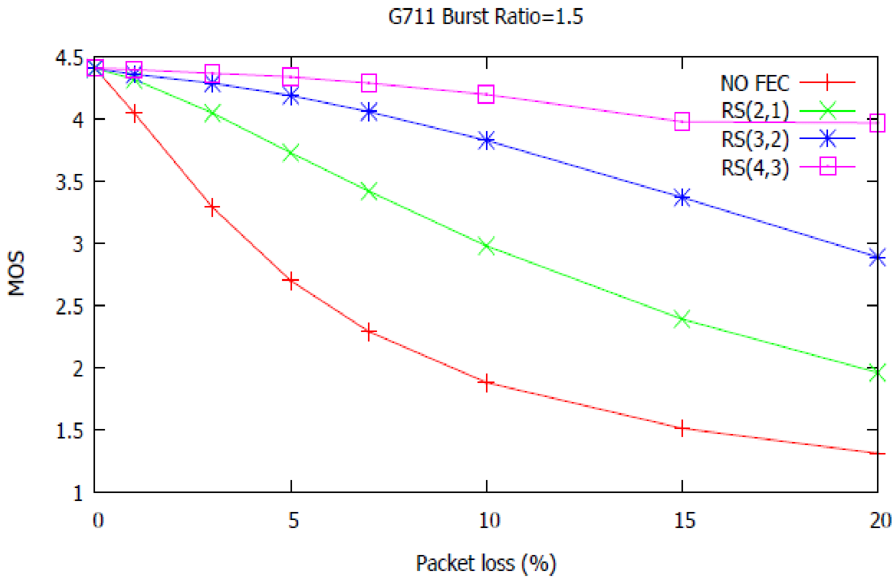


Figure 7-17 MOS score of G.711 codec under Burst ratio equals 1.5

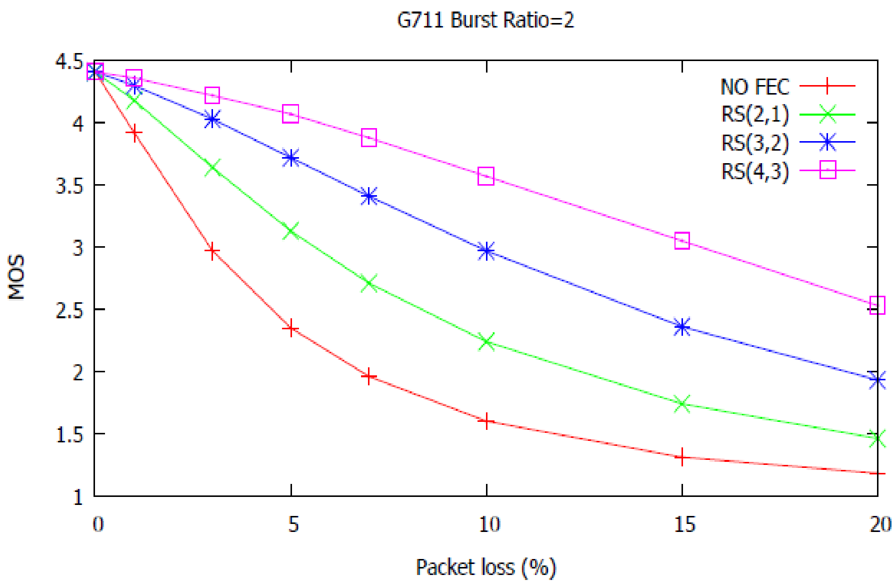


Figure 7-18 MOS score of G.711 codec under Burst ratio equals 2

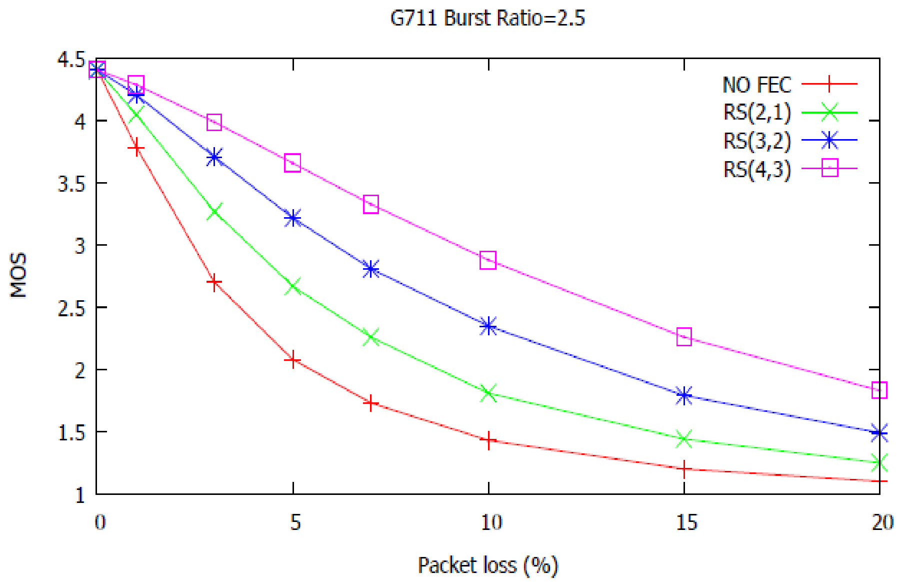


Figure 7-19 MOS score of G.711 codec under Burst ratio equals 2.5

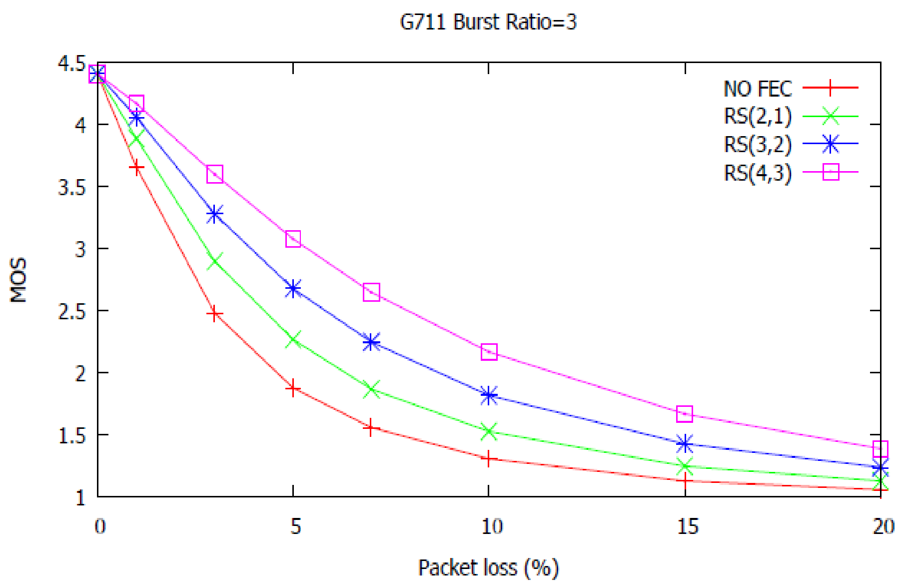


Figure 7-20 MOS score of G.711 codec under Burst ratio equals 3

7.4 Problem definition and objective

The network loss in an IP network is considered one of the most challenging aspects of IP QoS that is not guaranteed due to its complex behavior. This problem arises due to the complexity in predicting the probability of a packet to be lost. For the voice traffic, it is very difficult to tolerate a packet loss greater than 5% as it will be harmful to the voice quality [50]. There are some main factors that depend on the amount of packet loss that can be tolerated like the encoding algorithm and the sampling rate of the voice stream.

The importance of having low packet loss rates in VoIP applications to sustain high perceived voice call quality has led to a number of loss repair methods introduced (e.g.: FEC and LBR). It was shown previously that FEC is much preferred over LBR [63]. Forward Error Correction (FEC) recovers lost packets by transmitting redundant data. FEC schemes send redundant information along with the original information, so as to recover the lost original information. Nowadays, the Reed-Solomon codes are considered one of the most commonly used FEC coding schemes [66, 68]. Reed-Solomon codes are systematic block based codes that take digital data and add parity in order to recover from errors. Reed-Solomon codes are considered one of the most convenient techniques for VoIP applications, as every RTP packet can be represented as one of the data symbols of a codeword while the parity bits will hold some redundant information of previous packets.

In general, one of the major drawbacks of using FEC is increasing the delay in the communication process because the receiver cannot start the playback until it receives the parity data. Therefore, the level of FEC scheme applied to a stream of RTP packets must be addressed carefully by an adaptive redundant control algorithm. Thus, an optimization problem arises from here which is choosing an appropriate FEC scheme to be applied with the guarantee of higher perceived call quality. It is precisely the goal of the particular research in this chapter is to solve this problem, we have observed in current VoIP applications that the delay factor is not taken into consideration when adjusting the coding scheme which may lead to recover some of the packet losses at the expense of crossing the acceptable delay level, which may lead to a worse overall call quality. In this chapter, we have addressed this problem carefully, we have studied the effect of using different RS codes on the VoIP call quality, we have done a subjective interactive testing that leads us to deduce a single metric called *APU score* to rate the call quality. We then propose our redundant control algorithm (*APU algorithm*). The *APU algorithm* chooses the optimum RS code during the call according to the current/expected QoS parameters in order to attain the maximum interactive call quality that can be achieved.

7.5 Reviewing Related Work

Forward Error Correction has been introduced in Voice over IP (VoIP) applications to enhance the overall voice quality. Although these methods can reduce the effect of packet loss, they increase the amount of bandwidth and delay. Consequently, the FEC scheme applied to stream of RTP packets must be addressed carefully by an adaptive control algorithm. Research has been carried out to solve this optimization problem. While the literature is extensive, we review some of relevant previous work in this section and we show our contribution in this area.

J-C. Bolot and A.Garcia [69] introduce an adaptive FEC control algorithm named the "*Bolot algorithm*", this algorithm tries to maintain the loss rate after reconstruction at the receiver side. The Bolot algorithm will add redundancy only if the network loss rate is below the LOW mark, so it does not waste bandwidth. C.Padhye, K-J. Christensen and W.Moreno [70] show that the Bolot algorithm does not consider the change in network loss rate before reconstruction in its decision to change the amount of redundancy. This lead them to introduce the "*USF algorithm*" which is build on the Bolot algorithm. It detects loss bursts and considers the history of packet losses in the network before changing the amount of redundancy dynamically.

W.Jiang and H.Schulzrinne studied the loss repair methods applied to the VoIP applications. In [63, 71] they studied the relation between the packet interval and the FEC performance. They also perform subjective testing to see the influence of the bursty loss on the perceived quality. Finally, they show that the FEC is much preferred over LBR, if the main codec is already a low bit rate codec. In [71] they evaluate three commonly used codecs, G.729, G.723.1 and ILBC with FEC. They found that G.729 with FEC generally prevails.

M.Rousan and A.Nawasrah [72] introduced the Bandwidth Optimized Adaptive FEC (BOAFEC) approach to optimize the redundancy of the generated codewords from a Reed-Solomon (RS) encoder in order to save the bandwidth of the channel. This approach succeeded in saving 25% of the redundant bandwidth which allows a subscription for more clients on the same server. Moreover, this scheme responds to the high network losses by settling on the maximum allowed amount of redundancy.

T-Y.Huang, K-T.Chen, P.Hunang and P-J.Wang [66, 68] examines how much redundancy Skype adds to its voice streams. They show that Skype's control algorithm does not take the individual codec and bursty loss factors into consideration by comparing Skype's behavior under 3 codecs used in Skype, G.729, ISAC and SVOPC. This leads them to derive an optimal redundancy control policy for a desired VoIP quality under certain network conditions for G.711 and G.729 codecs. Although this methodology is considered useful, it misses adding the one-way delay as a main factor of adjusting the percentage of packets carrying

redundant information because they derived the optimum redundancy ratio based on the PESQ score which does not take into account other impairments such as transmission delay.
Although extensive work has been done in this area, we found that most of the work had focused on adjusting the redundancy ratio (the percentage of packets that carry redundant voice data [66]) based on the current or expected network conditions but no one addresses choosing the optimal and appropriate RS code according to the QoS parameters and switching between the codes dynamically in the call. Consequently, we have studied the effect of using three different commonly used RS codes in VoIP applications separately on the VoIP call quality and then we propose the *APU algorithm* to switch between those 3 different codes dynamically based on the current/expected network conditions. We left for the future, the merging of the *APU algorithm* with other redundancy control algorithms to adjust the percentage of packets that carry redundant data to save bandwidth.

7.6 APU Algorithm

Although FEC is useful for decreasing the effect of packet loss, the overall call quality may be reduced when it is used. This happens due to the increase of the delay that leads to a decrease in the overall call quality from the human perception point of view. This leads us to develop the APU (Acceptable, Poor and Unacceptable) algorithm that will restrict the transition from the current state to a desirable state only using different FEC RS coding schemes as the state reflects the expected call quality perceived at the end user. Given this, we derived a state diagram that shows the acceptable transition states from the current state of the call using one of the RS codes in order to achieve higher call quality.

7.6.1 APU model for MOS and one-way delay

In order to reflect most of the factors of the network in the APU state diagram, our state diagram is developed from several states; each is a function of the MOS score from the E-model (excluding the Id) and the one way delay. Thus, we can address the correlation between the one way delay and the other factors accommodated in the rating factor R (packet loss, burst ratio, codec impairment factors). By this method, we accommodate the factors mentioned before affecting choosing certain FEC RS code. Such factors are the packet loss, burst ratio, codec used and the one-way delay.

ITU-T G.107 Recommendation [25] provides the satisfaction level corresponding to a measured MOS. Since, we need to derive an algorithm dependent on the codec used, thus we cannot rely on such relation. For instance, the maximum achievable MOS of G.711 codec is 4.41 indicating, “Very satisfied” whilst the G.726 16K has a maximum MOS of 2.74 indicating, “Nearly all users are dissatisfied”. We want to assess the codecs subject to the performance of each codec. So in the prior example, 2.74 should be satisfying if the G.726 16K is used and would be unsatisfying if G711 codec is used. This can be illustrated from Figure 7-21 that shows the different performance of codecs under different packet loss rate range from 0-20%.

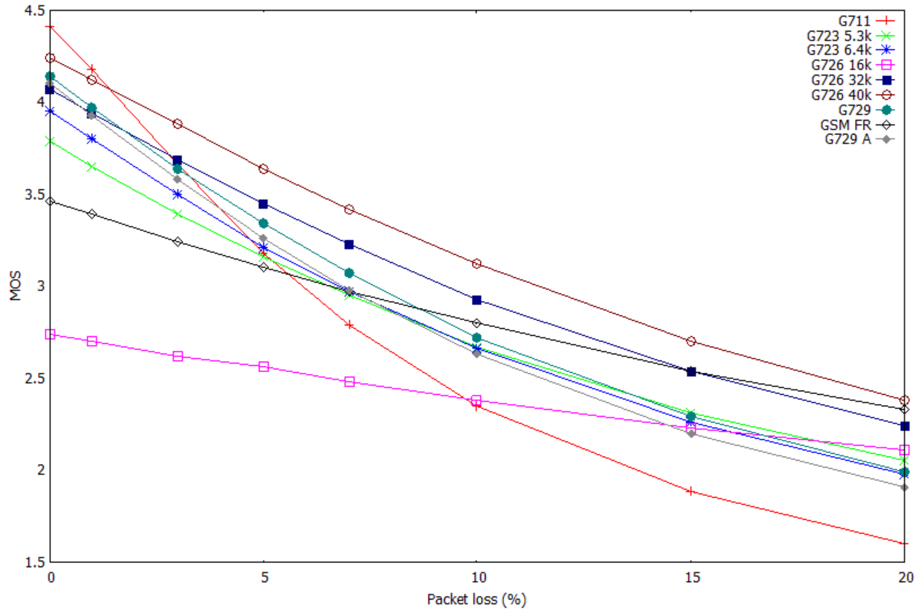


Figure 7-21 MOS comparative analysis of different codecs

It is obvious from Figure 7-21 that at 0% packet loss, the MOS score differs from one codec to another (e.g.: G.711: MOS=4.41, G.723 5.3k: MOS=3.79). Consequently, we first measured the maximum score that can be attained from each codec and then we divide this MOS score into three equal parts to derive the APU relative scores for each codec as shown in Table 7-2.

Table 7-2 APU MOS model

codec	Acceptable	Poor	Unacceptable
G.711	3.28-4.41	2.14-3.28	1-2.14
G.723 5.3k	2.86-3.79	1.93-2.86	1-1.93
G.723 6.3k	2.96-3.95	1.98-2.96	1-1.98
G.726 16k	2.16-2.74	1.58-2.16	1-1.58
G.726 32k	3.04-4.07	2.02-3.04	1-2.02
G.726 40k	3.16-4.24	2.08-3.16	1-2.08
G.729	3.26-4.13	2.13-3.26	1-2.13
GSM FR	2.64-3.46	1.82-2.64	1-1.82
G.729 A	3.06-4.10	2.03-3.06	1-2.03

Delay has a relatively significant impact on the VoIP call quality. Sometime it is slightly inconsistent. ITU-T Recommendation G.114 [73] for one-way delay transmission is:

- Under 150 ms: acceptable for all of the users.
- 150 to 400 ms: acceptable on the condition that the administrators are aware of the drawback of this delay in the overall quality.
- Over 400 ms: Unacceptable for all of the users.

Thus, we derive the APU model for mapping the one way delay to the user perception of the link quality as shown in Figure 7-22.

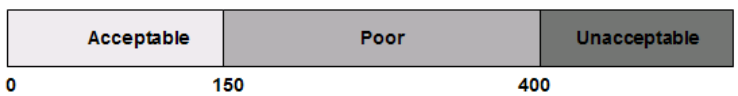


Figure 7-22 APU delay model [73]

In our research and particularly in this chapter, we will use notation $< d_{net}, MOS_{net} >$ to define the current state of the network conditions that will reflect the call quality.

7.6.2 Closed network testing

A novel closed-network test methodology that involves actual human subjective testing is performed to derive the APU state diagram. In our tests, human subjects are asked to rank their perception QoE (MOS score) of interactive VoIP calls for a different range of packet loss and delay configured using Dummynet [49].

The test cases can be listed as a result of all possible combinations between delay and MOS different levels as : <A, A>, <A, P>, <A, U>, <P, A>, <P, P>, <P, U>, <U, A>, <U, P> and <U, U>, where each test case is defined by a certain sequence of the network factor levels $< d_{net}, MOS_{net} >$. For example, the <A, P> test case corresponds to network conditions that results in an acceptable one way delay and Poor rating factor R according to the codec used.

We have carried conversation-opinion subjective tests according to the procedures provided in ITU-T P.800/P.920 Recommendation [41, 74]. Our tests are made on an isolated LAN with no cross traffic. Before the test, the people shared in the test were informed about its purpose, procedures and the benefits from the test. ITU-T recommends 16 persons as the minimum number for the accuracy required for the results [74].

In order to obtain a wide range of subjective quality scores from our testing, 20 human subjects shared in the test. We classified the human subjects into general and experienced users. General users are those who have moderate experience due to their occasional usage with the VoIP applications. Experienced users are considered those users who use VoIP applications frequently and they know their concern from using it. According to the ITU Recommendation P.920 [74] our tests were based mainly on the Name-Guessing task which is based on a question-answer game performed according to a fixed protocol. A base line test with no network impairments was executed before starting the tests. The human subjects are asked to rank their subjective perceptual quality for the test cases relative to the base line test. In our experiment, each test case is tested 5 times and then we obtain the MOS value range from 1(Bad) to 5 (Excellent) as shown in Figure 7-23. The final MOS score will be the arithmetic mean of all the individual scores. We use the midpoint of each range from the delay and MOS score in order to reproduce the tests (i.e.: delay = 275ms for the poor level).

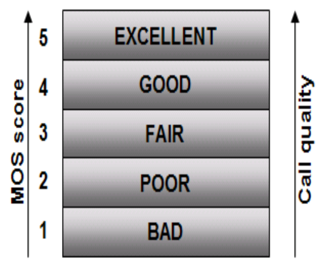


Figure 7-23 MOS scale

We note that our results from these tests can be generalized for any voice codec (e.g.: G.711, G.729, G.723, G.729A, etc). For simplicity, we focus on our testing on 2 common used codecs: G.711 and GSM FR. Our results are shown in Figure 7-24, the test case $< d_{net,} MOS_{net} >$ is represented on the x axis and its corresponding average MOS score is shown on the y axis. Interestingly, our tests show that human perception is more sensitive to packet loss than delay; this can be directly observed from the higher MOS score resulted from <P, A> test case than <A, P>. Contrary to our initial expectations, it was observed that the transition from acceptable level of delay to poor is not highly observed from the human perception point of view, while the unacceptable level of delay is more observable and an obvious difference was recognized by the listeners between the poor and unacceptable level of delay; this can be shown in Figure 7-25 by the small slope in the 2 codecs between the following test cases: <A, A>&<P, A>, <A, P>& <P, P>, <A, U>&<P, U> and the higher slope between <P, A>& <U, A>, <P, P>& <U, P>, <P, U>&<U, U>. On the other hand, the transition between acceptable rating factor R (indicating the packet loss level) to poor then unacceptable level is highly recognized by the human perception; this was shown by the large slope between <A, A> & <A, P>&<A, U>, <P, A>&<P, P> &<P, U>, <U, A> & <U, P> & <U, U> as shown Figure 7-25. From our tests, we have noticed also that people prefer in their conversation to have both poor delay and packet loss rather than having one of them with an unacceptable level as shown in Figure 7-26 where the MOS score of the <P, P> test case is greater than the <A, U> , <P, U>, <U, A> and <U, P>. Since both codecs (G.711, GSM FR) tested subjectively shown in Figure 7-24 show the same trend, thus we can generalize the same results for any voice codec because we are expecting to have same trend for the MOS of different test cases but with different value due to the variation of the performance of different codecs.

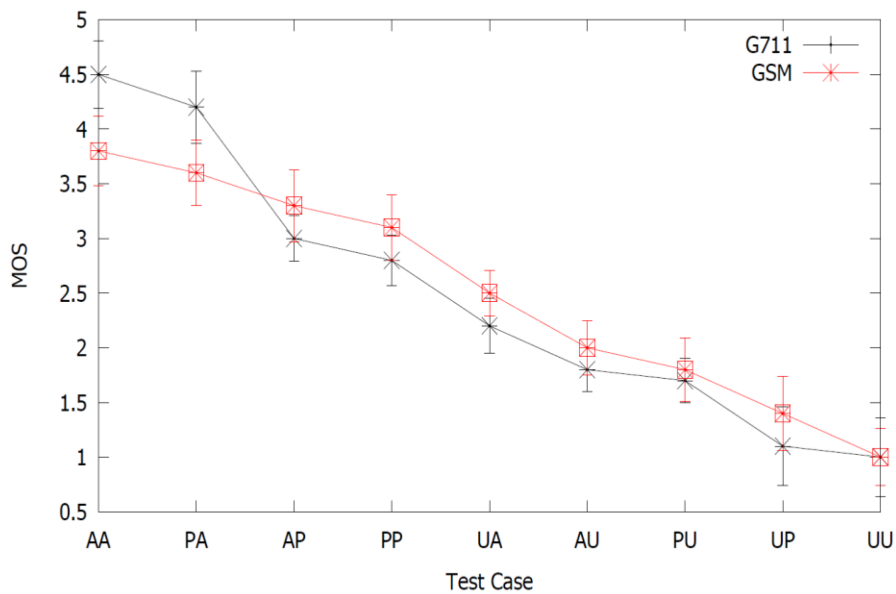


Figure 7-24 Subjective testing MOS scores of G711 and GSM codecs

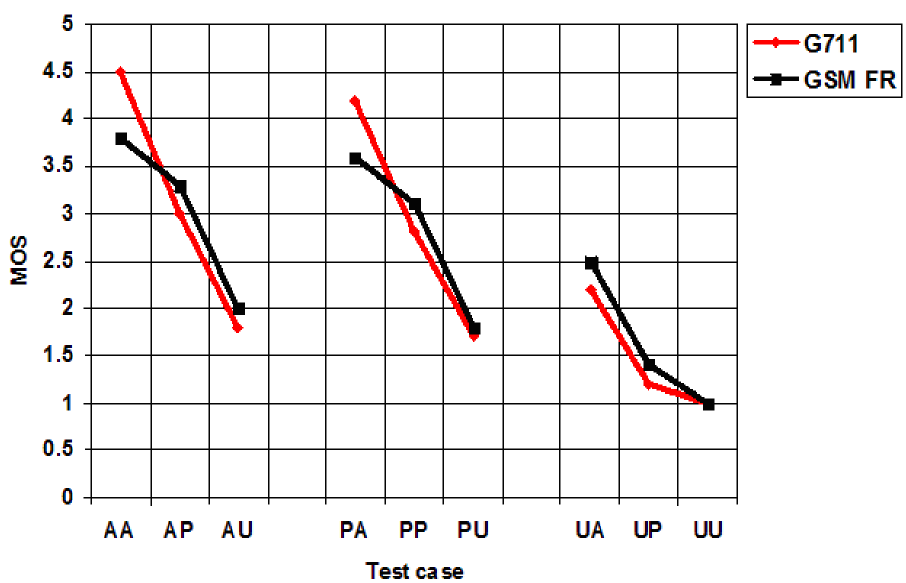


Figure 7-25 Subjective testing MOS scores (Packet loss transition effect)

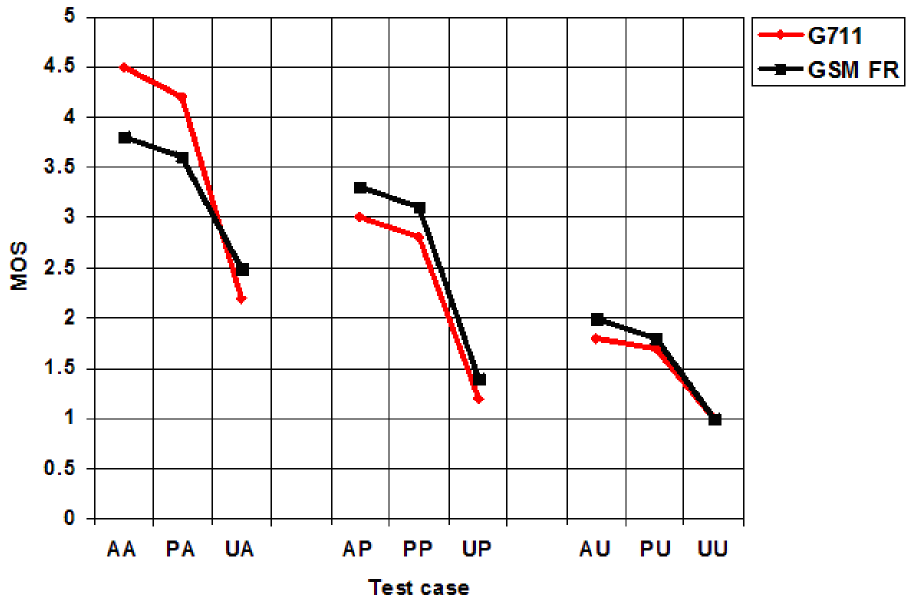


Figure 7-26 Subjective testing MOS scores (Delay transition effect)

In order to define a single metric for our subjective results, we define the term *"APU score"* as an indication of the conversational call quality from the human perception point of view of each test case that reflects different network conditions. It is now straightforward to deduce Table 7-3 from Figure 7-24. The APU score increases with the increase of the VoIP quality perceived.

Table 7-3 APU scores of each state

State	**APU score**	**State**	**APU score**
AA	9	AU	4
PA	8	PU	3
AP	7	UP	2
PP	6	UU	1
UA	5		

7.6.3 *Proposed APU algorithm*

We propose our *APU algorithm* based on the APU state diagram derived from Figure 7-27. The state diagram shows the desirable transitions from the current state according to our subjective testing. The APU state diagram is shown in Figure 7-27. Our objective cases to enhance is that with poor or unacceptable MOS score. Thus no transitions or enhancements for <P, A> or <A, A> or <U, A> states because they already have acceptable MOS based on the network losses indicating acceptable call quality.

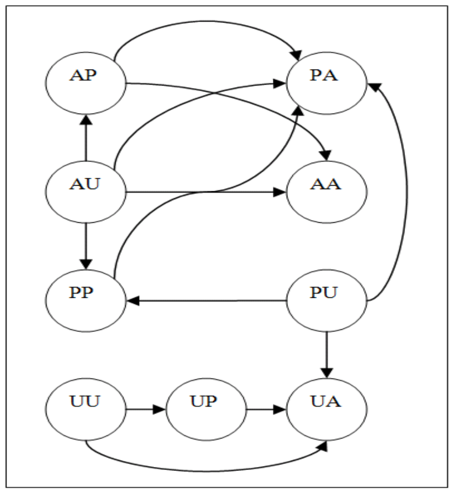


Figure 7-27 APU state diagram

The simplest way to build a control algorithm is to have a target residual loss rate at the receiver after applying FEC, but our algorithm is based on choosing the optimum FEC coding scheme from three different RS codes that are commonly used in the VoIP applications [66, 68] according to the current network conditions. In our algorithm, the network conditions are measured based on the RTCP receiver reports (RRs) from generated packets. Iperf [75] can be used also for the same purpose by generating the codec's traffic using UDP and measuring the current network conditions. The RTCP feedback is sent back by the receiver every 5 seconds [3] but unfortunately, RTCP report does not include information about random packet lost (p) and burst ratio (q) separately. Thus, two solutions are provided around this issue [76]. The first solution is to use other unused fields (e.g.: jitter) in RTCP RRs to include p and q separately. The second solution is to assume that the loss process is Bernoulli, and not Gilbert [76]. We applied the first solution as it is more preferable especially that FEC depends on the burst ratio as well as the random packet loss rate. In our algorithm, we wait for every 5 consecutive RRs (25 seconds) to take our decision. Then the rating factor R and the overall one way delay is measured so that the current state (i.e.: <P, P>) can be deduced. Our algorithm focuses on enhancing only the MOS score (packet loss level indication) with poor or unacceptable level as these states are considered our target to enhance. The transit state is deduced by calculating the R and delay values after reconstruction using three main commonly used RS codes in VoIP applications. It is now direct to determine whether the next state is a valid state or not by the usage of the state diagram that shows the desirable transitions deduced from Figure 7-27. We will choose the RS code to be used with the corresponding state having minimum *APU score*. It is possible to have 2 same states which means the same *APU score*, if this is the case then we will choose the smaller RS code as it will result in less delay.

A summary of the *APU algorithm* proposed in this chapter and applied at the sender side is shown in Figure 7-28.

```
if (! 5 RTCP RRs are received)
    Wait until all the 5RRs received;
else
    Calculate avgPacketLoss, avgBurstRatio, and avgDelay;
    // calculate current state without using any RS codes:
    Calculate rating MOS score using E-model ( MOS_network );
    Calculate overall Delay ( d_network );
    if (MOS != P||U)
        Nothing to be done;
    else
        Deduce current state < d_network , MOS_network >;
        Deduce equivalent APU score for the current state;
        Add "NO RS" and its APU score in list (validStates);
        Loop for i=0....2
            Calculate MOS after reconstruction for RS(2+i,1+i);
            Calculate overall delay after reconstruction
            Deduce next_state < d_network , MOS_network >;
            if (next_state [i] is valid transition state) // from state diagram
                Deduce APU_score for next_state[i];
                Add RS(2+i,1+i) and its APU score in list (validStates);
            else
                Exclude next_state;
            end if
        end Loop
    Sort the validStates list by APU score in descending order;
    if (2 states or more has same APU score)
        Sort them with minimum delay mode // RS(2,1)<RS(3,2)<RS(4,3)
    end if
Use the top RS mode in the validStates list;
end if
```

Figure 7-28 APU algorithm applied at the sender side

7.7 *Results and Discussion*

In this section, we present the simulation results for the *APU algorithm*, we show the response and the results of our *APU algorithm* under different network conditions and how it affects the overall conversational call quality perceived by the end user. Moreover, we show the APU adaptive algorithm results and compare then with the different pure Reed Solomon FEC codes.

Our results and test cases are based on varying the network conditions six times with different levels of delay and packet loss. We have measured our results within the boundaries of the APU model of the packet loss and delay so that we can move from one state to another after adding the overhead in the one-way delay and the percentage of packet loss recovered as a result of using certain RS code. This demonstrates the effect of the APU algorithm on the final perceived VoIP call quality.

We have tested the *APU algorithm* under two commonly used codecs: G.711 and G.723.1 6.4k. In our simulation setup, we vary the network conditions after 20, 45, 70, 95, 120 and 145 seconds (i.e.: before five seconds from taking the decision in the *APU algorithm based on the next received RTP report*). We give two more seconds which are needed to switch between different states having different APU scores. In our results, the time is represented on the x axis in seconds, while the *APU score* is represented on the y axis. The *APU score* gives an indication of the final quality perceived by the end user as it is directly proportional with the quality of experience (QoE). A sample of our results are described in four different test cases. The first two test cases are tested using G.711 codec while the last two test cases are tested using G.723 6.4k codec. In each test case, we compare the adaptive *APU algorithm* with the fixed RS(2,1), RS(3,2) and RS(4,3). In each test case, we have demonstrated our results in a table to show the different states indicating different RS codec used in our proposed adaptive redundancy control algorithm.

7.7.1 *First test case*

The first test case is conducted using the G.711 codec with burst ratio equal to 1.5 under different percentages of packet loss and different delay levels. The network conditions are changed as follows: after 20 seconds the packet loss and delay are changed to 15% and 170 ms respectively. After 45 seconds the packet loss is changed to 20% while the delay remained 170 ms. After 70 seconds the packet loss is returned to 15% while the delay is increased to 330 ms. After 95 seconds the packet loss is returned to 7% while the delay is decreased to 100 ms. After 120 seconds the packet loss is changed to 20% while the delay is changed to 330 ms; finally the packet loss is returned to 7% while the delay is increased to 330 ms. The response of the *APU algorithm* compared to the different pure RS codes is shown in Figures 7-29 – 7-31 and the comparative results of the this test case is shown in Table 7-4.

Table 7-4 Comparative results of the first test case

Time (sec.)	**APU algorithm**		**RS(2,1)**	**RS(3,2)**	**RS(4,3)**
	state	**code**			
0-25	AA	-	AA	AA	AA
25-50	PA	3,2	PP	PA	PA
50-75	PA	4,3	PU	PP	PA
75-100	PA	3,2	PP	PA	UA
100-125	AA	2,1	AA	PA	PA
125-150	PP	3,2	PU	PP	UA
>150	PA	2,1	PA	PA	UA

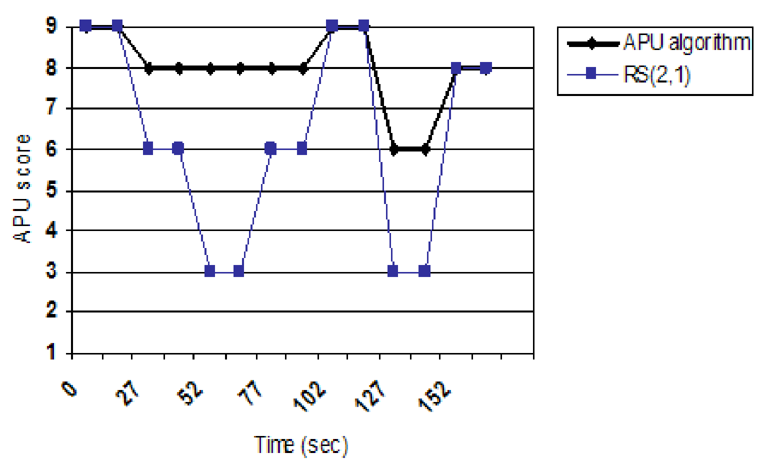


Figure 7-29 Comparative analysis of the first test case with RS (2, 1)

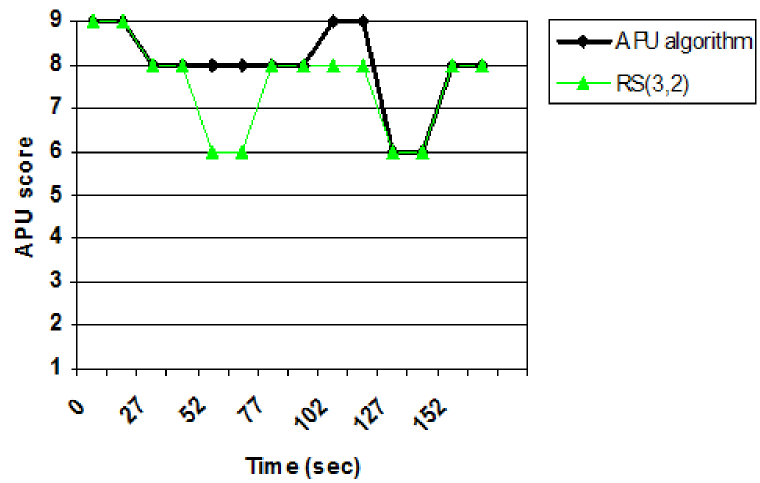


Figure 7-30 Comparative analysis of the first test case with RS (3, 2)

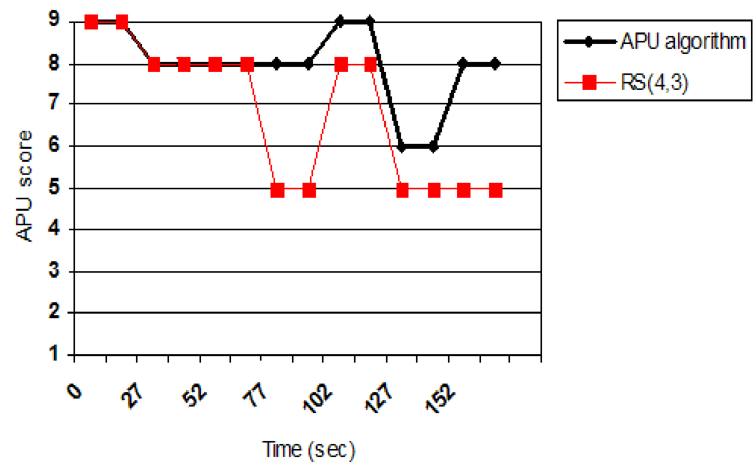


Figure 7-31 Comparative analysis of the first test case with RS (4, 3)

7.7.2 Second test case

The second test case is conducted using the G.711 codec with burst ratio equal to 2.5 under different percentages of packet loss and different delay levels. The network conditions are changed as follows: after 20 seconds the packet loss and delay are changed to 3% and 170 ms respectively. After 45 seconds the packet loss is changed to 5% while the delay remained 170 ms. After 70 seconds the packet loss is returned to 3% while the delay is increased to 330 ms. After 95 seconds the packet loss is raised again to 5% while the delay remained 330 ms. After 120 seconds the packet loss is reached 10% while the delay is changed to 100 ms; finally the delay is returned to 330 ms keeping the packet loss rate 10%. The response of the *APU algorithm* compared to the different pure RS codes is shown in Figures 7-32 – 7-34 and the comparative results of the this test case is shown in Table 7-5.

Table 7-5 Comparative results of the second test case

Time (sec.)	**APU algorithm**		**RS(2,1)**	**RS(3,2)**	**RS(4,3)**
	state	**code**			
0-25	AA	-	AA	AA	AA
25-50	PA	3,2	PP	PP	PP
50-75	PA	4,3	PP	PP	PA
75-100	PA	3,2	PP	PA	UA
100-125	PP	2,1	PP	PP	UA
125-150	PA	4,3	AU	PP	PA
>150	PP	3,2	PU	PP	UP

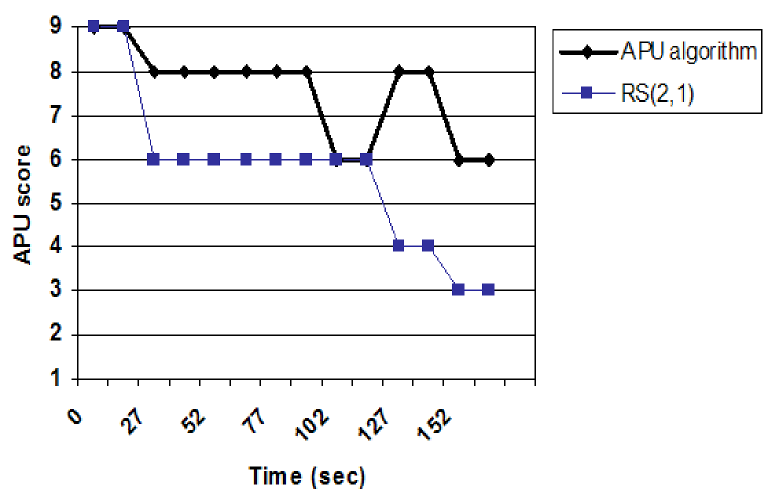


Figure 7-32 Comparative analysis of the second test case with RS (2, 1)

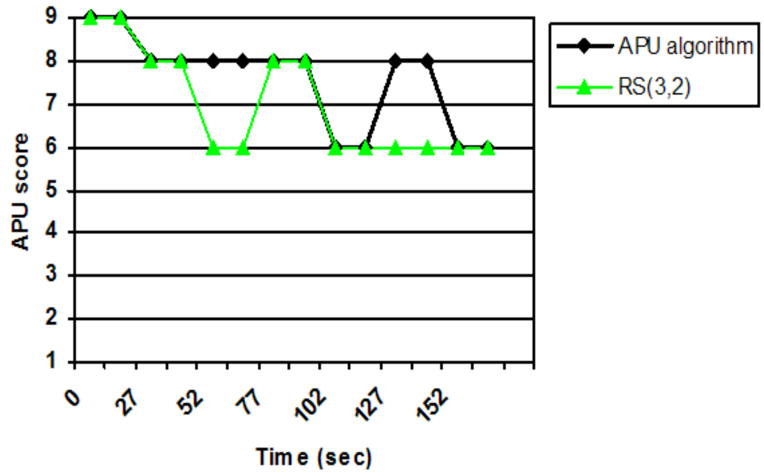


Figure 7-33 Comparative analysis of the second test case with RS (3, 2)

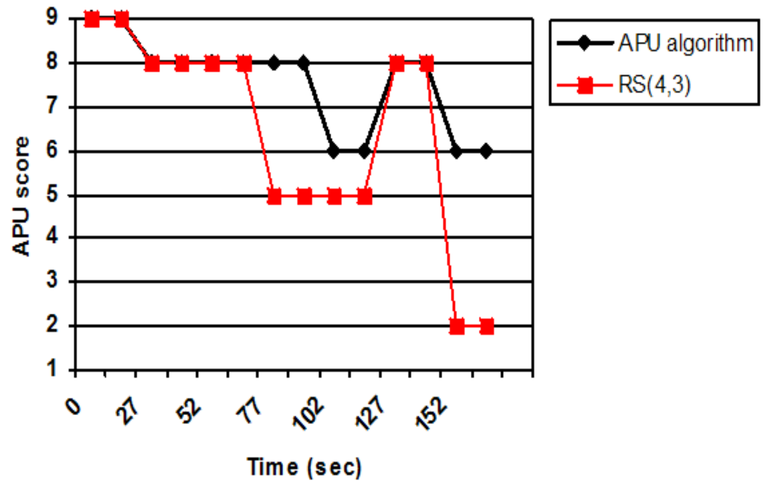


Figure 7-34 Comparative analysis of the second test case with RS (4, 3)

7.7.3 Third test case

The third test case is conducted using the G.723.1 codec with burst ratio equal to 2.5 under different percentages of packet loss and different delay levels. The network conditions are changed as follows: after 20 seconds the packet loss and delay are changed to 10% and 330 ms respectively. After 45 seconds the delay is changed to 100 ms while the packet loss remains 10 %. After 70 seconds the delay is increased to 170 ms with the same previous packet loss. After 95 seconds the packet loss is increased to 15% while the delay is set to 330 ms. After 120 seconds the packet loss is kept as it is, while the delay is changed to 1000 ms; finally the packet loss is decreased to 7% with preserving the same previous delay. The response of the *APU algorithm* compared to the different pure RS codes is shown in Figures 7-35 – 7-37 and the comparative results of this test case is shown in Table 7-6.

Table 7-6 Comparative results of the third test case

Time (sec.)	APU algorithm		RS(2,1)	RS(3,2)	RS(4,3)
	state	code			
0-25	AA	-	AA	AA	AA
25-50	PP	2,1	PP	PP	UA
50-75	PA	4,3	AP	AP	PA
75-100	PA	4,3	PP	PA	PA
100-125	PP	3,2	PU	PP	UP
125-150	AP	3,2	AU	AP	PP
>150	AA	3,2	AP	AA	PA

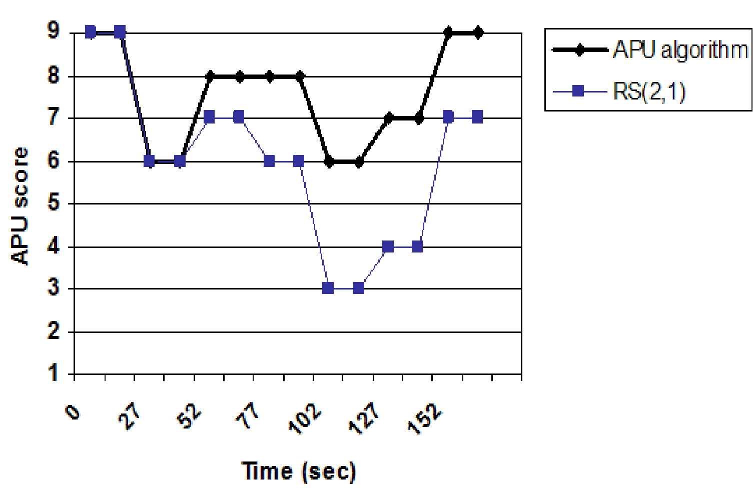


Figure 7-35 Comparative analysis of the third test case with RS (2, 1)

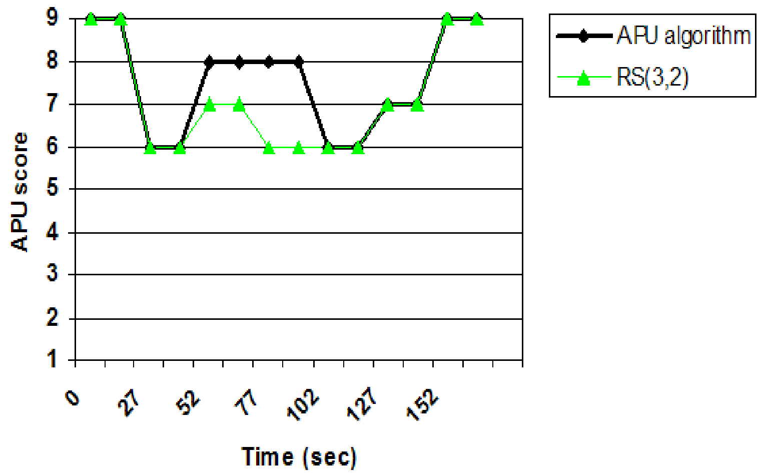


Figure 7-36 Comparative analysis of the third test case with RS (3, 2)

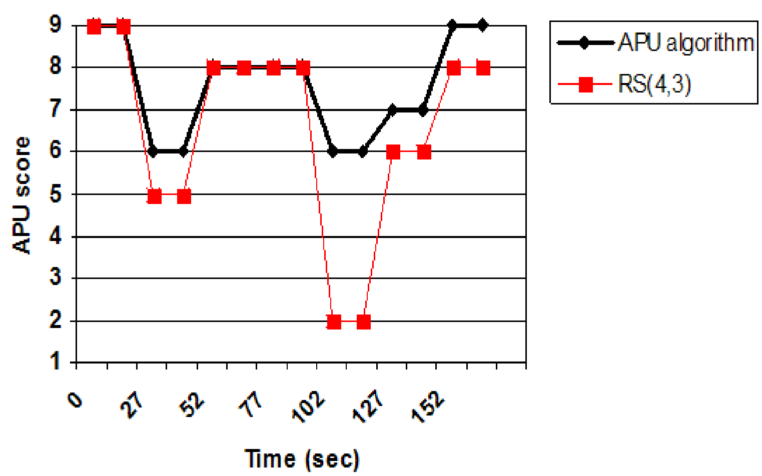


Figure 7-37 Comparative analysis of the third test case with RS (4, 3)

7.7.4 *Fourth test case*

The fourth test case is conducted using the G.723.1 codec with burst ratio equal to 3 under different percentages of packet loss and different delay levels. The network conditions are changed as follows: after 20 seconds the packet loss and delay are changed to 5% and 170 ms respectively. After 45 seconds the packet loss is changed to 10% while the delay remained 170 ms. After 70 seconds the packet loss is returned to 5% while the delay is decreased to 1000 ms. After 95 seconds the packet loss is returned to 10% while the delay is increased to 330 ms. After 120 seconds the packet loss is changed to 5% while the delay is changed to 330 ms; finally the packet loss is returned to 10% while the delay is decreased to 100 ms. The response of the *APU algorithm* compared to the different fixed RS codes is shown in Figures 7-38 – 7-40 and the comparative results of the this test case is shown in Table 7.7.

Table 7-7 Comparative results of the fourth test case

Time (sec.)	**APU algorithm**		**RS(2,1)**	**RS(3,2)**	**RS(4,3)**
	state	**code**			
0-25	AA	-	AA	-	AA
25-50	PA	4,3	PP	PP	PA
50-75	PP	3,2	PU	PP	PP
75-100	PA	4,3	AP	PP	PA
100-125	PP	3,2	PU	PP	UP
125-150	PP	2,1	PP	PP	UA
>150	PA	4,3	PP	PP	PA

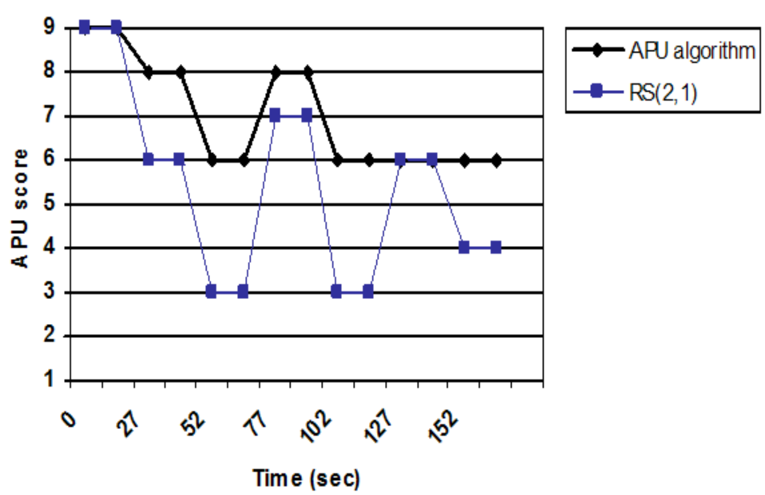


Figure 7-38 Comparative analysis of the fourth test case with RS (2, 1)

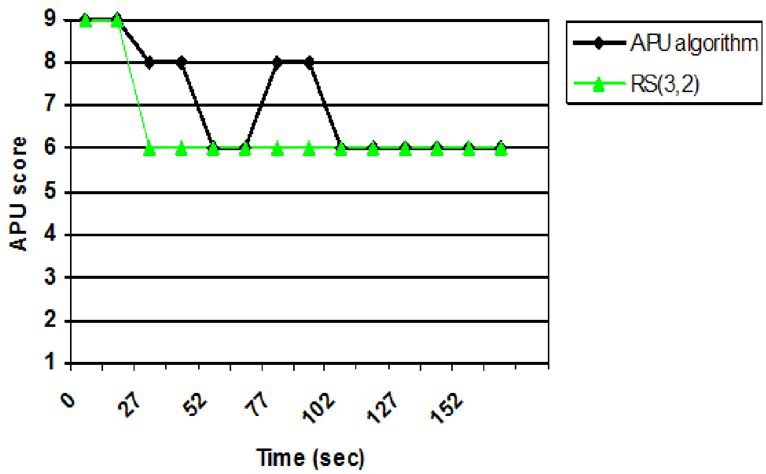


Figure 7-39 Comparative analysis of the fourth test case with RS (3, 2)

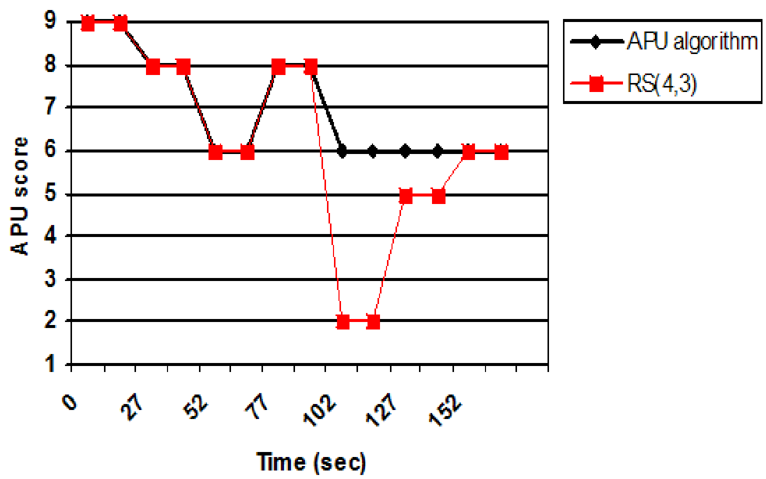


Figure 7-40 Comparative analysis of the fourth test case with RS (4, 3)

It was noticed from the results of the previous four test cases that the APU algorithm gives a higher APU score indicating higher quality compared to other pure RS codes. The previous figures show the response of the APU algorithm to different network conditions when varying the network loss rate and the delay level. Unlike other control algorithms, our algorithm does not focus on using certain RS code to achieve target loss but it takes into consideration the main drawback of using FEC which is the additional delay that may lead to a decrease in the overall call quality perceived by the end user. This can be observed from the previous transitions from RS(4,3) to RS(3,2) or RS(2,1). Theoretically the RS(4,3) code will get the least packet loss, however we found that it does not guarantee the best overall quality due to impacting negatively other QoS factors (i.e.: delay) of the media applications. Our algorithm solved this optimization problem by choosing the optimum RS code according to the current network state represented by 2 factors: First, the MOS score measured based on the E-model taking into account the packet loss, codec impairments and the burst ratio. Second, the one way delay based on the jitter buffer size and the round trip delay.

7.8 Summary

Although various redundant control algorithms have been proposed recently, nearly all of the previous work focused on adjusting the number of packets that carry redundant voice data during the call according to the current QoS and the target QoE. In our research, we found that each RS code performs differently according to the present network conditions. This leads us to develop a new adaptive redundant algorithm that switches between different FEC coding RS coding schemes according to the current QoS in order to enhance the overall QoE compared to the use of pure RS code. We have derived our algorithm (*APU algorithm*) from performing a various subjective testings to understand the drawbacks of using different RS codes on the perceived end user quality. Since, increasing the delay is the main drawback when using higher FEC RS codes, we addressed this carefully by splitting the delay factor from the E-model and studying its impact separately, then we derived our single metric (i.e.: *APU score*) that determine the decision. We found that the decision of choosing the optimum RS code during the call must be based on 4 factors: packet loss rate, burst ratio, codec used and the overall one way delay. At the end of this chapter, we show our simulation results comparing our algorithm to the pure RS codes. We have passed through four test cases using 2 codecs and different network impairments. Although our results show that the *APU algorithm* will outperform the QoE compared to the use of pure RS codes under different QoS factors. Our optimal scheme cannot provide guaranteed call quality given the best effort model for the current internet. However, it puts us one step closer to getting good quality over connections with unstable network conditions.

We are looking forward in the future to merging our algorithm with an adaptive algorithm that will change the percentages of packets having redundant data to save more bandwidth according to the target perceived quality.

Chapter 8

Conclusions and Future Directions

In this chapter, we conclude the thesis by highlighting the main contribution and achievements of our work. We discuss also the future work aimed at further assessing and improving VVoIP call quality.

8.1 Topics of the Thesis

This thesis has presented a study of the Voice and Video over IP (VVoIP) call quality. Moreover, it highlights the challenges in managing VoIP call quality under different network conditions especially as PSTN users are accustomed to experiencing a certain quality; hence they are unwilling to accept a lower quality. The first objective of such a study was to present testing methods for assessing the VVoIP call quality. We have introduced an improved method to monitor the VoIP call quality with higher accuracy results using a simplified E-model. In addition, we have introduced a VVoIP automated testing framework to measure the VVoIP call quality expected at the end user to save time and decrease computation required for processing the audio and video sequences typically required for such testing. The second objective in this thesis was to present new algorithms that have the potential to yield a significant improve in the VoIP call quality.

This chapter summarizes our contributions and outlines potential directions for future work.

8.2 Contributions of the Thesis

In this section we summarize the main challenges addressed in this thesis and our contribution for facing such challenges.

8.2.1 Assessing the VVoIP call quality

From our engagement with our industrial partner IBM, we noticed that the process of assessing VoIP call quality still faces challenges. Such challenges can be expressed in the complexity of the current models provided by the ITU-T (e.g.: E-model), the accuracy of the results from such models when compared with the intrusive testing methods (e.g.: PESQ), the limitation of the usage of such models to ITU codecs only, the time required for testing the VVoIP call quality, and the high computational power required for processing the audio/video sequences.

Through the thesis, we have addressed these challenges. In the fourth chapter, we proposed an improved simplified E-model for measuring the VoIP call quality and we showed how we derived the coefficients used in the model for 4 common used codecs nowadays (G.711, G.723.1, G.726, and G.729A). We demonstrated the results of the derived model by implementing it in a complete monitoring system; our proposed system analyzed the impact of voice quality encoding factors under different network conditions and uses the derived model. The main advantage of our work in this is that we provided an improved simplified model for estimating the VoIP call quality which is less complex than the original E-model and is more accurate than the simplified versions used before. Through this thesis and specifically in Chapter 6, we have studied the behavior of different codecs under varying network conditions, in doing so deriving the impairment factors for non ITU-T codecs so that the E-model can be used to assess the voice call quality for them.

We have also considered the cost in terms of time and computational power required for testing VVoIP calls. In the fifth chapter, we proposed a testing framework that provides an online estimate for both audio and video call quality on network paths without end-user involvement and without requiring any audio/video sequences or network traces. In our proposed framework, we presented a tool that emulates the audio and video traffic of IP calls in order to estimate the end user perception call quality. We have seen acceptable results compared to the quality of an interactive real time voice and video calls using the most commonly used industry standards for objective voice and video quality testing: PESQ and PSNR respectively.

8.2.2 Improving the VoIP call quality

Users are accustomed to the quality of service (QoS) they have enjoyed for years with the public switched telephone network (PSTN). Consequently, the VoIP applications should meet the high expectations of the end users. However, Voice over IP is based on IP networks that may not provide perfect network conditions. Thus, the majority of work reported in this area has focused on improving the call quality by enhancing existing algorithms or introducing new techniques to sustain high perceived voice call quality under different network conditions. Through this thesis, we have focused on introducing new algorithms to improve the VoIP call quality.

In Chapter 6, we presented and evaluated an algorithm that performs in-call selection of the most appropriate audio codec given prevailing conditions on the network path between the end-points of a voice call. Unlike any previous work in this area, we have studied the drawbacks of codec switching from the end user perception point of view; our algorithm is the first that seeks to minimize this impact. Our results, presented in Chapter 6, show that in many typical network scenarios, switching codecs mid-call results in better call quality compared to the use of an initial fixed codec throughout the call.

We presented the second main contribution in improving the call quality in the seventh chapter. In the first part of this chapter, we have introduced several loss repair methods used in VoIP applications. We highlighted the most used method, studied it and introduced the drawbacks when using it. In the second part of the chapter, we show our subjective testing results indicating performance from the human perception point of view. Based on our findings, we proposed a new adaptive FEC mechanism (APU algorithm) for voice calls based on the generated codewords from a Reed-Solomon (RS) encoder. Our mechanism chooses the optimum RS code from three different codes to improve the conversational call quality. At the end of the chapter, we show that our algorithm improved the QoE at the end user compared to the used fixed RS codes under variable QoS factors.

8.3 Possible Future Directions

The central focus of the work in this thesis and its main contributions is shown in assessing the VVoIP call quality and improving the VoIP call quality. Despite our explorations being focused on assessing the audio call quality under different network conditions in different scenarios, the algorithms presented in this thesis are potentially generic enough to be applied in any VoIP application. Our research has the potential for different future directions, including possible improvements to the algorithms presented, and also to apply these algorithms in different VVoIP applications. Potential future work includes:

- The current models available for monitoring the VoIP call quality need some improvement to provide higher accuracy in measuring the VoIP call quality. The current methods that provide accurate algorithms for assessing the VoIP call quality are intrusive testing methods; such methods require recording from both sender and receiver sides (Refer to Chapter 3). Consequently, such algorithms cannot be applied in real time and cannot be used in monitoring purposes. Such algorithms include the PESQ and recently POLQA (Perceptual Objective Listening Quality Assessment). Our work in Chapter 4 introduced a simplified E-model corrected with the PESQ score to provide a simplified and more accurate version to monitor the call quality. We have proven our concept by deriving such correction coefficients for four commonly used codecs. We believe that such work should be extended for more Narrowband and Wideband codecs. However, PESQ is not designed to work with wideband codecs, so it might be necessary to use a recently developed extension, such as POLQA.

- The process of testing the Voice and Video call quality is time consuming and computationally intensive, due to the requirements of processing audio/video sequences first. Thus, we have proposed a complete testing framework in Chapter 5 to do it in an automated way. We have supported nearly all of the Narrowband audio codecs. In the future, our framework can be extended to support the WB audio codecs. Video testing could be extended also to support more codecs, including but not limited to: VP7, VP8 and H263. The proposed work in Chapter 5 opens the directions in research to work on simulating the audio and video conference calls and measuring the QoE expected at each user.

- Recently and due to the increase demand of the communication between more than one party in different locations, a conferencing VoIP system was introduced and became more mature. We have noticed that there is lack of the work carried in the conferencing system for measuring the call quality for both video and audio using non-intrusive testing methods. In the future, we are intending to study the QoE of the VoIP conferencing systems and provide an analysis of the accuracy of the E-model for multi-party VoIP sessions when all audio is processed by a centralized focus node.

- Despite the fact that our codec switching algorithm (Chapter 6) showed an improve in the call quality, there is still a possible potential extension as a future work by merging it with a path switching algorithm such as that proposed by Shu Tao et al. in [90]. Such an algorithm could be executed on a gateway; the gateway will find the optimal combination of path and codec to achieve maximum call quality that can be attained under certain network conditions. The gateway will provide the codec information to the sender side which will switch to this codec while it will manage the path switching task.

- VoIP call quality still needs much research to improve it by providing different algorithms that will improve the VoIP call quality under different network conditions. In Chapter 6 and 7, we provide new algorithms that improve the call quality but still they need more improvement in the future. In Chapter 7, we introduced the APU algorithm which is considered a new adaptive FEC mechanism for VoIP calls based on the generated codewords from a Reed-Solomon (RS) encoder. Our algorithm switches between the optimum RS code from three different codes to account for the variation of the network conditions including packet loss and delay. Our proposed algorithm assumes that there is sufficient bandwidth always, which is not the case in reality. In the future, further work is needed to merge our derived algorithm with an adaptive algorithm that will change the percentages of packets having redundant data to save more bandwidth according to the target perceived quality. Moreover, in our algorithm we switch between 3 RS codes. Future work might include more RS schemes that might be more optimal in different packet loss rates.

8.4 Closing Remarks

We believe the work presented in this thesis has developed new techniques for assessing and improving the VVoIP call quality. The central focus of the work is on providing new methods for improving VoIP call quality, and in addition we believe that the process of assessing the VVoIP call quality still needs research attention because of its importance in the industry. We hope that this work will act as a starting point for other researchers to continue investigations on the problems we addressed, trying to further improve the algorithms and methods presented in this thesis.

Bibliography

[1] Junsheng Zhang and Xiaohua Sun "The VoIP phone QoS protection in the wide area network", Computer learning, 2006 no. 6, pp. 17-18.

[2] RFC 768, J. Postel, UDP: User Datagram Protocol, August 1980.

[3] RFC 1889, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, January 1996.

[4] RFC 3550, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, July 2003.

[5] RFC 3551, H. Schulzrinne, S. Casner, RTP Profile for Audio and Video Conferences with minimal control, July 2003.

[6] RFC 3261, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: Session Initiation Protocol, June 2002.

[7] RFC 4566, M. Handley, V. Jacobson, C. Perkins, SDP: Session Description Protocol, july 2006

[8] http://www.telecomspace.com/vop-sip.html

[9] Patrick Le Callet, Sebastian Möller and Andrew Perkis, (Eds.). Qualinet White Paper on Definitions of Quality of Experience (2012). European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland, Version 1.1, June 3, 2012. Online available: http://www.qualinet.eu.

[10] ITU-T Rec. P.10, "Vocabulary for performance and quality of service, Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100", Int. Telecomm. Union, Geneva, 2008.

[11] ITU-T Rec. E.800, "Definitions of terms related to quality of service'', 2008.

[12] A. Khan, L. Sun, E. Jammeh, and E. Ifeachor, "Quality of experience-driven adaptation scheme for video applications over wireless networks, *In Special Issue on Video Communication over Wireless Networks*, 2009, pp. 1337-1347.

[13] H. A. Tran and A. Mellouk, "QoE model driven for network services," in *Wired/Wireless Internet Communications*, E. Osipov, A. Kassler, T. M. Bohnert, and X. Masip-Bruin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6074, pp. 264–277. Online Available: http://www.springerlink.com/content/7159q510q378m234/

[14] G. Van der Auwera and M. Reisslein, "Implications of smoothing on sta- tistical multiplexing of H.264/AVC and SVC video streams," *Broadcasting, IEEE Transactions on*, vol. 55, no. 3, sept. 2009, pp. 541-558.

[15] T. N. Minhas and M. Fiedler, “Impact of disturbance locations on video quality of experience,” in *Quality of Experience for Multimedia Content Sharing, EuroITV2011*, June 2011.

[16] T. N. Minhas, O. G. Lagunas, P. Arlos, and M. Fiedler, “Mobile video sensi- tivity to packet loss and packet delay variation in terms of QoE. (accepted),” in *Packet Video Workshop (PV), 2012 19th International*, May 2012.

[17] C.-H. Lin, C.-H. Ke, C.-K. Shieh, and N. Chilamkurti, “The packet loss effect on mpeg video transmission in wireless networks,” in *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, vol. 1, april 2006, pp. 565–572.

[18] D. Loguinov and H. Radha, “End-to-end internet video traffic dynamics: Statistical study and analysis,” *IEEE INFOCOM*, 2002, pp. 723–732.

[19] M. Li, M. Claypool, and R. Kinicki, “MediaPlayerTM versus RealPlayerTM: a comparison of network turbulence,” in *Proceed-ings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, ser. IMW ’02. New York, NY, USA: ACM, 2002, pp. 131–136. [Online]. Available: http://doi.acm.org/10.1145/637201.637221.

[20] P. Pragtong, K. Ahmed, and T. Erke. Analysis of speech data rate in voice over IP conversation. TENCON 2004. 2004 IEEE Region 10 Conference. 2004. Volume A, pp. 143–146.

[21] ITU-T Rec. P.861, “Objective quality measurement of telephone-band (300-3400 Hz) speech codecs,” 1998.

[22] ITU-T Rec. P.862, “PESQ an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs”, 2001

[23] ITU-T Rec. P.862.2, “Wideband extension to Recommendation P.862 for the assessment of wide band telephone networks and speech codecs”,2001

[24] ITU-T Rec. G.107, “The E-Model, a computational model for use in transmission planning,” 1998.

[25] ITU-T Rec. G.1070: “Opinion model for video-telephony applications”, April 2007

[26] ITU-T Rec. P.563: “Single-ended method for objective speech quality assessment in narrow-band telephony applications”, May 2004.

[27] W. Rix, and M.P. Hollier, “The perceptual analysis measurement system for robust end-to-end speech quality assessment”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 3, 2000, pp. 1515–1518.

[28] M. Keyhl, C. Schmidmer, and H. Wachter, “A combined measurement tool for the objective, perceptual based evaluation of compressed speech and audio signals”, Audio Engineering Society (AES) Convention, 4931 (M3), May 1999, pp. 4931-4935.

[29] www.**pevq**.org/

[30] A.D. Clark, “Modeling the effects of burst packet loss and recency on subjective voice quality”, 2nd IP-Telephony Workshop, Columbia University, New York, April 2001.

[31] R.G. Cole, and J.H. Rosenbluth, “Voice over IP performance monitoring,” ACM SIGCOMM Computer Communication Review, vol. 31, no. 2, April 2001, pp. 9-24.

[32] C.M. Chernick, S.L. Leigh, K.L. Mills, and R. Toense, “Testing the ability of speech recognizers to measure the effectiveness of encoding algorithms for digital speech transmission”, IEEE International Military Communications Conference (MILCOM), 1999, pp. 1468-1472.

[33] W. Jiang, and H. Schulzrinne, “Speech recognition performance as an effective perceived quality predictor”, IEEE International Workshop on Quality of Service, 2002, pp. 269-275.

[34] “3SQMTM advanced non-intrusive voice quality testing”, White Paper, Opticom GmbH, Germany, 2003.

[35] S. Mohamed, F. Cervantes-Perez, and H. Afifi, “Integrating networks measurements and speech quality subjective scores for control purposes”, Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2001, pp. 641-649.

[36] L.Sun, and E.C. Ifeachor, “Perceived speech quality prediction for voice over IP-based networks,” IEEE International Conference on Communications (ICC), vol. 4, 2002, pp. 2573-2577.

[37] A. Tarraf, and M. Meyers, “Neural network-based voice quality measurement technique”, IEEE International Symposium on Computers and Communications, 1999, pp. 375-381.

[38] Kazuhisa Yamagishi and Takanori Hayashi, “Opinion Model for Estimating Video Quality of Videophone Services”, IEEE Global Telecommunications Conference, Nov. 2, 2006, pp. 1-5.

[39] Takanori Hayashi, Kazuhisa Yamagishi, Toshiko Tominaga, and Akira Takahashi, “Multimedia Quality Integration Function for Videophone Services”, IEEE Global Telecommunications Conference, Nov. 2007, pp. 26-30.

[40] John Q. Walker, “Assessing VoIP Call Quality Using the E-model”, NetIQ Corporation.

[41] ITU-T Rec. P.800, “Methods for subjective determination of transmission quality”, Geneva, 08/1996.

[42] Chunlei Jiang and Peng Huang, “Research of Monitoring VoIP Voice QoS”, International Conference on Internet Computing and Information Services, 2011, pp. 499-502.

[43] C. Hoene, H. Karl, and A. Wolisz, “A perceptual quality model for adaptive VoIP applications”, Int. Symp. Performance Evaluation of Computer and Telecommunication Systems(SPECTS’04), SanJose, CA, pp. 299-316.

[44] L. Sun, “Speech Quality Prediction for voice Over Internet Protocol Networks”, Ph.D dissertation, Univ. Plymouth, UK., Jan 2004.

[45] ITU-T Rec. G.113, “Transmission impairments due to speech processing”, 2001.

[46] ITU-T Rec. P.833, “Methodology for derivation of equipment impairment factors from subjective listening-only tests”, 2001.

[47] R.G.Cole and J.Rosenbluth, “Voice over IP performance monitoring”, ACM comput. Commun. Rev., vol. 31, no. 2 , April 2001, pp. 9-24.

[48] S.Pracht and D.Hardman: "Voice Quality in Converging Telephony and IP Networks, Agilent Technologies", White Paper, http://literature.agilent.com/litweb/pdf/5980-0989E.pdf

[49] Marta Carbona and Luigi Rizzo, "Dummynet Revisited", ACM SIGCOMM Computer Communication ReviewVolume 40 Issue 2, April 2010, pp. 12-20.

[50] S. Agrawal, J. Ramamirtham, and R. Rastogi. Design of active and passive probes for VoIP service quality monitoring. In Proc. 12th International Telecommnications Network Strategy and Planning Symposium (NETWORKS 2006), 2006, pp. 1-6.

[51] I. Aktas, F. Schmidt, E. Weingrtner, C.-J. Schnelke, and K. Wehrle, An adaptive codec switching scheme for SIP-Based VoIP. In S. Andreev, S. Balandin, and Y. Koucheryavy, editors, Internet of Things, Smart Spaces, and Next Generation Networking, number 7469 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, Jan. 2012, pp. 347-358.

[52] Robustelli, L., Loreto, S., Fresa, A., Longo, M., & Spinelli, D., "Prototype of an Adaptive Voice Coder for IP Telephony", International Conference on Software, Telecommunications and Computer Networks–SoftCom 2003, pp. 7-10.

[53] N. Costa and M. S. Nunes. Adaptive Quality of Service in Voice over IP Communications. In Proc. 5th International Conference on Networking and Services, ICNS '09, Washington, DC, USA, 2009, IEEE Computer Society, pp. 1924-1929.

[54] M. Walterman, B. Lewcio, P. Vidales, and S. Moller, "A Technique for Seamless VoIP-codec Switching in Next Generation Networks", In Proc. 2008 IEEE International Conference on Communications (ICC 2008), 2008, pp. 1772-1776.

[55] M. Sulovic, D. Raca, M. Hadzialic, and N. Hadziahmetovic, "Dynamic codec selection algorithm for VoIP", In Proc. 6th International Conference on Digital Telecommunications (ICDT 2011), 2011, pp. 74-79.

[56] L. Sun and E. C. Ifeachor, "Voice quality prediction models and their application in VoIP networks", IEEE Transactions on Multimedia, 2006, pp. 809- 820.

[57] Jitsi. http://www.jitsi.org, 2012.

[58] P. T. Brady, "Effects of Transmission Delay on Conversational Behavior on Echo-Free Telephone Circuits," *Bell Sys. Tech. J.*, vol. 50, Jan. 1971, pp. 115–34.

[59] T-k. Chua and D-C. Pheanis, "QoS Evaluation of Sender-Based Loss-Recovery Techniques for VoIP", IEEE Network Journal, Issue 6, Dec. 2006, pp. 14-23.

[60] C. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," *IEEE Network*, vol. 12, no. 5, Sept.–Oct. 1998, pp. 40–48.

[61] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring The Internet*, 3rd Ed. (Addison Wesley, 2004).

[62] T. K. Chua and D. C. Pheanis, “Comparative Analysis of Audio Coders and Packet-Loss Recovery”, *Proc. Commun. Systems and Applications*, CSA 2005, Banff, Alberta, Canada, July 2005, pp. 176–81.

[63] W. Jiang and H. Schulzrinne, “Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss,” *Proc. 12th Int’l. Wksp. Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 2002*, Miami, FL, May 2002, pp. 73–81.

[64] H. P. Sze, S. C. Liew, and Y. B. Lee, “A Packet-Loss- Recovery Scheme for Continuous-Media Streaming Over the Internet,” *IEEE Commun. Letters*, vol. 5, no. 3, Mar. 2001, pp. 116–18.

[65] J. Nonnenmacher, E. Biersack, and D. Towsley, “Parity-Based Loss Recovery for Reliable Multicast Transmission”, *Proc. ACM Special Interest Group on Data Communications*, *SIGCOMM 1997*, Cannes, France, Sept. 1997, pp. 289–300.

[66] T-Y.Huang, P-Huang, K-T.Chen and P-J.Wang,”Could Skype be more Satisfying?”, Journal IEEE Network, The Magazine of Global Internetworking, Volume 24 Issue 2 , March/April 2012, pp.42-48.

[67] An Introduction to Reed-Solomon codes: principle, architecture and implementation. (http://www.4i2i.com/reed_solomon_codes.htm).

[68] T-Y.Huang, P-Huang and P-J.Wang, “Tuning Skype’s Redundancy Control Algorithm for user satisfaction”, INFOCOM 2009, IEEE, April, pp.1179-1187.

[69] J-C. Bolot and A. Garcia, “Control Mechanisms for Packet Audio in the Internet,” Proceedings of IEEE INFOCOM, March 1996, pp. 232 – 239.

[70] C-Padhye, K-J.Christensen and W.Moreno, “A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications”, 2000, pp. 307-313.

[71] W-Jiang and H.Schulzrinne, “Comparisons of EC and codec robustness on VoIP quality and bandwidth efficiency”, world scientific, june 5, 2002.

[72] M.Al-Rousan, A.Nawasrah, “Adaptive FEC technique for multimedia applications over the internet”, Journal of emerging technologies in web intelligence, vol. 4, No2, May 2012, pp. 142-147.

[73] ITU-T Rec. G.114, “Delay variation on unshared access lines”, 2009.

[74] ITU-T Rec. P.920, “Interactive test methods for audiovisual communications 2000”.

[75] Iperf’s website, www.iperf.sourceforge.net .

[76] J-C.Bolot, S-F.Parisis and D.Towsley, “Adaptive FEC-Based Error Control for Internet Telephony”, INFOCOM 1999, pp. 1453-1460

[77] ITU-T Recommendation, Bt.500-10: Methodology for the subjective assessment of quality for television pictures”, 2002.

[78] H. Kim and S. Choi, “Traffic quality monitoring system between different network providers,” in Proc. 12th International Conference onAdvanced Communication Technology (ICACT 2010), vol. 2., 2010, pp. 1153–1158.

[79] J. da Silva and R. Lins, "Analyzing the QoS of VoIP on SIP in java", in Proc. 2006 InternationalTelecommunications Symposium. IEEE, 2006, pp. 576–581.

[80] L. Carvalho, E. Mota, R. Aguiar, A. Lima, and J. de Souza, "An E-model implementation for speech quality evaluation in VoIP systems", in Proc. 10th IEEE Symposium on Computers and Communications (ISCC 2005). IEEE, 2005, pp. 933–938.

[81] Y. Gong, F. Yang, L. Huang, and S. Su, "Model-based approach to measuring quality of experience," in 1st International Conference on Emerging Network Intelligence. IEEE, 2009, pp. 29–32.

[82] P. Hershey, J. Pitts, and R. Ogilvie, "Monitoring real-time applications events in net-centric enterprise systems to ensure high quality of experience", in Proc. 2009 IEEE Military Communications Conference (MILCOM 2009). IEEE, 2009, pp. 1–7.

[83] P. Calyam, E. Ekici, C. Lee, M. Haffner, and N. Howes, "A GAP model based framework for online VVoIP QoE measurement", Journal of Communications and Networks, vol. 9, no. 4, 2007, pp. 446-456.

[84] A.D.Clark, "Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality", Columbia University Telephony Workshop, 2001

[85] R.G.Cole and J.H.Rosenbluth, "Voice over IP Performance Monitoring", ACM SIGCOMM, Vol. 31, Issue 2, 2001, pp. 9-24.

[86] L.C.G.Lustosa, L.S.G.Carvalho, P.H.A. Rodrigues, and E.S.Mota, "E-Model Utilization For speech quality Evaluation Over VoIP Based Communication Systems", SBRC, 2004

[87] RFC 792, J. Postel, ICMP: Internet Control Message Protocol, September 1981.

[88] "IBM,SUT,"2012.[Online].Available:http://www01.ibm.com/software/lotus/products/sametime/unifie dtelephony.

[89] J. Ohm, Multimedia communication technology: Representation, transmission and identification of multimedia signals. Springer Verlag, 2004.

[90] K. Xu, A. Estepa, T.F.L. Gao, R.Guerin, J.Kurose, D. Towsley, Z-L.Chang, "Improving VoIP quality through Path Switching", INFOCOM 2005, vol. 4, pp. 2268 – 2278.

[91] ITU-T Rec. G.723: "Extensions of Recommendation G.721 adaptive differential pulse code modulation to 24 and 40 kbit/s for digital circuit multiplication equipment application", November 1988.

[92] ITU-T Rec. G.729: " Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)", June 2012.

[93] ITU-T Rec. G.711: "Pulse code modulation (PCM) of voice frequencies", Novemeber 1988.

[94] RFC 6716, JM.Valin, k.Vos, T. Terriberry, Definition of the Opus Audio codecs, September 2012.

[95] RFC 5574, G.Herlein, J.Valin, A.Heggestad, A.Moizard, RTP pacyload format for the speex codec, June 2009.

[96] RFC 3951, S.Anderson, A.Duric, H.Astrom, R.Hagen, W.Kleijn, J.Linden, Internet Low Bit Rate codec, December 2004

[97] http://en.wikipedia.org/wiki/Video_codec

[98] RFC39547, Negotiation of NAT-Travsersal in the IKE, T.Kivinen, B.Swander, A.Huttenen, V.Volpe, Janurary 2005.

[99] H.Yoshino, N. Kitawaki, "Perceptual QoS assessment technolgies for VoIP", IEEE Communications Magazine, July 2004., pp. 28-34.

[100] http://www.cisco.com/image/gif/paws/5125/delay-details.pdf

[101] http://www.packetizer.com/ipmc/diagnostics/papers/TelchemyVoiceQualityMeasurement.pdf

[102] E.Gilbert, "Capacity of a Burst-Noise Channel", The Bell System Technical Journal, vol. 39, 1960, pp.1253-1265.

[103] http://voip.about.com/od/voipbasics/a/MOS.htm

[104] http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml

[105] http://www.polycom.com/solutions/solutions-by-business-uc-environment/solutions-for-ibm.html

[106] R. Barbosa, C. Kamienski, D. Mariz, A. Callado, S. Fernandes, and D. Sadok, "Performance evaluation of P2P VoIP application", ACM NOSSDAV '07, June 2007.

[107] Ted Wallingfors, (2005). Switching to VoIP. Retrieved from http://books.google.com

Appendix A

Screenshots of VVoIP QoE measurment framework

In this appendix, we show some of the screenshots of our framework proposed in Chapter 5. First, we show the network emulation screenshot that is used to emulated different network conditions using Dummynet. Second, we show an example of the audio testing screenshot after being run on certain network conditions. Finally, we show an example of the video testing screenshot on certain network conditions.

A.1 Network Emulation Screenshot

Figure A-1 shows the screen shot of the network emulation. The user can emulate different network conditions, the line bandwidth can be changed in the Kbps unit, one way delay can be increased in milliseconds, the queue size can be changed, the probability of losing packet can be increased, Finally the burst ratio defined as “the ratio of the average length of consecutive losses under bursty loss to that under random loss” can be configured by the user.

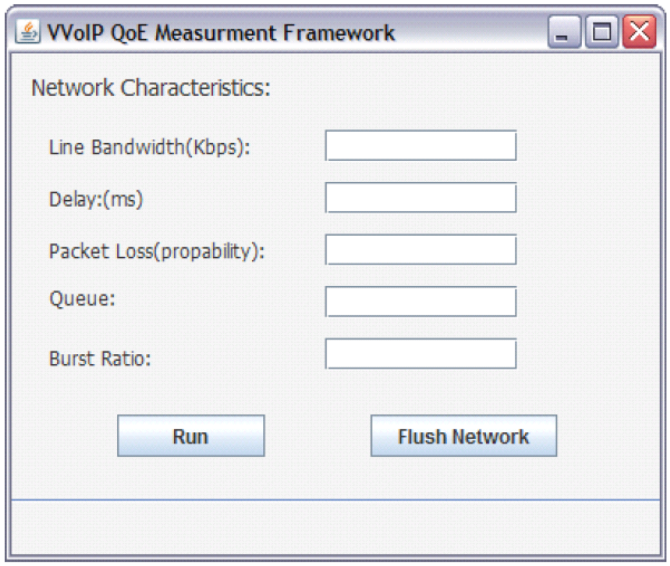


Figure A-1 Network Emulation Screenshot

A.2 Audio Screenshot

Figure A-2 shows the screenshot of the audio testing in our proposed framework, the destination IP and port no are required to be entered by the user. Moreover, one of the following supported codecs should be selected: G.711, G.723.1 5.3k, G.723.1 6.4k, G.726, G.729, G.729 A, GSM FR, SILK, ILBC and SPEEX. The number of frames per packet could be changed for testing purposes. After running the test, the QoS factors of the current network conditions according to the codec used will be measured as shown in the below screenshot. Such factors will be mapped to a single metric MOS score and user satisfaction level will be given indicating the degree of satisfaction at the end user with a progress bar representation.

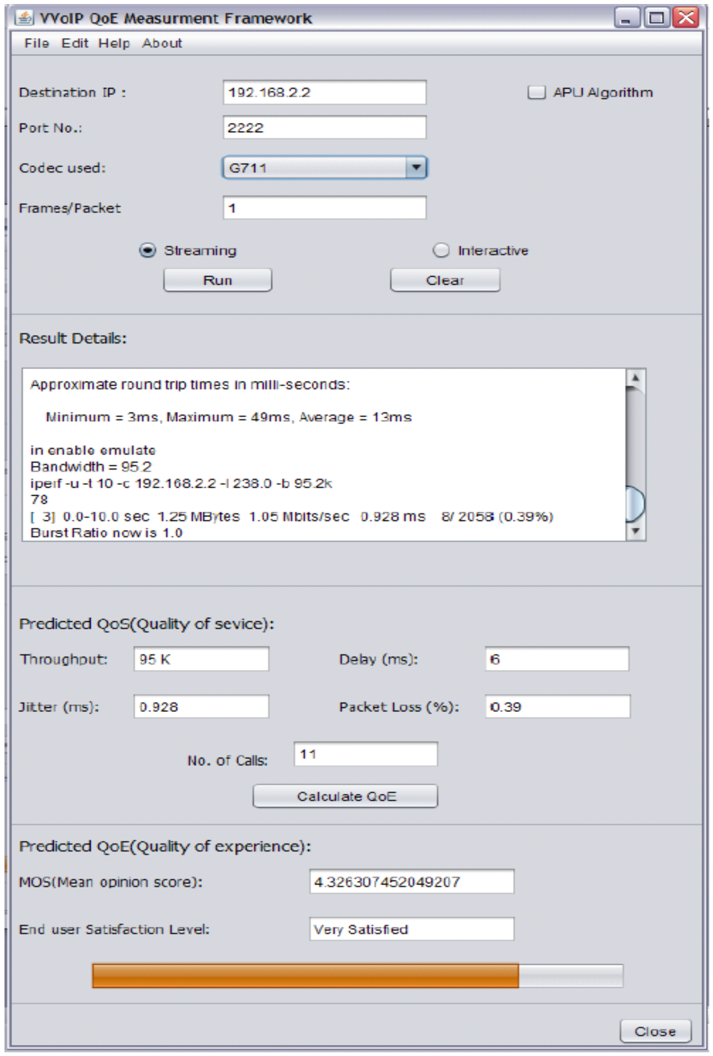


Figure A-2 Audio Testing Screenshot

A.3 Video Screenshot

Figure A-3 shows the screenshot of the video testing in our proposed framework, the destination IP and port no are required as an input by the user. H.264 codec is used in our framework with two different modes. First, the video format is QQVGA representing 2.1 inch display size; second, the video format is QVGA representing 4.2 inch display size. The number of frames per second can be changed by the user from 1-30 fps. The user should select one of the required call quality either HD or low or high call quality which will reflect the required bandwidth. After running the test, the QoS factors of the current network conditions are measured as shown in the below screenshot. Such factors will be mapped to a single metric MOS score and user satisfaction level will be given indicating the degree of satisfaction at the end user with a progress bar representation.

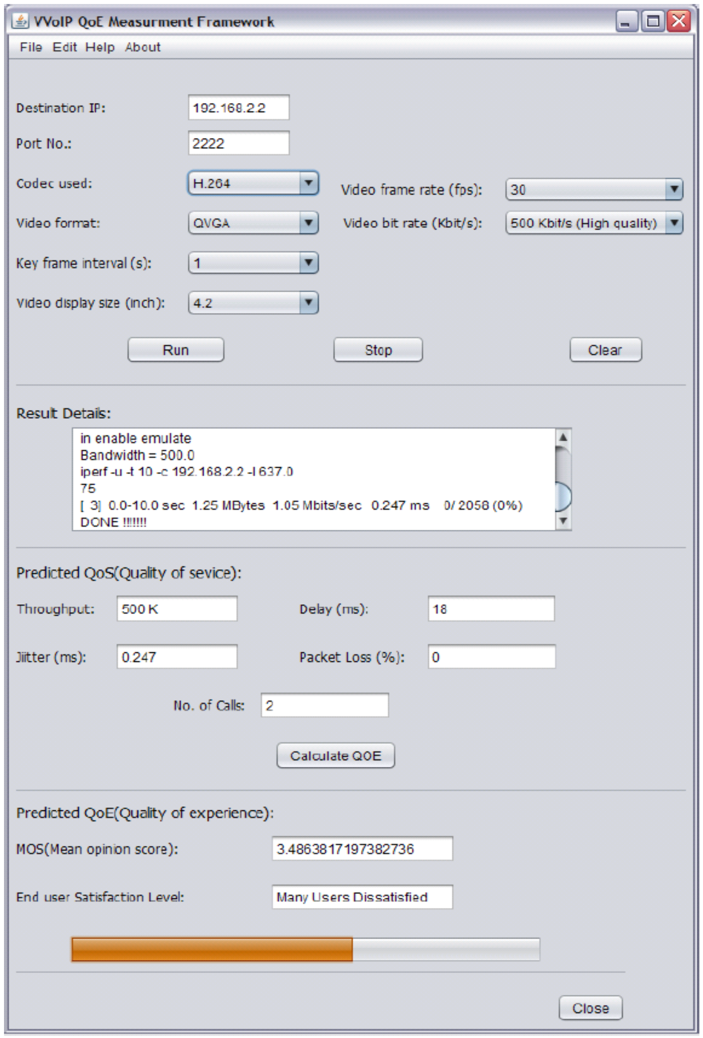


Figure A-3 Video Testing Screenshot