Assessment and Improvement of the Quality of Voice-over-IP Communications

Mohamed Adel Mahmoud Mohamed, BSc

Department of Computing, Mathematics and Physics

Waterford Institute of Technology

Thesis submitted in partial fulfilment of the requirements for the award of

Masters by Research

Supervisor: Dr. Brendan Jennings

August 2013

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Masters by Research is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Signed:...
Student ID: 20055419
Date: August 2013

To my parents and my sister for their love, support, and belief in me.

Acknowledgements

I would like to thank my supervisor, Dr. Brendan Jennings, for his dedication, support, and attention to detail. I would like to especially acknowledge his unique method of identifying most challenging and important problems and offering guidance to address these problems efficiently. Also, Dr. David Malone from NUI Maynooth for his precious advice and cooperation.

I would like to thank my supervisor in IBM, Jonathan Dunne, for his valuable suggestions and support. I would like to thank my team in IBM, Haytham Assem, Himanshu Dadheech, and Yi Han for their dedication to team values and for their continuous effort towards the best for the whole team.

I would like to thank my parents, and my sister, who provided me with continuous support throughout my studies. Also, all my friends in Egypt, especially Ahmed Moeness, Mahmoud Ashraf, Kareem Nagy, Mohamed Tawakol, Ahmed Omar, Mohamed Adel Nour, and Ayman Ahmed.

I also would like to thank the members of our research group in TSSG for their insightful comments. This research was supported by Science Foundation Ireland (SFI) grants 07/SK/I1216a and 08/SRC/I1403 and by IBM Software Lab in Dublin.

Publications

- Haytham Assem, Mohamed Adel, Brendan Jennings, David Malone, Jonathan Dunne, and Pat O'Sullivan. **A Generic Algorithm for Mid-call Audio Codec Switching.**, pages 1276-1281. In Proc. 1st IFIP/IEEE International Workshop on QoE Centric Management (QCMan 2013) . IEEE, 2013.

- Haytham Assem, Mohamed Adel, Brendan Jennings, David Malone, Jonathan Dunne, and Pat O'Sullivan. **Online Estimation of VVoIP Quality-of-Experience via Network Emulation**. In Proc. 24th IET Irish Signals and Systems Conference (ISSC 2013) , 2013.

- Mohamed Adel, Haytham Assem, Brendan Jennings, David Malone, Jonathan Dunne, and Pat O'Sullivan. **Improved E-model for Monitoring Quality of Multi-Party VoIP communications.** 2nd IEEE Workshop on Quality of Experience for Multimedia Communications (QoEMC2013) - Submitted.

Abstract

This dissertation addresses real-time communications using Voice-Over-IP (VoIP) technology, which is nowadays widely used by enterprises and individual users. The focus is on the assessment of the Quality-of-Experience (QoE) from the end-user's perspective and the development of algorithms and techniques to improve the overall QoE.

One of the main contributions is a generic testing tool than can be used for any Voice and Video-Over-IP (VVoIP) application in any environment. The tool employs network emulation techniques to provide estimates for the perceived voice and video quality on different network paths. Importantly, the tool operates without the need for use of traditional quality assessment techniques which are known to be time and resources consuming as they require end-user involvement to collect audio/video sequences and network traces. Our tool emulates the audio and video traffic and employs the E-model and video quality opinion model to estimate audio and video quality respectively, with the advantage of emulating various network conditions to run experiments in multiple scenarios.

Secondly, we present a generic adaptive algorithm for switching audio codecs throughout an ongoing call. Codecs are known to have different behaviours under various network conditions; we study the behaviour of five of the most commonly used codecs (including some non-ITU codecs), deriving models for them so that E-model can be used to assess the Quality-of-Experience. Furthermore, we analyse the negative of codec switching from the end user perspective, so that this impact can be minimized as much as possible. We describe results of tests of the algorithm under different network scenarios; these results suggest that the algorithm can deliver better Quality-of-Experience than would have been achieved by employing one codec only during the call.

Lastly, we study different multi-party conferencing architectures with a focus on the centralised architecture which is most commonly used. We analyse the degradation to the quality that results from the need of passing of every packet in the conference through the focal node, and to further decode and then encode those packets to be sent again to their intended destinations. An extended E-model is presented to be used with multi-party calls—we introduce a correction formula to three of the most used codecs so that E-model can be valid when used to estimate the quality of experience of multi-party conferencing calls.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Nowadays, establishing means of communications between people in distant locations is crucial and is gaining more importance. Voice communications is one of the most valuable means of communications, when it is not possible to perform ordinary face-to-face communications. There have been traditional approaches to meet this growing need by implementing specialized networks that are designed to transmit voice signals only between certain equipments, such as landlines phones.

The rapid growth of IP networks around the globe has provided a means of communications to a wider scope of users with non-expensive costs. Voice over IP (VoIP) technology has evolved rapidly, providing real-time voice and video communications between users through the existing IP network, in a manner that closely approximates face-to-face interactions. Voice over Internet Protocol (VoIP) applications have gained wide acceptance by general Internet users and are increasingly important in the enterprise communications sector. Furthermore, VoIP provides a seamless way of integration between the traditional telephony services as PSTN, hand-in-hand with the modern collaborative services, such as instant messaging, desktop sharing and voice mail, which have made it the most convenient technology for the present and the future. However, achieving voice quality levels for VoIP remains a significant challenge, as IP networks typically do not guarantee delay, packet loss, jitter and bandwidth levels.

In IP networks, it is a main challenge to achieve business and technical requirements as knows as service level agreements (SLAs) for audio and video quality in VoIP applications. Consequently, the need has arisen to accurately measure and quantify network and application impairment factors that affects the overall perceived Quality-of-Experience (QoE) in both peer-to-peer, and multi-party VoIP communication models. Providing techniques and frameworks for monitoring and estimating the VoIP call quality in prevailing network conditions is essential to mitigating issues that can significantly reduce the QoE as experienced by end users.

1.2 Research Questions

- Can we use open source tools to construct a framework suited for comprehensive and repeatable scenarios of QoE of VVoIP applications?

  Assessing QoE of VVoIP requires performing extensive sets of experiments, due to the fact that several factors can influence the overall perceived QoE at end-users. There are many open source tools available for different purposes, such as measuring network performance, and emulating network parameters. Putting this tools together in a reliable framework to essential in order to estimate QoE of VVoIP accurately and efficiently.

- Can network emulation provide an appropriate means of assessing QoE for VVoIP applications?

  VoIP codecs have different characteristics. Their performance is dependent on multiple factors. Emulating the behaviour of audio and video codecs is a novel approach in order to estimate QoE of VoIP calls, instead of using current assessment methodologies which are known to be time consuming and computationally intensive.

- Under what circumstances can in-call switching of audio codes improve user-perceived QoE?

  Audio codecs show different performance levels under varying network conditions. Codec switching can be a method to improve QoE, by consistently switching to the codec that provides the best QoE at the present network conditions. However, codec switching can have some negative impact on the overall QoE.

- How can we assess the QoE of multi-party VoIP conference calls realised using a centralised architecture?

  Multi-party VoIP communication sessions are more complicated than peer-to-peer sessions. It can be implemented using different models. The current assessment methods of QoE were designed for peer-to-peer sessions, which are not necessarily valid for multi-party conferencing sessions.

1.3 Contributions of Research

The first contribution of this dissertation, is the identification of the latest methodologies and tools that provides objective measurements for conversational quality perceived by VoIP calls. Some of them have been employed in characterizing network conditions offline, others can be used to obtain online measures during the run-time of VoIP calls. Furthermore, we clearly define the advantages and disadvantages of each of these measures in order to maximize the usage of them to obtain accurate results which reflects the performance of VoIP applications during various network conditions.

The second contribution is a generic testing tool that can be used for evaluating VoIP quality in multiple network paths. The uniqueness of the developed tool, comes from the fact that it does not require establishing the excessive steps of other tools which often involves establishing calls, recording from participants' sides, manipulating network conditions,.etc. The tool operates by emulating the behaviour of audio codecs, then employing standard objective audio and video measurement models to estimate the QoE.

The third contribution, is the algorithm developed to dynamically switch between a set of multiple codecs during VoIP audio calls. Audio codecs are known to have different performance that varies according to several factors. Our algorithm leverage the difference between codecs in a positive way, in order to attain the best possible QoE during the call, while considering degradations in quality or irritations that might happen to end-users as a result of excessive switching.

The last contribution is providing a correction for the standard metric for estimating audio QoE (The E-Model) for multi-party VoIP conference calls. The E-model is typically used effectively with peer-to-peer calls. In multi-party, the flow of the traffic depends on a focal point of the conference. This focus is responsible for receiving audio data packets from the involved parties, decoding them, further encode them again and forward them to their intended recipients. Extra processing done on the focus cause unexpected degradations in the overall quality perceived by the conference participants, which makes the standard E-model not suitable for estimating the QoE of multi-party calls.

1.4 Outline of the Thesis

The thesis is organized as follows, in Chapter 2 we provide a background of VoIP technology, in addition to the state of the art network protocols used by modern VoIP applications, the latest methodologies to assess audio and video quality, and various techniques to improve the quality-of-experience of VoIP calls, such as codec switching. The architecture of softphone applications is explained in Chapter 3 with a focus on two widely used softphones; IBM SUT, and Jitsi. Moreover, we describe in detail the testbed built to carry out a comprehensive set of experiments for evaluating quality of VoIP applications.

Chapter 4 describes a novel tool for testing QoE via network emulation without requiring the typical steps of other testing methodologies which were known to be costly and time consuming. Furthermore, we present a comparison between our tool and other objective measures.

In chapter 5, we introduce the concept of audio codec switching within an ongoing call. We provide a comprehensive analysis of behaviour of different codecs. Additionally we develop a generic codec switching algorithm which considers typical drawbacks of codec switching so as to maximize QoE.

The architecture of multi-party conferencing VoIP systems is presented in Chapter 6. We present a correction function for the standardized objective model for estimating audio QoE of peer-to-peer calls. We introduced an improved version of E-model for three widely used audio codecs in order to make it valid for assessing the QoE of multi-party calls.

Finally, in chapter 7 we summarize our conclusions and outline topics for future work.

Chapter 2

State of the Art

In this chapter we provide background information on Voice and Video over IP(VVoIP) technology, and communication protocols required to deliver VVoIP services, particularly Session Initiation Protocol(SIP), Real-Time Protocol (RTP), and Real-time Control Protocol (RTCP). Additionally, we provide a literature review in the field of VVoIP, which covers a comparison for the performance of multiple audio and video codecs, the latest methodologies and testing frameworks to assess the quality-of-experience from the end-user prospective with a detailed analysis for the advantages and disadvantages of each approach, and finally, the various techniques proposed to improve the perceived quality-of-experience.

2.1 Voice-over-IP Technology Overview

2.1.1 Voice Over Internet Protocol (VoIP)

Voice-over-IP is the combination of communication technologies, methods, protocols and transmission techniques which are required for the delivery of telephony services over the IP networks like the Internet. Telephony in general has dramatically developed in the last decade. It has initially started with Plain Old Telephone Service (POTS) or Public-Switched Telephone Network (PSTN) which carried digital data where one PSTN link supports bandwidth of 64 Kbps which is typically carried over traditional copper cables. The rapid evolution of internet starting from the early 90s has challenged strongly PSTN to a great extent, as the newly developed VoIP offer almost the same quality but with much lower costs. This growing trend is exemplified by the fact that Skype (sky, 2013), one of the most widely used VoIP applications, has 405 million registrars and 15 million online users (Wu *et al.*, 2009). Communication services transported over the internet such as voice, video, SMS, and voice-messaging are often refereed to as Internet Telephony which mainly relies on VoIP for the delivery of those services. Due to high increase of usage of VoIP services, offering services that meets with high standards of reliability and quality has become of a great importance to the VoIP service providers.

VoIP carries digital data, where voice signals are digitized and packetized at the sender before its transmission over the IP network to the receiver as shown in Figure 2.1. At the receiver, packets are decoded and played out to the listener with the usage of playout buffer which can hold the packets until its scheduled playout time to prevent long delays, silent gaps or unclear speech.

There are various implementations for VoIP according to the environment and the nature for the users of the service. Figure 2.2 (Goode, 2002) shows a deployment model for VoIP within an enterprise where many services are incorporated together to serve the enterprise needs. This implementation provides the enterprise with the choice to make all of its calls through the IP network using VoIP or to divide the traffic among the IP network and the PSTN according to costs of undertaking different paths which can be configured in the Private Branch Exchange (PBX).

Another implementation of VoIP in Figure 2.3 which doesn't use PBX and employs

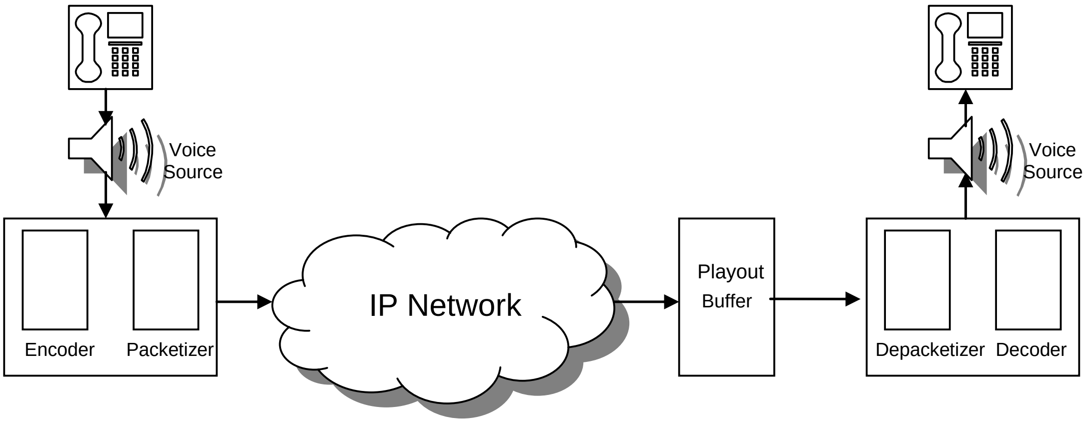


Figure 2.1: Basic VoIP System Architecture. Analogue signal is converted into digital one at the sender side by passing through the process of encoding and packetization. At the receiver side, digital signal is converted back into analogue signal.

more usage for the IP networks. Physical IP phones and computer installed softphones[1] are connected to smaller Local Area Networks (LANs) which are in turn connected to a Wide Area Network (WAN). IP phones can establish calls locally over the LAN, as they include codecs required to encode and decode the transmitted voice, whereas the best design for packet network is done when the speech near the speaker is encoded once, and decoded once near the listener. Connections to traditional switched networks can be made through the PSTN gateways.

Internet applications typically rely on TCP/IP. IP is the most widely deployed connectionless protocol for network communication, whereas TCP is a connection oriented protocol in the network transport layer which confirms packets' arrival by using acknowledgements and retransmissions. TCP/IP is a combination of both resulting a reliable connection-oriented communication protocol set. However, the main dependency of TCP/IP over acknowledgements and retransmissions causing many delays makes it difficult to be used for real-time communications, such as VoIP. On the other hand, UDP offers connectionless services using IP to transmit end-to-end messages through internet. Real-time Transport Protocol (RTP) (described in §2.1.3) and UDP are used

[1] A softphone is a software installed on a computer to establish phone calls. It is designed to have the same behaviour of ordinary physical phones, with a graphical user interface containing buttons and call options.

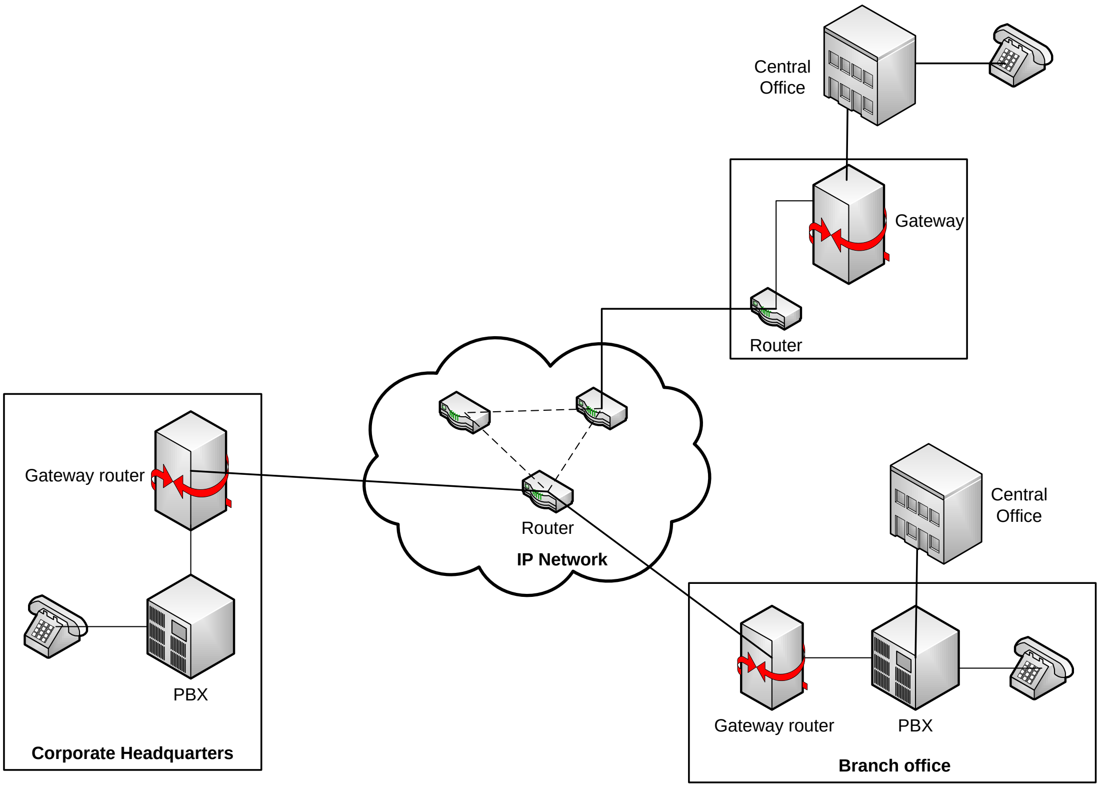


Figure 2.2: VoIP for enterprise use

together for the delivery of real-time data, such as audio and video, consequently VoIP operates over RTP/UDP/IP.

It is not an easy process to design a VoIP system that fulfils the required needs by humans for high quality and reliability. The conversion of analogue signals to digital ones is done using packetization and coding which will typically produce delays more than what users experience in traditional circuit switched networks. Lossy networks may result in high rates of packet loss leading to irritating silent gaps and lowering quality of received voice. Furthermore, application specific factors, like error concealment techniques, playout buffer size and codec algorithms has a non-trivial effect over the overall perceived quality. Standard audio codecs have coding rates from 5 kb/s to 64 kb/s. Mostly, if the output rate is decreased, then the complexity of the codec algorithm increases. Hence, in order to engineer a reliable system, a tradeoff has to be made between various factors to achieve acceptable levels of delay with the usage of

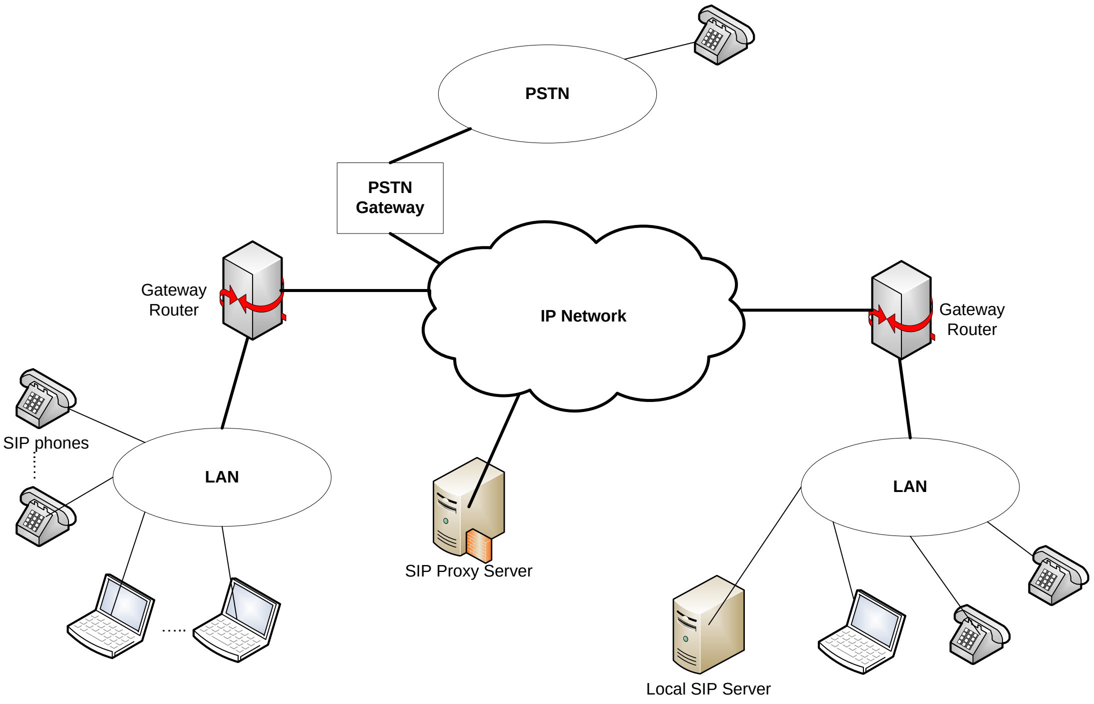


Figure 2.3: End to end VoIP

relatively uncomplicated coding algorithms, hand in hand with maximizing bandwidth efficiency.

For establishing calls using VoIP, a signaling protocol is required between the participants involved in the call in order to negotiate the call parameters between all the parties. There are many signalling protocols, such as SIP (Rosenberg *et al.*, 2002), H.323 (Toga & Ott, 1999), Megaco/H.248 (Taylor, 2000) and MGCP (Arango *et al.*, 1999). SIP and H.323 are used for peer-to-peer systems, whilst MGCP and Megaco are based on the master-slave model. MGCP is typically used for the PSTN telephony model. H.323 and Megaco are connection oriented protocols which can support video conference as well as the basic audio service. On other hand, SIP was designed specifically to be used with IP networks to support smart terminals used for establishing real-time communications. H.323 was the early leader for VoIP, however recently SIP has become the most popular. Moreover, SIP accommodates a wide range of service beside the basic telephony, such as instant messaging and presence services making it more appropriate for the current needs of Internet users with its fast evolving nature.

In the following sections, we are going to present with more details the protocols and techniques used by VoIP applications and methods of measuring call quality and the various parameters affecting it.

2.1.2 Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) is a signaling (control) protocol which operates in the network application layer on top of many different transport protocols. There exist a lot of real-time protocols which convey multiple forms of data such as audio, video or instant messaging; SIP works hand-in-hand with these protocols to enable participants to negotiate with each other in order to agree on a certain session of data transfer. It is used for initiation, modification and termination of sessions with one or more participants (Rosenberg *et al.*, 2002). A session is a method of data exchange between a set of participants, whereas media sessions can carry only audio or video or both of them for peer-to-peer and multi-party calls. Moreover, SIP provides the capability of creating new services together with replication of telephony services; examples of such services are presence and instant messaging.

SIP is not a standalone protocol. It is used in conjunction with other protocols to provide a full multimedia architecture. This architecture will typically contain protocol to transport real-time data such as Real-time Transport Protocol (RTP) (Schulzrinne *et al.*, 2003), and Real-time Control Protocol (RTCP) (Schulzrinne *et al.*, 2003) for providing network Quality of Service (QoS) feedback. Whilst, Media Gateway Control Protocol (MEGACO) (Cuervo *et al.*, 2000) is used to control the gateways to the PSTN network. SIP relies on Session Description Protocol (SDP) (Handley & Jacobson, 1998) to negotiate matching parameters between participants that describe call session such as audio,video, size of packets, codec type, etc.

SIP is based on client-server model. It consists of many functional entities that can be deployed separately or together in the same environment. Below is some of these entities:

- **User agent:** It contains both user agent client which creates SIP requests, and user agent server which responds with SIP responses.

- **Registrar:** Responsible for receiving REGISTER requests from SIP clients, and then authenticate and approve them.
- **SIP Proxy:** It is a sort of relay, which receives requests from user agents, and may modify these requests, demand authentication, calculate routes, and finally forward these requests to their intended targets whether they are other proxies, registrars or user agents.
- **Location Server:** Saves user information in a database in order to set to which IP address the request should be sent to.
- **Redirect Server:** Answers SIP requests with an address, which the request originator require to contact the targeted entity directly.

SIP is designed to be similar to HTTP request/response transaction model. Any transaction is performed by sending a request from the client that invokes a specific function, or a method, and then a response is received from the server. The main request methods defined in SIP are :

- **REGISTER:** Used to register user agent IP address and port with a SIP server(Registrar).
- **INVITE:** It creates call signaling process, it can also be used to update codec, IP address and port to which packets would be sent (RE-INVITE).
- **ACK:** It is used to acknowledge the creation of a call session on the client side.
- **CANCEL:** Cancels sessions in progress.
- **BYE:** Terminates session.
- **OPTIONS:** Acquires information about server capabilities.

The main response methods defined in SIP are :

- **1xx:** Indicates a provisional response, such as 100 Trying, and 180 Ringing.
- **2xx:** Indicates the successful completion of the request, such as 200 OK.
- **3xx:** Redirection response, in order to redirect the request to another party.
- **4xx:** Client Failure responses, such as 400 Bad Request.

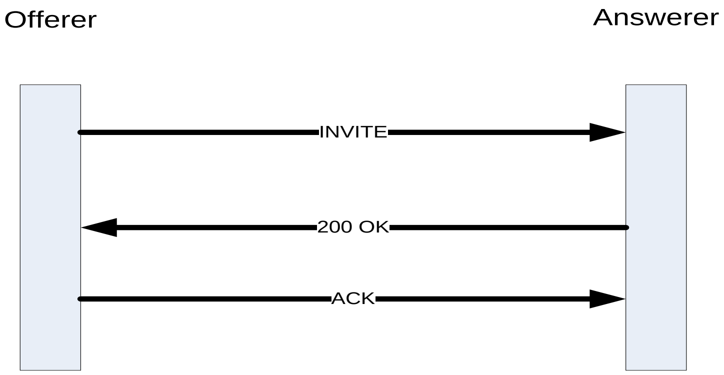


Figure 2.4: SIP/SDP session negotiation

- **5xx:** Server Failure responses, such as 500 Server Internal Error.
- **6xx:** Global Failure responses, such as 600 Busy Everywhere.

Each transaction consists of:

- exactly one request.
- one or several provisional responses.
- exactly one final response.

Session negotiation is performed by a handshake process which is typically initiated by sending an INVITE message to the targeted participant as shown in Fig. 2.4 This message includes the initial SDP offer that contains the specific parameters supported by the sender, such as codecs list. If the targeted participant agrees with the offer, it will reply by 200 OK. Consequently, the initiator confirms the session by sending an acknowledgement message ACK. Then the selected codec agreed upon from the two parties will be selected throughout the call session, unless any of the parties send a RE-INVITE message to switch to a different codec within the existing session.

2.1.3 Real-time Transport Protocol(RTP)

Real-time Transport Protocol (RTP) (Schulzrinne *et al.*, 2003) is used in the network transport layer to provide real-time services like establishing voice and video calls. The main function is to ensure the transmission and delivery of audio and video packets between source and destination, whether it is peer-to-peer call, or a multi-party conference that involves multiple senders and receivers in the same call. Apart from its main function, RTP is used for other services which contributes to the delivery of real-time data, such as timestamping, sequence numbering, determining payload type and monitoring of delivery. RTP typically runs over UDP to ensure multiplexing of different services at the same time on different ports.

Real-time communications generally do not guarantee quality of service (QoS). Consequently, RTP relies on associated services such as RTCP to monitor the QoS factors and to provide a relative degree of reliability. For example, sequence numbering is used to identify the correct placement of a packet and to avoid out-of-order reception of packets. Moreover, encryption of the transmitted packets is possible using various techniques in order to provide an acceptable level of security and privacy for the participants of the call. Beside audio and video streaming, RTP can be useful for other applications such as control and measurement applications, interactive simulation and continuous data storage.

RTP clearly establishes separation between different media types. In order to transmit audio and video sequences in a conference, they have to be sent in two different RTP sessions where the RTP and RTCP packets are sent through two unique UDP ports. An advantage of this technique is to provide conference participants with the capability to either activate audio or video streaming , or to activate both. Regardless of the separation, audio and video synchronization is done by the use of timing data which are present in RTCP packets for both audio and video sessions. Furthermore, for the sake of achieving a seamless transmission of sequences among the conference parties, VoIP applications hold the responsibility of adjusting the rate of transmission according to the bandwidth allowed in each of the reception nodes in order to prevent network congestion problems.

Figure 2.5 illustrates RTP packet format.Transmitted audio and video data is represented by RTP headers where both data and headers are combined in a UDP packet during the transmission. SSRC field indicates the source of stream of RTP packets, while CSRC represents the contributing sources for the payload included in the packet. Encoding type is typically listed in the RTP header which is attached to every packet sent from source to destination and vice versa, so that senders could modify the encoding type within the existing session, and hence allowing receivers to change their

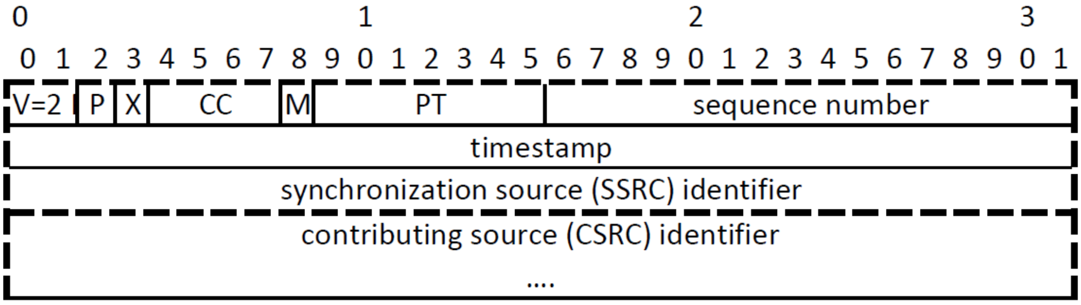


Figure 2.5: RTP packet format

decoder to accommodate with the newly changed codec. Moreover, RTP header holds sequence number and timing information in order to let the receivers to rebuild and synchronize the transmitted packets in their appropriate locations. Furthermore, the sequence number is useful for calculating the rate of packet loss and to restore packet sequence during the current session. Moreover, timestamp can be further used for measuring jitter (delay variation). The above characteristics in RTP provide a reliable way of communication in the today networks which do not guarantee quality of service and might be often subject to congestion, leading to packet losses and delays (especially in the wireless networks). However, it is important to note that RTP itself is not able to guarantee reliability; applications play a major role to achieve that by the way they are designed and developed to adapt with various network changes.

2.1.4 Real-time Control Protocol(RTCP)

The RTP control protocol(RTCP) (Schulzrinne *et al.*, 2003) is sent in parallel with RTP stream. It periodically sends control packets to all parties in the session. It employs the same distribution technique used for sending data packets through RTP. The VoIP application intending to make use of RTCP must provide an underlying mechanism to provide multiplexing of data and control packets, such as providing separate UDP ports for each type of packets.

RTCP is mainly used to provide statistics on the quality of data transmission. This statistics represent a useful feedback that can be used for many purposes. Estimating the overall call quality in MOS level primary depends on the information provided

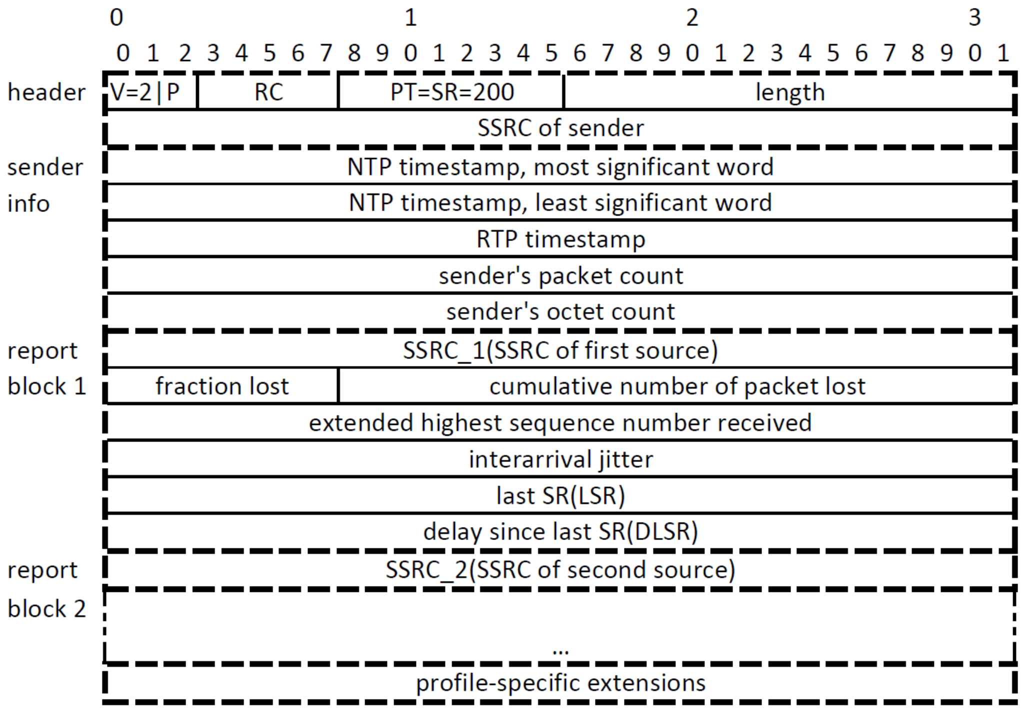

0
1
2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
header
V=2|P
RC
PT=SR=200
length
SSRC of sender
sender
info
NTP timestamp, most significant word
NTP timestamp, least significant word
RTP timestamp
sender's packet count
sender's octet count
report
block 1
SSRC_1(SSRC of first source)
fraction lost
cumulative number of packet lost
extended highest sequence number received
interarrival jitter
last SR(LSR)
delay since last SR(DLSR)
report
block 2
SSRC_2(SSRC of second source)
...
profile-specific extensions

Figure 2.6: Sender report packet format

by RTCP reports. Taking decisions to improve call quality, such as switching codecs dynamically within the call takes the feedback from RTCP as in input to the employed adaptive algorithms. Sender and receiver reports are responsible for performing the feedback functionality. RTCP makes it possible for third party hosts which are not involved in the session to be a receiving node of those reports in order to diagnose and monitor network performance. Additionally, RTCP has the ability to keep track of all the participating hosts in the session, and it provides a separate identifier for it called CNAME. This field is often used by receivers to link multiple data packets from a specific host in the session to synchronize audio and video.

There are multiple RTCP packet types to hold several control data. They are defined as follows:

- **SR:** Sender report, responsible for sending and reception of statistics from active participants in the session;

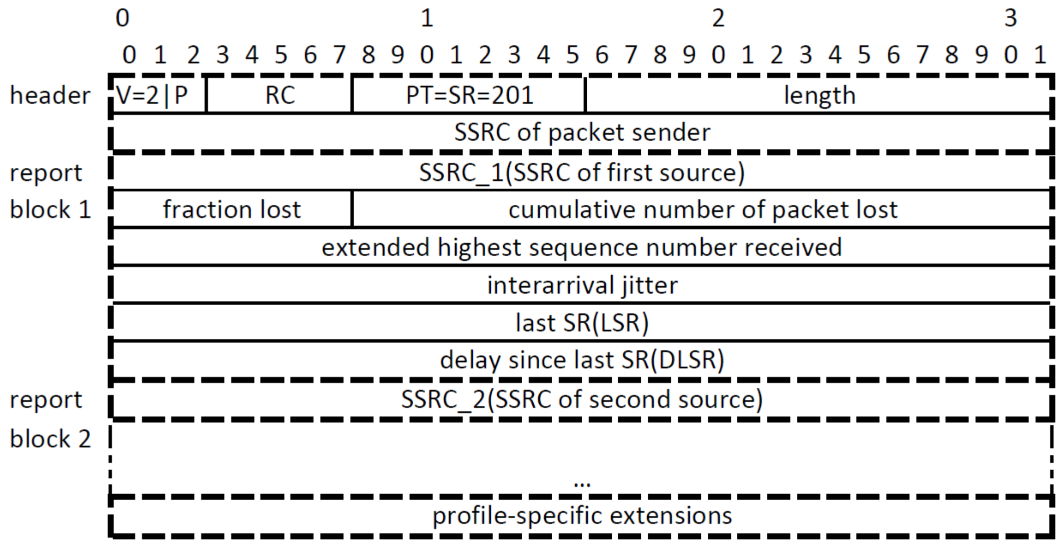


Figure 2.7: Receiver report packet format

- **RR:** Receiver report, accounts for reception of information from inactive participants in the session;
- **SDES:** Source description fields such as CNAME;
- **BYE:** Flags end of participation;
- **APP:** Application-specific functions.

The rate at which reception statistics are sent from each node is sent is constrained by the allowable bandwidth and it changes dynamically within the session according to the interval between RTCP packets transmission, it should be also sent as often as possible by the bandwidth constraints to maximize the usage of these information. Session bandwidth is divided between data traffic and control traffic, where control traffic takes a small portion of it (5%) so that the main function of delivering data traffic will not be distorted. Based on that, time intervals between packets transmission had a minimum limit of 5 seconds in order to prevent having bursts of packets that surpass the existing bandwidth.

Production of feedback statistics is done using sender report (SR) and receiver report (RR). The packet structure of both reports are almost the same except that the sender report has an extra 20-bytes information to be used by active senders. The format of sender report and receiver report are illustrated in Figures 2.6 and 2.7 respectively. Below, we explain the most important fields in the report packets which contain valuable information about the network performance:

- **Fraction lost:** Represents the fraction of lost RTP data packets from source since the last SR or RR packet. It is the calculated by dividing the number of packets lost by the number of packets expected;
- **Cumulative number of packets lost:** Indicates the total number of lost RTP data packets since the start of the session. It is the difference between the number of packets expected and the number of packets that were actually received;
- **Interarrival jitter:** Contains integer value that represents the variation of packets interarrival time;
- **Last SR timestamp (LSR):** It contains the time when the newest sender report was sent from source;
- **delay since last LSR:** It has the delay which arises from the difference between receiving the last sender report packet from source and sending the receiver report packet back.

2.1.5 Audio Codecs

A VoIP codec ("coder-decoder") is an algorithm that compresses the digital audio data by reducing number of bits so that they can be transmitted easier in the VoIP data channel (coder), the compressed data are then expanded at the receiver's side (decoder) so that audio data will be heard. It is important that participants involving in a VoIP call must agree on the same codec to be used throughout the call session, this agreement is achieved when the first INVITE request is sent which contains the SDP offer that includes the available codecs supported by the inviting party. Mostly, VoIP phones are equipped with a number of codecs that vary in their performance levels and bandwidths.

Analogue sounds can vary from low pitch sounds such as sonic boom or kettledrum, to very high pitches as a plucked guitar or a cymbal. The human ear can listen to sounds of frequencies starting from 20 to 20,000 Hz, human voice has important content beyond 14 kHz. Codecs are generally classified into narrowband and wideband codecs. Narrowband codecs were designed to be used with the traditional PSTN services for analogue phones (Ulseth & Stafsnes, 2006). The voice signal is sampled at 8,000 Hz, leading to an effective voice pass-band of about 200 to 3,300 Hz. Examples of narrowband codecs are :

- **G.711:** It is an ITU-T (International Telecommunication Union- Standardization Sector) standard codec (Rec, 1988), considered as the native language of all the modern digital telephony. Also known as Pulse Code Modulation (PCM) with its two versions μ-law and A-law. It provides audio quality at 64 kbit/s with low processing power as it doesn't perform compression, however, it needs higher bandwidth than other codecs (up to 84 kbps including all IP overhead).

- **G.729 AB:** It is an extension for G.729 (Rec, 1996) which compresses packets of 10 ms duration. It has low bandwidth requirements and operates at bit rate of 8 kbit/s.

- **G.726:** Another ITU codec that employs Adaptive Differential Pulse Code Modulation (ADPCM) algorithm (Rec, 1990). It covers transmission of audio at rates of 16, 24, 32 and 40 kbit/s. The most used mode is 32 kbit/s which doubles the usage of network capacity by using half the rate of G.711. It doesn't need high processing power as it doesn't carry out compression to the audio data like G.711.

On the other hand, wideband codecs (from 7 kHz - 20 kHz) are capable of delivering a higher fidelity audio than it used to be for traditional analogue phones resulting in a better user experience (Barriac *et al.*, 2004). Wideband codecs are often called "High Definition Codecs" or "HD codecs". Below we are mentioning the most common VoIP wideband codecs used nowadays :

- **G.722:** It is the most widely used wideband codec. It applies ADPCM algorithm at low and high frequencies separately (CCITT, 1988), resulting a 7 kHz audio that works well with speech or music.

- **G.722.1:** A new 7 kHz audio codec (Rec, 2005), mostly used in the current video conferencing systems. It is a transform codec that operates by eliminating redundant frequencies in audio. Hence, it has a lower bit rate and higher efficiency that G.722. It is also known as “Siren 7”.

- **G.722.2:** It is known as “AMR-WB” (Rec, 2003a), it is an extension of the commonly used adaptive multi-rate cellphone algorithm (AMR). It is a 7 kHz wideband that applies Algebraic Code Excited Linear Prediction (ACELP) algorithm which is more optimized to human speech, producing a high quality audio at the lowest bit rates.

- **Speex:** It is a lossy open-source codec optimized for speech (Valin, 2006). It supports narrowband (8 kHz), wideband (16 kHz), and ultra-wideband (32 kHz) compression in the same bit stream.

- **G.719:** It is a modern ITU-T codec which provides great quality for both music and speech with low values of latency, acceptable processing power and bit rates (Xie *et al.*, 2009).

2.2 Literature Review

2.2.1 Performance of Audio Codecs

Several studies has been made to compare the performance of audio codecs under varying network conditions. Kim & Choi (2011) authors compared the performance of different codecs, such as G.711, iLBC, GSM, and Speex in different conditions in a wireless network in order to estimate the capability of wireless mesh networks in handling VoIP calls. Similarly in Narbutt & Davis (2005), an assessment has been performed for G.711, G.723.1 and G.729A for WLAN networks.

There are different factors which are specific for each codec (Rodman, 2009), and has a great role in the comparison of codecs’ performance such as:

- **Audio Bandwidth:** It is a measure of audio fidelity. The higher the bandwidth, the better the codec is. Most of the available codecs today support 7 kHz audio, as 7 kHz offers easily a noticeable improvement in the audio delivered.

- **Complexity:** Codecs’ algorithms vary in their complexity. Codecs like G.711 are not complex doesn’t required high processing power, whilst for G.719 or G.722, faster processors and more memory are needed.

Table 2.1: Bit rate with respect to audio bandwidth.

Bandwidth	**Bit rate**
kHz	*kbps*
3.3	8 (G.729), 56 (G.711)
7	10 (G.722), 24 (G.722.1), 64 (G.722)
20	32(G.719)

- **Bit Rate:** The rate at which the audio packets are sent when the available network bandwidth is limited. The less the bit rate in the same audio bandwidth, the better the codec is considered. Table 2.1 lists values for the some of the common codecs. For example, at narrowband range G.729 is considered better than G.719 as it utilizes less bit rate at the same level of audio bandwidth (3.3 kHz).

- **Audio Quality:** Using standard methods of measuring quality such as subjective testing or objective testing (PESQ and E-model), codecs might result different values for each method. It is important to unify testing environment and samples used in order to have an accurate way of comparison between codecs.

- **Latency:** It is the defined as the time taken when the sender says a word till it is heard by the receiver "mouth-to-ear delay". Typically in VoIP systems, delay is composed of (Tao *et al.*, 2005):

  1. **Network Delay:** Results from propagation delay and queueing delay.
  2. **Codec-related Delay:** The delay produced as a result of packetization and encoding which is specific for each codec (Lutzky *et al.*, 2004) as shown in Figure 2.8.
  3. **Play-out Buffer Delay:** Play-out buffer usually add delays as they act as an intermediate step at the receiver to absorb jitter effect during data transfer (Atzori & Lobina, 2006).

- **Packet Loss Rate:** Packet loss in the IP network is considered one of the most important factors that cause degradation in the overall voice call quality. Packet loss greater than 5% has been shown to have a very detrimental effect on voice

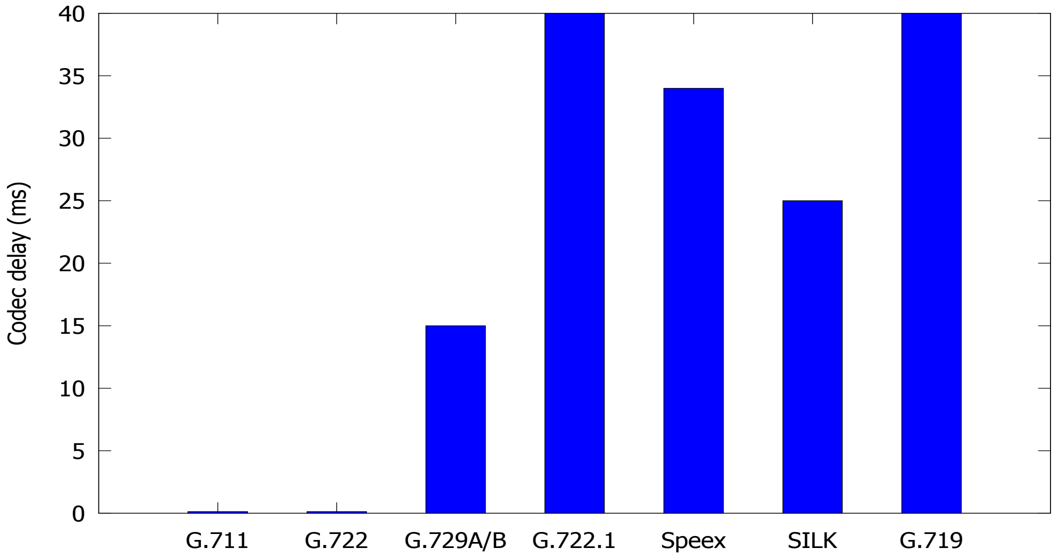


Figure 2.8: Codec-specific Delay

quality (Agrawal *et al.*, 2006). The maximum quality that can be achieved differs from codec to codec under different packet loss rates.

- **Availability and Cost:** ITU is the worldwide organization for standardization and evaluation of codecs used in telecommunications, and they are totally free to use. Codecs starting with prefix “G” such as G.711 and G.722 are ITU codecs which are subjected to open and extensive multi-vendor evaluation before being released. On the other hand, proprietary codecs, such as SILK which was developed internally by Skype, mostly require licensing and they are not standardized by ITU which might affect their reliability.

2.2.2 Measuring Audio Call Quality

VoIP offered an alternative for traditional circuit switched network, however, it does not guarantee the same degree of reliability and stability of the quality offered by the traditional telephony networks. The absence of a dedicated end-to-end guaranteed connections in VoIP communications might cause a negative effect on data transmission (Huntgeburth *et al.*, 2011). ITU defined Quality of Service (QoS) as the set of characteristics of a telecommunication service that bear on its ability to satisfy the implied needs of the user of the service (Rec, 1994). Furthermore, Quality of Experience (QoE) denotes the overall acceptability of an application or a service, as perceived subjectively by the end user (Rec, 2007). In other words, it is the measure of pleasure or annoyance of a service or an application as a result of the achievement of his expectations in terms of usability and enjoyment with respect to the user's personality and state (Jekosch, 2005; Möller, 2010; Qualinet, 2012; Roto *et al.*, 2010). QoE consists of a set of perceptual features which accounts for a multi-dimensional perceptual space (Raake, 2007; Wältermann, 2013).

There are several clear differences between QoS and QoE (Siller & Woods, 2003). QoE has a wider scope and a broader domain than QoS which typically focuses on telecommunication services only. Moreover, the QoE concept is closely related to the system performance with respect to users' point of view, unlike QoS which focuses on the performance factors of systems. Finally, QoS mainly depends on analytical methods and simulative approaches for measurements, whilst QoE relies more on multi-disciplinary approaches which involves various factors for measurement. However, it is important to consider that in many cases QoS can have a high impact on multiple dimensions of QoE, such as perceptual quality dimension (da Silva *et al.*, 2008; Fiedler *et al.*, 2010).

Enforcing QoS for voice calls in unmanaged data networks is not always possible, changes in network configuration might lead to a degradation in quality, even if the VoIP traffic doesn't share available resources such as bandwidth with other services, such as video streaming or Peer-to-Peer traffic. Hence, network operators require methods for monitoring and estimating QoS and QoE in an accurate and efficient way to achieve technical and commercial requirements. These methods are classified into subjective and objective methods as they are explained as follows.

2.2.2.1 Subjective Testing

ITU-T standard P.800 (Rec, 1996) has defined the conditions and the environment to carry out subjective testing methods. Subjective testing can be performed in two modes, listening-opinion test and conversation-opinion test. Listening test is a one way test which does not reach the same level of interactivity of conversational tests, it is used to assess applications of physical systems which are unidirectional, such as recorded announcements devices, broadcast circuits and public address systems where

Table 2.2: MOS Scale.

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

some factors can degrade the listening quality like noise, loss, and distortion. On the other hand, conversational test involves interactivity, and they are intended to reproduce as far as possible the real conditions experienced by call participants to measure the conversational quality. Typically, tests are carried out using 12-24 participants who listen separately to an audio stream of few seconds to rate the quality on a scale of 1 to 5 called as shown in Table 2.2. The arithmetic mean of the collected opinion score is called mean opinion score (MOS). MOS is a world-wide accepted standard for measuring voice quality as it correlates well with the actual voice quality perceived by end-users. The potential difficulties of performing subjective testing is that it is expensive, time consuming, difficult to repeat and can't be used for monitoring large-scale network infrastructure on long-term basis (Sun & Ifeachor, 2006).

2.2.2.2 Objective Testing

Due to the limitations of subjective testing, objective testing provides an alternative for measuring voice quality. It is classified into intrusive and non-intrusive methods. Intrusive methods as PESQ (Rec, 2001) are carried out offline, they are more accurate, however they are not suitable for monitoring live streams. Otherwise, non-intrusive techniques such as ITU-T E-model consists of computational models that can be used online to monitor live traffic based on the current network conditions and related system parameters (e.g. delay, jitter, loss rates and codec type).

PESQ: Perceptual Evaluation of Speech Quality (PESQ) (Rec, 2001) is the most commonly used method for intrusive assessment of the quality in VoIP applications. The PESQ algorithm performs a comparison between the original signal X(t) and the degraded signal Y(t) which results from passing X(t) to the VoIP application as shown

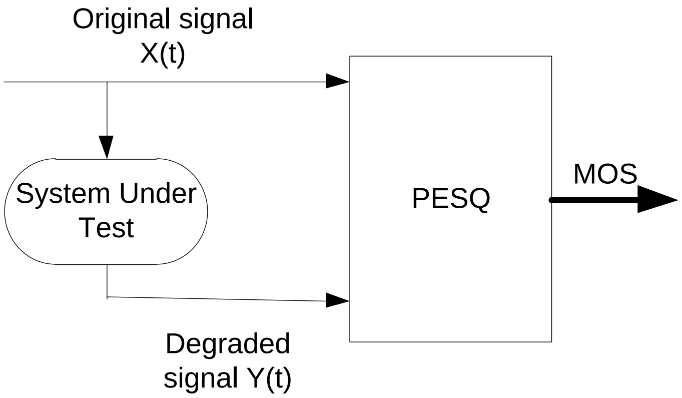


Figure 2.9: PESQ procedure

in Figure 2.9. The output of the algorithm is MOS value which represents an estimation of the perceived quality at the end user if Y(t) is tested using subjective tests.

The algorithm starts by computing a set of delays between the two signals, for each of these delay intervals a start and stop point is set. Then the alignment part of the algorithm is carried out based on the fact of having single delays for the corresponding interval. The main step of the procedure is the mapping of the original and degraded signals to a model that resembles the psychophysical models of audio signals in the human hearing system, while considering factors that might have a deterministic effect over the perceived quality, such as linear filtering and local gain variations. This is achieved by several steps that include: time alignment, level alignment to a calibrated listening level and time-frequency mapping.

ITU-T has further modified PESQ by introducing MOS-LQO (Listening Quality Objective) as in Rec (2003b). Raw MOS values are mapped to MOS-LQO by Equation 2.1 in order to provide a closer estimation of human perception and more accurate results, where x represent the raw MOS value and y is MOS-LQO value.

$$y = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.4945 * x + 4.6607}} \tag{2.1}$$

A comparison between subjective listening quality (LQ) MOS and PESQ has been done in Rix (2003). They provided a performance analysis of PESQ-LQ scale, which gave good results in varying network conditions and across different languages. On the other hand, MOS scored from subjective testing can vary significantly between different languages. Pennock (2002) investigated the accuracy of PESQ algorithm and stated its limitations in competitive analysis; when comparing performance of networks with similar quality and in system optimization. Furthermore, it is not suitable for use in legal contracts specifying requirements for speech quality(e.g SLA). Han *et al.* (2013) provided a comprehensive analysis for the behaviour of PESQ algorithm when tested in enterprise networks and compared it to the output of non-intrusive methods as E-model.

It is important to take into account that PESQ was designed to measure the quality of VoIP applications that use narrow band codecs for one-way speech. Moreover, it does not consider impairment factors relevant to two-way conversations, such as delay, echo, loudness loss and sidetone which might mislead the results of quality measurement. The newly developed tool POLQA (ITU-T, 2011) was designed to handle these limitations of PESQ, however it is not open to public yet and needs licensing for evaluation and testing purposes.

E-Model: ITU-T E-Model (Rec, 2009) is a computational model that gathers all the impairment factors affecting voice quality into a total value that represents the perceived quality by end-users. It is a non-intrusive method that does not require a reference and degraded signal, and can be used as an end-to-end tool for transmission planning.

E-Model is designed on the fact that psychological factors are additive on the psychological scale (Rec, 2009). In other words, each impairment factor that influence the quality like delay, loss rate,..etc. can be calculated separately as their contribution on the estimated quality can be separated (Cole & Rosenbluth, 2001). The input parameters are shown in Figure 2.10 where the background room noise at sender and receiver are represented by Ps and Pr respectively. D-factor is an indication to the distortion caused by the microphone and the loud speaker at both sides. A, qdu, SLR, RLR, and their sum OLR are considered as values for the connection overall. Other parameters, such as Ie, and $WEPL$ are associated only with the receiver's side.

$$R = R_0 - I_s - I_d - I_{e-eff} + A \tag{2.2}$$

R_0 is the signal to noise ratio at 0 dBR (decibels relative to reference level), I_s is the speech voice impairment factor, I_d indicates the impairments due to the delay, I_{e-eff} is the impairments caused by codecs, the values of R_0 and I_s are defined as 94.77 and 1.41

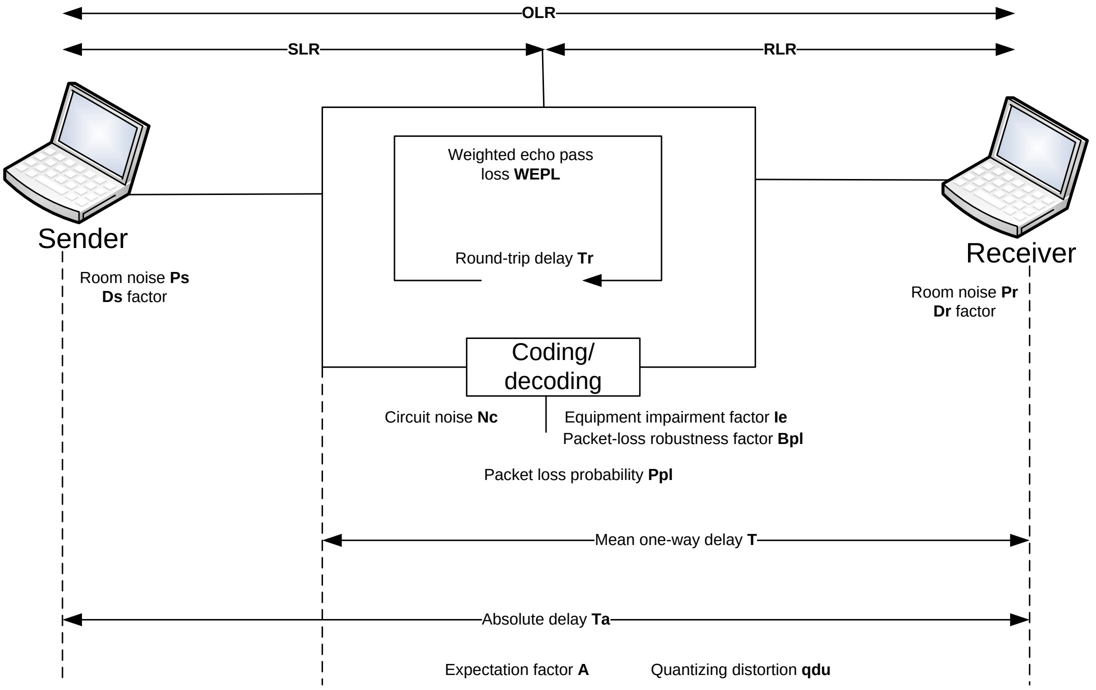


Figure 2.10: E-model connection.

respectively (Rec, 2009) and A is the advantage factor, assuming our communication system is conventional, then we neglect A value. As outlined in Clark *et al.* (2001); Cole & Rosenbluth (2001); Lustosa *et al.* (2004) 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 as.

$$R = 93.2 - I_d - I_{e-eff} \tag{2.3}$$

I_d is a function of one way delay only(d); it can be calculated using a 6th order polynomial (Sun & Ifeachor, 2006).

$$\begin{split} I_d = -2.46 \times 10^{-14} \times d^6 + 5.062 \times 10^{-11} \times d^5 - 3.903 \times 10^{-8} \times d^4 + \\ 1.344 \times 10^{-5} \times d^3 - 0.001802 \times d^2 + 0.103 \times d - 0.1698 \end{split} \tag{2.4}$$

I_{e-eff} is the packet loss dependent effective equipment impairment factor and can be expressed as:

$$I_{e-eff} = I_e + (95 - I_e)\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{2.5}$$

Table 2.3: Relationship between R and Mean Opinion Score.

R	**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

I_{e-eff} is derived using a codec-specific value (I_e) which represents the impairment factor given by codec compression, and by a packet loss robustness factor (Bpl) that represents the codec robustness against random losses. The values of I_e and Bpl for several codecs are provided by ITU in G.113 recommendation (ITU-T, 2001); they were deduced using subjective MOS tests and network experience. Ppl 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.

Once calculated, the R value can be used to estimate MOS using the following equation and as illustrated in Table 2.3:

$$MOS = \begin{cases} 1 & \text{if } R < 0 \\ 1 + 0.035R + & \\ \quad R(R-60)(100-R)\times & \\ \quad 7\times 10^{-6} & \text{if } 0 \leq R \leq 100 \\ 4.5 & \text{if } R > 100 \end{cases} \tag{2.6}$$

The E-model provides the most convenient way nowadays for online measurement of voice quality, however it is only exclusive to ITU codecs and to certain network conditions. In order to apply E-model on non-ITU codecs, subjective tests are needed to derive mathematical models for these codecs. This limitations might prevent its use in modern and emerging systems. Moreover, it is based on a complex set of formulas which might not be suitable for real-time monitoring, particularly when measuring performance of non-ITU codecs.

Corrected versions of the E-model have been proposed to simplify the calculations and focus on the most important factors required for monitoring the call quality (Assem *et al.*, 2013). Paulsen & Uhl (2010) introduced a new parametrized QoS measurement method for VoIP applications. Their proposed improved E-model use the "glass box" principle. They took into account typical IP-Environmental parameters whilst the original E-model is designed for circuit-switching networks and can not really take such factors into account. They compared their results to the PESQ and the original E-model results and they show higher accuracy in measuring the MOS compared to the original E-model. Ren *et al.* (2010) studied how the jitter affects the VoIP quality and how to model such effect into the E-model. They used the PESQ algorithm to measure such effect and as a result, they introduced a new I_j formula which is added to the original E-model representing jitter impairment factor. Zhang *et al.* (2011) came afterwards to use the prior extended E-model in order to compare the performances of the original and extended E-model (including the jitter impairment factor) by applying them both on different VoIP systems (Skype, Google Talk and Windows live messenger). They concluded that Windows live messenger outperforms in terms of listening, Skype has the largest MOS, and Google Talk generally has the least MOS. Obafemi *et al.* (2011) studied the E-model with a focus on the effect of the ignorant parameter jitter playout buffer on the accuracy of the call quality resulted from the E-model. Their results shows that the adaptive play out buffer should not be ignored when evaluating the perceived call quality. They suggest modifying the original E-model to include measurements of an adaptive playout buffering. Zhang *et al.* (2005) proposes a new algorithm to measure the packet loss burstiness to be included in the E-model as a replacement of the random probability of the packet loss to calculate the MOS value. They show that their improved E-model have a higher accuracy under bursty packet loss conditions. Authors in Halas *et al.* (2012) proposed an improved E-model for providing better estimates of MOS VoIP call quality, they included jitter buffer size, codec packetization and network jitter into E-model. Chen *et al.* (2006) proposed a model to quantify user satisfaction levels called USI model, which is initially designed for Skype, but can be generalized to other VoIP applications. USI model is different from other objective matrices that it is based on call duration, rather than speech quality which includes additional factors such as listening volume and conversational delay.

The main difference between our work and those reviewed above is that we noticed that the accuracy of the E-model has not been tested before in the multi-party VoIP conferencing system, where several participants are involved in the same VoIP call session. There is surprisingly limited work in the multi-party QoE area, compared to the wide literature on QoE for person-to-person calls. Thus, in Chapter 6 we perform a detailed analysis for the accuracy of monitoring the call quality using the original E-model. Based on our analysis, we proposed a correction for the current original E-model in order to be used in the VoIP conferencing systems with centralized architecture.

2.2.3 Testing Frameworks for Monitoring Quality-of-Experience

Real time Voice and Video over IP applications are sensitive to network conditions—variations in metrics including end-to-end delay, packet loss rates and jitter have a significant impact on the quality as perceived by end users. Given this, monitoring and estimating call quality is an important task that has been extensively studied by the research community. We briefly review some of the relevant previous work on this topic.

Jiang & Huang (2011) introduce a voice quality monitoring system based on the SIP protocol, which uses RTP statistics to get MOS score using the simplified E-model. Kim & Choi (2010) propose a network performance monitoring method that uses RTCP statistics to monitor multimedia services like VoIP and IPTV. da Silva & Lins (2006) analyze the QoS provided by SIP for voice traffic by measuring the delay, jitter and packet losses. Carvalho *et al.* (2005) propose three corrections to the E-model in order to give more accurate results indicating the QoE expected at the end user; they also describe a measurement tool based on these corrections. Gong *et al.* (2009) propose a pentagram model to measure the QoE based on service integrability, service retainability, service availability, service instantaneousness and service usability. Due to also the lack of QoE monitoring systems, Hershey *et al.* (2009) propose 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 event that causes noticeable application delay. Calyam *et al.* (2007) propose the GAP-Model, which assess VVoIP QoE via an offline model of QoE that is expressed as a function of bandwidth, delay, jitter, and packet loss.

There has been many attempts to provide the user with an indication of the current call quality. There are existing commercial solutions that analyze the network for several factors and not specified for VoIP services only. Other solutions as “Smart RTP monitoring Prope” offered by VOIPFUTURE (voi, 2013), and “NetIQ Vivinet

Assesor" (net, 2013) are focused on VoIP quality of service, but they are not free or open source. Huntgeburth *et al.* (2011) introduced a distributed setup for measuring the voice quality on pre-defined points in the network path in order to capture the network conditions, and hence estimate the call quality to be sent later to an administrative instance. The solutions provided is open source, however it requires complicated setups as it has to be installed on a separate entity, which will not be always feasible for simpler VoIP communications. Furthermore, it uses the E-model to estimate the call quality, but it is only restricted to ITU codecs, to which the E-model was developed for.

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 developed a novel testing framework to estimate the voice/video call quality in advance by emulating the audio/video traffic as explained later in Chapter 4. Such estimates can then be used to select the most appropriate codec to use for upcoming calls. Crucially, this processes does not require the transfer of source audio/video sequences and does not require the end user to provide quality rankings.

2.2.4 Improving Quality of Voice Communications

Real-time applications like VoIP requires minimum service guarantees that exceeds the best-effort structure of current IP networks. Network performance and codecs' behaviour have a vital effect on the perceived quality.

2.2.4.1 Improving Network Performance

There are two possible approaches to ensure that the network is capable of meeting service requirements (Tao *et al.*, 2005). The first approach is done by making changes in the network itself by introducing service differentiation mechanisms, and configuring them to add service guarantees through resources dedication to VoIP traffic. The main issue with that approach, is the added complexity to the network. It has showed success in limited cases, such as allocating bandwidth on a link to provide VoIP traffic with a sufficient share from the overall bandwidth in that access link. However, it is infeasible to apply that approach on large multi-operator networks as the size of Internet.

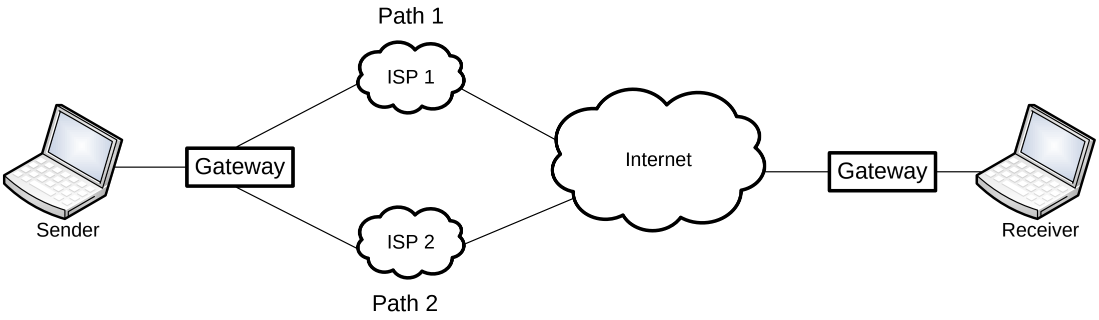


Figure 2.11: Path Switching algorithm is implemented on the gateway machine in order to periodically select the best path available to achieve the best possible QoE.

The second option is to rely on the diversity of paths that the network can offer, and to provide applications to select the best path according to its performance (Fei *et al.*, 2006; Nguyen & Zakhor, 2003; Tao *et al.*, 2004).Multi-homing (Akella *et al.*, 2003) and overlay network (Andersen *et al.*, 2001) are two widely used approaches aimed at leveraging path diversity to enhance end-to-end application performance and availability. VoIP quality can be improved when the VoIP traffic operates through a path switching mechanism that consistently picks the best path with the highest performance measures. Tao *et al.* (2005) have developed an application-driven path switching algorithm that has to be implemented on a gateway as shown in Figure 2.11. Zhang *et al.* (2009) proposed a more economical solution based on SIP+P2P system, where P2P overlay is organized via a set of proper nodes, which can also provide path diversity for end-to-end communications. Similar work has been done for video communications in (Apostolopoulos, 2001).

2.2.4.2 Codec Switching

In current VoIP applications codec switching is typically achieved via the Session Initiation and Session Description Protocols (SIP/SDP) (IETF, 2002, 2006). 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 is undergone to select a new codec; in this scenario it is important that both peers synchronize with each other in order to avoid data misinterpretation (Walterman *et al.*, 2008).

In Osipov (2006), an algorithm is developed to calculate the average delay, the compare it to the current delay in order to select the appropriate codec, algorithm was verified on different network bandwidths, resulting an increase in MOS value when using the adaptive codec selection technique. Aktas *et al.* (2012) compare the speech quality of a set of standard codecs under different network conditions, and propose an adaptive end-to-end based codec switching scheme based on available bandwidth—the codec is chosen accordingly. However, they only evaluate their scheme using two codecs: PCMU and SPEEX. Sulovic *et al.* (2011) propose an algorithm for adaptive adjustment of VoIP sources transmission rate based on voice quality estimated at the receiver. They switched between three codecs in their algorithm: G711, G729A and G723.1 5.3k, showing that their algorithm maintains high MOS values during network congestion. Robustelli *et al.* (2003) propose a voice coder that performs automatic codec switching according to packet loss. They mainly switch between GSM and PCMU to in three different network scenarios, showing that voice quality will increase if compared to using only one codec. Sfairopoulou *et al.* (2007) described an algorithm which extracts network statistics from RTCP packets and MAC layer, then adapt dynamically according to changing network conditions. Similar techniques were employed in (Servetti & De Martin, 2003), (Trad *et al.*, 2004), (Kawata & Yamada, 2006) and (Ng *et al.*, 2005).

Costa & Nunes (2009) describe an adaptive codec switching technique embodied in their "NCVoIP" application. NCVoIP starts to monitor and analyze the quality of the voice, changing to a lower or higher codec transmitted rate according to predefined threshold values for each codec. They demonstrate 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 network congestion exists, results in a significant voice quality improvement. Walterman *et al.* (2008) introduce 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 Chapter 5 we present a generic adaptive codec switching algorithm to improve overall perceived QoE. 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 special care should be taken to avoid the negative impact of switching.

2.2.5 Video Telephony

Video transmission has been used for several applications, such as cable TV, satellite broadcasting, DVD storage and terrestrial transmission channels. These systems are distinguished by having a constant format of video signal (Schwarz *et al.*, 2007), and were typically encoded using H.220.0—MPEG.2 systems (Rec, 2000). These system were characterized by their reliability, as they either work fully or don't work at all. The rapid evolution of the Internet has led to modern video transmission systems which totally rely on IP networks by employing RTP for providing real-time services to ordinary platforms and to mobile devices. The quality of connections in IP networks is known by its variability due to connection sharing and links' capabilities. Consequently, modern video transmission requires another set of codecs different than the traditional set to cope with the characteristics of modern IP networks in order to provide end-users with the best possible quality of experience.

An example of modern video codecs used in VoIP services is H.263 and H.264. H.263 is video compression technique standardized by ITU-T (Recommendation, 1998), it is a low-bit-rate codec which doesn't require high processing power, unlike H.264 (Draft, 2003) which employs more complex algorithms and is far more efficient in bandwidth utilization to deliver good video quality, however, it needs higher processing power.

Quality of video transmission is not guaranteed in IP networks. In most cases, transmission of video can be subjected to a lot of losses. Moreover, delay can cause unwanted pauses in the received signal, as the receiver might need to pause it processing, while the buffer refills. Consequently, both packet loss and delay will cause degradation in the interactive video call quality between the end-users. Hence, establishing methods of assessment to evaluate the quality of video services over IP is indispensable. Video's assessment methods are quite similar to those of audio, they are classified into subjective testing and objective testing.

Table 2.4: Relationship between $PSNR$ and Mean Opinion Score.

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

ITU-R BT.500 (Rec, 2002) introduced a methodology for carrying out subjective testing through a panel of human observers to evaluate the video quality using MOS values. Peak Signal to Noise Ratio (PSNR) is a method of offline objective testing, it operates in a similar way to PESQ. This method assesses the performance of video transmission systems by calculating PSNR between the original and the received (degraded) video; it is a differential metric which is computed using images. Mapping of PSNR to MOS values has been provided by Ohm (2004) as shown in Table 4.2, the resulted values would give a close indication to the human quality perception for videos with relatively low motion (Klein & Klaue, 2009).

ITU-T G.1070 (rec, 2007) specified a video quality model for telephony services. Video quality V_q is defined as:

$$V_q = 1 + I_{coding} \exp(\frac{-P_{plv}}{D_{Pplv}}) \tag{2.7}$$

Where 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]) is represented by I_{coding}. D_{Pplv} is the measure of robustness for the video quality against packet loss, where the percentage of packet loss rate is defined by $Pplv$[%]. I_{coding} and D_{Pplv} are further defined in rec (2007) by the following set of equations:

$$I_{coding} = I_{Ofr} \exp(-\frac{(\ln(Fr_v) - \ln(O_{fr}))^2}{2{D_{FrV}}^2}) \tag{2.8}$$

Where O_{fr} is the optimal frame rate where video quality is the maximum. I_{ofr} is

Table 2.5: Conditions of deriving coefficients.

Factors	**#1**	**#2**	**#3**
Codec type	MPEG-4	MPEG-4	H.264
Video format	QVGA	QQVGA	QQVGA
Key frame interval (s)	1	1	1
Video display size (inch)	4.2	2.1	2.1

the maximum quality at each video bit rate(Br_v). They are expressed as:

$$O_{fr} = v1 + v2 \times Br_v \tag{2.9}$$

$$I_{ofr} = v3 - \frac{v3}{1 + (\frac{Br_v}{v4})^{v5}} \tag{2.10}$$

D_{FrV} defines the robustness of video quality due to frame rate (Fr_v):

$$D_{FrV} = v6 + v7 \times Br_v \tag{2.11}$$

The degree of video quality robustness against packet loss is defined by D_{PplV} expressed in (2.12).

$$D_{PplV} = v10 + v11 \times \exp(-\frac{Fr_v}{v8}) + v12 \times \exp(-\frac{Br_v}{v9}) \tag{2.12}$$

Finally, coefficients $v1, v2...v12$ are defined according to codec type, key frame interval, video display size and video format as shown in Tables 2.5 and 2.6. They were derived in Yamagishi & Hayashi (2006a) through as set of subjective quality assessment experiments. Thirty two non-experienced individuals were involved in the experiments, where an image which has diagonal measurement of 4.2 or 8.5 inches was presented on a 17-inch LCD screen with a resolution of 1280×1024. The format of the displayed video was video graphic array (VGA: 640×480) or quarter video graphic array (QVGA: 320×240). The parameters of the experiment were frame rate, packet loss rate, and coding bit rate. (Yamagishi & Hayashi, 2006b) applied the same model for video quality estimation of H.264 codec.

Table 2.6: Provisional coefficients for video quality estimation function.

Coefficients	**#1**	**#2**	**#3**
$v1$	1.431	7.160	7.160
$v2$	2.228 x 10^{-2}	2.215 x 10^{-2}	2.215 x 10^{-2}
$v3$	3.759	3.461	3.8
$v4$	184.1	111.9	0.29
$v5$	1.161	2.091	1.2
$v6$	1.446	1.382	1.382
$v7$	3.881 x 10^{-4}	5.881 x 10^{-4}	5.881 x 10^{-4}
$v8$	2.116	0.8401	0.8401
$v9$	467.4	113.9	113.9
$v10$	2.736	6.047	6.047
$v11$	15.28	46.87	46.87
$v12$	4.170	10.87	10.87

2.3 Summary

We have presented the latest models and techniques to measure the perceived QoE at end-users, they are known to have several deficiencies, as they are time consuming, and require expensive resources. We focused in our research in finding novel ways of estimating QoE in a manner that is more efficient than the current methods. Furthermore, we provided a comparison between the existing techniques to improve the overall QoE during VoIP calls. Our work involved developing a generic codec switching algorithm in order to attain the best possible QoE, based on the fact that codecs' performance is different under varying network conditions. Finally, we studied the existing architecture of multi-party VoIP conferencing systems, which are more complicated than the ordinary peer-to-peer systems. Current methods of assessing QoE for peer-to-peer calls are not necessarily valid for multi-party calls.

Chapter 3

Framework and Test Environment

We present an overview of the architecture of VoIP softphone applications which were used during our experiments, in particular, IBM Sametime Unified Telephony which is intended for enterprises and business purposes, and Jitsi, which is an open-source application intended for personal use. Furthermore, we describe the components used in our testbed to measure and evaluate the quality of VoIP applications, including tools which can emulate network and manipulate network conditions—such as Dummynet, and that tools can send network streams between multiple network nodes, and consequently evaluate network performance—such as Iperf. We also describe additional tools that we used to emulate various network scenarios and to monitor network characteristics—such as Imunes and Wireshark. Finally, we show how these components are deployed together in our testbed in order to have a reliable framework that has the capability to perform multiple test cases with various scenarios for the network, and hence estimate the perceived quality-of-experience accurately.

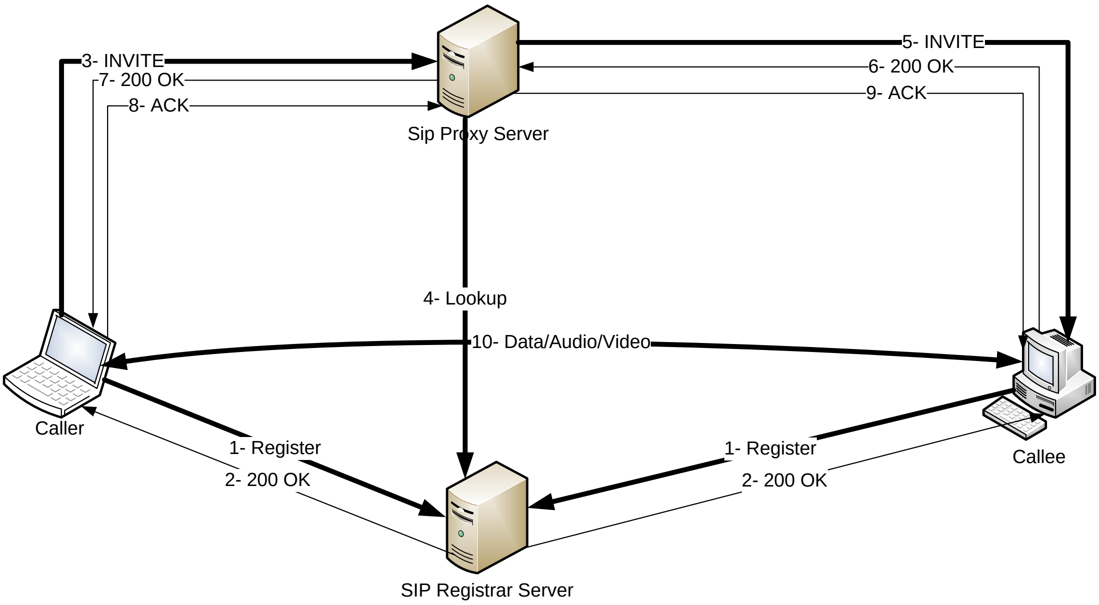


Figure 3.1: Basic SIP-based System Architecture.

3.1 Architecture of Softphone Applications

A softphone is a phone that enables establishing calls without the need to have a physical device. It is a software component which is designed mainly to be a part of a VoIP application. Its main function is to act as a friendly interface between users and the complicated VoIP system, where users can dial numbers and establish calls as a primary aim. Other functions might exist in the softphone application as sending SMS or Instant Messaging and presence services. Most likely, softphone graphical interface is designed to look like an ordinary phone with buttons representing the keys, where input devices as mouse, keyboard, keypad or touch screen can be used to make calls. Whereas for speaking and listening, a headset and microphone will be required.

Softphones that implement SIP as the protocol for establishing calls has an architecture as shown in Figure 3.1. The application must support SIP methods as INVITE, 200 OK, ACK,...etc. Proxy Server is the intermediate entity which plays the role of routing while Registrar is the server that receives REGISTER requests from both of the caller and callee, consequently it stores the information it receives in the location service for the domain it handles.

SIP-based softphone application at the caller and callee side must be able to carry out the following procedure for establishing calls.

1. REGISTER request is sent from both caller and callee to the Registrar server.
2. 200 OK is sent back from the Registrar server, where the name address (URI) is contained in the message.
3. When it is intended to establish a call, Caller sends INVITE request to the proxy sever.
4. Proxy server looks up the Registrar server to find the callee's address.
5. INVITE is forwarded from caller to callee through the Proxy server.
6. Callee sends 200 OK as a response to INVITE.
7. 200 OK is forwarded to the caller by the Proxy Server.
8. Caller confirms that the session has been established by sending ACK message to the Proxy server.
9. Proxy sever forwards ACK to the callee.
10. Flow of the RTP stream starts between the caller and the callee.
11. For the termination of the call, one of the call terminals will send a BYE request.
12. The other terminal will reply by 200 OK to confirm the termination of the call session.

The previous procedure is considered the basic and main functionality to be present in the softphone in order to carry out its main role which is establishing and receiving calls. However, architecture of different softphones are not the same, they differ due to several factors which includes the environment where the softphone is going to be deployed, the potential users of the service which varies from internet users who uses the service for free for personal use to enterprise users who use the service more frequently for important meetings which require another level of security and quality of service provided, and the extra targeted functionality also has a major rule in defining the architecture of the softphone. Some softphones offer services like sending SMS or making landline calls, others add more complicated functionality as integrating with email clients and calendar services.

In the next section, we are describing the architecture of two softphone applications, which we used extensively during our experiments to demonstrate the effect of various network conditions on the call quality. The first one is Jitsi (jit, 2013) which is an open source application intended for personal use, the second is IBM Sametime Unified Telephony (SUT) (ibm, 2013) which provides telephony services for enterprises.

3.1.1 Jitsi

Jitsi is an application that provides free service for making audio and video calls, sharing desktops, and transfer of files and messages. More importantly, it allows users to do those services through various protocols including standard ones as Extensible Messaging and Presence Protocol (XMPP), SIP, and proprietary protocols like Yahoo! and Windows Lives Messenger (MSN). Most of it is written in Java, however some parts are written in native code.

The most important factors which were considered when designing Jitsi is to make it developer friendly, to support multiple protocols and finally to work on different platforms. Jitsi runs on Windows, Mac, Linux and FreeBSD. Jitsi is built using OSGI framework (Brown & Wilson, 2012) where the whole design is divided into smaller modules, and features are separated into bundles. Jitsi is simply a collection of these bundles. There is one bundle responsible for handling SIP calls, another one that controls the GUI, yet another one that does XMPP. All these modules need to run together in an evirnoment provided, Jitsi used Apache Felix (apa, 2013) as an open source implementation for OSGI framework.

OSGI consists of two main parts, services and its implementation “Impl”. OSGI services are Java interfaces which represents the parts of the bundle that are visible to everyone. They allow the use of certain functionalities like making calls or sending messages without knowing the actual implementation of the functionalities. Implementation of those functionalities is done separately in other classes and it is called Impl. OSGI provides this advantage for developers to hide service implementation and to assure that they are never accessible from outside the bundle they are in, accordingly other bundles can make use of it through the service interface. For example, service interfaces for all protocols will be in this package `net.java.sip.communicator.service.protocol`

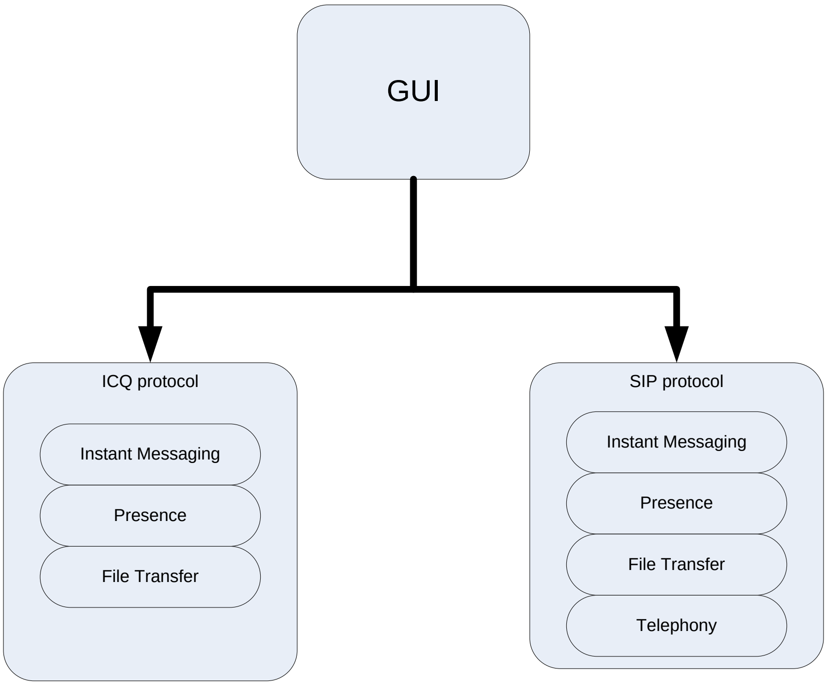


Figure 3.2: Jitsi Architecture.

while different implementation for each protocol will exisit at separate package, SIP implementation will have the name of `net.java.sip.communicator.impl.protocol.sip`.

As illustrated in Figure 3.2 Jitsi is composed of operation sets that provide the interface for the protocols implementation. The main module of GUI is connected to all the interfaces so that when the GUI wants to update for example the presence (online status of contacts) it checks which protocol is being used and accordingly it calls the appropriate method for this protocol. Not all protocols support all features, as shown in the figure ICQ protocol does not support telephony services while SIP does.

A large set of audio codecs is supported ranging from narrowband codecs as iLBC, Speex 8Khz, GSM and G729 to wideband codecs that delivers high audio quality like wideband Speex, G.722 and SILK. Moreover Jitsi supports the traditional codec G.711 with its two versions (μ-law) and (a-law). For video calls, H.263 is supported. In addition to, the popular H.264 which delivers great video quality, however, it needs higher requirements of bandwidth, and CPU processing power.

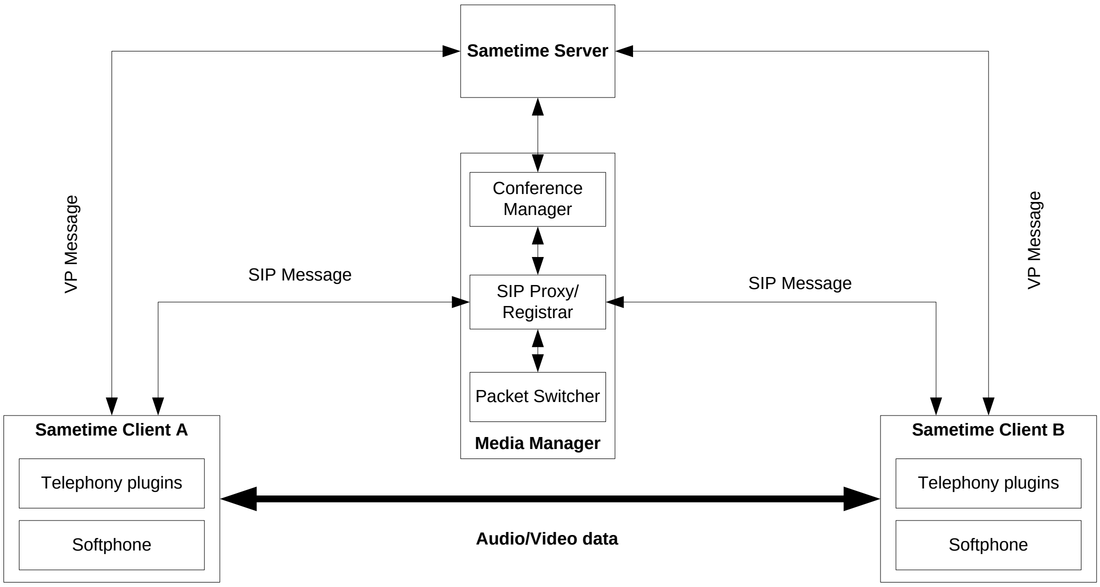


Figure 3.3: IBM Sametime Unified Telephony Architecture.

3.1.2 IBM Sametime Unified Telephony (SUT)

IBM Sametime Unified Telephony (ibm, 2013) is a family of collaboration products targeted for enterprises with high capacity of users working in different environments: offshore and on-site. SUT provides many services as real-time awareness, instant messaging, screen-sharing capabilities, file transfers and IP audio/video communications, in addition to private branch exchange (PBX) and legacy time-division multiplexing (TDM) systems. It offers flexibility and efficiency of real-time communications into business world by connecting employees, customers and partners in a way that guarantees the security and the integrity for the flow of information between the interacting parties.

Figure 3.3 presents an overview of SUT architecture. It is composed of two main components: Client and Server. Sametime Client contains the softphone application and other plugins that can be modified for extending the features according to different business requirements; the client can be stand-alone client or embedded in another application (e.g. Lotus Notes) or a meeting client (embedded as a web browser plugin). All client-to-client communications will pass though Sametime server. This design guarantees the security of all communications. Sametime server contains three application servers which interact with each other and are responsible for performing various functions.

- **Community Server:** is responsible for delivering services as login, presence and instant messaging.
- **Meeting Server:** its function is to handle meeting services like screen sharing and audio and video calls.
- **Domino Server:** provides core functions for SUT such as directory access and authentication.

SUT supports a reasonably good set of audio codecs such as G.722.1, G.711, iLBC and the high performance codec iSAC, which is a wideband codec that is adaptive to the available bandwidth and has variable bit rate(10 Kbit/s to 52 Kbit/s). For video, SUT uses H.263 and H.264 for peer-to-peer calls, as well as multi-party calls.

3.2 Tools and Methods of Network Emulation

In this section we are presenting the tools used to generate different network scenarios through out our experiments to test the accuracy of the results and the efficiency of the produced algorithms under various conditions and setups for the network.

3.2.1 Dummynet

Dummynet (Carbone & Rizzo, 2010) is a network emulator used mainly for testing network protocols, it has been extended to emulate and test various network scenarios by simulating network conditions, such as delay, packet loss, jitter, bandwidth limitation, and queues. It also implements many queue management policies and packet scheduling algorithms with parameters which can be configured in run time. Moreover, Dummynet has the capability to create multiple paths between source and destination which allows traffic to be directed through the path which user selects or to be randomly directed to one of these paths. Dummynet is one of the main components of FreeBSD and Mac OS X, furthermore, it is supported on Windows XP and Linux. It can be used as bridge as shown in Figure 3.4 by installing it on a separated machine

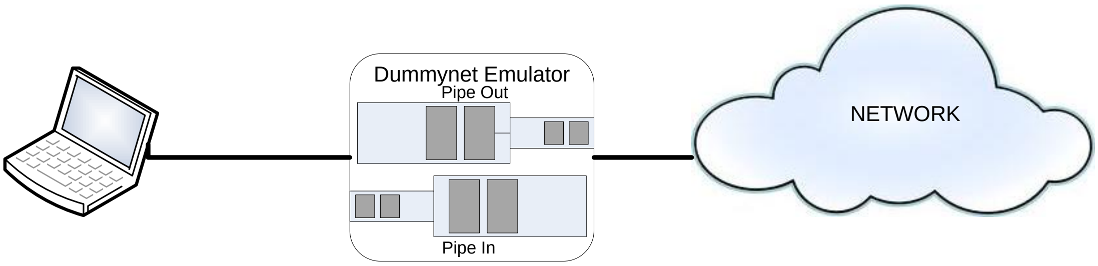


Figure 3.4: Dummynet insertion in an existing network without introducing any changes.

without changing any of the existing software installation or disrupting the current network setup.

Previous network emulators had mostly applied an approach where the total factors affecting network as (delays, loss patterns, reordering,...etc.) are emulated through certain network configurations. That approach cause too large approximations when modelling those factors separately, as those factors are substantially dependent on the actual traffic patterns. Consequently, Dummynet was designed based on a different approach, which is to emulate the basic components of the IP network and to provide tools to connect these components in a simple and flexible way. Therefore, in order to emulate conditions like congestion loss or multiple paths management, Dummynet components and tools are used to get the required behaviour and the expected results.

Pipe (Figure 3.5) is the main object of Dummynet. It consists of a queue, and a communication link whose bandwidth (bw), delay and packet loss rate (plr) are programmable. Dummynet enables the creation of multiple pipes at the same running instance to meet with the required design to be emulated. One-line commands are used to reconfigure pipe parameters dynamically, for example:

Listing 3.1: pipe example

```
ipfw pipe 1 config delay 20ms bw 300Kbit/s plr 0.1
```

Where pipe 1 is configured to delay the packets that pass through it by 20ms in a bandwidth of 300 Kbit/s with packet loss percentage of 10%. `ipfw` is the packet

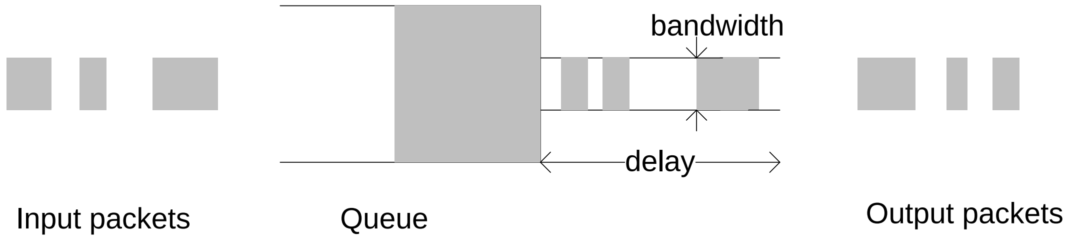


Figure 3.5: Pipe is the main component of Dummynet

classifier, its function is to pass the traffic to pipes by applying a list of numbered rules(ruleset).

Listing 3.2: Asymmetric pipe configuration

```
ipfw pipe 1 config bw 512Kbit/s delay 13ms
ipfw pipe 2 config bw 3000Kbit/s delay 2ms

ipfw add 100 pipe 1 in src ip serverx.com
ipfw add 200 pipe 2 out dst ip serverx.com
```

In the previous example, we create two rules with numbers 100 and 200, each is responsible for a different direction. Each direction has a different settings which emulates asymmetrical link as ADSL. We have used different forms of the previous commands extensively during our experiments to emulate various network scenarios and conditions in order to be able to measure the impact on the quality of audio and video signals transmitted in the network.

3.2.2 Iperf

Iperf (ipe, 2013) is an open source tool, written in C++. It is used mainly for measuring and monitoring the performance of network links. Iperf is widely used as it runs on various platforms including Windows, Unix and Linux. Furthermore, it can be installed over any network, and it produces performance measurements that meets with the standards. Additionally, it offers the capability to compare between wireless and wired networking technologies and devices.

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are transport protocols which represent one of the core protocols of the network protocol

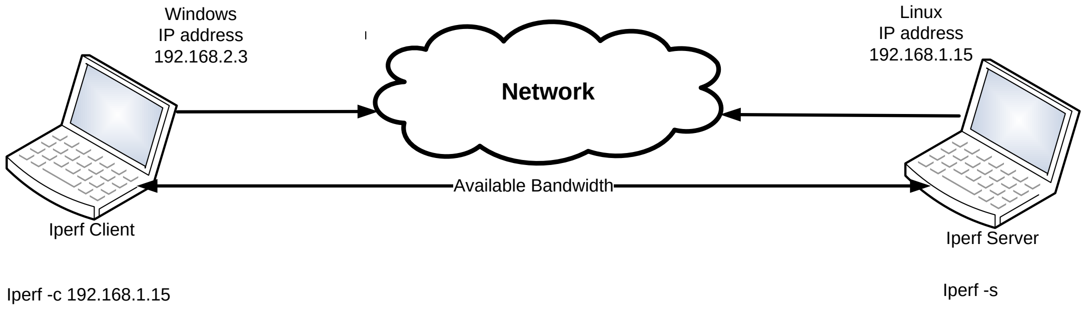


Figure 3.6: Iperf setup

suite. Iperf employs the usage of both protocols in performing its tests in order to provide information about the capability of different network links.

Iperf is based on client-server architecture as shown in Figure 3.6, where the network link is represented by two hosts running Iperf. In order to test the performance of network links, the main factors which are considered are delay, jitter, packet loss rate, and throughput. Delay (RTT Round Trip Time) is calculated using Ping Command by sending Internet Control Message Protocol (ICMP) Echo Request packets to the destination host and waiting for a response. Iperf UDP test can be carried out to measure jitter (delay variation) and for calculating the packet loss rate. Moreover, Iperf can measure the bandwidth between two ends, either unidirectionally or bi-directionally. Consequently, it can be used for optimizing and tuning IP networks.

We have used Iperf during our experiments to emulate the traffic of voice and video data over the networks, as Iperf provides the functionality of setting the size of packets so it that it could be configured in accordance with the used codec to establish the call, then based on that we can measure the network parameters, and hence estimate the quality of calls.

3.2.3 Imunes

Imunes is a network emulator (Zec & Mikuc, 2004) based on the FreeBSD operating system. It is considered as a valuable alternative for live testbed networks which often require real hardware as servers, routers and physical links to be connected together and configured to shape a network of the required topology, and to produce different

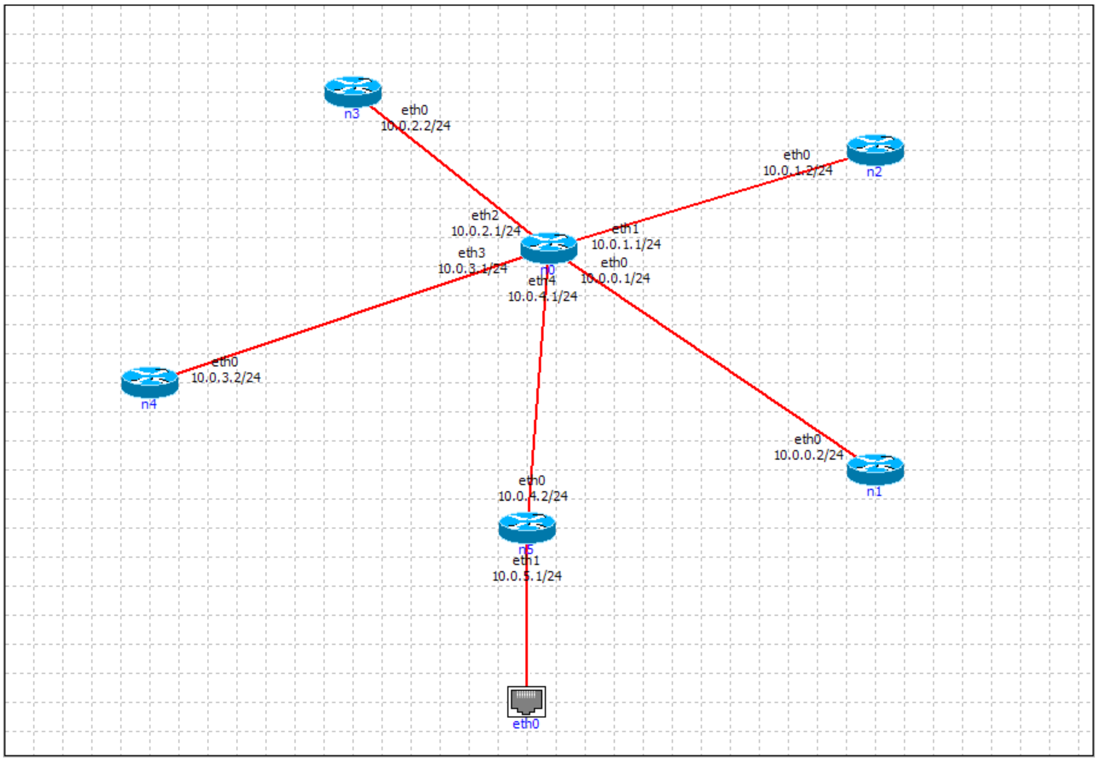


Figure 3.7: Star Topology emulated by Imunes

network characteristics such as latency, bandwidth and packet loss rate. They offer a degree of realism which as a result can lead to more accurate results; however, they are hard to implement and maintain, costly, and time consuming. Network emulators are typically a combination of testbeds and simulators which can provide real network traffic to a virtual network environment. Single hop emulators as Dummynet can provide synthetic network conditions, but it doesn't offer the ability to emulate various network topologies unlike Imunes.

It provides the functionality of designing the common network topologies such as star, chain, cycle and wheel topologies. Figure 3.7 illustrates star topology. What Imunes offer here is the interconnection between this virtual nodes in the star topology and physical real devices. By using the ethernet in the design (eth0) which is available at the machine running Imunes, connecting virtual nodes with external physical devices is made possible.

Virtual nodes represents the main component of Imunes. Each virtual node consists of an instance of network stack and user space processes. Each network stack instance has its own set of private properties such as routing tables, network interfaces and set of communication sockets. Each node has it own independent copy of the full network stack, moreover, they are interconnected together via kernel-level links. Every virtual node can operate either as UNIX end host, or as an IP router that contains all features of routing algorithms like Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) without making any degradation in the throughput or performance if compared to the real physical devices. Furthermore, it supports different queueing techniques including FIFO and Deficit Round Robin (DFR).

Virtual links are used for communication between virtual nodes based on netgraph framework which is a component of FreeBSD. Virtual links enables features such as bandwidth limiting, latency simulation and simulation of bit error-rate (BER) which can be used hand-in-hand with Dummynet to offer a full configurable emulated network.

3.2.4 Wireshark

Wireshark (wir, 2013) is the most widely used network protocol analyzer. It captures network packets and lists all the information presented in them in an appropriate user friendly manner. Furthermore, it provides live capturing for many network media types including Wireless LAN, Ethernet, and many others. It used for a lot of purposes such as network administration, testing security of networks and for debugging and testing of protocols implementations. Wireshark does not introduce any changes in the network and it does not manipulate network by sending additional packets from its own—this is why it is considered a reliable and secure method of monitoring networks. It can filter certain packets from the network flow, packets like RTP, and SIP messages are very useful to separate and study the contained info. Moreover, it can provide important statistics, such as the loss rate, packet length, and packet sequence numbers.

3.3 Testbed Setup

The main concept of our testing procedures is to establish large number of VoIP calls with different codecs, and under varying network conditions. The varying parameters of network are: packet loss rate, delay, bandwidth, and jitter. And then estimate the QoE whether offline using PESQ or online using the E-Model.

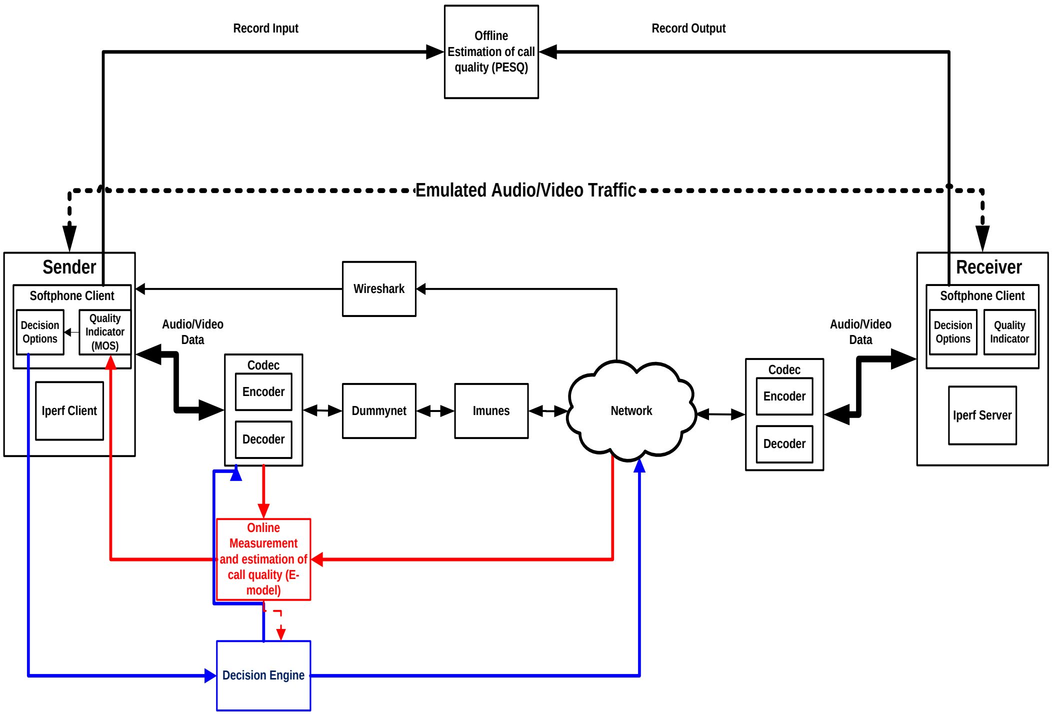


Figure 3.8: Testbed Architecture

The experiments were performed in a real enterprise network of IBM, using their VoIP product (IBM SUT), and the open source VoIP application (Jitsi). We ensured to use a lightly weighted network (clean) with 1 Gbps bandwidth, 0% packet loss rate, average round trip delay of 14 ms, and jitter range between 3-7 ms.

Audio test samples used in the experiments were taken from Rec (2004) in order to stick to the standard recommendations of testing telephony applications to avoid misleading results. The total sample duration varies from 8 to 12 seconds, and consists of a pair of sentences separated by a silent gap. Sentences are selected to be short, simple and meaningful. The audio sample are encoded in PCM format, with a bit rate of 256 kbps, sample size of 16 bit, and sampling rate of 16 kHz.

Figure 3.8 illustrates how we placed the previous components together in order to have a reliable framework to establish calls in various network scenarios and topologies, then to estimate and monitor the perceived call quality. Finally, to take accurate decisions based on the current measures to improve the call quality without the need of dropping the current call and establish new one. Decision like switching the codec used to achieve the best possible quality-of-experience(QoE) to users.

The framework was developed as a plugin of IBM SUT to be used automatically through running JAVA code, or manually in Jitsi. Audio streams at both sender, and receiver sides are recorded in order to run PESQ test. Sender size takes audio files as an input, which are then encoded using the specified codec then streamed through the IP network to the receiver machine. Before sending audio packets, network characteristics can be manipulated to add different degradations to the network. Meanwhile, on the receiver side, all the incoming packets are monitored so as to acquire packet loss rate, delay, and jitter by analysing the packet trace in Wireshark. These network parameters are then fed into the E-Model in order to estimate the QoE online, and compute the MOS score.

Chapter 4

Online Estimation of VVoIP Quality-of-Experience via Network Emulation

We describe a testing tool that can provide online estimates of audio and video call quality on network paths, without requiring either end-user involvement or prior availability of audio/video sequences or network traces. The tool includes a tool that emulates the audio and video traffic of IP calls and employs an extended E-Model to measure the audio quality and VQM to estimate video quality. Additionally, it can emulate network impairments to run experiments in different network conditions. Our experiment results show that the quality measurements acquired using the tool compare well to the most commonly applied industry standard for objective voice and video offline testing—PESQ and PSNR respectively.

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 tool that emulates the traffic of multiple audio and video codecs in order to estimate the voice/video call quality in advance. Such estimates can then be used to select the most appropriate codec to use for upcoming calls. Crucially, this processes does not require the transfer of source audio/video sequences and does not require the end user to provide quality rankings.

4.1 Tool components

This tool was implemented using Java programming language. Distributed application always uses Java programming which it split up with the client-server model and provides real distributed processing that is appropriate for developing Internet applications (da Silva & Lins, 2006). We used NetBeans Integrated Development Environment (IDE) version 7.1. We have chosen NetBeans among other Java editors (e.g.: Eclipse, BlueJ, etc.) as it provides the capability for Graphical User Interface (GUI) development as others need source codes. Architecture of the tool is shown in Figure 4.1.

Our tool uses Iperf (ipe, 2013) to measure packet loss, jitter and throughput. Iperf is a networking tool that creates TCP and UDP data streams of specified size; it runs on various platforms including Linux, UNIX and Windows.

The voice and video packets are sent using UDP. Consequently, the exact measurement of the delay between the sender and destination is not directly measured. We use Ping to send Internet Control Message Protocol (ICMP) echo request packets to the target destination and wait for the ICMP response. To get an accurate measurement of the delay we emulate the Ethernet-layer bandwidth according to the codec using Dummynet (Carbone & Rizzo, 2010) and set the ping parameters based on the codec used.

We use Dummynet in our tool for two purposes. First, 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, to set the bandwidth with the Ethernet

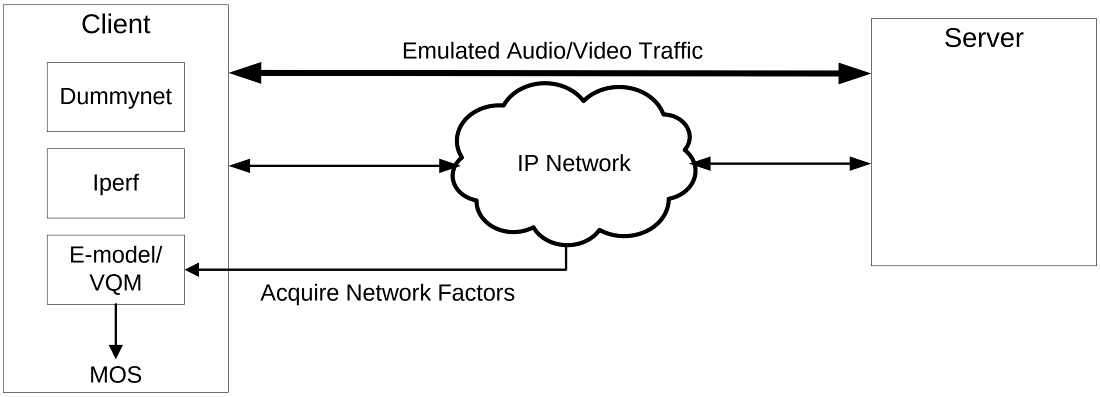


Figure 4.1: Tool Architecture

bandwidth according to the codec emulated in order to measure accurate delay results with the current browsing sessions if any on the computer.

4.2 Development of the Tool

The proposed tool 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. The Packet size (Ps) and the Ethernet bandwidth (Eb) varies from codec to another. In our tool we calculated them as:

$$Ps = Fs \times framesPerPacket + ipHeader + eOverHead \tag{4.1}$$

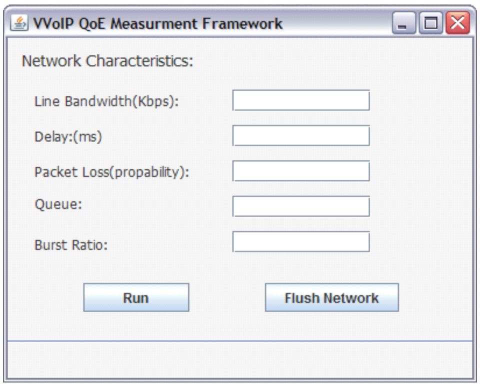
$$Eb = Ps \times (\frac{bw}{Fs})_{Codec} \tag{4.2}$$

Ps is the total packet size, Fs is the frame size according to the codec (see Table I), $framesPerPacket$ 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. Roughly,

Table 4.1: Mean Packet Length Estimates for H.264.

Bandwidth	**Mean packet length**
Kbit/s	*Bytes*
300 (Low Quality)	316
500 (High Quality)	637
1500 (HD)	885



VVoIP QoE Measurment Framework

Network Characteristics:

Line Bandwidth(Kbps):

Delay:(ms)

Packet Loss(propability):

Queue:

Burst Ratio:

Figure 4.2: Inputting Data for Network Emulation.

for transmission of low quality video, 300 Kbit/s of available bandwidth is needed, whilst for high quality 500 Kbit/s would be required. HD video requires a minimum of 1.5 Mbit/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 4.1.

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. Fig 4.2 shows the dialogue box for inputting this data. The IP destination address, port number, codec used and frames/packet are the main inputs before running the testing tool; Dummynet will then emulate the network conditions. The delay is measured using Ping command taking in its account the packet size and the sending bit rate of the codec used as calculated in Eq.4.1,4.2. Iperf is called to measure the packet loss percentage, throughput and jitter by specifying Datagram size (Eq.4.1) and Ethernet Bandwidth (Eq.4.2) for audio, or by using Table 4.1 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:

$$nOfCalls = \lfloor \frac{throughput}{Eb} \rfloor \tag{4.3}$$

$nOfCalls$ 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 At the end the QoS parameters measured are mapped to QoE MOS score using E-model and Video Quality Model(VQM) described in Chapter 2. The pseudocode for this process is shown in

Algorithm 1.

```
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
    Step1: Emulate Network.
    Initialize Dummynet emulator by loading kernel module;
    Emulate network conditions (Line Bandwidth, Delay, Random Packet loss, Burst Ratio, Queue
    length);
    Step2: Initialize Test.
    Check codec selected for call;
    Specify packet size, inter-packet time and sending bit rate;
    Step3: Begin Test.
    Counter = 0;
    while Counter less than 5 do
        Start Packet trains from source to destination;
        Measure one-way delay using ICMP request;
        Measure packet loss, throughput and jitter using Iperf;
        Increment Counter;
    end
    Calculate average results for one way delay, packet loss, throughput and jitter;
    Calculate link capability (No of Calls);
    Step4: Display Measured QoS factors.
    Display previous extracted data;
    Calculate QoE MOS score using E-model for audio and VQM for video;
    Display MOS score and user satisfaction level;
    Step5: End Test.
    Flush all inbound/outbound pipes of Dummynet;
end
```

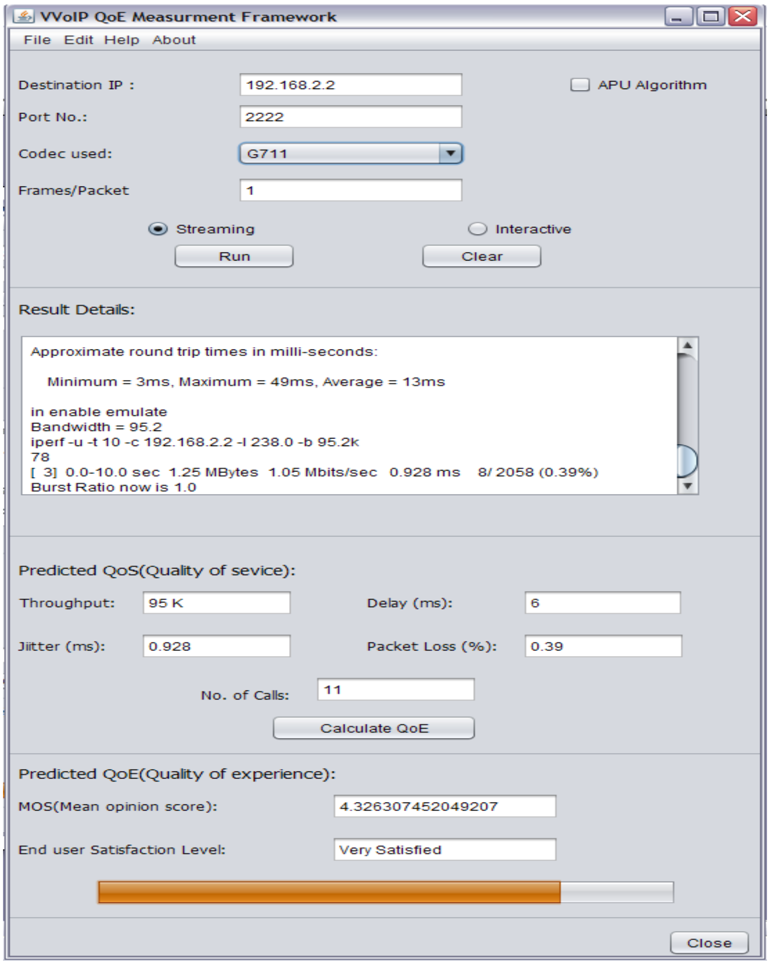
Algorithm 1: QoE Estimation Process.

4.3 Results and Discussion

In this section, we provide the results of our QoE estimation process for voice and video in comparison 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 results, we used a beta version of IBM Sametime Unified Telephony (SUT) (ibm, 2013) product, measuring the audio/video call quality under different packet loss rates using Dummynet. We have compared these to offline audio and video testing using PESQ and PSNR respectively. We first outline the results for audio and then outline the video results.

4.3.1 Audio Testing

A screenshot of the configuration dialog box for audio testing is shown in Figure 4.3. Tests are carried out on several codecs: G711, G723.1 5.3k, G723.1 6.4k, G726, G729, G729 A, GSM FR, SILK, ILBC and SPEEX. We show a sample of our results in Figure 4.4 and 4.6. The x-axis represents the packet loss rate ranges from 0-20% and the y-axis indicates the MOS from the tool and PESQ algorithm. Our results match well with the PESQ scores, confirming the accuracy of our approach. We observe in our results that we slightly underestimate MOS compared to scores produced from PESQ. This can be explained by the observation that we take into our account the delay impairment factor (conversational call quality) while the intrusive methods as PESQ do not take it into consideration.



Destination IP : 192.168.2.2

□ APU Algorithm

Port No.: 2222

Codec used: G711

Frames/Packet 1

● Streaming ○ Interactive

Result Details:

```
Approximate round trip times in milli-seconds:

   Minimum = 3ms, Maximum = 49ms, Average = 13ms

in enable emulate
Bandwidth = 95.2
iperf -u -t 10 -c 192.168.2.2 -l 238.0 -b 95.2k
78
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.928 ms   8/ 2058 (0.39%)
Burst Ratio now is 1.0
```

Predicted QoS(Quality of sevice):

Throughput: 95 K

Delay (ms): 6

Jitter (ms): 0.928

Packet Loss (%): 0.39

No. of Calls: 11

Predicted QoE(Quality of experience):

MOS(Mean opinion score): 4.326307452049207

End user Satisfaction Level: Very Satisfied

Figure 4.3: Screenshot of Audio Testing GUI.

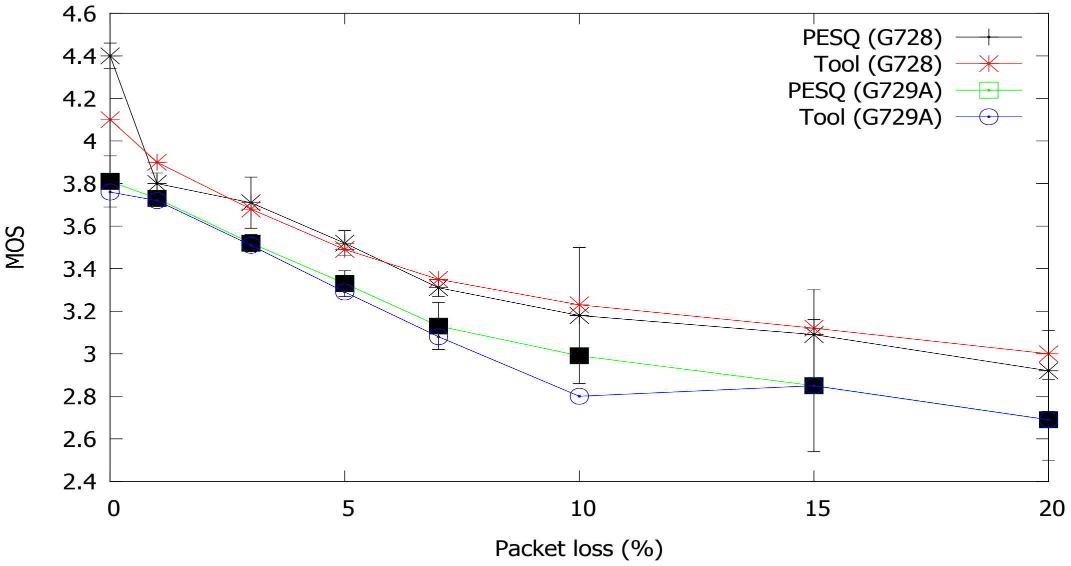


Figure 4.4: MOS estimations for G.728 and G.729A audio codecs.

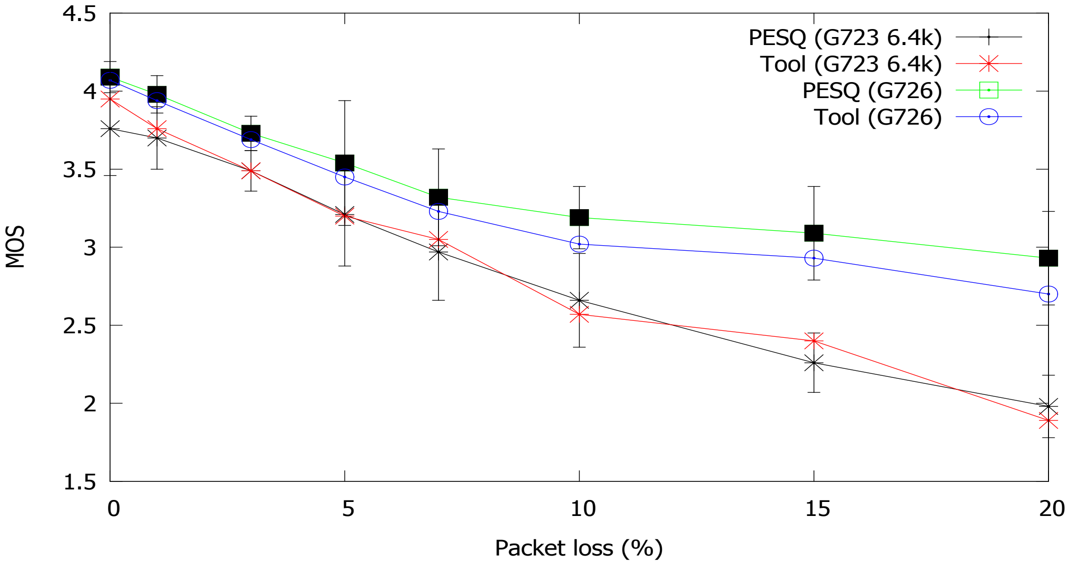


Figure 4.5: MOS estimations for G.723 6.4K and G.726 audio codecs.

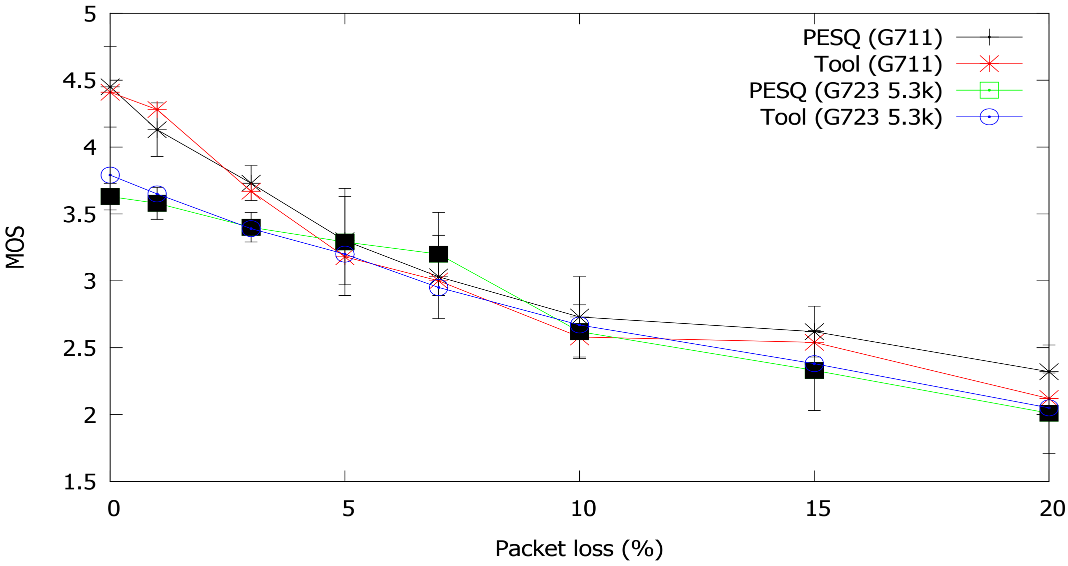
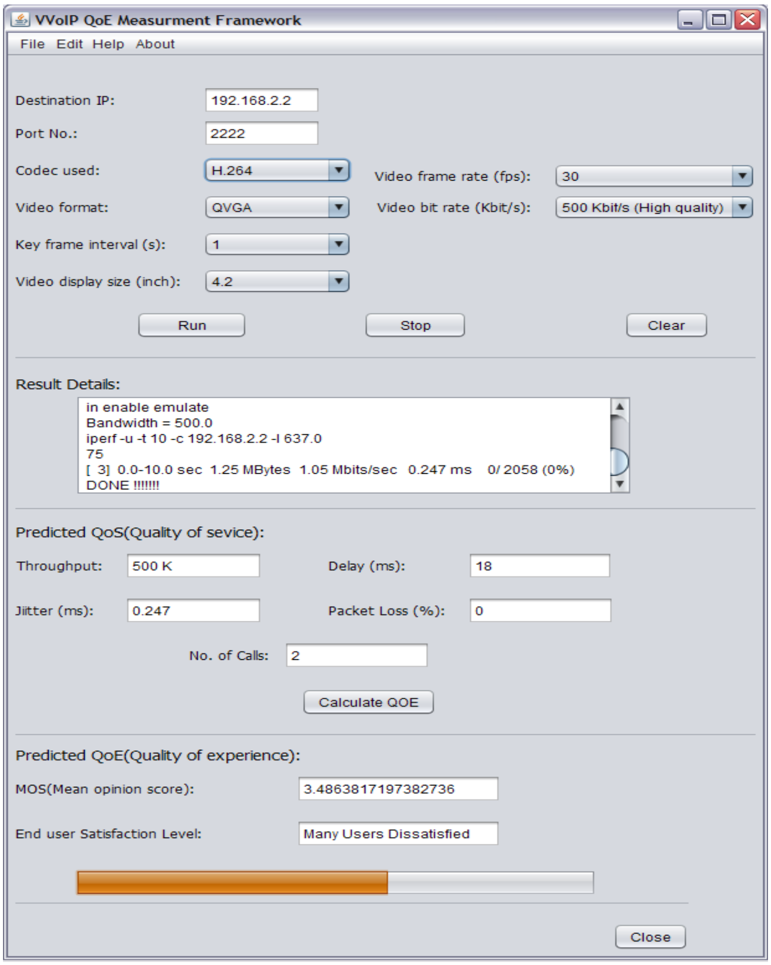


Figure 4.6: MOS estimations for G.711 and G.723 5.3K audio codecs.

4.3.2 Video Testing

Figure 4.7 shows a screenshot of our configuration dialog box for video testing. We compared our results to real time PSNR values of H.264 codec after converting them to MOS values. Table IV (derived by Ohm (2004)) 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 4.2 by assuming that the relation between MOS and PSNR inside these regions is linear.



Destination IP: 192.168.2.2

Port No.: 2222

Codec used: H.264

Video frame rate (fps): 30

Video format: QVGA

Video bit rate (Kbit/s): 500 Kbit/s (High quality)

Key frame interval (s): 1

Video display size (inch): 4.2

Result Details:

```
in enable emulate
Bandwidth = 500.0
iperf -u -t 10 -c 192.168.2.2 -l 637.0
75
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.247 ms    0/ 2058 (0%)
DONE !!!!!!!
```

Predicted QoS(Quality of sevice):

Throughput: 500 K

Delay (ms): 18

Jitter (ms): 0.247

Packet Loss (%): 0

No. of Calls: 2

Predicted QoE(Quality of experience):

MOS(Mean opinion score): 3.4863817197382736

End user Satisfaction Level: Many Users Dissatisfied

Figure 4.7: Screenshot of Video Testing GUI

Table 4.2: PSNR to MOS

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

$$MOS = \begin{cases} 5 & \text{if } PSNR > 37 \\ 0.15 \times PSNR - 0.65 & \text{if } 31 \leq PSNR \leq 37 \\ 0.153 \times PSNR - 0.813 & \text{if } 25 \leq PSNR \leq 31 \\ 0.184 \times PSNR - 1.673 & \text{if } 20 \leq PSNR \leq 25 \\ 1 & \text{if } PSNR < 20 \end{cases} \tag{4.4}$$

We show sample of our results for two resolutions, QQVGA (160x120) and QVGA (320x240), with frame rates of 15 fps and 25 fps respectively. The comparison is presented in the Figs 4.8 and 4.9. The x-axis represents the packet loss rate ranges from 0-6% and the y-axis represent the MOS score of the tool and equivalent PSNR values.

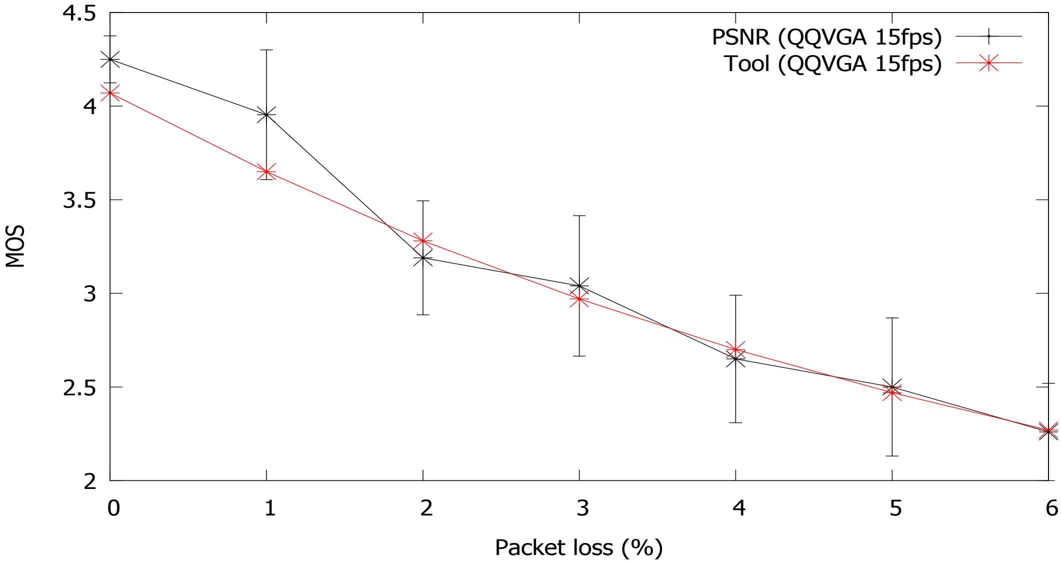


Figure 4.8: QQVGA at 15 fps and bitrate of 300 Kbit/s.

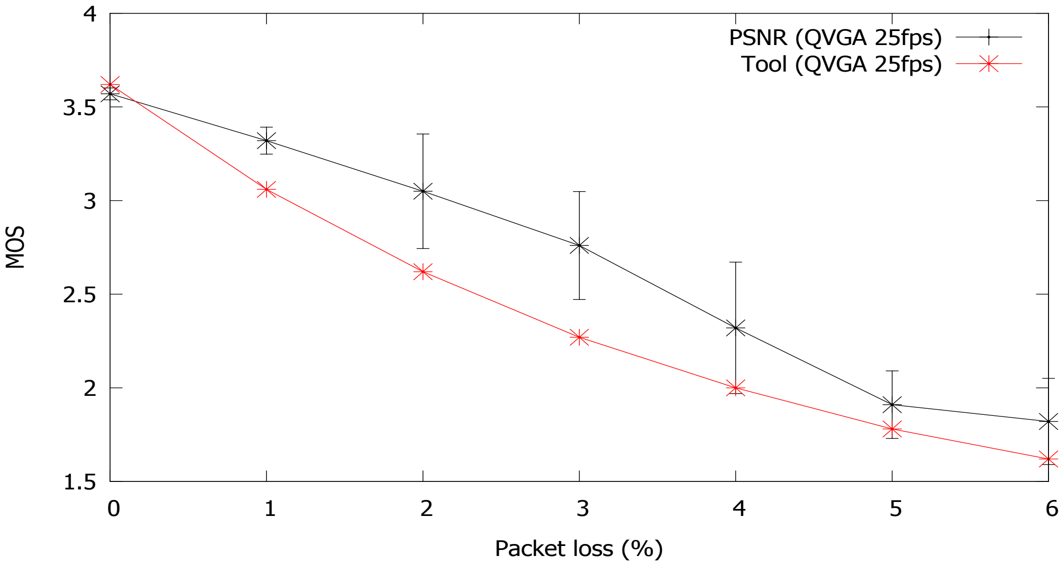


Figure 4.9: QVGA at 25 fps and bitrate of 500 Kbit/s.

Our tool produces acceptable results for multiple audio and video codecs under different packet loss rates compared to the commonly-used objective testing methods for estimating audio and video quality.

4.4 Summary

Since processing audio/video sequences is time consuming and computationally intensive, existing objective QoE estimation techniques are not suited for online use. Furthermore, because audio/video codes each have different characteristics it is very difficult to use these techniques to assess in advance which is the codec most appropriate for use giving the prevailing network conditions. To address these limitations we have developed a QoE estimation tool for audio/video that does not require transfer of audio/video sequences or end user involvement. Our experiments show that our tool can achieve acceptable results in comparison to those achieved using the most commonly used industry techniques for audio and video quality testing; PESQ and PSNR respectively.

Chapter 5

A Generic Algorithm for Mid-call Audio Codec Switching

We present and evaluate 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. We have studied the behaviour 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 voice call quality for them. Moreover, we have studied the drawbacks of codec switching from the end user perception point of view—our switching algorithm seeks to minimise this impact. We have tested our algorithm on different packages that contain a selection of the most commonly used codecs: G.711, SILK, ILBC, GSM and SPEEX. Our results show that in many typical network scenarios, our switching codecs mid-call algorithm results in better Quality of Experience (QoE) than would have been achieved had the initial codec been used throughout the call.

As described in Chapter 2, there are different methods to measure the voice quality accurately in the VoIP networks. E-model, specified in ITU-T Rec. G.107 (Rec, 2009), 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 a so-called "impairment factor" of the codec, which ITU-T provide for codecs they specify, but which is not specified for a range of other commonly used codecs. In this chapter we drive the impairment factors for 4 widely used non-ITU codecs, furthermore, we present a generic codec switching algorithm that can respond to changing network conditions during an ongoing call and switch to the most appropriate codec.

5.1 Deriving E-model for non ITU-T codecs

Although the new objective E-model (Equations 5.1,5.2) 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 provided nor can be calculated easily. Recently, there has been a great progress in the non-ITU codecs which are used widely now in VoIP applications (e.g.: Skype, G-talk). Thus we seek to derive the codec factors for some widely used non-ITU-T codecs.

$$R = 93.2 - I_d - I_{e-eff} \tag{5.1}$$

$$I_{e-eff} = I_e + (95 - I_e)\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{5.2}$$

ITU-T recommendation G.113 (ITU-T, 2001) does not provide codec I_e, Bpl values for the most well know 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. We then calculate the E-model R value using the following 3$^{\text{rd}}$ order polynomial fitting from (Sun, 2004):

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.06MOS - 57.336 \tag{5.3}$$

The MOS (PESQ) factor converted to rating factor R does not consider delay impairments (I_d value). Hence, we consider only the equipment impairment, I_{e-eff},

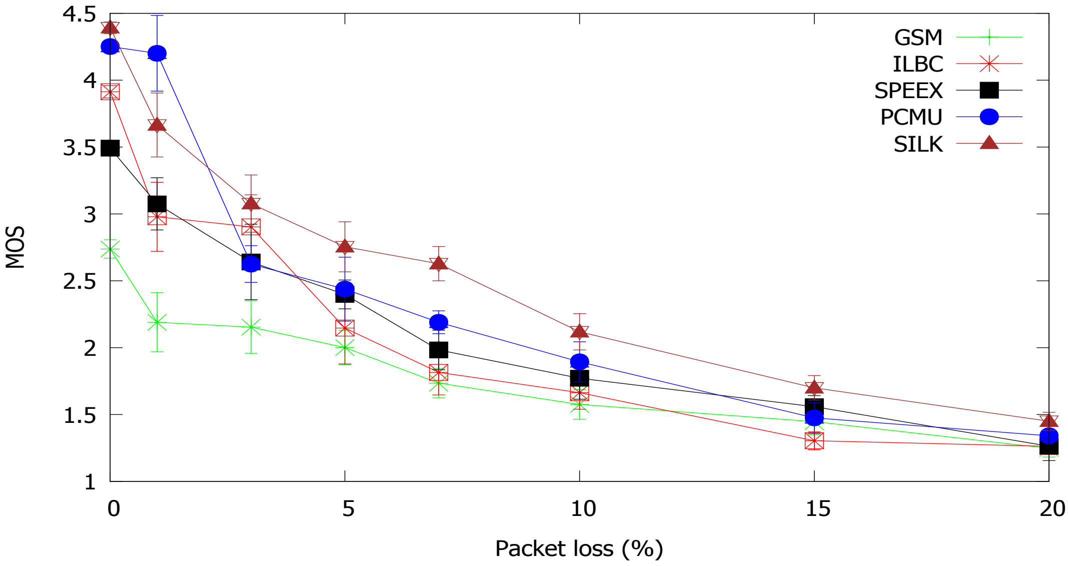


Figure 5.1: codecs performance.

which results from the codec compression rate and packet loss. Therefore, following from (2.3), R can be converted to I_{e-eff} as

$$I_{e-eff} = 93.2 - R \tag{5.4}$$

We used PESQ to estimate the MOS for the popular G711, ILBC, SILK, GSM and SPEEX codecs at different packet loss rate ranges from $0-20\%$. We have established 5 new calls at each packet loss rate of each codec. We recorded the audio signal from the sender and receiver side removing any delay effect as a result of recording. We apply the PESQ algorithm for each original and degraded pair to measure the MOS score. Each MOS estimation at each percentage of packet loss was measured 5 times and we took the average in order to increase the accuracy of our results. We used Dummynet (Carbone & Rizzo, 2010) to embed random packet loss rates during the session. Our results are shown in Figure 5.1 with the packet loss on the x-axis and the PESQ MOS score on the y-axis.

We observe that the performance of the codecs is different under packet loss rates. For example, SILK out performs the other codecs at 0% packet loss rate. PCMU gives the best performance in the range $0-3\%$ packet loss. Starting nearly from 4% packet

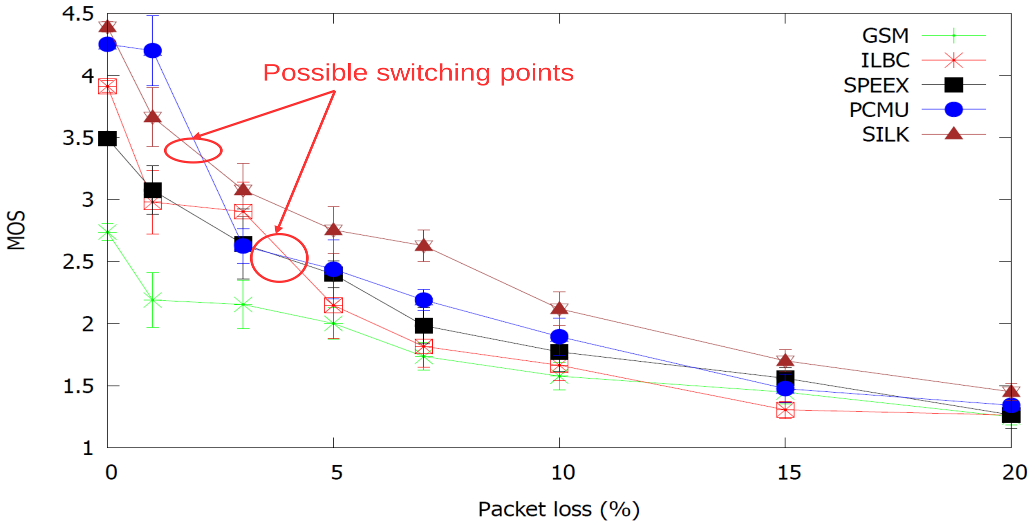


Figure 5.2: Possibilities of codec switching.

Table 5.1: Derived Linear Regression Model Parameters for Different Codecs.

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

loss, we found that SILK over performs until 20% packet loss. These observations suggest that switching codecs mid-session in response to increased in detected packet loss rate has the potential to deliver an improved QoE as illustrated in Figure 5.2.

In Figure 5.3, a non linear regression model (similar to the logarithmic function in Sun & Ifeachor (2006)) can be derived for each codec by the least squares method and curve fitting. The derived I_{e-eff} model has the following form:

$$I_{e-eff} = a \log(1 + b \times Ppl) + c \tag{5.5}$$

The Ppl in (5.5) is the packet loss rate in percentage and the parameters (a, b, and c) are shown in Table 5.1 for the different codecs.

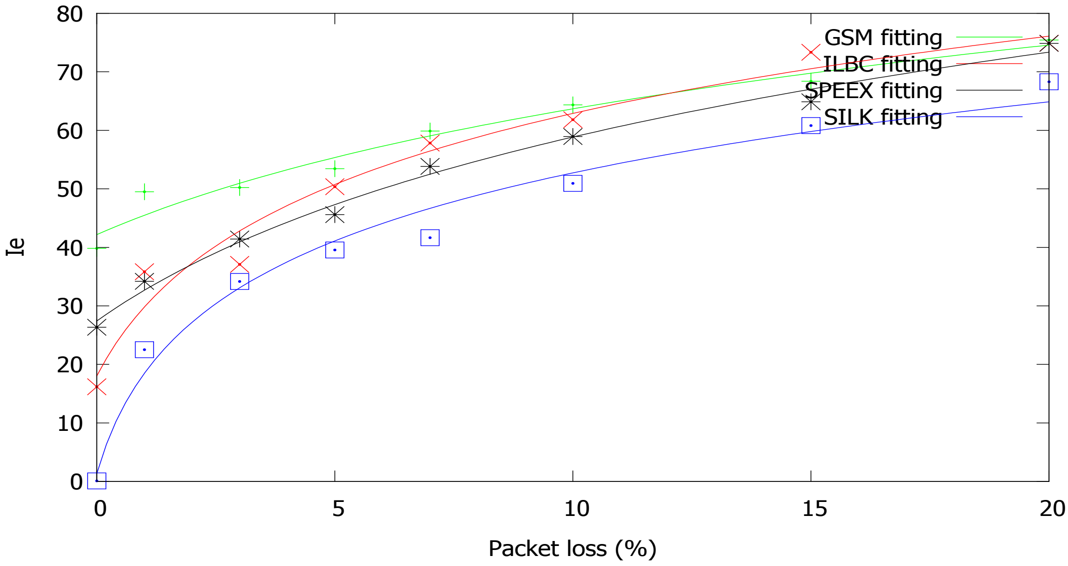


Figure 5.3: Deriving I_e factor for four non-ITU codecs.

5.2 Impact of Codec Switching

Codec switching is done through Session Initiation and Session Description Protocols (SIP/SDP). 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 handshake procedure explained in Chapter 2 §2.1.2 is used in order to agree on a common codec and other session parameters when establishing a call.

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.

In the following sections we study the impact of the codec switching process itself. Two factors can lead to degraded quality: the "switch-over gap" when codecs are switched and the overall number of switches during a session.

5.2.1 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 for high switching gap which will lead 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 till the message reaches the intended receiver, and the same also will happen for the 200 OK and ACK messages, therefore 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, thus we have measured the switch-over time between G711 and ILBC with a packet loss rate ranges from $0 - 40\%$. At each packet loss rate, we have measured 10 values for the switching-over gap measured in $msec$. For doing so, we used Wireshark to capture the SIP packets sent during the re-negotiation of using a new codec during the call, we measured the time difference between the RE-INVITE and ACK messages captured by Wireshark. Figure 5.4 shows our results indicating the packet loss percentage on the x-axis while the switch-over gap on the y-axis.

We divided the y axis in Figure 5.4 into three distinct regions based on the average of the switch-over gap. The first region which is between 0-10% packet loss 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 ranges 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

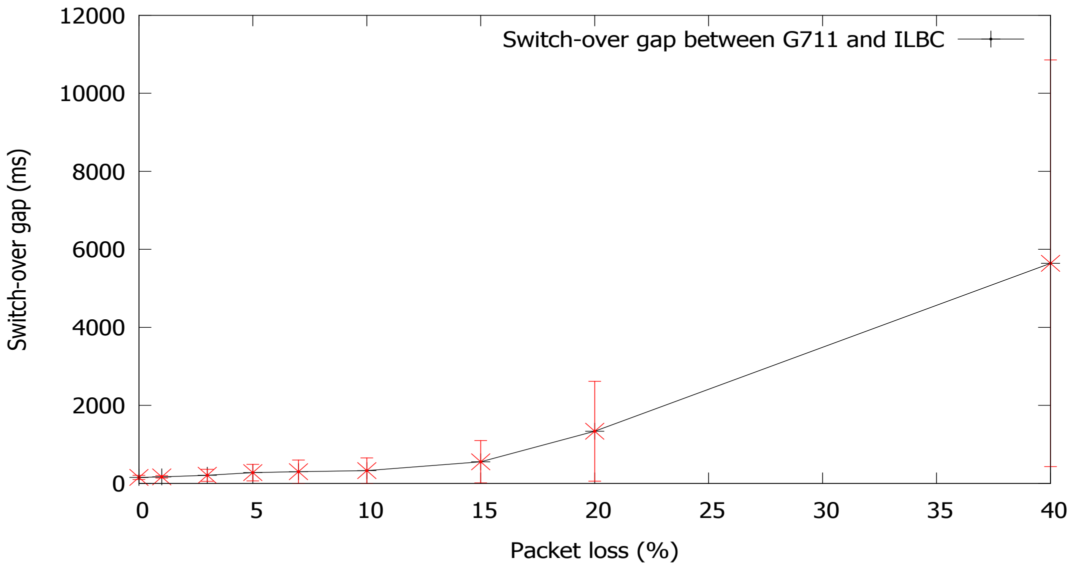


Figure 5.4: Switch-over Gap Effect.

region between 20 – 40% packet loss, it is not recommended to switch as the switching-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 in order to increase the responsiveness of our algorithm.

5.2.2 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 (Schulzrinne *et al.*, 2003).

In order to measure the 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 or almost same performance. From Figure 5.1, we observe that at 0% percent packet loss

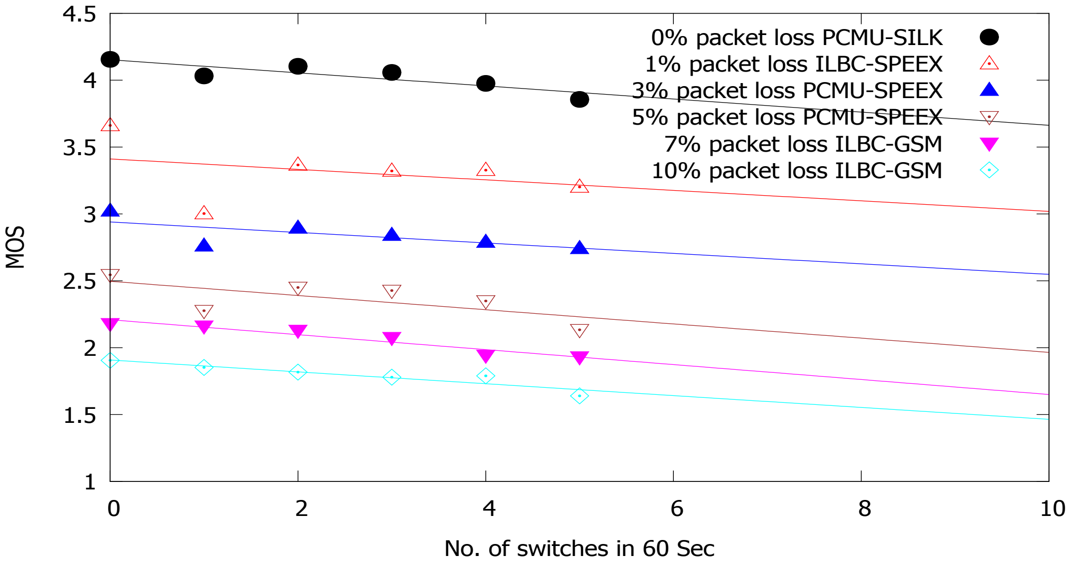


Figure 5.5: Effect of Number of Switching on MOS Score.

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 established different calls with a 60 seconds playing audio file using the stated pair of codecs. We switched several times between them during the call, recording from both ends the audio signals and finally applying the PESQ algorithm to measure the MOS score. The results are shown in Figure 5.5: we see that the relation between the number of switches and the MOS score is well matched by 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 $0 - 10\%$. 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 5.5 from the degradation in the MOS when there is no switching compared to 1 switch.

5.3 Codec Switching Algorithm

We now specify a codec switching algorithm that can be used in conjunction with an arbitrary set of codecs available to a VoIP application. The algorithm is based on the use of the E-model to estimate MOS during an ongoing voice session. Thus, it requires knowledge of the impairment factor for different codecs; we have derived these values for a range of common codecs above and ITU-T specifies them for their codecs.

The algorithm, specified in Algorithm 2, operates as follows. It assumes the call starts with a default codec set in the VoIP application. It then waits for every 2 successive RTCP reports (a control period of $10s$), each time calculating the current average packet loss rate. If the packet loss rates is in the $0-10\%$ range, the algorithm estimates the predicted call quality using the E-model for all of the other codecs available to the VoIP application. Once this is done, the MOS scores for the other codecs are compared to the one currently in use and, if the score would be improved by making a switch this is done. This decision takes into account the potential degradation in MOS due to frequent codec switching.

5.4 Experimental Analysis

We implemented our codec switching algorithm in Jitsi (jit, 2013), an open source audio/video Internet phone and instant messenger written in Java. We used Dummynet to emulate a range of typical network conditions. To test the codec switching algorithm, we evaluate its use with three "packages" of codecs with one default codec, as specified in Table 5.2. For our experiments we played a sample audio file for $3mins$, with the potential for switching a codec being assessed every $10s$.

5.4.1 First Package

In this experiment and as shown in Figure 5.6, we started the call using GSM codec at 0% packet loss; it took the algorithm $10s$ to switch to SILK which has the highest R at this loss rate. For the next $60s$ the MOS for all codecs is degraded by 0.1 as a result of switching. After the end of the previous $60s$, the MOS recovered from the negative

```
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 $\leq$ 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
```

Algorithm 2: Codec Switching Algorithm.

effect of the switching and returns back to its value as shown in the time slice between 1:20 and 1:30. At 1:20 we emulated 1% packet loss, so the switching occurred at 1:30 to PCMU. After 40s and although the packet loss was increased to 5%, switching didn't occur at the 2:20 as one switch was already done in the previous 60s (-0.1 MOS) and the gain from such switch between PCMU and SILK (+0.1 MOS) was not worthwhile

Table 5.2: Codec Packages

Package	Codecs Present	Default Codec
First	SILK PCMU GSM	GSM
Second	SPEEX ILBC GSM	GSM
Third	ILBC GSM	ILBC

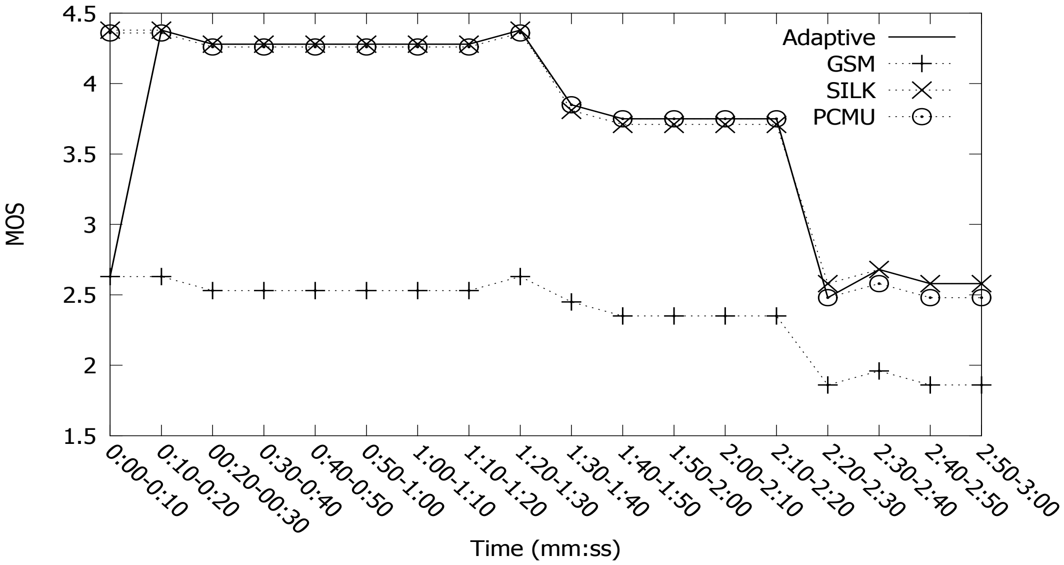


Figure 5.6: MOS values for the First Package. Switching between GSM, SILK, and PCMU results better MOS than using only one of them through out the call.

in the context of overall call quality. Finally, at 2:30 the codec was switched to SILK.

5.4.2 Second Package

In this experiment, as shown in Figure 5.7, we started the call using GSM codec at 0% packet loss; it took the algorithm 10*s* to switch to ILBC. At the 1:30, we applied a packet loss of 6%. Thus, the codec was switched to SPEEX in the next slice. At

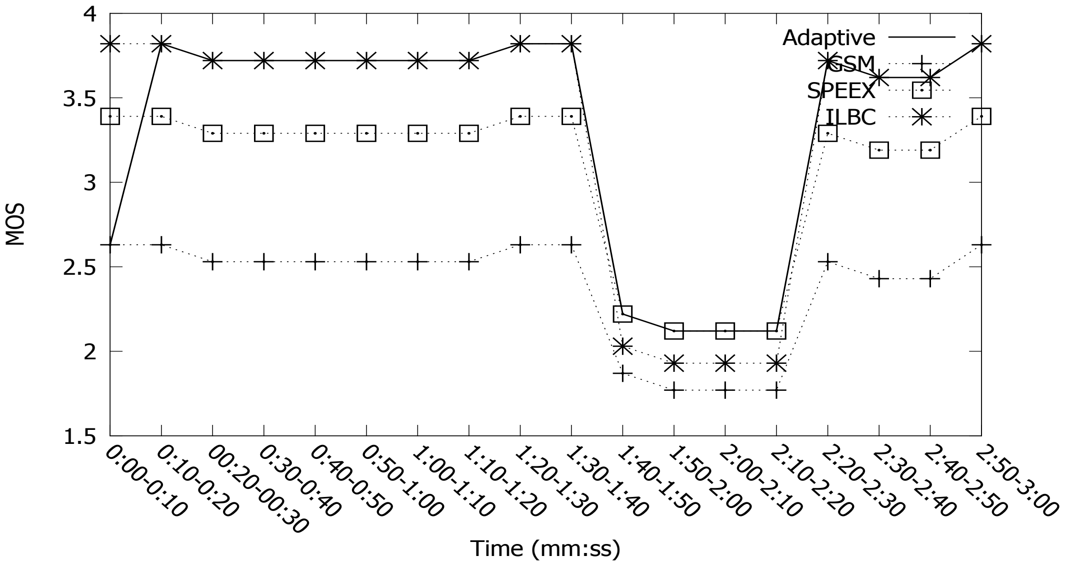


Figure 5.7: MOS values for the Second Package.

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*s* (-0.1MOS) it is worth switching as the total gain expected from such switch will be +0.33 MOS. In 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*s* from the first switch at 1:40, the MOS returned back to its normal value at current packet loss rate.

5.4.3 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 5.8 no switching occurred at all as GSM has always a lower R value which will not be worth at any point to switch to the ILBC codec.

5.5 Summary

Switching codecs during an ongoing voice session can improve users perceived Quality-of-Experience due to the fact that different codecs behave differently under different

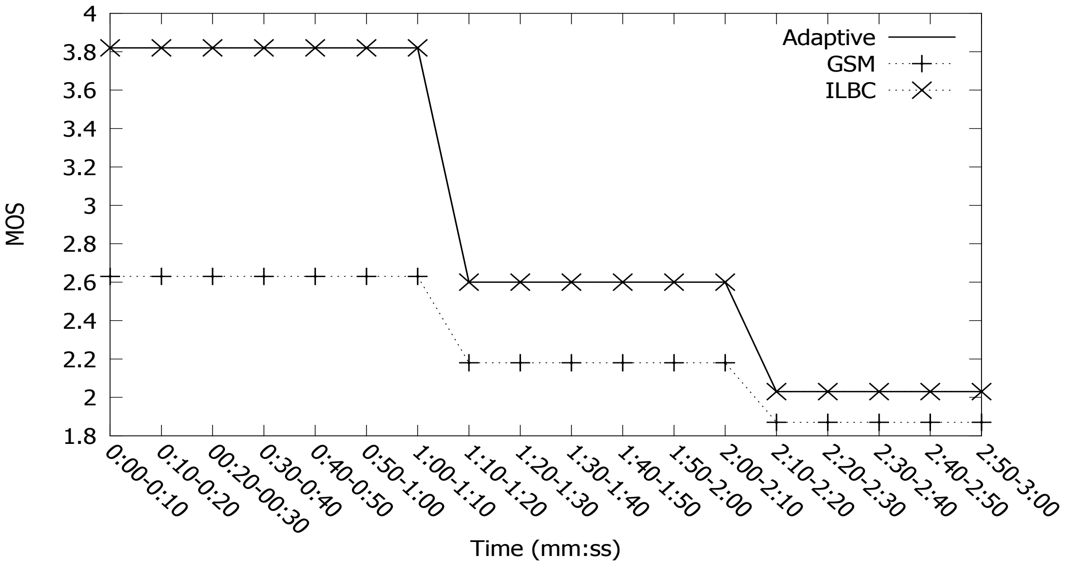


Figure 5.8: MOS values for the Third Package.

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 gap and switch-over gaps of different lengths depending on the prevailing pack loss rates. 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.

Chapter 6

Improved E-model for Monitoring Quality of Multi-Party VoIP communications

Maintaining good Quality-of-Experience (QoE) is crucial for Voice-over-IP (VoIP) applications, particularly those operating across the public Internet. Accurate online estimation of QoE as perceived by end users allows VoIP applications take steps to improve QoE when it falls below acceptable levels. ITU-T recommendation G.107 introduced the E-model, which provides a means to assess QoE levels for two-party VoIP sessions. In this chapter we provide an analysis of the accuracy of the E-model for multi-party VoIP sessions when all audio is processed by a centralised focus node. We analyse the impact of what we term the "Focus Transcoding Effect (FTE)," the "Focus Forwarding Effect (FFE)," and the number of end-points participating in the session. Through comparison to QoE metrics produced by the offline PESQ method for three common audio codecs, we show that the standard E-model does not provide accurate QoE assessment for multi-party VoIP sessions. We then introduce an improved E-model for these codecs for multi-party VoIP sessions. We describe the implementation of the improved E-model in a QoE monitoring application, showing that it produces results similar to actual PESQ scores.

In recent years VoIP has become an extremely important application class, with VoIP clients being very widely used by businesses and individuals. The success achieved by the basic two-party VoIP communications in terms of reliability and the cutting of costs has encouraged the emergence of multi-party VoIP conferencing facilities. Intuitively, it is more difficult to ensure QoE in multi-party sessions since, at different times during a sessions different people, connecting via different network paths, will be speaking. In this chapter we examine whether the E-model for online QoE estimation model which was developed for two party VoIP sessions are applicable to multi-party sessions. We find that it is not—for three commonly used audio codecs we find that it consistently over-estimates MOS values for a range of network-path packet loss conditions. We specify an enhanced E-model, describes its realisation in an online VoIP QoE monitoring tool and show results that indicate that it provides a more accurate QoE estimation.

6.1 Multi-party VoIP Conferencing Systems

The increasing demand for people to interact across different locations for business or personal needs has led to further developments in the VoIP systems which basically were supporting the two-party communications only. The success achieved by the basic two-party VoIP communications in terms of reliability and cutting of costs encouraged engineers to implement a multi-party VoIP conferencing system to match with the growing demand, and in the same time to provide a reliable and cheap way of communication between different parties.

SIP supports establishing communication sessions with multiple participants (Rosenberg, 2006). SIP dialogs are responsible for managing the communication sessions, typically dialog is between two parties(useragents), so signaling using SIP is quite straightforward. However, multi-party communication sessions are more complicated, and it have different models of implementation. Currently, VoIP conferencing system can be implemented through three possible connection topologies (Sat *et al.*, 2007):

1. **Decentralized Model:** all conference clients are connected to each other via unicasts or multicasts. Each client interacts with the rest of the clients using SIP. There is no focal point or centre for the conference, however the flow of data is distributed among all the clients. Figure 6.1 is an example of a full mesh

architecture, where each of the 4 clients transmits the data to the rest of the 3 listening clients by unicasts. Furthermore, every client will have 3 jitter buffers and decoders for processing audio signals which are sent from them.

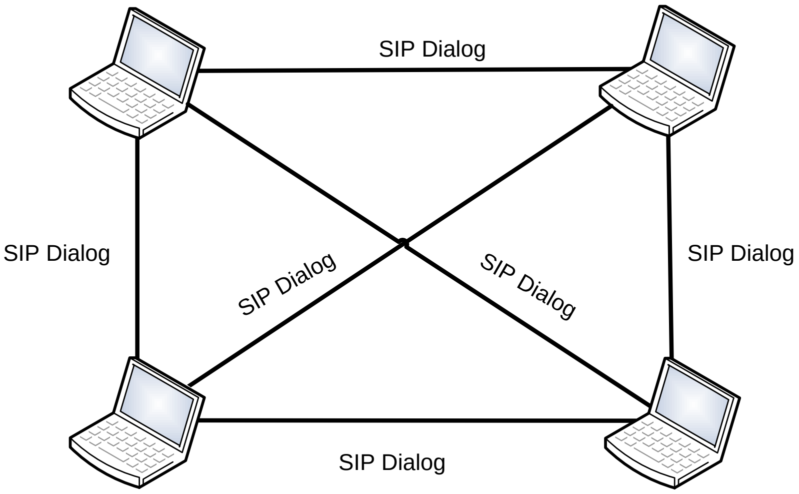


Figure 6.1: Decentralized Architecture for VoIP Conferencing System

2. **Centralized Model:** It is based upon a central point of control called "focus". The focus point can be a dedicated server called Media Server as used in IBM SUT (ibm, 2013), or one of the conference clients can perform this task, such as what is being used in Skype (sky, 2013) and Jitsi (jit, 2013). The focus is typically responsible for SIP signalling between all the conference participants. Moreover, all the transmitted data in the conference call must pass first through the focus to be decoded, mixed (if more than one client is speaking) and finally re-encoded and sent to the rest of clients. An example of centralized architecture is shown in Figure 6.2.

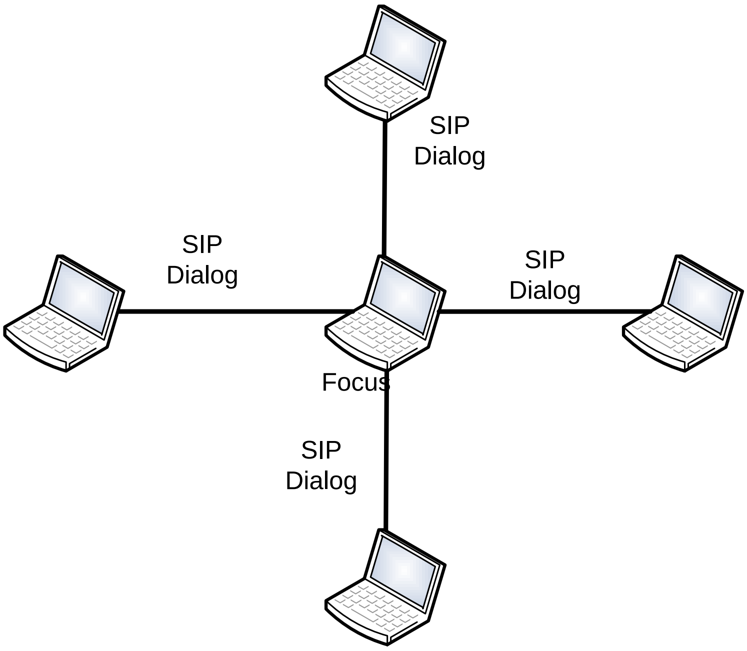


Figure 6.2: Centralized Architecture for VoIP Conferencing System

3. **Hybrid Model:** is a combination of centralized and decentralized architectures (see Figure 6.3). It relies on the underlying overlay network, where nodes A,B, and C are the parent nodes which are fully connected to each other. While nodes D, E, F, and G are child nodes which are only connected to one parent node.

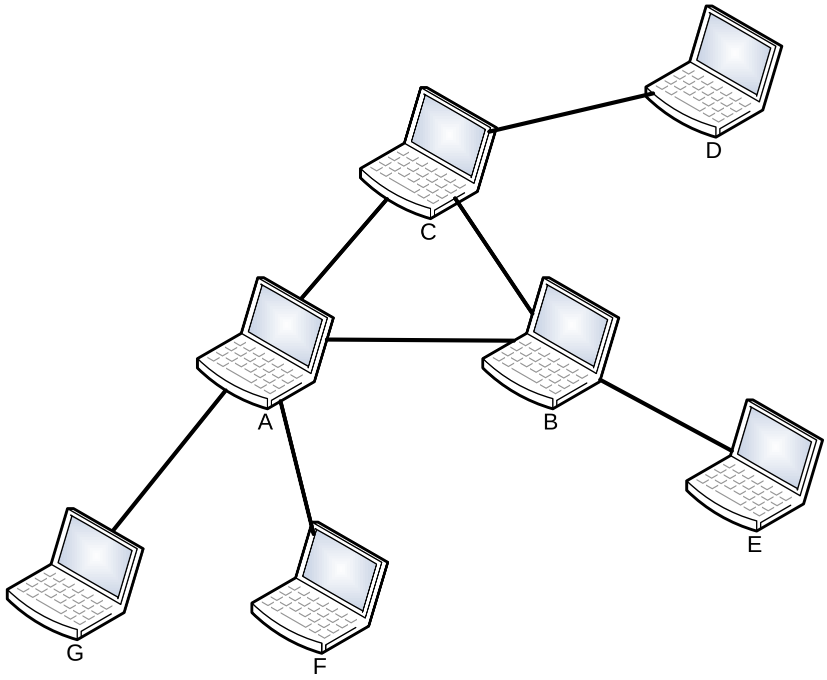


Figure 6.3: Hybrid Architecture for VoIP Conferencing System

We focus on the Centralized model since it is common in designing VoIP multi-party conferencing systems. Each endpoint is connected directly to the focus, and it has no current knowledge of other connections between other endpoints and the focus. Multiple links to the focus are often subjected to different degradation factors.

6.2 QoE Analysis of Multi-Party Calls

In this section, we describe an analysis of QoE of multi-party VoIP calls initiated using a centralized multi-party VoIP application. We estimate MOS scores using both PESQ and the E-model. PESQ is an intrusive method, requiring both the original and the degraded signal, so we take it as a benchmark for the E-model estimates as it should achieve a high degree of estimation accuracy. In our analysis, we study the performance of three commonly used codecs: G711, SILK and ILBC, under different network conditions. Figure 6.4 shows the testbed used in our experiments; we establish different VoIP multi-party calls between three users. In the figure the labels L1 and L2 indicate the links between user B, which acts as the central focus, with user A and

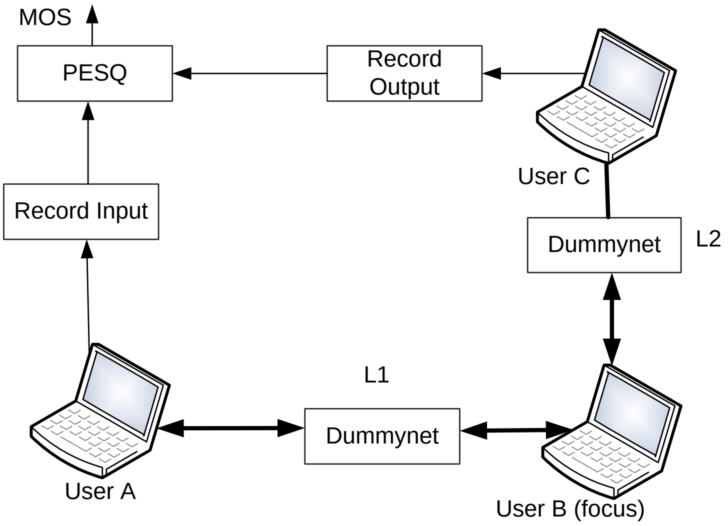


Figure 6.4: Centralized Multi-party VoIP Setup.

user C respectively. We use Dummynet (Carbone & Rizzo, 2010) to emulate different packet loss rates in L1 and L2 in the range from 0-5%, with 0.5% increments. In our analysis and to unify our comparison's parameters, we consider user A (speaking), user B (focus) and user C (listening).

We record the original and degraded audio signals from user A and user C respectively; these signals were then used as input to the PESQ algorithm, which produces MOS values in the range 1-5. In order to have accurate measurements and scores, we have taken more than 200 PESQ MOS values under different network condition for each codec. We also developed an online monitoring module that employs the E-model to estimate the MOS score but we excluded the delay factor from our calculation since the PESQ does not take it into account when estimating MOS. For the E-model, we do include the packet loss rate from both links when we repeat such experiment for the three codecs using Jitsi as a VoIP application.

From Figure 6.5 we see that the PESQ score under different packet loss rate of L1 and L2 with focus transcoding differs from the PESQ score of the call without focus transcoding. Also these PESQ scores differ compared to the PESQ score that is resulted from a single link with the sum of the packet loss rate of the two links. For instance, the PESQ score of the conference call between A and C passing by focus B having packet loss rate of L1 and L2 equals to 1% and 3% respectively differs when compared

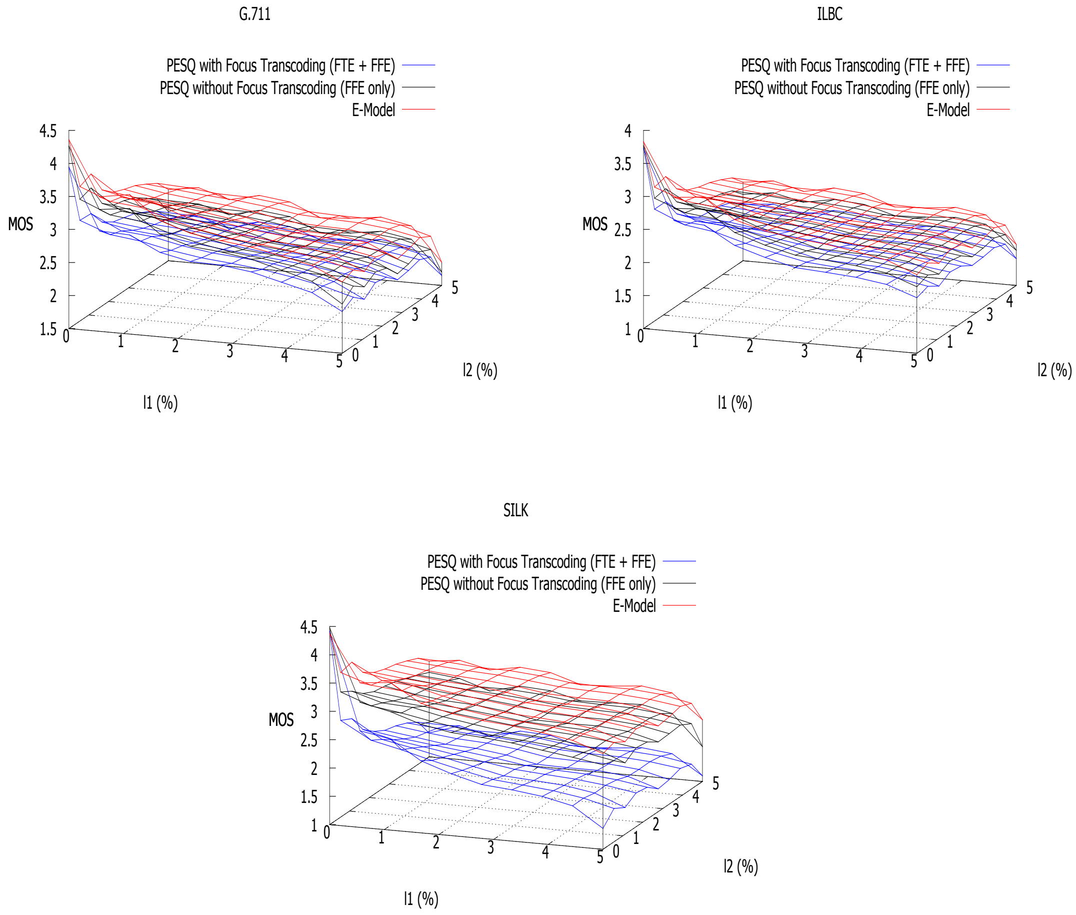


Figure 6.5: QoE for Multi-party call for G.711, ILBC, and SILK. The x and y axis indicates the percentage of the packet loss of the links L1 and L2 respectively, whilst the z axis indicates the MOS score.

to a single link between 2 users having a packet loss rate of 4%. Specifically, using the G.711 codec, the PESQ MOS score was 2.526 when having the 2 links while it was 2.81 when having single link. When using ILBC codec, the PESQ score was 1.82 using the 2 links while it was 2.37 when having single link with the sum of the packet loss rate of L1 and L2. We also observe that changing the order of introducing packet loss to conference links produces very similar results; in other words, adding 1% packet loss rate to L1 and 3% to L2 would give almost the same result as 3% to L1 and 1% to L2.

These observations leads us to study the effect of the presence of the central focus in order to be able to model its effect and to develop a corrected E-model that can be used in monitoring the call quality of multi-party calls. We address the following three effects of introducing a central focus: the *Focus Transcoding Effect*, the *Focus Forwarding Effect*, and the *Number of Users*.

6.2.1 Focus Transcoding Effect (FTE)

In the centralized model, all of packets are forwarded to the node that acts as a central focus. In order to understand the signal, this node decodes the packets back in to an audio signal. Then this signal is re-encoded and forwarded to the rest of the users in the conference call after re-negotiation of the used codec. The process of the decoding/re-encoding of the packet is called the transcoding process. This process has an influence on the QoE perceived the end user. We have studied the FTE effect by measuring the PESQ MOS score of a conference call with the setup shown in Figure 6.4 using the three different codecs under different packet loss rates. The resulting PESQ scores are shown in Figure 6.5 for the G711, ILBC and SILK codecs respectively labelled as PESQ with transcoding effect.

6.2.2 Focus Forwarding Effect (FFE)

We have studied the forwarding process of the packets from the focus to the rest of users and its influence on the QoE at the end user. In order to study such effect only, a typical peer-to-peer call is established between A and C, which are connected to Internet through the same gateway. We have used the testbed shown in Fig 6.4 by adding another node between user A and user C so that all of the packets forwarded from A to C are forced to pass by a gateway node first. This process emulates the forwarding effect only without the transcoding effect. We have measured the PESQ scores of different established calls under different packet loss rates using 3 different codec. These can be shown in Figure 6.5 for G711, ILBC and SILK codecs respectively, labelled as PESQ without focus transcoding.

6.2.3 Number of Users Effect

In order to study the impact of the number of users in the call we start a conference call with 3 users, and we increase the number of users from 3 to 6, adding a single user each time. At each time of increasing one more user, we have measured the PESQ MOS score at a certain user. First, the call was initiated with user A, user B, and user C using G.711 codec where user B is acting as the central focus in the call. In our experiment, we measure and track the call quality of user A using PESQ algorithm, so user A is considered as a speaker whilst we consider user C is the listener. We ensured that there is no network losses when measuring the call quality at user A. When 3 users were participating in the call, the MOS PESQ was 3.98. We found that this score is constant when increasing the users every time from 3 to 6 users. This shows that, at least for a reasonably small number of users, the number of users in the centralized model of the multi-party call has no appreciable effect on the end user perceived call quality.

6.2.4 Accuracy of the E-model

In order to study the accuracy of the original E-model in assessing QoE of multi-party audio calls, we have employed the original E-model in our monitoring system at the end user C, using the same testbed shown in Figure 6.4. The measured MOS resulted from the E-model is shown in Figure 6.5 for the G711, ILBC and SILK codecs respectively labelled as E-model. We clearly see that the standard E-model consistently overestimates call quality. It is therefore clear that the E-model needs to be correct to take the FTE and FFE into account. Moreover, we have noticed that such gap between the PESQ score with FFE and FEE with the E-model is codec dependent. Thus, codec dependent coefficients need to be derived for such correction of the E-model.

6.3 Corrected E-Model for Multi-Party Calls

In this section, we derive a correction function to the ITU standard E-model in order to make it suitable for evaluating QoE of multi-party calls. In Figure 6.6, we mapped the values of the original E-Model (x-axis) to the actual quality estimated by PESQ algorithm (y-axis). For example for SILK, at 4% packet loss, the audio MOS value estimated by E-model is equal to 2.87, while the actual MOS perceived as calculated

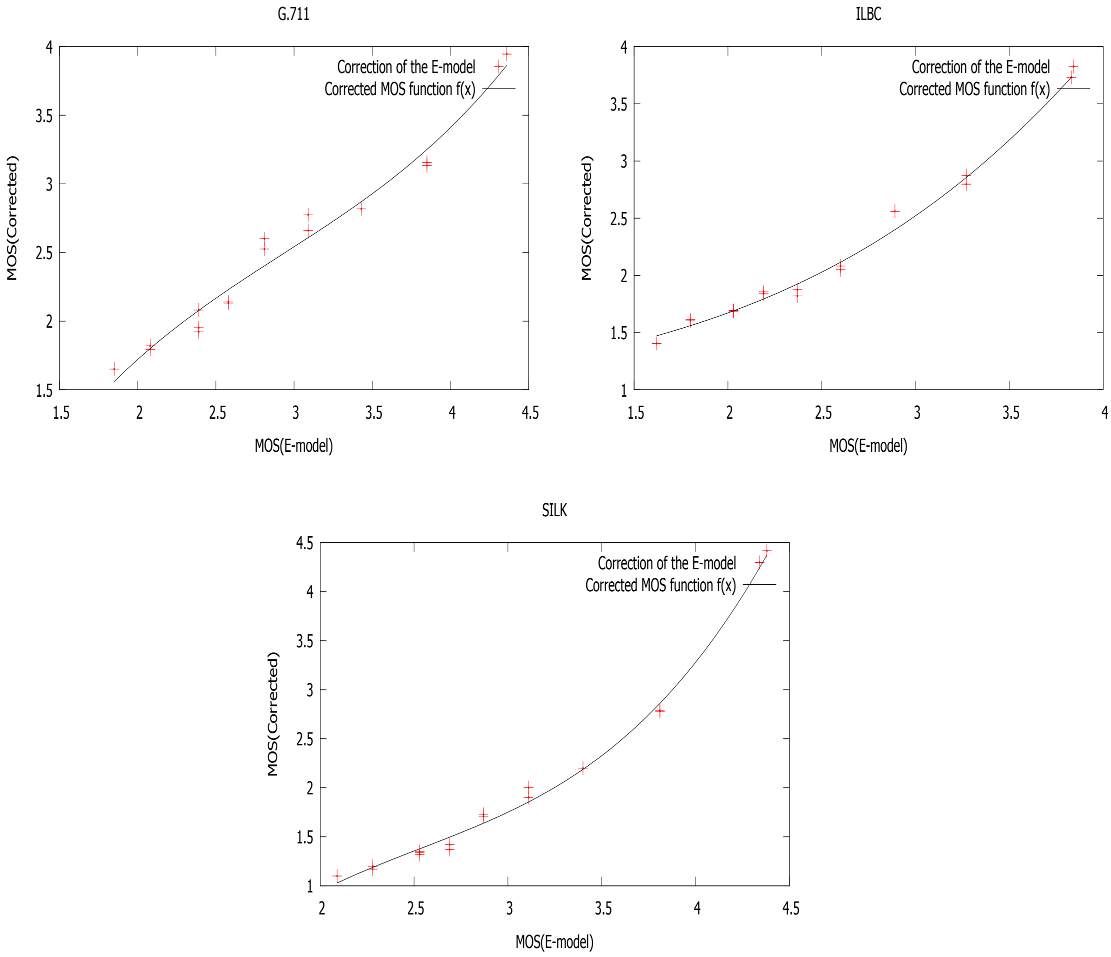


Figure 6.6: Correction Function for G.711, ILBC, and SILK

by PESQ algorithm should be equal to 1.73. By applying curve fitting by using the least squares method, the points fit well with a third degree function MOS_C. It indicates the actual QoE in a multi-party session as perceived by end-users, considering the focus degradation factors FTE and FFE. It is derived for each of the three codecs with parameters x1, x2, x3 and x4 as shown in Table 6.1. In Equation 6.1, MOS is the standard E-model function as explained.

$$MOS_C = x1 \cdot MOS^3 + x2 \cdot MOS^2 + x3 \cdot MOS + x4 \tag{6.1}$$

In order to estimate QoE online, the corrected E-model can be employed by first capturing the network characteristics (packet loss rate is the sum of loss rates of L1 and L2), acquiring the codec robustness factor then calculating the R which is then mapped to MOS. This MOS value resulted from the standard E-model is then used in Equation 6.1 to calculate MOS_C, the estimated MOS perceived by the end-users of multi-party VoIP session.

6.4 Monitoring System Design and Results

We have developed a monitoring system based on our corrected improved E-model for monitoring the VoIP call quality for the multi-party calls. Our monitoring system targets specific number of RTP packets to capture and perform an effective MOS value calculation based on our corrected E-model. Our system uses a coefficient database according to the codec used in the call, see Table 6.1. It is based at the network terminals, and the environment could be a personal or family network with voice quality monitoring. Our monitoring system works as follows. First, the system uses a network capturing module to capture a certain number of packets to certain IP and port. The non-RTP packets will be filtered. After this process is finished, the system will then starts to anlyze the data, delay and packet loss rate. Finally, the measured network conditions is converted into the MOS_C to indicate the call quality at the end user in the multi-party call. We took our results on-line by introducing random packer loss rates in the network in the range from 0-10% using Dummynet. For comparisons our system also computes MOS values using PESQ and the standard E-model.

We established conference calls using Jitsi, then applied the modified E-model under various packet loss rates; the results are shown in Figure 6.7. We have tested three codecs G.711, iLBC, and SILK. The correction made for the E-model has resulted in accurate results, very similar to what PESQ estimates. Crucially, our model can be

Table 6.1: Derived 3rd Degree Parameters for Different Codecs for Multi-party calls.

Parameters	**G.711**	**ILBC**	**SILK**
x1	0.111	0.045	0.26
x2	-0.978	-0.068	-1.982
x3	3.597	0.326	5.769
x4	-2.451	0.929	-4.748

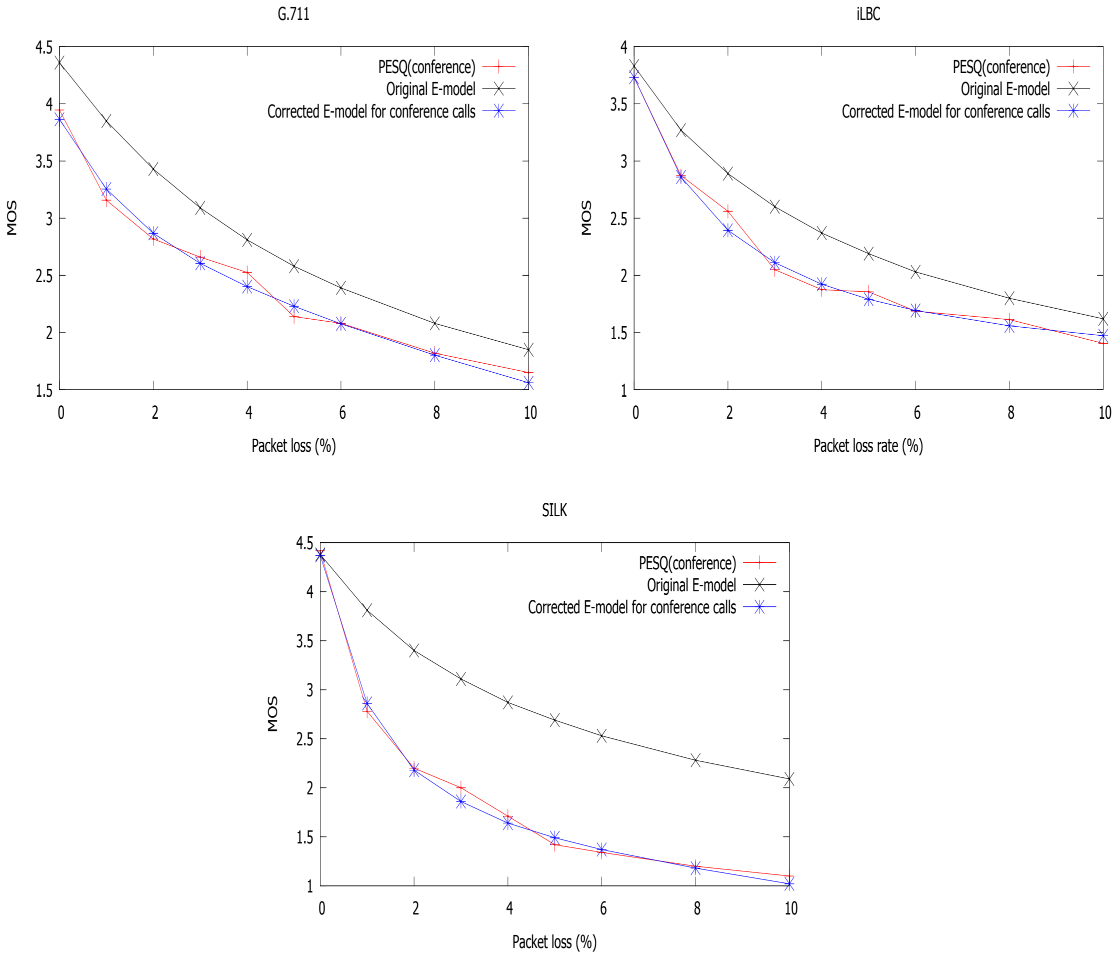


Figure 6.7: Corrected E-model against Packet Loss Rate for G.711, ILBC, and SILK

used online to estimate QoE, unlike PESQ which has required to be performed offline by recording on both sides and then comparing both original and degraded signals.

6.5 Summary

ITU-Recommendation G.107 introduces the E-model which brings a new approach to estimate the VoIP call quality of the person-to-person calls. The main advantage of this model that it can be applied in real time which enables monitoring the call quality during the call. Recently and due to the increase demand of the communication between more than one party in different locations, conferencing VoIP system emerged and became more mature. In this chapter, we have studied the QoE of the VoIP conferencing systems that use the centralized model. We found that there is a negative impact on the call quality from using the centralized model in the conferencing system. We have studied such effects and introduced them as Focus Transcoding Effect (FTE) and Focus Forwarding Effect (FFE). We found that such effects are are not taken into account in the E-model which will lead to estimating inaccurate multi-party call quality when using E-model. Consequently, we have corrected the original E-model in order to be used in live monitoring the multi-party call quality. We have proposed an improved corrected E-model and show how we derived the coefficients used for 3 commonly used codecs (G.711, ILBC and SILK). 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 corrected E-model to assess the multi-party voice call quality in real-time.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Voice-Over-IP has been considered as a technology prospect for several years; a great potential for networking and communications industry. Nowadays, most of the telecommunication companies and organizations are migrating gradually from the traditional telephony services to the newly developed IP data networks. VoIP services provide a cheap and efficient alternative to the traditional PSTN telephony services. Unfortunately, despite the widespread of its implementation, the technology is still considered to be in its adolescence. Furthermore, achieving voice quality levels for VoIP remains a significant challenge, as IP networks typically do not guarantee delay, packet loss, jitter and bandwidth levels. Nevertheless, VoIP is the fastest growing technology in the past decade, considering the flexibility and cost saving it can offer.

Existing tools and techniques for assessing QoE are limited in several aspects; PESQ for example requires end-user involvement to record audio samples at the sender's and the receiver's sides; the algorithm is then executed offline to obtain MOS value for the recorded session. For extensive testing, performing these steps is time consuming and requires expensive resources, moreover the unsuitability for online estimation of QoE is considered a major drawback. In order to overcome these limitations, we developed a QoE estimation tool based on a novel approach that does not require prior existence of audio/video recorded samples or offline processing. It works by emulating audio and video traffic of VoIP calls, and then estimates QoE of audio and video by using E-model and VQM respectively.

Improving user's quality-of-experience of audio calls can be achieved by codec switching. Codecs are known to have different behaviour when subjected to various network characteristics (e.g. packet loss rate). Nevertheless, codec switching doesn't always have a positive impact, a silent gap and switch-over gap is produced as a result of different loss rates experienced by the network. Furthermore, excessive switching in a certain time interval cause user's annoyance and degrade the overall perceived quality. We designed a codec switching algorithm to improve the overall perceived QoE, by taking in consideration negative effect of it. We show that by using the algorithm, switching between codecs like PCMU and SILK will result a better performance than using only one of them.

Multi-party VoIP conferencing systems has gained a lot of popularity recently due to the growing need of having communications between multiple users in different locations. The centralized architecture is the most commonly used model for establishing multi-party calls, where all the traffic is forced to pass through the focal node(the focus) of the conference. QoE perceived at end-users during conference sessions is quite different than the traditional peer-to-peer calls. The focus has a negative impact on the overall QoE, we presented a comprehensive analysis of that degradation which is mainly caused due to two factors which we defined as Focus Transcoding Effect(FTE) and Focus Forwarding Effect(FFE). We proved that the standard ITU E-model is not suitable for estimating the QoE, as it does not consider the negative impact of the focus. Consequently, we introduced an improved corrected E-model which takes into account extra degradation factors, and can be used in real-time to monitor the quality of multi-party calls.

7.2 Future Work

For future work, we are intending to target Web Real-Time Communication (WebRTC) framework (Bergkvist *et al.*, 2012). It brings VoIP capabilities to web browsers via Javascript and HTML5. Estimating the QoE for VoIP functionality on web browsers over various platforms, such as tablets and smart phones would be challenging and interesting. Moreover, we can apply the newly developed, Perceptual Objective Listening Quality Analysis (POLQA) ITU-T (2011), which is considered as the next generation for audio quality testing. POLQA supports modern wideband codecs, and it can be used for analysing audio quality in mobile networks, such as 3G, and 4G/LTE networks.

Furthermore, we intend to measure and quantify the degradation factors for video communications when using the centralized multi-party architecture over the common video codecs: H.263 and H.264. We can extend our work to take certain decisions at the end-users side based on the actual quality perceived in order to minimize the degradation effect caused by the focus. Codec switching can be a solution, re-negotiating a new codec at certain links can lead to minimizing the Focus Transcoding Effect and improving the QoE for end-user at that link.

References

(2007). ITU-T recommendation G.1070. Opinion model for video-telephony applications. 34

(2013). Apache felix - index. http://felix.apache.org/site/index.html. 40

(2013). IBM sametime unified telephony. http://www.ibm.com/. 40, 42, 56, 79

(2013). Iperf. http://perf.sourceforge.net. 45, 52

(2013). JITSI. http://www.jitsi.org. 40, 72, 79

(2013). NetIQ Corp. http://www.netiq.com. 30

(2013). Skype. http://www.skype.com. 6, 79

(2013). VOIPFUTURE GmbH, Hamburg. http://www.voipfuture.com. 29

(2013). Wireshark. http://www.wireshark.org. 48

Agrawal, S., Ramamirtham, J. & Rastogi, R. (2006). Design of active and passive probes for VoIP service quality monitoring. In *Proc. 12th International Telecommnications Network Strategy and Planning Symposium (NETWORKS 2006)*, 1–6, IEEE. 21

Akella, A., Maggs, B., Seshan, S., Shaikh, A. & Sitaraman, R. (2003). A measurement-based analysis of multihoming. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 353–364, ACM. 31

Aktas, I., Schmidt, F., Weingrtner, E., Schnelke, C.J. & Wehrle, K. (2012). An adaptive codec switching scheme for SIP-Based VoIP. In S. Andreev, S. Balandin & Y. Koucheryavy, eds., *Internet of Things, Smart Spaces, and Next Generation Networking*, no. 7469 in Lecture Notes in Computer Science, 347–358, Springer Berlin Heidelberg. 32

Andersen, D., Balakrishnan, H., Kaashoek, F. & Morris, R. (2001). *Resilient overlay networks*, vol. 35. ACM. 31

Apostolopoulos, J.G. (2001). Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *Visual Communications and Image Processing*, vol. 4310, 392–409. 31

Arango, M., Elliott, I., Pickett, S., Huitema, C. & Dugan, A. (1999). Media gateway control protocol (MGCP) version 1.0. 9

Assem, H., Malone, D., Dunne, J. & O'Sullivan, P. (2013). Monitoring VoIP call quality using improved simplified E-model. In *International Conference on Computing, Networking and Communications(ICNC 2013)*, 927–931. 27

Atzori, L. & Lobina, M. (2006). Playout buffering in IP telephony: a survey discussing problems and approaches. *Communications Surveys & Tutorials, IEEE*, **8**, 36–46. 20

Barriac, V., Sout, J. & Lockwood, C. (2004). Discussion on unified objective methodologies for the comparison of voice quality of narrowband and wideband scenarios. In *Proceedings of Workshop on Wideband Speech Quality in Terminals and Networks: Assessment and Prediction*. 18

Bergkvist, A., Burnett, D.C., Jennings, C. & Narayanan, A. (2012). WebRTC 1.0: Real-time communication between browsers. *Working draft, W3C*. 91

Brown, A. & Wilson, G. (2012). *The Architecture of Open Source Applications, Volume II*, vol. 2. Kristian Hermansen. 40

Calyam, P., Ekici, E., Lee, C., Haffner, M. & Howes, N. (2007). A "GAP-model" based framework for online VVoIP QoE measurement. *Journal of Communications and Networks*, **9**, 446. 29

CARBONE, M. & RIZZO, L. (2010). Dummynet revisited. *ACM SIGCOMM Computer Communication Review*, **40**, 12–20. 43, 52, 66, 82

CARVALHO, L., MOTA, E., AGUIAR, R., LIMA, A. & DE SOUZA, J. (2005). An E-model implementation for speech quality evaluation in VoIP systems. In *Proc. 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, 933–938, IEEE. 29

CCITT, R. (1988). G. 722,7 khz audio-coding within 64 kbit/s. 18

CHEN, K.T., HUANG, C.Y., HUANG, P. & LEI, C.L. (2006). Quantifying skype user satisfaction. In *ACM SIGCOMM Computer Communication Review*, vol. 36, 399–410, ACM. 28

CLARK, A., IEE, P. & ET AL. (2001). Modeling the effects of burst packet loss and recency on subjective voice quality. 26

COLE, R.G. & ROSENBLUTH, J.H. (2001). Voice over IP performance monitoring. *ACM SIGCOMM Computer Communication Review*, **31**, 9–24. 25, 26

COSTA, N. & NUNES, M.S. (2009). Adaptive Quality of Service in Voice over IP Communications. In *Proc. 5th International Conference on Networking and Services*, ICNS '09, 1924, IEEE Computer Society, Washington, DC, USA. 32

CUERVO, F., GREENE, N., RAYHAN, A., HUITEMA, C., ROSEN, B. & SEGERS, J. (2000). RFC 3015: Megaco protocol version 1.0. *Internet Engineering Task Force*. 10

DA SILVA, A.P.C., VARELA, M., DE SOUZA E SILVA, E., LEÃO, R.M. & RUBINO, G. (2008). Quality assessment of interactive voice applications. *Computer Networks*, **52**, 1179–1192. 22

DA SILVA, J. & LINS, R. (2006). Analyzing the QoS of VoIP on SIP in java. In *Proc. 2006 InternationalTelecommunications Symposium*, 576–581, IEEE. 29, 52

DRAFT, I. (2003). Recommendation and final draft international standard of joint video specification (ITU-T rec. h. 264— iso/iec 14496-10 avc). *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050*. 33

FEI, T., TAO, S., GAO, L. & GUERIN, R. (2006). How to select a good alternate path in large peer-to-peer systems? In *25th IEEE International Conference on Computer Communications, Proceedsing IEEE Infocom 2006*, vol. 1, 1106–1118. 31

FIEDLER, M., HOSSFELD, T. & TRAN-GIA, P. (2010). A generic quantitative relationship between quality of experience and quality of service. *Network, IEEE*, **24**, 36–41. 22

GONG, Y., YANG, F., HUANG, L. & SU, S. (2009). Model-based approach to measuring quality of experience. In *1st International Conference on Emerging Network Intelligence*, 29–32, IEEE. 29

GOODE, B. (2002). Voice over internet protocol (VoIP). *Proceedings of the IEEE*, **90**, 1495–1517. 6

HALAS, M., KOVAC, A., ORGON, M. & BESTAK, I. (2012). Computationally efficient E-model improvement of MOS estimate including jitter and buffer losses. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*, 86–90, IEEE. 28

HAN, Y., FITZPATRICK, J., MURPHY, L. & DUNNE, J. (2013). Accuracy analysis on call quality assessments in voice over IP. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, 1–7. 25

HANDLEY, M. & JACOBSON, V. (1998). RFC 2327. *SDP: session description protocol*. 10

HERSHEY, P., PITTS, J. & OGILVIE, R. (2009). 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)*, 1–7, IEEE. 29

HUNTGEBURTH, B., MARUSCHKE, M. & SCHUMANN, S. (2011). Open-source based prototype for quality of service (QoS) monitoring and quality of experience (QoE) estimation in telecommunication environments. In *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011 5th International Conference on*, 161–168, IEEE. 21, 30

IETF (2002). Session Initiation Protocol, RFC 3261. 31

IETF (2006). Session Description Protocol, RFC 4566. Tech. rep. 31

ITU-T (2001). Transmission Impairments due to Speech Processing. Tech. Rep. Recommendation G.113. 27, 65

ITU-T (2011). P. 863,perceptual objective listening quality assessment (POLQA). *Int. Telecomm. Union, Geneva*. 25, 91

Jekosch, U. (2005). *Voice and speech quality perception*. Springer. 22

Jiang, C. & Huang, P. (2011). Research of monitoring VoIP voice QoS. In *Proc. 2011 International Conference onInternet Computing & Information Services (ICICIS 2011)*, 499–502, IEEE. 29

Kawata, T. & Yamada, H. (2006). Wlc24-5: Adaptive multi-rate VoIP for IEEE 802.11 wireless networks with link adaptation function. In *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, 1–5, IEEE. 32

Kim, H. & Choi, S. (2010). Traffic quality monitoring system between different network providers. In *Proc. 12th International Conference onAdvanced Communication Technology (ICACT 2010)*, vol. 2, 1153–1158, IEEE. 29

Kim, K. & Choi, Y.J. (2011). Performance comparison of various VoIP codecs in wireless environments. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, 89, ACM. 19

Klein, A. & Klaue, J. (2009). Performance evaluation framework for video applications in mobile networks. In *Advances in Mesh Networks, 2009. MESH 2009. Second International Conference on*, 43–49, IEEE. 34

Lustosa, L., Carvalho, L., Rodrigues, P. & Mota, E. (2004). E-model utilization for speech quality evaluation over VoIP-based communication systems. *Proc. 22nd SBRC*. 26

Lutzky, M., Schuller, G., Gayer, M., Krämer, U. & Wabnik, S. (2004). A guideline to audio codec delay. In *AES 116th convention, Berlin, Germany*, 8–11. 20

Möller, S. (2010). *Quality Engineering: QualitaEt Kommunikationstechnischer System*. Springer DE. 22

NARBUTT, M. & DAVIS, M. (2005). An assessment of the audio codec performance in voice over WLAN (VoWLAN) systems. In *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, 461–467, IEEE. 19

NG, S.L., HOH, S. & SINGH, D. (2005). Effectiveness of adaptive codec switching VoIP application over heterogeneous networks. In *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, 7–pp, IEEE. 32

NGUYEN, T. & ZAKHOR, A. (2003). Path diversity with forward error correction (pdf) system for packet switched networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1, 663–672, IEEE. 31

OBAFEMI, O., GYIRES, T. & TANG, Y. (2011). An analytic and experimental study on the impact of jitter playout buffer on the E-model in VoIP quality measurement. In *ICN 2011, The Tenth International Conference on Networks*, 151–156. 28

OHM, J. (2004). *Multimedia communication technology: Representation, transmission and identification of multimedia signals*. Springer Verlag. 34, 60

OSIPOV, A.E. (2006). *Dynamic voice codec determination mechanism for VoIP*. Ph.D. thesis, Wichita State University. 32

PAULSEN, S. & UHL, T. (2010). Adjustments for QoS of VoIP in the e-model. In *Telecommunications: The Infrastructure for the 21st Century (WTC), 2010*, 1–6, VDE. 27

PENNOCK, S. (2002). Accuracy of the perceptual evaluation of speech quality PESQ algorithm. *Proc. Of MESAQIN*. 25

QUALINET (2012). Qualinet white paper on definitions of quality of experience (QoE) and related concepts. In *White paper*, Qualinet. 22

RAAKE, A. (2007). *Speech Quality of VoIP: Assessment and Prediction*. Wiley. 22

REC, I. (1988). G. 711: Pulse code modulation (PCM) of voice frequencies. *International Telecommunication Union, Geneva*. 18

Rec, I. (1990). G. 726,40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM),. *International Telecommunication Union, Geneva*. 18

Rec, I. (1994). E. 800, terms and definitions related to quality of service and network performance including dependability. *International Telecommunication Union*. 22

Rec, I. (1996). G. 729: Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear-prediction (CS-ACELP). *Mars*. 18, 22

Rec, I. (2000). H. 262— iso/iec 13818-2. *Information technologyGeneric coding of moving pictures and associated audio informationVideo*. 33

Rec, I. (2001). P. 862,. *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*. 23

Rec, I. (2002). Bt. 500-11,. *Methodology for the subjective assessment of the quality of television pictures*, **22**, 25. 34

Rec, I. (2003a). G. 722.2. *Wideband coding of speech at around*, **16**. 19

Rec, I. (2003b). P. 862.1: Mapping function for transforming p. 862 raw result scores to MOS-LQO. *International Telecommunication Union, Geneva*. 24

Rec, I. (2004). P. 501 amendment i, test signals for use in telephonometry. *International Telecommunication Union, Geneva, Switzerland*. 49

Rec, I. (2005). G. 722.1, low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss. *International Telecommunication Union, Geneva, Switzerland*. 18

Rec, I. (2007). P. 10 (2007) vocabulary for performance and quality of service. *International Telecommunication Union, Geneva*. 22

Rec, I. (2009). G. 107. *The E-model, a computational model for use in transmission planning*. 25, 26, 65

Recommendation, I. (1998). H. 263. *Video coding for low bit rate communication*, **2**, 30. 33

Ren, J., Zhang, C., Huang, W. & Mao, D. (2010). Enhancement to E-model on standard deviation of packet delay. In *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on*, 256–259, IEEE. 28

Rix, A.W. (2003). Comparison between subjective listening quality and p. 862 pesq score. *Proc. Measurement of Speech and Audio Quality in Networks (MESAQIN03), Prague, Czech Republic*. 24

Robustelli, L., Loreto, S., Fresa, A., Longo, M. & Spinelli, D. (2003). Prototype of an adaptive voice coder for IP telephony. In *International Conference on Software, Telecommunications and Computer Networks–SoftCom 2003*, 7–10, Citeseer. 32

Rodman, J. (2009). VoIP to 20 khz: Codec choices for high definition voice telephony. In *White paper*, Polycom. 19

Rosenberg, J. (2006). RFC 4353A framework for conferencing with the session initiation protocol. 78

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E. *et al.* (2002). SIP: session initiation protocol. Tech. rep., RFC 3261, Internet Engineering Task Force. 9, 10

Roto, V., Law, E., Vermeeren, A. & Hoonhout, J. (2010). User experience white paper. In *Report from the Dagstuhl Seminar on Demarcating User Experience*. 22

Sat, B., Huang, Z. & Wah, B. (2007). The design of a multi-party VoIP conferencing system over the internet. In *Multimedia, 2007. ISM 2007. Ninth IEEE International Symposium on*, 3–10, IEEE. 78

Schulzrinne, H., Casner, S., Frederick, R. & Jacobson, V. (2003). RFC 3550 RTP: A transport protocol for real-time applications. 10, 12, 14, 70

Schwarz, H., Marpe, D. & Wiegand, T. (2007). Overview of the scalable video coding extension of the h. 264/avc standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, **17**, 1103–1120. 33

SERVETTI, A. & DE MARTIN, J.C. (2003). Adaptive interactive speech transmission over 802.11 wireless lans. In *Proc. IEEE Int. Workshop on DSP in mobile and Vehicular Systems*. 32

SFAIROPOULOU, A., MACIÁN, C. & BELLALTA, B. (2007). VoIP codec adaptation algorithm in multirate 802.11 wlans: distributed vs. centralized performance comparison. In *Dependable and Adaptable Networks and Services*, 52–61, Springer. 32

SILLER, M. & WOODS, J. (2003). QoS arbitration for improving the QoE in multimedia transmission. In *Visual Information Engineering, 2003. VIE 2003. International Conference on*, 238–241, IET. 22

SULOVIC, M., RACA, D., HADZIALIC, M. & HADZIAHMETOVIC, N. (2011). Dynamic codec selection algorithm for VoIP. In *Proc. 6th International Conference on Digital Telecommunications (ICDT 2011)*, 74–79. 32

SUN, L. (2004). *Speech Quality Prediction for Voice over Internet Protocol Networks*. PhD thesis, University of Plymouth. 65

SUN, L. & IFEACHOR, E.C. (2006). Voice quality prediction models and their application in VoIP networks. *Multimedia, IEEE Transactions on*, **8**, 809–820. 23, 26, 67

TAO, S., XU, K., XU, Y., FEI, T., GAO, L., GUERIN, R., KUROSE, J., TOWSLEY, D. & ZHANG, Z.L. (2004). Exploring the performance benefits of end-to-end path switching. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, 304–315, IEEE. 31

TAO, S., XU, K., ESTEPA, A., GAO, T.F.L., GUERIN, R., KUROSE, J., TOWSLEY, D. & ZHANG, Z.L. (2005). Improving VoIP quality through path switching. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, 2268–2278, IEEE. 20, 30, 31

TAYLOR, T. (2000). Megaco/h. 248: a new standard for media gateway control. *Communications Magazine, IEEE*, **38**, 124–132. 9

Toga, J. & Ott, J. (1999). ITU-T standardization activities for interactive multimedia communications on packet-based networks: H. 323 and related recommendations. *Computer Networks*, **31**, 205–223. 9

Trad, A., Ni, Q. & Afifi, H. (2004). Adaptive VoIP transmission over heterogeneous wired/wireless networks. In *Interactive Multimedia and Next Generation Networks*, 25–36, Springer. 32

Ulseth, T. & Stafsnes, F. (2006). VoIP speech quality-better than PSTN? *Telektronikk*, **102**, 119. 18

Valin, J.M. (2006). Speex: a free codec for free speech. In *Australian National Linux Conference, Dunedin, New Zealand*, Citeseer. 19

Walterman, M., Lewcio, B., Vidales, P. & Moller, S. (2008). A Technique for Seamless VoIP-codec Switching in Next Generation Networks. In *Proc. 2008 IEEE International Conference on Communications (ICC 2008)*, 1772–1776, IEEE. 32

Wältermann, M. (2013). *Dimension-based Quality Modeling of Transmitted Speech*. Springer. 22

Wu, C., Chen, K., Huang, C. & Lei, C. (2009). An empirical evaluation of VoIP playout buffer dimensioning in skype, google talk, and msn messenger. In *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*, 97–102, ACM. 6

Xie, M., Chu, P., Taleb, A. & Briand, M. (2009). ITU-T G.719: A new low-complexity full-band (20 khz) audio coding standard for high-quality conversational applications. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, 265–268, IEEE. 19

Yamagishi, K. & Hayashi, T. (2006a). Qrp08-1: opinion model for estimating video quality of videophone services. In *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, 1–5, IEEE. 35

Yamagishi, K. & Hayashi, T. (2006b). Verification of video quality opinion model for videophone services. In *2nd ISCA Tutorial & Research Workshop on Perceptual Quality of Systems*, 143–148. 35

Zec, M. & Mikuc, M. (2004). Operating system support for integrated network emulation in imunes. In *Proc. of the 1st Workshop on Operating System and Architectural Support for the on demand IT InfraStructure (OASIS), Boston, MA*. 46

Zhang, H., Xie, L., Byun, J., Flynn, P. & Shim, C. (2005). Packet loss burstiness and enhancement to the E-model. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on*, 214–219, IEEE. 28

Zhang, H., Gu, Z. & Tian, Z. (2011). QoS evaluation based on extend E-model in VoIP. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, 852–854, IEEE. 28

Zhang, X., Lei, W., Yu, B. & Jia, J. (2009). Using P2P overlay to improve VoIP quality in SIP+ P2P system. In *Information Engineering, 2009. ICIE'09. WASE International Conference on*, vol. 1, 255–259, IEEE. 31