ADAPTIVE SPEECH QUALITY IN VOICE-OVER-IP COMMUNICATIONS

by
Eugene Myakotnykh

Ph.D. Dissertation
Submitted to Faculty of the Telecommunications Program,
Graduate School of Information Sciences, University of Pittsburgh
in Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

University of Pittsburgh
2008

University of Pittsburgh

School of Information Sciences

Department of Information Sciences and Telecommunications

Dissertation Defense

Name of Student: Eugene Myakotnykh

Dissertation Title: Adaptive Speech Quality in VoIP Communications

Committee:	Signature:
Dr. Richard Thompson (Advisor)	_____
Dr. Joseph Kabara	_____
Dr. David Tipper	_____
Dr. Stephen Walters	_____
Dr. Martin Weiss	_____

Date: _____

ADAPTIVE SPEECH QUALITY IN VOICE-OVER-IP COMMUNICATIONS

Eugene Myakotnykh
University of Pittsburgh, 2008

The quality of VoIP communication relies significantly on the network that transports the voice packets because this network does not usually guarantee the available bandwidth, delay, and loss that are critical for real-time voice traffic. The solution proposed here is to manage the voice-over-IP stream dynamically, changing parameters as needed to assure quality.

The main objective of this dissertation is to develop an adaptive speech encoding system that can be applied to conventional (telephony-grade) and wideband voice communications. This comprehensive study includes the investigation and development of three key components of the system. First, to manage VoIP quality dynamically, a tool is needed to measure real-time changes in quality. The E-model, which exists for narrowband communication, is extended to a single computational technique that measures speech quality for narrowband and wideband VoIP codecs. This part of the dissertation also develops important theoretical work in the area of wideband telephony.

The second system component is a variable speech-encoding algorithm. Although VoIP performance is affected by multiple codecs and network-based factors, only three factors can be managed dynamically: voice payload size, speech compression and jitter buffer management. Using an existing adaptive jitter-buffer algorithm, voice packet-size and compression variation are studied as they affect speech quality under different network conditions. This study explains the relationships among multiple parameters as they affect speech transmission and its resulting quality.

Then, based on these two components, the third system component is a novel adaptive-rate control algorithm that establishes the interaction between a VoIP sender and receiver, and manages voice quality in real-time. Simulations demonstrate that the system provides better average voice quality than traditional VoIP.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Abbreviations and Glossary

3GPP	Third Generation Partnership Project
ACELP	Algebraic Code Excited Linear Prediction
AMR	Adaptive Multi-Rate Codec
AMR-WB	Adaptive Multi-Rate Wideband Codec
CBR	Constant Bit Rate
CBWFQ	Class-based Weighted Fair Queuing
CS-ACELP	Conjugate Structure Algebraic Code Excited Linear Prediction
DSCP	DiffServ Codepoint Field
EFR	Enhanced Full Rate codec
ETSI	European Telecommunications Standards Institute
FEC	Frame Error Correction
GoB	Good or Better
GSM	Global System for Mobile Communications
IETF	Internet Engineering Task Force
iLBC	Internet Low Bit Rate Codec
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
ITU	International Telecommunications Union
ITU-T	Telecommunications Standardization Sector of ITU
LAN	Local Area Network
LRD	Long-Range Dependency
LSP	Label Switched Path
LSR	Label Switched Router
MNB	Measuring Normalizing Blocks
MOS	Mean Opinion Score
MPLS	Multi-protocol Label Switching
NB	Narrowband
PAMS	Perceptual Assessment of Speech Quality

PBX	Private Branch Exchange
PCM	Pulse Code Modulation
PESQ	Perceptual Evaluation of Speech Quality
PL	Packet Loss
PLC	Packet Loss Concealment
PoW	Poor or Worse
PSQM	Perceptual Speech Quality Measure
PSTN	Public Switched Telephone Network
QI	Quality Indicator
QoS	Quality of Services
RED	Random Early Detection
RF	Radio Frequency
RFC	Request For Comments
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
TCP	Transmission Control Protocol
TME	Terminate the Call Early
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VAD	Voice Activity Detection
VBR	Variable Bit Rate
VoIP	Voice over IP
WAN	Wide Area Network
WB	Wideband
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection

Chapter 1

Introduction

This Chapter is organized as follows. Section 1.1 provides background information related to the VoIP quality area. The problem statement and research objectives are given in Section 1.2. Section 1.3 provides a brief overview of organization of this dissertation.

1.1 Background

Over the last several years, experience with the voice-over-IP technologies has shown that the quality of voice transmission over the Internet remains a primary obstacle to the broader adoption of VoIP services. VoIP has moved from being an interesting and cheap application for enthusiasts to a public service for everybody, where speech quality requirements have significant importance. Many people are not satisfied with the quality of service offered by VoIP providers, which is often lower than the quality of the traditional toll-grade PSTN telephony. One of the main causes of the problem is that the Internet was initially designed to transport bursty data, and was not optimized for real-time traffic. Voice requires real-time handling from the network and from the end-points, and very sensitive to many factors.

The quality of VoIP communications depends on two major factors: (1) network conditions (that is the situation in the network along a path of a given call in terms of delay, jitter, packet loss) and (2) codec settings (that is how, a given voice stream is formed in terms of the compression algorithm, packet size, signal frequency range, etc). Most of the previous research in the VoIP quality has concentrated on networking issues, such as QoS management. Researchers have developed many different algorithms to improve the transport of packetized voice traffic, including traffic classification (Differentiated Services technology [21, 22]), bandwidth reservation (Integrated Services architecture [24], Resource Reservation Protocol [23]), congestion avoidance algorithms (for example, Random Early Dropping, Weighted Random Early Dropping [25]), Multi-Protocol Label Switching (MPLS)

technology [20], and others. These approaches use different techniques to decrease transmission delay and/or probability of voice packet congestion in the network and make the Internet more suitable for voice transmission (network optimization). However, these methods often do not provide acceptable results nor do they solve the problem completely because (1) not all equipment supports multiple QoS protocols and (2) the Internet is a dynamic media and the technologies often cannot react to changing network conditions and manage the quality of every call in real-time.

Besides network-level solutions of voice quality management, sender-initiated approaches exist. Multiple speech coding techniques were developed to provide a choice between a desired level of quality and bandwidth resources required per call. For example, the G.711 codec [6] uses 64 kbps of audio bandwidth and provides toll-grade quality in the absence of packet loss and significant end-to-end delay. The G.729 codec [10] uses 8:1 compression; the quality of this codec is worse than the quality of the G.711 codec but it requires just 8 kbps of audio bandwidth to transmit a voice stream. These codecs are not adaptive: an initial codec is chosen in the beginning of a communication session and is not changed during the conversation.

Recently, the idea of adaptive codecs, which can change their parameters depending on some factors, was proposed. This idea implies that, based on the investigation and analysis of different parameters and characteristics that affect speech quality, it is potentially possible to form a voice stream with a given or optimal quality under given network conditions and to improve average quality of VoIP communications. The choice of optimal end-user parameters under given network conditions may enhance the quality of VoIP because it can change a configuration of a VoIP system, so that the system matches the current state of the network. For example, the lower bandwidth requirements for the G.729 codec may reduce the likelihood of congestion in the network and thus delay and/or packet loss. As a result, the quality of this codec in some cases can be better than the quality of the G.711 codec. So, intelligent management of codec settings can potentially improve average speech quality. Noticeable improvement of voice quality over an IP network requires continuous monitoring of the quality of every call (or a group of calls) made through this network so that quality issues can be detected and resolved immediately. This approach

proposes to adjust a voice stream to the network and change multiple parameters of the stream in real-time, depending on the state of the network.

Adaptive Multi-Rate codec (AMR) [26, 8] is an example of adaptive codec, which can change a bit rate and manage the quality of communications depending on situation in the network. But this codec was designed for GSM and UMTS networks and it not used in VoIP because philosophy behind the AMR is to lower a codec rate based on a level of interference. Different decision parameters and management criteria should be used by this codec in VoIP networks.

Adaptive speech quality management is a recent but very promising approach. Intelligent speech encoding algorithms can choose the most suitable VoIP stream settings under given network conditions thus optimizing speech quality or managing a quality of a communications session in real-time depending on users requirements. This area is in the early stage of its development and investigation of many questions related to VoIP quality measurement and management, dependencies between multiple parameters and the resulting quality, is necessary to develop intelligent and efficient adaptive VoIP codecs. There is a need to investigate these questions in details and to propose a new solution to improve average quality of VoIP communications.

1.2 Problem Statement and Research Objectives

Figure 1.1 demonstrates the traditional (non-adaptive) VoIP system. The system consists of three main components: a sender (a source of VoIP traffic), a receiver (a destination point of a VoIP stream) and the network. The sender can be represented by individual users as well as by a group of calls. Speech is encoded on the sender side and a voice stream typically goes through the traditional data network together with other types of traffic or through a dedicated VoIP-only network. The jitter buffer eliminates a delay variation on the destination side. Different voice decoding mechanisms are used to convert the digital stream to an analog form.

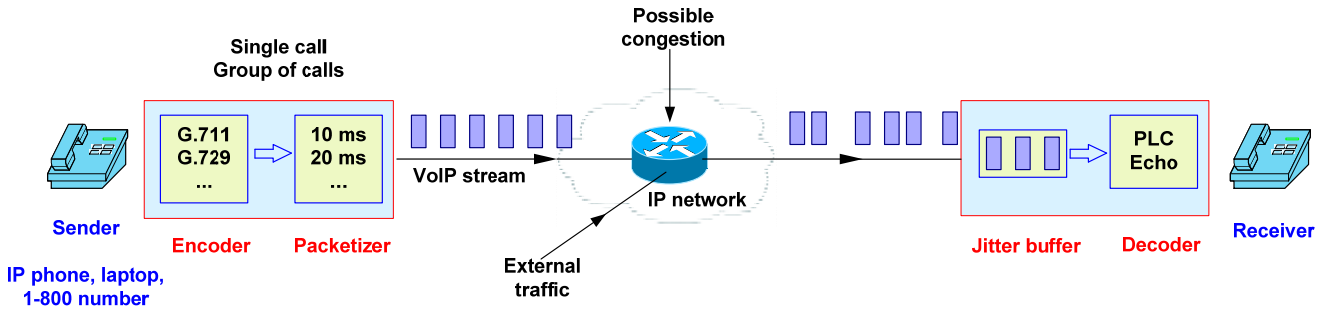


Figure 1-1: Non-adaptive VoIP system

To make this system adaptive (that is to manage a quality of communications on the sender side in real-time depending on some criteria), it is required to design two components: (1) objective mechanisms of real-time speech quality assessment and (2) adaptive speech quality management algorithms (Figure 1.2).

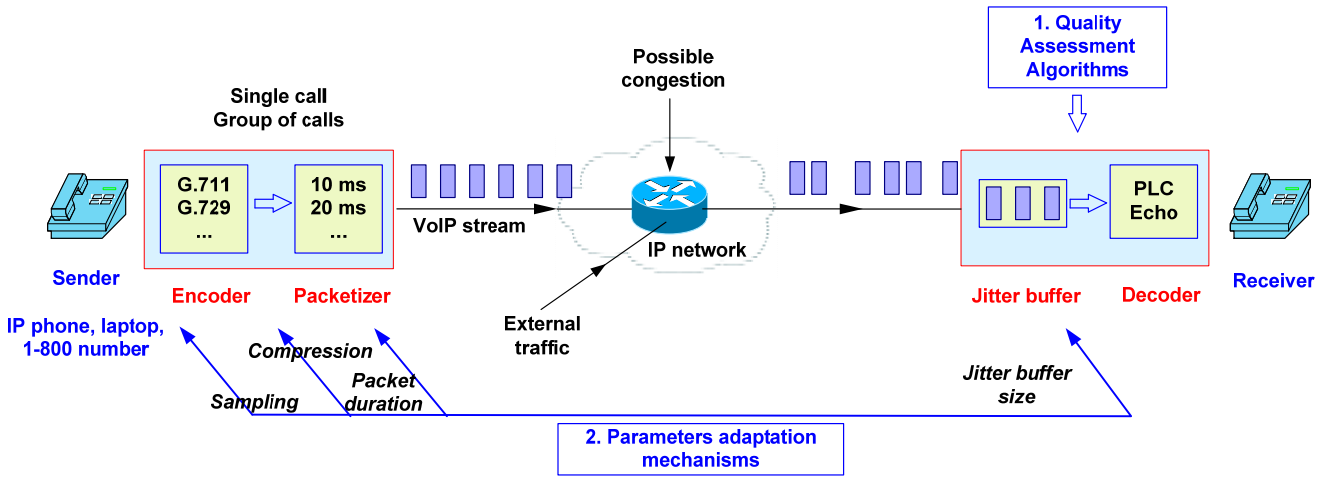


Figure 1-2: Adaptive VoIP system

The main goal of this dissertation is the development of algorithms of objective speech quality measurement and adaptive real-time speech quality management and the investigation of several related questions. The problem statement of the project is given by the following main research questions:

- How to estimate the quality of a communication session objectively and in real-time in the case of variable speech quality encoding?
- How does the variation of speech encoding parameters affect the quality of VoIP communication under different network conditions?
- How to manage speech quality dynamically depending on specified criteria?

The first part of this dissertation investigates questions related to voice quality measurement. Speech quality is a managed parameter and that is why a relatively precise technique is required to estimate VoIP quality "on the fly". The computational model called the E-model [1] and proposed by ITU, can be used to estimate quality of communications based on received packet statistics. Many papers and commercial quality testing solutions mentioned in Chapter 2 use this algorithm. But this model can be used only for narrowband telephony. Several years ago the minimization of bandwidth resources required for every call was one of the primary goals. Now the situation has changed: although the effectiveness of VoIP coding algorithms is still important, the problem of bandwidth became less critical. Now we can even speak about adaptive wideband VoIP communications. The deployment of wideband VoIP telephony together with efficient speech encoding algorithms is a potential solution to improve the average quality of VoIP technologies. Adaptive speech quality management requires the deployment of real-time objective quality measurement mechanisms valid not only for standard narrowband codecs but also for the wideband telephony and for any set of speech encoding parameters. There is a need to develop a new methodology to measure/predict voice quality in these situations. The first part of this dissertation proposes such a model. This model can be used to monitor voice quality in the adaptive algorithms. Also, the quality measurement part of the dissertation addresses several other important questions from the wideband telephony area:

- How to compare the quality of narrowband and wideband telephony? Is it possible to propose and use a single MOS scale to compare qualities directly?
- How to extend the traditional R-scale from the narrowband E-model to characterize the quality of wideband telephony?
- How will the multiple factors included to the traditional narrowband E-model change in the case of wideband VoIP telephony? What will the computational wideband model look?

The traditional E-model can be used only for a fixed set of codecs (it includes several parameters defined only for specific codecs). Speaking about adaptive changes of codec parameters, that is assuming that codec parameters can change dynamically and can have any values, we have to answer the question:

- How to choose parameters of the computational model in the case of arbitrary codec settings (in other words, in the case of "non-standard" codecs)?

The answers to all these questions will provide methodologies to estimate a quality of a given communication session objectively and in real-time in the case of fixed and adaptive speech quality encoding.

To manage VoIP quality changing sender parameters, it is necessary to know how a variation of different speech encoding parameters (packet size, compression, encoding scheme, signal frequency range) affects a quality of communications under different network conditions. It is required to understand how, for example, to improve a quality of a given voice stream if necessary. Is it better to improve signal quality (signal frequency range)? Or it is better to decrease packet size? Or it would be better to change compression? Theoretical and practical investigations have to be provided to answer the question:

- How do multiple codec and network parameters affect speech quality under different network conditions and different quality management scenarios?

Based on these results and developed quality measurement methodologies, the possibility of a sender-based real-time speech quality management is investigated. Here are several especially interesting questions:

- How to choose parameters to make decisions about speech quality adaptation? Which parameters besides the computational mechanisms proposed in the previous part have to be considered?
- How can information be delivered to a sender side? How often is it necessary to do it?
- How to manage voice quality dynamically depending on specified criteria? How to prove an effectiveness of the algorithms and to confirm their stability?
- How to characterize network behavior? Is it possible to use a statistical model to estimate network state and to make predictions about potential future network behavior based on current and previous network states?

- How adaptive quality management algorithms be different in the case of VoIP-only and traditional data networks and different types of speech quality degradations?

The outcome of this dissertation provides new methodologies of objective speech quality measurement in the cases of fixed wideband codecs, of variable codecs, investigates a perceptual quality measurement metrics. It also investigates effects of multiple codec parameters on speech quality and proposes dynamic quality adaptation mechanisms. All these results are interesting and useful both from scientific (theoretical) and practical perspectives.

1.3 Dissertation Outline

Chapters 2 and 3 of the dissertation provide background information in the area of VoIP quality measurement and management. These Chapters analyze previous work and explain how this work is used is the research project. Chapter 4 investigates a set of questions related to real-time computational speech quality measurement. We develop a computational quality model for wideband telephony and provide an analysis of how the model can be used in the case of variable codec parameters. Chapter 5 includes investigation of effects of multiple speech encoding parameters on quality of VoIP communications in different network conditions. Chapter 6 describes an approach of adaptive speech quality management developed based on results from Chapters 4 and 5.

Chapter 2

Speech Quality Measurement Techniques

This Chapter is organized as follows. Sections 2.1 and 2.2 provide an overview and analysis of current subjective and objective speech quality measurement methodologies. Section 2.3 discusses some aspects related to the wideband telephony. Section 2.4 describes a concept of perceptual quality and existing computational methodologies of perceptual speech quality estimation.

2.1 Subjective Speech Quality Measurement

Over many years, voice quality measurement has been very a subjective issue: people could pick up the phone and give their impression about the quality of speaker's voice. Different users could hear the same telephone call but their impressions could be different. The qualities of masculine, feminine and children's voice heard over a telephone network in the same conditions are also different [33, 34]. After years of research, human responses have been recorded and scored, establishing a rather objective measurement of call quality. The leading subjective criterion of voice quality measurements is the Mean Opinion Score (MOS), which is defined in the ITU-T Recommendation P.800 [11]. Mean Opinion Score is a subjective score of voice quality as perceived by a large number of people listening to speech over a communication system. To determine the MOS for a particular phone connection, a statistically valid group of males and females rate the quality of special test sentences read aloud over the connection. The testing considers a number of factors, including packet loss, circuit noise, talker echo, distortion, propagation time, end-to-end delay, and other transmission problems. This Recommendation uses the scale from one to five and the MOS of some session of voice transmission is the average estimate of voice quality rates assigned by this statistical group (1 – bad, 2 – poor, 3 – fair, 4 – good, 5 – excellent).

An MOS of 4.0 is considered *toll quality* within the telephone industry. Toll quality

is the telephone conversations quality level typically heard on a wired land-line from a local telephone company. Anything below MOS of 4.0 would then be *below toll quality* level. The narrowband G.711 codec achieves MOS score between 4.0 and 4.4 on a clean LAN. On a WAN link however, the impact of jitter, latency and packet loss can significantly drop the voice quality. Just for a comparison, a "typical" cell phone call might achieve an MOS score of 3.0 to 3.5. Call qualities below 3.0 on the MOS scale are generally unacceptable for communications.

Figure 2.1 shows that the acceptable range of MOS scores (between 3.0 and 4.4) occupies about 30% of the 1-to-5 MOS scale; an achievable and relatively good quality (between 3.6 and 4.4) occupies about 15% of the scale; the achievable toll-grade quality range is between 4.0 and 4.4 (less then 10% of the scale).

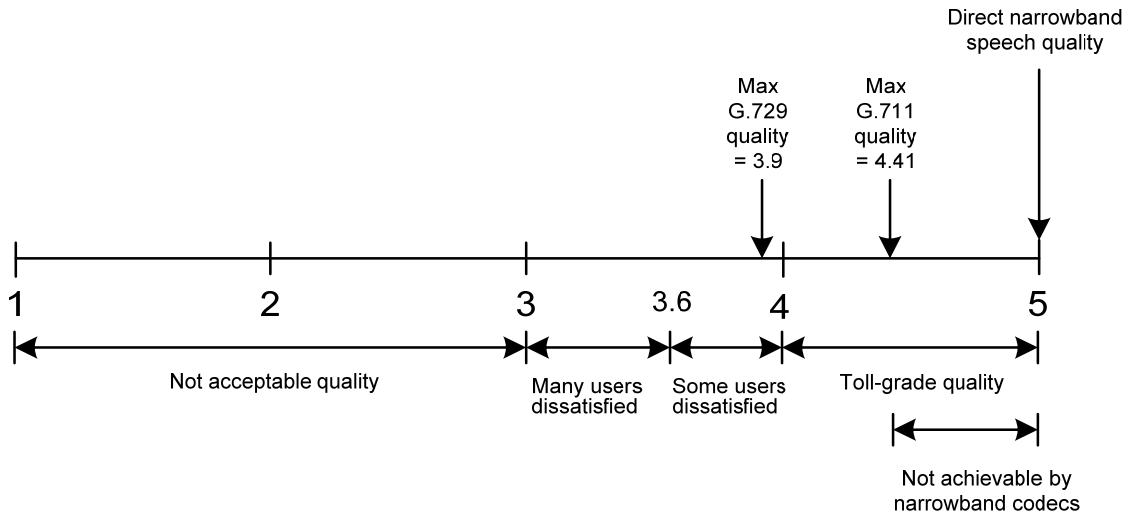


Figure 2-1: Traditional MOS scale

It is possible to conclude that a communications quality is a relatively abstract concept. It cannot be defined exactly and often depends on conditions of a given experiment. Subjective (human) voice quality testing is considered as a good way to evaluate speech quality because it uses live listeners. But this process is costly and time consuming because the testing process, which could provide us accurate results, is very complex and it is not acceptable for real-time voice quality handling. Good objective mechanisms are required for this purpose.

2.2 Objective Speech Quality Measurement

2.2.1 Signal-based Approaches

Several techniques were developed to measure voice quality objectively: Perceptual Speech Quality Measure (PSQM) [13], Perceptual Evaluation of Speech Quality (PESQ) [14, 15], Perceptual Assessment of Speech Quality (PAMS), Measuring Normalizing Blocks (MNB) and several others. PESQ is the most recent and the most advanced ITU standard; it has higher accuracy than any other model and it is often used in different hardware solutions designed for objective voice quality testing. PESQ uses a model to compare an original, unprocessed signal with a degraded version of this signal received on a destination side. This speech quality measurement technique takes into account not only codec but also network parameters, which influence the voice quality: coding distortions, errors, packet loss, variable delay, filtering in analog network components, several others. There is a list of conditions for which the ITU recommendation is known to provide inaccurate predictions and not intended to be used: the effects of loudness loss, delay, sidetone, echo, and other impairments related to two-way interaction are not reflected in the PESQ scores.

The resulting PESQ quality score can be converted to the subjective Mean Opinion Score using a mapping function described in the Recommendation P.862.1 [14]. Also there is a wideband extension of the PESQ algorithm described in the Recommendation P.862.2 [15], which uses a different model to measure wideband quality and different mapping function between a wideband PESQ score and the same 1-to-5 MOS scale.

The ITU experiments described in [14, 16] investigate the accuracy of the PESQ algorithm. The benchmark experiments, performed by the ITU covered a wide range of conditions (random and bursty loss, different codecs, etc.) and have demonstrated a relatively high correlation with human testing results (the correlation coefficient is 0.935). But this high correlation with subjective testing does not necessary mean high accuracy of the objective approach because correlations ignore absolute differences between subjective and objective scores. The ITU experiments demonstrate that the difference $|MOS_{human} - MOS_{PESQ}|$ is below 0.25 for 70% of test results and below 0.5 for 90% of test results [35]. The Scott Pennock from Lucent Technologies [36] provides more detailed subjective experiments

and makes even less optimistic conclusion saying that “P.862 indicate higher accuracy that what will be experienced from real-world use by the telecommunications industry” and that “there are limitations to using PESQ for verification of voice quality performance, competitive analysis, and system optimization".

So, the results indicate that PESQ is not a perfect measure of speech quality. But this is the best available tool and it is often used in commercial voice quality testing solutions. PESQ is relatively good for comparing the qualities of signals in the same conditions. High correlation coefficient with human testing results means that the model is accurate under the same experimental conditions: the results can change from test to test but the PESQ score is a good indicator of quality within a given experiment. We are able to benefit from the relative accuracy of this predictor without suffering from absolute measurement error.

The next few chapters of this dissertation discuss the adaptive algorithms to estimate and manage speech quality in real-time. PESQ cannot be used for this purpose because it requires both original and degraded signals to estimate voice quality and we do not have an original signal on a destination side. So, it is not a good technology to analyze speech quality “on the fly”. For dynamic voice quality, a computational model is required to estimate the level of voice quality at a given moment of time (or during a given period of time).

2.2.2 Computational Speech Quality Measurement: the E-model

The International Telecommunications Union (ITU) has developed and standardized a computational quality measurement mechanism called the E-model. The “original” version of this model is very complex (20 input parameters representing terminal, network and environmental quality factors) and can be applied to telephony in general (not necessary to VoIP). It is described in the ITU-T standards G.107 [1], G.113 [4]. Besides “traditional” factors affecting speech quality like delay, packet loss, codec quality, the original version of the E-model includes several other parameters shown in Figure 2.2.

The simplified version of this model is often used in the Voice-over-IP area. The “VoIP version” of the E-model assumes that default values are chosen for all but a few parameters (all except for delay and packet loss). For example, it is assumed that there is no

echo effect, loudness level is constant and sufficient, no environmental noise on a sender and receiver sides, etc. These assumptions are reasonable and suitable for this dissertation.

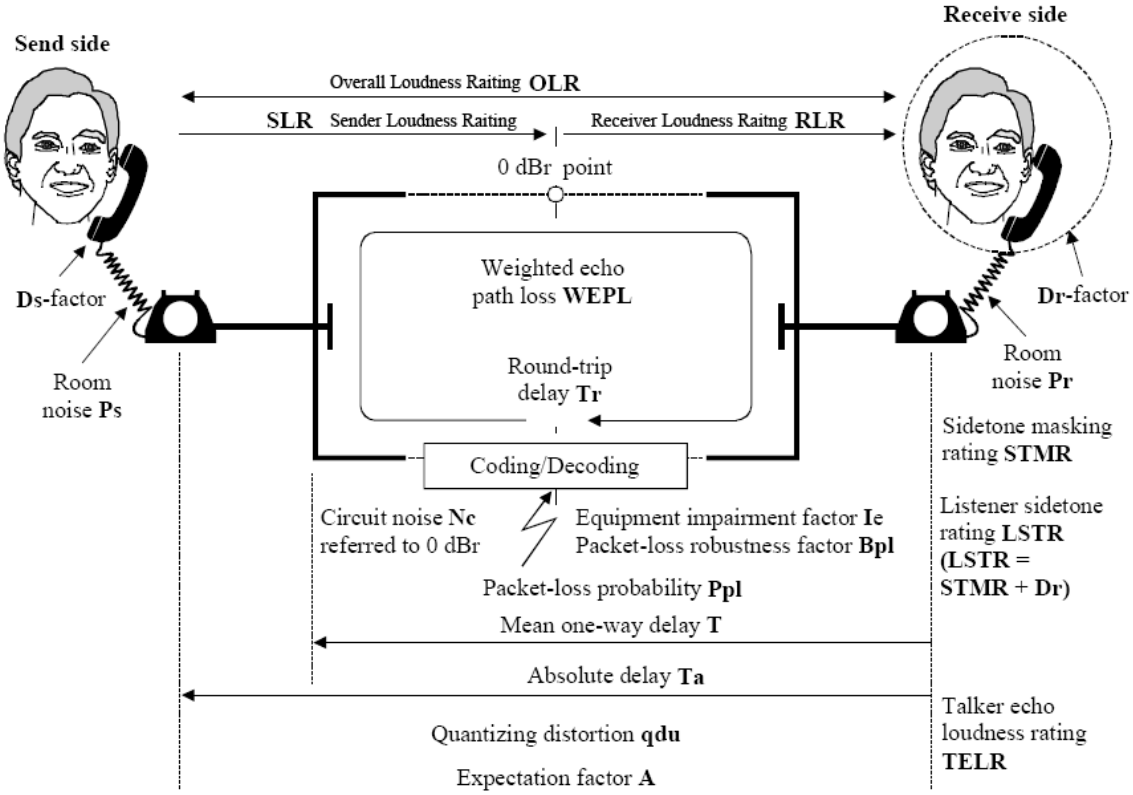


Figure 2-2: The E-model

The "narrowband VoIP version" of the E-model includes several components [1, 37]:

$$R = R_0 - I_d - I_{e\text{-}eff} \tag{2.1}$$

- R is the indicator of the resulting voice quality (from 0 to 100 scale)
- R_0 = 93.2 is the maximum score which can be achieved by narrowband codecs and which corresponds to the narrowband MOS = 4.41
- I_d is the impairment factor caused by end-to-end delay (the function of delay)

- $I_{e\text{-}eff}$ is the effective equipment impairment factor, which depends on equipment (codec, which is used to form a voice stream) and also on packet loss and packet loss robustness. This factor is expressed by Equation 2.2:

$$I_{e-eff} = I_e + (95 - I_e)\frac{Ppl}{Ppl + Bpl} \tag{2.2}$$

- I_e – equipment impairment (codec quality) - codec specific characteristics defined in Appendix I / G.113 [4]. The methodology of derivation is explained in [12].
- Bpl – packet loss robustness (an effectiveness of packet loss concealment algorithms; codec specific characteristics also)
- Ppl – packet loss rate in percent

The E-Model is based on the concept that "psychological factors on the psychological scale are additive" [1]. This means that each impairment factor from the E-model can be computed separately, even so this does not imply that such factors are uncorrelated, but only that their contributions to the estimated impairments are separable [37]. The model was designed for narrowband communications only and 100 points on the R-scale corresponds to a quality level of direct (mouth-to-ear) narrowband speech.

The delay impairment factor, I_d, in the E-model includes three components: delay itself and the effects of talker and listener echo. Large end-to-end delay is one of the main problems of voice transmission over IP networks. When this delay becomes significant, long pauses in conversations occur. To provide high quality voice, the VoIP network must be capable of guaranteeing low latency. The ITU-T G.114 recommendation [5] defines the acceptable round trip delay time between two VoIP gateways: 150 ms one-way delay. Delays between 150 ms and 400 ms make the conversations possible but the excessive delay becomes annoying. Delays of more than 400 ms are unacceptable for general network planning purposes. The traditional narrowband E-model uses a very complex equation to calculate the effect of delay on voice quality. R. Cole and J. Rosenbluth [37] propose a simpler linear function to define the delay impairment:

$$I_d = 0.024D + 0.11(D - 177.3)\cdot H(D-177.3) \quad (2.3)$$

- D – end-to-end (mouth-to-ear) delay
- H(x) – Heaviside function: H(x) =0 if x<0; H(x)=1 if x>0

The graph of the I_d function is shown in Figure 2.3, where D is a mouth-to-ear delay. It includes not only propagation delay but also packetization, transmission and even jitter buffer delay. Normally, packets arrive to a receiver side with different delays; the jitter buffer is used to eliminate the delay variation and all packets after jitter buffer have the same delay. If variable jitter buffer mechanisms are used and end-to-end delay changes over time, we have to apply the E-model to time intervals with constant delays or use an average delay value.

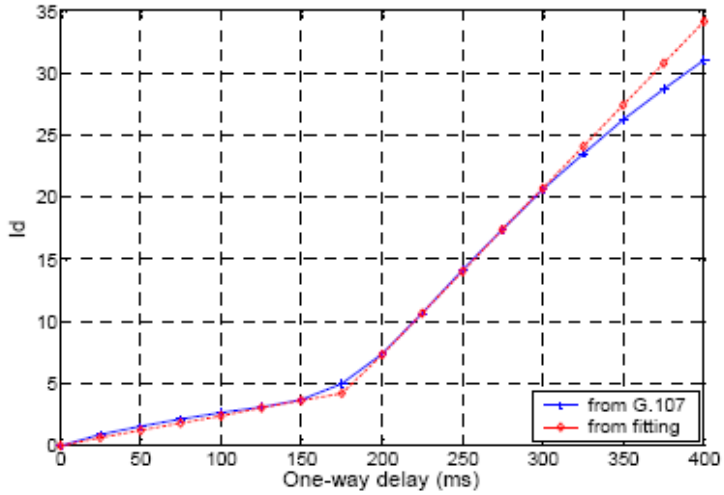


Figure 2-3: Delay impairment

The effective packet loss robustness factor $I_{e\text{-}eff}$ depends on the equipment impairment factor I_e and also on packet loss rate and effectiveness of packet loss concealment algorithms. Packet loss includes loss caused by the network and also by excessive delay variation. Lost packets are restored using packet loss concealment (PLC) mechanisms. The Equation 2.2 assumes that the packet loss is random. In the case of bursty packet loss, $I_{e\text{-}eff}$ is calculated using Equation 2.4.

$$I_{e-eff} = I_e + (95 - I_e)\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \quad (2.4)$$

BurstR is the so-called burst ratio, which is defined as the ratio of average length of observed bursts in an arrival sequence to an average length of bursts expected under random loss. When packet loss is random (i.e., independent) *BurstR* = 1; and when packet loss is bursty (i.e., dependent) *BurstR* > 1. The E-model standard says that this Equation is accurate for relatively small values of packet loss (no more than 2-3 %).

Different codecs use different PLC techniques. The ITU has proposed special tables, which contain I_e and Bpl values for some specific codecs. Codec qualities I_e are measured with respect to the quality of the best narrowband codec G.711 (I_e for this codec is 0). Worse codecs have higher values of I_e. These values for different codecs are based on different ITU's experimental results.

Table 2-1: Provisional planning values for the equipment impairment and for packet loss robustness factors

Codec	PLC type	Ie	Bpl
G.711	None	0	4.3
G.711	App.I/G.711	0	25.1
G.726 (32 kbps)	Native	7	23
G.723 + VAD	Native	15	16.1
G.729A + VAD	Native	11	19.0
GSM-EFR	Native	5	10.0

The E-model proposes a special mapping function to establish a relationship between the R-scale and the traditional MOS scale. The range of corresponding narrowband MOS values is from 1 to 4.5 (the maximum achievable narrowband MOS = 4.41, which corresponds to R = 93.2).

$$\begin{aligned} &\text{For } R < 0: && MOS = 1 \\ &\text{For } 0 < R < 100: && MOS = 1 + 0.035R + R(R-60)(100-R) \times 7 \cdot 10^{-6} \\ &\text{For } R > 100: && MOS = 4.5 \end{aligned} \tag{2.5}$$

The graph of this function is shown on Figure 2.4.

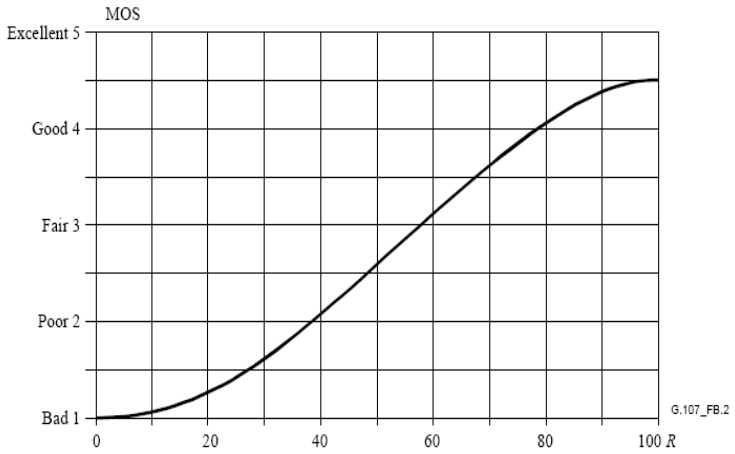


Figure 2-4: The relationship between an R-factor and MOS in the narrowband E-model

This equation was derived based on the assumption that the E-model uses a statistical estimation of quality measures. It uses the Gaussian Error function

$$E(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt \tag{2.6}$$

to establish the relationship between an R-factor and the percentage of subscribers, i.e., users, that would typically regard the call as being Good (GoB – Good or Better), Poor (PoW – Poor or Worse) or Terminate the Call Early (TME). The equations for GoB and PoW are:

$$\begin{aligned} GoB &= 100E\left(\frac{R-60}{16}\right)\% \\ PoW &= 100E\left(\frac{45-R}{16}\right)\% \end{aligned} \tag{2.7}$$

E(x) is defined by the Equation 2.6. The graph of these functions is shown below:

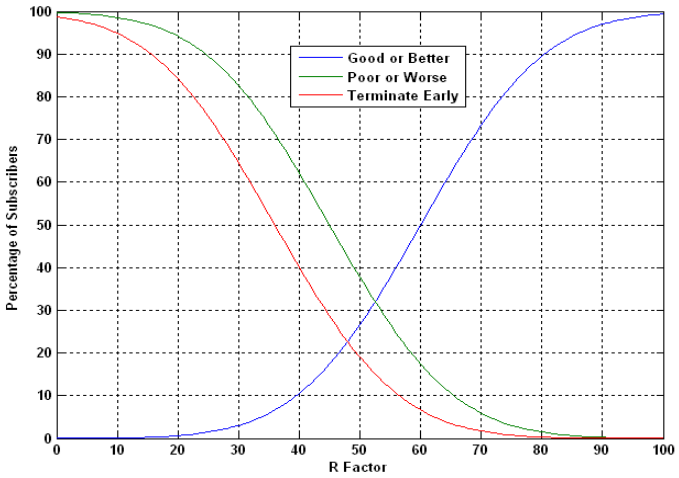


Figure 2-5: GoB, PoW, TME curves for the narrowband E-model

For example at an R-factor of 60, about 50% of subscribers would regard the call quality as "good", nearly 20% of subscribers would regard the call quality as "poor" and almost 10% would terminate the call early. The assumption about Gaussian distribution is empirical but it is also based on ITU experimental results.

Tables 2.2 and 2.3 demonstrate relationships between R-values and MOS and provide examples of multiple codec performances depending on delay and packet loss.

Table 2-2: Relationship between R-value and MOS

R-value	MOS (lower limit)	User Satisfaction	Quality
90-100	4.34	Very satisfied	Toll
80-90	4.03	Satisfied	Toll
70-80	3.60	Some users dissatisfied	Below-Toll
60-70	3.10	Many users dissatisfied	Unacceptable
50-60	2.58	Nearly all users dissatisfied	Unacceptable

Table 2-3: Feasible Combinations of Equipment, Impairment & Delay

Codec	Delay	R-value	Quality
G.711	Up to 250 ms	80 and higher	Toll
G.726 & G.728	Up to 200 ms 250 ms	87-80 73	Toll Below-Toll
G.729	50-150 ms 200-250 ms	83-80 77-70	Toll Below-Toll
G.729A + 2% loss	50-150 ms	74-71	Below-Toll
G.723.1 @ 6.3 kbit/s + 1 % loss	100-200 ms	77-72	Below-Toll
G.723.1 @ 5.3 kbit/s + 1 % loss	100-150 ms	73-71	Below Toll
GSM FR & IS-54	100-150 ms	72-70	Below-Toll

The ITU-T G.107 standard [1] says that the E-Model has not been fully verified by field surveys or laboratory tests for the very large number of possible combinations of input parameters. For many combinations of high importance to transmission planners, the E-Model can be used with confidence, but for other parameter combinations, the E-Model predictions have been questioned and are currently under study. Detailed investigations of this question by NTT Lab (Japan) [38] concluded that correlation coefficient of results provided by the E-model with subjective human testing results is about 80%. This high correlation coefficient shows the effectiveness of the algorithm if we make experiments under the same conditions. The E-Model was originally developed as a transmission planning tool for telecommunication networks; however, it is widely used for VoIP service quality measurement. Many commercial quality testers (for example, VQmon developed by Telchemy, DirectQuality by Minacom) use the E-model as an objective measure of quality.

In this project a similar computational model is developed and used for real-time voice quality analysis and management. But, besides conventional narrowband VoIP, the project also addresses issues with wideband (7-kHz channel) speech (see Section 2.3 contains the discussion about narrowband and wideband telephony). The traditional E-model was developed for narrowband communications only and there is nothing similar for wideband communications. Chapter 4 proposes such a model.

2.3 Narrowband and Wideband Telephony

The deployment of efficient wideband encoding algorithms in VoIP telephony is a very promising approach to improve average quality of VoIP communications. The human ear can hear sound waves in the range of frequencies approximately between 20 Hz and 20 kHz. So, it would be ideal sending all frequencies up to 20 kHz over a connection to get perfect communications. In reality, the traditional narrowband PSTN telephony transmits signals up to 4 kHz thus creating problems related to sound recognition and speaker identification. The example of such a problem is demonstrated on Figure 2.6, which is taken from [39]. It shows that the differences between "s" and "f" sounds are practically indistinguishable in narrowband telephony. A similar picture can be generated to compare some other sounds.

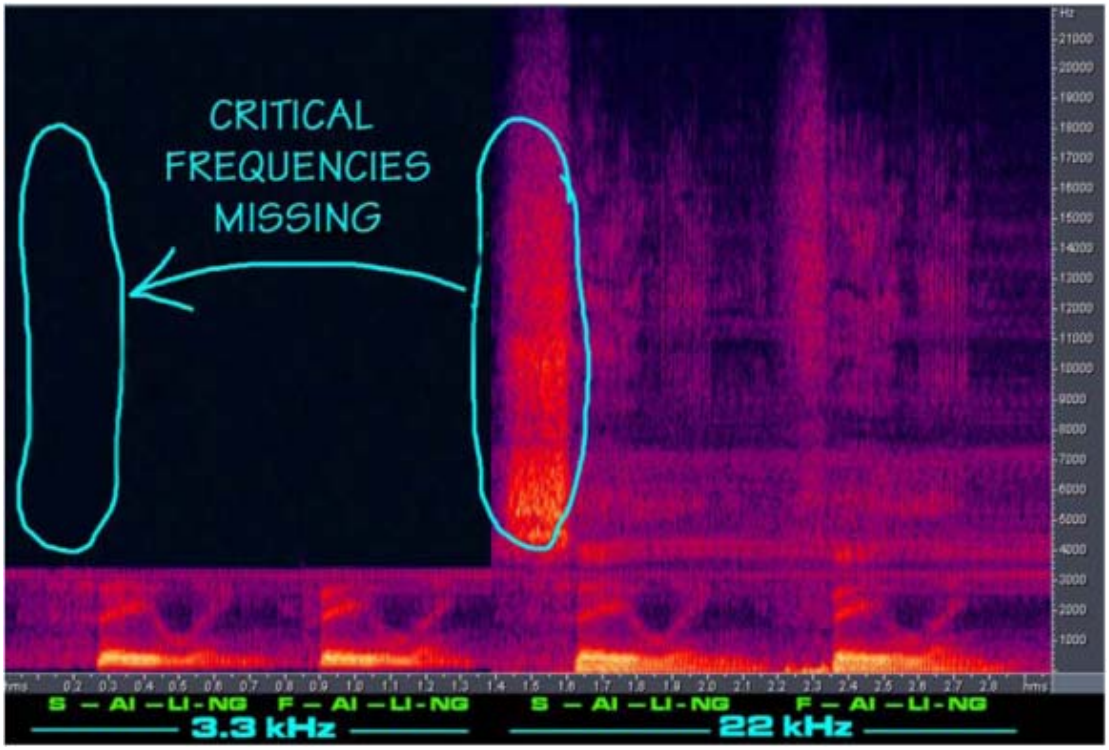


Figure 2-6: Speech spectra of "sailing" and "failing" at 3.3 kHz and 22 kHz

How much bandwidth is required to provide comfortable communications? The exact answer does not exist. Wideband telephony extends the frequency range and this

dissertation speaks about 8-kHz telephony as a way to improve average quality of VoIP communications. But many people think that even 8-kHz is also not enough to provide perfect (or very good) quality, although it is still much better than 4-kHz. 12-kHz speech transmission is considered to be optimal [40], but it requires more bandwidth resources.

We would like to explain several terms used in this dissertation. It was said that the traditional narrowband speech uses 4-kHz frequency range and 8-kHz sampling rate to convert this speech to a digital form. It is necessary to remember and to understand that the PSTN telephony uses frequency band between 300 Hz and 3400 Hz (a little more then 3 kHz) but not 4 kHz exactly. But the 8-kHz sampling rate is still used in this situation. Similarly, this dissertation speaks about wideband telephony as telephony with the twice the maximum frequency range and twice sampling rate (8 kHz and 16 kHz respectively). But most existing wideband speech encoding mechanisms do not use 8-kHz signal frequency range but a 7-kHz range (50 Hz – 7000 Hz) and 16-kHz sampling. This is done because of convenience: for simplicity it is assumed that narrowband and wideband speech use 8 and 16-kHz sampling respectively. This assumption does not affect our results or conclusions.

Although multiple wideband codecs have been developed and provide a significant improvement in quality, narrowband codecs are still widely used mainly because of two reasons: the quality of IP-to-PSTN communications is limited by 4-kHz of the PSTN bandwidth and wideband codecs do not provide increasing quality in these situations; also, wideband codecs provide higher communication quality but require more bandwidth (bandwidth consumption is still important in many situations).

2.4 Perceptual Quality Measurement

This Section explains one more concept often used to estimate quality of communications. Computational quality management mechanisms should be used periodically and rather often (later it will be defined what “often” means) to provide relatively accurate and reliable information about speech quality changes. In real networks, packet loss is not uniform, delay is variable, especially if variable jitter buffer algorithms are used, and end-user parameters

can be changed. As a result, the speech quality level may change very quickly and significantly and it is necessary to monitor these changes. It is assumed that the network is stable and that quality does not change noticeably during these short periods of time. So, using the E-model and the extension of the model proposed in Chapter 4, it is possible get a sequence of instantaneous (for example, during several hundred milliseconds) values of speech quality (Figure 2.7).

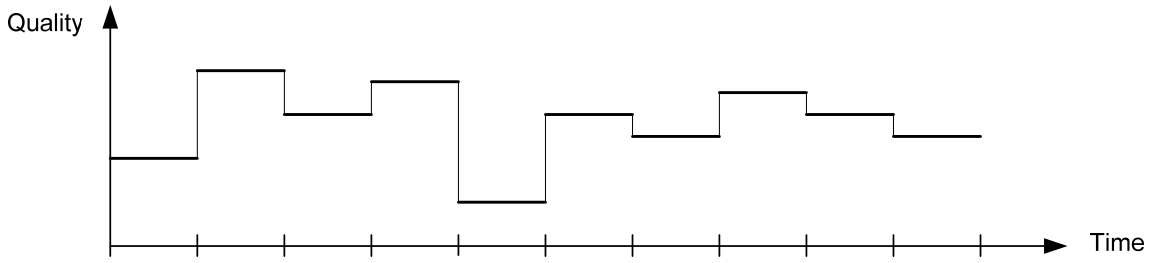


Figure 2-7: Instantaneous speech quality variation

Measurable voice quality can change significantly and immediately but it takes some time for users to understand that the quality has changed. Perceptual (real) speech quality is different from an instantaneous computational quality. Perceptual quality takes into account factors not only during the last short period of time (last second or several seconds), but all quality values and quality variation history starting at the beginning of a call. So, in addition to the computational quality model, it would be useful to estimate (1) instant perceptual speech quality at any moment of time during a call and (2) integral call quality at any moment of time during a call. The term "call quality" means average user opinion about the quality of a conversation. It does not take into account call blocking or dropping, time to establish connection, etc. Perceptual call quality metrics provides one more parameter to make decisions about real-time speech quality management.

The effect, which reflects the way that a listener remembers call quality, is called "recency" effect. This effect implies that periods of low or high quality positioned at the end of a speech sample have a stronger influence on the overall session quality then when such periods are positioned in the beginning of the sample. In tests conducted by AT&T [42], a

burst of noise was created and moved from the beginning to the end of a 60-second call. When the noise was at the start of the call, users reported a higher MOS score than when the noise was at the end of the call. Tests reported by France Telecom [41] showed a similar effect. The effect is believed to be due to the tendency for people to remember the most recent events or possibly due to auditory memory, which typically decays over a 15-30 second interval [41]. Clark [43] proposed a computational model to describe this effect. If the instantaneous voice quality changes from "good" to "bad" at some moment during a call, then it can be expected that a user initially would not be too concerned. However, after some time, the listener would become annoyed with the voice quality degradation. In [43] this recency effect is modeled by an exponential curve with time constant of 5 seconds in the transition from "good" to "bad" and 15 seconds in the transition from "bad" to "good" (Figure 2.8).

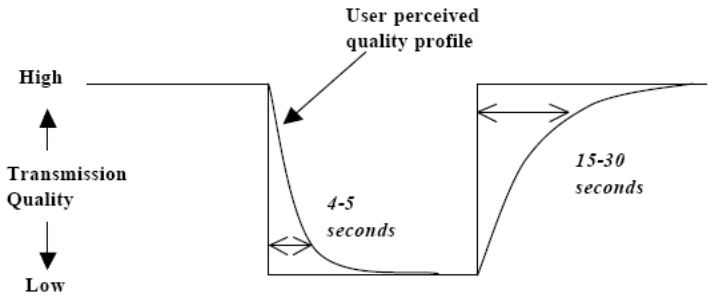


Figure 2-8: Exponential perceptual quality model (1)

The experiments in paper [44] concluded that the exponential constant is 14.3 seconds in the case of quality improvement and 9 seconds in the case of decrease in quality. Using these models, information about perceptual speech quality can be obtained at any moment of time during a call (an instantaneous perceptual call quality level) (Figure 2.9).

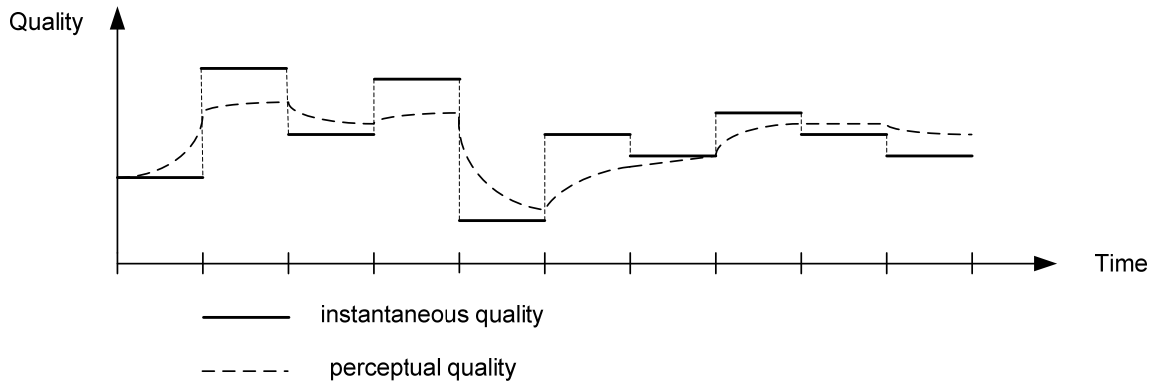


Figure 2-9: Exponential perceptual quality model (2)

To estimate the integral call quality (quality of communication session, which takes into account all history of speech quality variations), Clark [43] proposed

$$I_e(end\_of\_call) = I_e + (k(I_{end} - I_e))e^{-y/t} \tag{2.8}$$

where I_e is the average impairment factor during the call; I_{end} is the impairment after the decrease of quality caused by packet loss; k is constant (assumed to be 0.7); t is also constant (typically 30 seconds); y is the time delay since the last burst of packet loss. The paper proposes to calculate average equipment impairment factor I_e during a communication session and to find the effect of degradation after period of time *y*. One degradation period caused by bursty packet loss is assumed. The R-factor is determined from the expression R = 93.2 – I_e (end of call) – I_d.

Papers [42, 45, 46] indicate that average impairment factor cannot be used as an indicator of integral perceptual quality. Some weighting of impairment events is required both with regard to intensity of degradation and its position. Rosenbluth in paper [42] proposes to use the weighting average with weights

$$W_i = \max\left[1,1 + (0.038 + 1.3 \cdot L_i^{0.68}) \cdot (4.3 - MOS_i)^{\left\{0.96 + 0.61 \cdot L_i^{1.2}\right\}}\right] \tag{2.9}$$

$$MOS_I = \frac{\sum_i W_i \cdot MOS_i}{\sum_i W_i} \quad (2.10)$$

MOS_I is the integral perceptual call quality; MOS_i is the MOS during a smaller measurement period; L_i is a location of a degradation period (measured on 0-to-1 scale; 0 indicates the beginning of a conversation, 1 is the end of a conversation; the parameter changes proportionally to time starting from the beginning of a call).

This perceptual call quality metrics can potentially be used as one of decision parameters for adaptive speech quality management. This question will be investigated in more details in Chapter 5.

Chapter 3

Speech Quality Management

This Chapter describes the previous work in the area of voice encoding and adaptive speech quality management. Section 3.1 provides an overview of the well-known Adaptive Multi-Rate codec (AMR) used in GSM/UMTS networks. It also discusses several of the most popular VoIP codecs (narrowband and wideband), performance characteristics of these codecs and their bandwidth requirements. Section 3.2 reviews existing adaptive jitter buffer strategies. Sections 3.3 and 3.4 analyze potential effects of two novel technologies on quality of VoIP communications: IPv6 and MPLS.

3.1 Speech Encoding Techniques Overview

3.1.1 Adaptive Multi-Rate Codec (AMR)

The Adaptive Multi-Rate codec (AMR) is an audio data compression scheme optimized for speech coding and used in GSM and UMTS networks. An AMR codec uses adaptation technique to manage the quality of communications. The philosophy behind AMR is to lower a codec rate as the interference increases and thus enabling more error correction to be applied. If conditions in the network are bad, the source coding (data rate) is reduced and the channel coding (error correction) is increased. This improves the quality and robustness of the network connection while sacrificing some voice clarity.

AMR can select one of eight different bit rates: 12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15 and 4.75 kbps [26, 17]. The variable bit rate is achieved by using an adaptive codebook technique (that is different encoding schemes based on the ACELP compression algorithm) and can be specified for each frame: the frame length is the same (160 samples; 20 milliseconds), but the number of bits per frame is variable. There is a wideband specification for this codec [8, 18, 47] with rates of 6.60, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85, 23.05

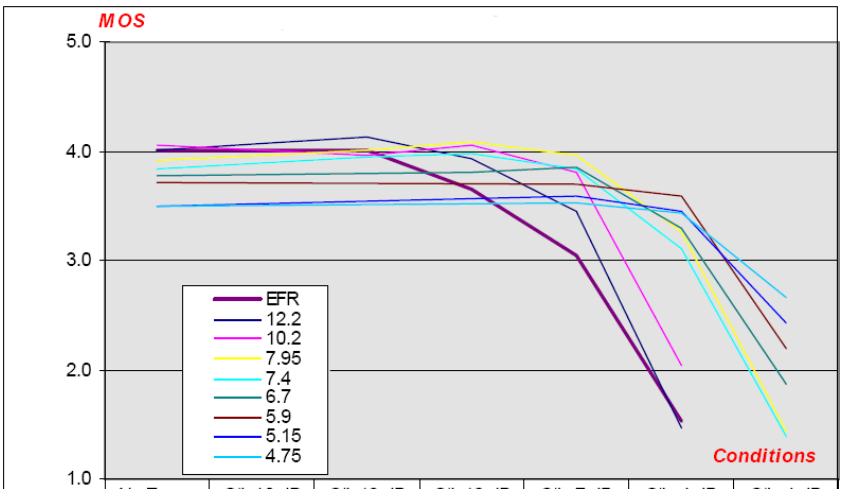
and 23.85 kbps.

To choose a bit rate, the receiving side measures quality of incoming radio channel. The quality indicator (QI) is used for this purpose and is defined as an equivalent carrier-to-interference (C/I) ratio. which is the ratio of power in an RF carrier to the interference power in the channel. The QI is then compared to a set of pre-defined thresholds to decide which codec mode has to be used. The thresholds are normally fixed during a call, but the system can initiate a change to these parameters. The thresholds depend on the radio condition, frequency hopping scheme, network configuration and other factors and the process of threshold choice can be complicated. Also, network conditions vary over time and it is likely that even well-selected adaptation thresholds will not be the best.

The specification [17] provides the narrowband AMR codec quality estimates. The quality testing was based on a set of subjective experiments: clear channel testing, testing with background noise in the channel (several noise patterns), testing with errors in the channel. The results of the experiment are measured based on the traditional 1-to-5 MOS scale. The authors emphasize that MOS scores depend on the conditions in which they were recorded (listening conditions, language, cultural background of the listening subjects) and can be different in other experiments.

The codec quality and the codec bit rate depend on carrier-to-interference ratio and on error and noise levels. The testing under ideal conditions gives Mean Opinion Score of 4.13 (full-rate codec, fixed rate 12.2 kbps), compared to 4.41 for G.711. The MOS scores for other C/I ratios are shown in Figure 3.1. Also, the experiments were designed to evaluate the AMR performances with the turned on adaptation. Multiple C/I profiles were generated simulating different behavior of the radio channel and different effects. The specification shows that the bit rate variation can provide better quality than in the case of the fixed-rate GSM EFR codec, which is also shown in Figure 3.1.

The Specification [18], the ITU standard G.722.2 [8] and the paper [47] analyze the quality of the AMR-WB codec. The subjective experiments are similar to the previous case and include multiple scenarios using clear channel, multiple noise patterns, music performance, etc. The results show that the quality of the 23.85 kbps AMR-WB codec exceeds the quality of the narrowband 64 kbps G.711 codec (4.5 vs. 4.4). The performance of the 12.65 kbps AMR-WB codec is about 4.2 on the 1-to-5 MOS scale.



	No Errors	C/I=16 dB	C/I=13 dB	C/I=10 dB	C/I= 7 dB	C/I= 4 dB	C/I= 1 dB
EFR	4.01		4.01	3.65	3.05	1.53	
12.2	4.01	4.06	4.13	3.93	3.44	1.46	
10.2	4.06		3.96	4.05	3.80	2.04	
7.95	3.91		4.01	4.08	3.96	3.26	1.43
7.4	3.83		3.94	3.98	3.84	3.11	1.39
6.7	3.77			3.80	3.86	3.29	1.87
5.9	3.72				3.69	3.59	2.20
5.15	3.50				3.58	3.44	2.43
4.75	3.50				3.52	3.43	2.66

Figure 3-1: AMR codec quality (clean channel)

AMR and AMR-WB were originally designed for circuit-switched mobile radio systems, but they may also be suitable for other real-time speech communication services over packet-switched networks such as the Internet. Currently fixed-rate AMR codecs are used for speech encoding in VoIP. Evidently, the process of adaptive quality management in IP networks with AMR codec will be different than in the case of GSM/UMTS: there is no channel interference, there are IP packets instead of radio signals, and the threshold choice and management process will be different. There are several papers investigating this area. For example, J. Seo *et al.* [48] use network jitter as the indicator of network quality. Eight states are assigned (20-ms jitter increments); each state corresponds to one of the eight bit rates. No detailed experimental confirmation of the assumption was provided. J. Matta *et al.* [49] and Y. Huang *et al.* [50] propose schemes to optimize the speech quality in VoIP applications using the AMR codec mode switching, FEC and retransmissions based on the E-model. But the algorithms presented in these papers do not take into account multiple factors and parameters. For example, in the case of speech quality degradation, they propose using

redundant packets, to increase the Frame Error Correction bits and even retransmission to decrease the number of lost packets. If this approach is applied to a group of calls, a significant traffic increase can make the situation even worse.

This dissertation is going to explore a similar concept: managing speech quality adaptively based on some criteria. But, in addition to variable bit rate achieved by variable compression, as in the case of the AMR codec, a set of other parameters is used to manage speech quality. These parameters include not only compression and jitter buffer adaptation, but also changes of signal frequency range (signal quality) and packet duration. A detailed explanation will be provided below. Like these papers, the algorithms proposed in this dissertation use the E-model as one of the decision making parameters. In addition to this model, the next Chapter of this dissertation proposes an extended version of the E-model, which is valid not only for the narrowband but also for the wideband telephony.

3.1.2 Speech Encoding and Packetization

Theoretical investigations and simulations provided in this dissertation, use several different codecs, for example: the G.711 codec [6], the G.726 [9], the G.729 [10] and several others. The goal of this Section is to provide an overview of the main speech encoding parameters used by these codecs, to explain how a voice packet is formed, and to address bandwidth-related issues.

G.711 is one of the oldest codecs, but it is still finds considerable use in telephony. The ITU-T Standard [6] defines two main encoding algorithms: the μ-law and the A-law algorithms. The first algorithm is mainly used in the United States and in Japan; the second algorithm is widely deployed in Europe. More details about the differences in the algorithms can be found in [6]. The codec does not use any compression, it has 8-kHz sampling rate, requires 64 kbps of audio bandwidth and provides very good (higher then the toll-grade) quality level. The wideband version of this codec (16-kHz sampling rate, no compression) is not used in practice because of very significant bandwidth consumption. Instead, the G.722 wideband speech codec operation at 48-64 kbps and defined by [7] is used. The quality of the wideband codec is noticeably higher than the quality of the standard G.711 and the compression is not that great. Considering the fact that it has the same bit rate as the

narrowband G.711 codec and delivers much more realistic sound, the G.722 is one of the best codecs to use. But the G.711 codec is not replaced by the G.722 and will not be replaced in the near future. Although multiple wideband codecs for VoIP telephony were developed, narrowband codecs are still widely used for three main reasons: (1) IP-to-PSTN call quality is limited by the 4-kHz of PSTN bandwidth, so a user will not get any quality improvement using a wideband codec and making a call through the PSTN; (2) not all IP phones support wideband codecs (although many of them do); (3) because of bandwidth limitations, narrowband codecs with high compression ratio are often used.

The G.726 and G.729 codecs are also very popular. The G.729 codec is computationally complex, but provides significant bandwidth savings. It has 8:1 compression and requires just 8 kbps of audio bandwidth. But the quality level, provided by this codec, is generally lower then the toll-grade quality level: the maximum achievable MOS is about 3.9. Currently, researchers try to develop a wideband version of this codec, which will use 16 kbps of audio bandwidth to encode a wideband signal. It is expected that the quality of this codec will be higher than the quality of the G.711 codec, but the G.729-WB does not exist now. The G.726 is an ITU-T ADPCM speech codec standard covering the transmission of voice at rates of 16, 24, 32, and 40 kbps. Instead of 8-bit PCM encoding used in the G.711 codec, it uses 2-bits, 3-bits, 4-bits and 5-bits respectively. In this dissertation we use the G.726 codec with 2:1 compression (32 kbps of audio bandwidth), which still provides toll-grade quality of communications (it has about 4.1 MOS in the absence of significant delay and packet loss).

There is a variety of other non-commercial codecs used for voice encoding in different VoIP applications: for example, Speex, Vorbis, iLBC, etc. These codecs are not analyzed in this Chapter because no systematic studies regarding the quality of these codecs, their loss robustness and complexity exist.

Below, the voice packet structure is presented as the packet travels through the Internet. The analysis also estimates the effect of overhead (service information) on the total voice stream bandwidth consumption. Figure 3.2 demonstrates the structure of a VoIP packet. To form one IP packet, an overhead with control information is added, and the length of this overhead is 40 bytes: the IP header is 20 bytes; the User Datagram Protocol (UDP) header is 8 bytes; and the Real-Time Transport Protocol (RTP) header is 12 bytes. So,

for every voice packet, 40 bytes of RTP, UDP and IP headers are added plus, for example, 38 bytes of Ethernet overhead if we speak about VoIP over the Ethernet. If the G.711 codec is taken, the length of one 10 ms packet is 80 bytes. So, the length of the overhead is 50% of the packet length and the total bandwidth required for a voice stream transmission is 96 kbps. The G.729 codec has the frame length of 10 bytes (8:1 compression ratio); the total packet size is 50 bytes and the total bandwidth required per call is 40 kbps. The bandwidth required to send an audio streams using the G.711 is 8 times higher then that with the G.729 codecs, but the total bandwidth (including overhead) is just 2.4 times higher (Figure 3.3).

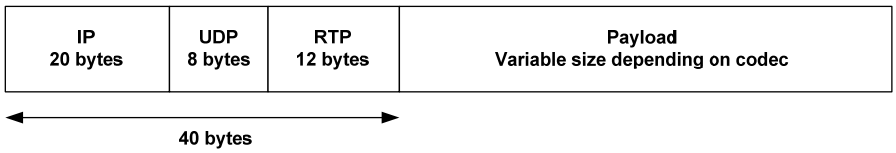


Figure 3-2: VoIP packet structure

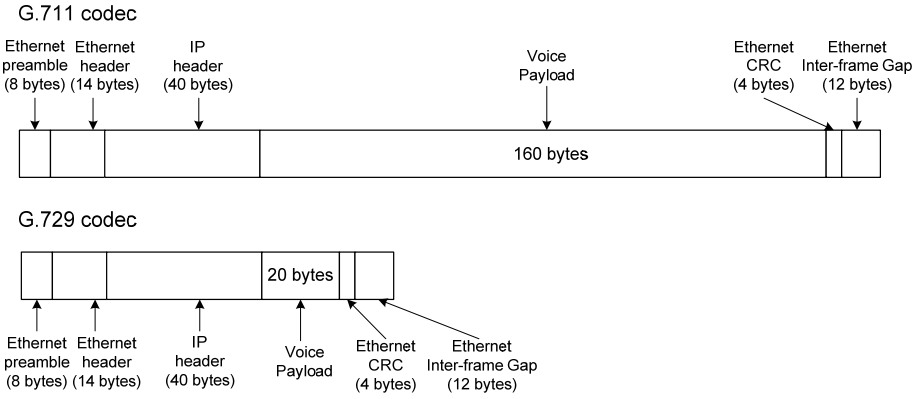


Figure 3-3: VoIP frame structure for G.711 and G.729 codecs (20 ms packet size)

Evidently, this approach is not efficient because the overhead is a significant part of the total packet length and voice traffic. To make the process of voice transmission more efficient from the bandwidth consumption point of view, it is possible to put several frames to one IP

packet. But it is not recommended to put too many frames into one packet because it will be required to wait longer to form one IP packet; the end-to-end delay will increase significantly, and the quality of a voice stream will degrade. Usually, two to three packets are included to one frame.

3.2 Adaptive jitter buffer strategies

The variation of packet delays in the network is called jitter. Jitter is eliminated by jitter buffers, which temporarily store arriving packets and send them to a receiver in equal intervals. If jitter buffer is too small, a lot of packets may be discarded because of a significant delay variation. This will negatively affect speech quality. Increasing the jitter buffer allows waiting longer for delayed packets but increases the overall end-to-end delay, which also negatively affects speech quality. A lot of research focuses on adaptive jitter buffer strategies to find some optimal point in the tradeoff between the end-to-end delay and packet loss and to optimize speech quality dynamically.

The basic adaptive playout algorithm of Ramjee *et al.* [51] uses two statistics to make a decision: delay and delay variance (jitter). Specifically, the delay estimate for packet *i* is computed as $d_i = \alpha \cdot d_{i-1} + (1-\alpha) \cdot n_i$, where n_i is the *i*-th packet delay. The variation is computed as $\upsilon_i = \alpha \cdot \upsilon_{i-1} + (1-\alpha) \cdot | d_i - n_i |$. Packet playout time in this algorithm is calculated as $p_i = t_i + d_i + \beta\upsilon_i$; where t_i is the time the packet was sent. In [51] $\alpha = 0.998002$ and $\beta=4$. The second algorithm proposed in [51] uses two values of α: one for increasing network delays and the other for decreasing network delays. The value of α determines how rapidly the delay estimate adapts to fluctuations in the network. If a current network delay n_i is larger then d_{i-1}, the equation of d_i is set as $d_i = \alpha \cdot d_{i-1} + (1-\alpha) \cdot n_i$ with $\alpha = 0.998$; if it is smaller, then $\alpha = 0.75$.

The paper [54] proposes to adjust α depending on the variation in the network delay: α is low when the variation is high and vice versa.

While these equations estimate d_i and υ_i for each packet, playout time p_i is adjusted only at the beginning of a new talkspurt. The same idea is used by Moon *et al.* [52], who proposed to use silence intervals between talkspurts (word or phrases) to adapt to the delay

variation. Silence periods between phrases can be modified (by either expanding or contracting) to improve speech quality capturing more late-arriving packets. The difference from the previous algorithms is that this methodology chooses playout time by finding a delay, which represents the given packet delay quantile among the last several talkspurts. This determination is based on a continuously updated histogram of packet delays. When a new packet arrives, the delay of this packet replaces the delay of the oldest packet in the histogram and a required percentile is computed. The paper also assumes that no adaptation exists within a talkspurt.

Ramjee [51] has also proposed an algorithm based on a spike detection mechanism. A spike is a sudden large increase in the end-to-end network delay followed by a sequence of packets arriving almost simultaneously. If a spike is detected, the delay d_i is computed as $d_i = d_{i-1} + n_i - n_{i-1}$; if the algorithm is not in the spike mode, the equation is similar to the first algorithm $d_i = \alpha \cdot d_{i-1} + (1-\alpha) \cdot n_i$, with $\alpha = 0.875$. [56] provides a different algorithm with a spike detection mechanism.

S.Huang *et al.* [57] and L. Atzori, M. Lobina [55] proposed an adaptive model based on equations from [51] and [54] and the E-model [1]. They try to choose a jitter buffer size that optimizes computational speech quality level. The significant problem of these algorithms is that it is very difficult to find an optimal jitter buffer value because of continuously changing network conditions.

There are many other adaptive jitter buffer mechanisms. It is difficult to say, which of them provides the best performance and no studies investigating this question are available. To choose an adaptive jitter buffer method for simulations in this dissertation, the classical algorithm presented in [51] and described above was implemented. The moving average delay is calculated using the equation $d_i = \alpha \cdot d_{i-1} + (1-\alpha) \cdot n_i$; the variation is $\upsilon_i = \alpha \cdot \upsilon_{i-1} + (1-\alpha) \cdot | d_i - n_i |$; packet playout time is $p_i = t_i + d_i + 4\upsilon_i$ and adjusted in the beginning of a new talkspurt. The paper [51] proposed using $\alpha = 0.998002$ and $\beta = 4$. Because of the high value of α, this algorithm mainly relies on the moving average delay and relatively slowly adapts to sudden changes of instantaneous packet delays. In congested networks, this slow adaptation may cause a significant packet loss due to high jitter. This value of α does not provide an optimal quality level. Smaller values of α (for example, 0.75) put higher wait on instantaneous packet delay n_i. This helps to decrease packet loss but significantly increases

the end-to-end delay. In this project the intermediate value of $\alpha = 0.875$ is used: according to the preliminary simulation study, the algorithm with this value of α has better performance (provides better quality calculated using the E-model) than with $\alpha = 0.75$ or with α close to 1.

3.3 IPv6 and Quality of VoIP Technologies

Multiple technologies can be used to manage the quality of real-time traffic: different network-based QoS schemes (IntServ, DiffServ, RSVP, traffic shaping, multiple queuing algorithms), adaptive jitter buffer, etc. None of these technologies completely solves the numerous problems related to communication quality. This dissertation does not have as its goal the comparison of the efficiencies of these technologies or to propose even better algorithms of dynamic speech quality management. This project explores an alternative way of communication quality management and analyzes the performance of sender-based codec adaptation. Of course, it would be nice to compare the efficiencies of the algorithms proposed in this project with all other existing technologies and quality management approaches but (1) this study is very complex and can probably form the background for a separate dissertation; (2) we do not have sufficient technical facilities; (3) there is no enough information about performance of these technologies, and results of existing studies are often contradictory. This dissertation uses "traditional" assumptions about the network. It assumes that the IPv4 protocol is used, that data and voice are transmitted through the same channel, and that packets do not carry information about priorities and quality requirements. All assumptions and the network setup will be discussed in more detail in Chapter 5.

The remaining two Sections of this Chapter discuss two relatively novel technologies, which can potentially improve the quality of real-time traffic transmission in the Internet: MPLS technology and the IPv6 protocol. This Section has two main goals: (1) to analyze very briefly the advantages and potential problems of these technologies and (2) to demonstrate that the algorithms and investigation results, proposed in this dissertation, can still be used even in the case of the wider adoption of these two technologies.

IPv6 is the "next generation" protocol designed to replace the current version 4 Internet Protocol [58]. This protocol provides not just an increase in addressing space, but also includes QoS management. IPv6's very large addressing space allows the allocation of

large address blocks to ISPs and other organizations. This enables organizations to aggregate the prefixes of all its users into a single prefix and announce this one prefix to the IPv6 Internet. The implementation of a multi-leveled address hierarchy provides more efficient and scalable routing [59]. The deployment of a special field in the IP header will give processing and routing priority to the VoIP packet streams, which will reduce the delay in the transmission of the real-time packets. So, the average quality of communications will likely increase. But we firmly believe that the deployment of this technology will not eliminate the need for our research results and algorithms because of several reasons:

(1) One of benefits of adaptive algorithms is the ability to manage the efficiency of communications and voice traffic load in the network. The situation will not change in the case of IPv6 protocol. IPv4 uses overhead of 20 bytes (160 bits); IPv6 overhead is 40 bytes (320 bits) [58]. IP header size will increase twice. VoIP packets contain not just IP overhead but also RTP header (12 bytes) and UDP header (8 bytes). So, the total size of the voice-over-IPv6 overhead is 60 Bytes and it is 50% larger than the voice-over-IPv4 overhead. The efficiency of transmission will still be important and adaptive encoding can still be used to manage this efficiency if this is required.

(2) Compatibility issues are also very important. Most hardware and software that are currently available are tied to the IPv4 address structure. The whole Internet is not expected to fully upgrade to IPv6 soon, although the process of this transition has already started. It will take some time before a complete transition from IPv4 to IPv6 is made. During the transition period, full compatibility must be maintained between IPv4 and IPv6 addresses to avoid disruptions and provide a continuous service to all clients. Voice over IPv6 requires IPv4 tunneling over IPv6, and this translation may have negative effects on speech quality. So, one should not expect that the quality of the network and the quality of real-time traffic transmission will increase significantly.

(3) More efficient routing algorithms provided by IPv6 will probably not eliminate the main reason of real-time traffic degradation - congestion. This congestion can still cause significant additional delay and delay variation and/or packet loss. So, the intelligent choice and management of voice stream parameters will still be important.

In conclusion, it is not evident that the deployment of IPv6 with additional QoS mechanisms will automatically significantly increase the quality and eliminate all problems

related to real-time traffic transmission in the network. Effects of IPv6 on speech quality have to be investigated and the clear answer to this question does not exist now. There is a lot of concentration on the engineering of IPv6, but not much is discussed about its performance. Much related to IPv6 is still in the planning stage. If speech processing in the network becomes more efficient because of IPv6, this is good. But the adaptive encoding algorithms, proposed in this dissertation, can still be used together with other network management schemes.

3.4 MPLS and Quality of VoIP Communications

Multi-Protocol Label Switching (MPLS) technology, described in RFC 3031 [20], is considered as a very promising way to improve average quality of real-time communications. This is a network traffic engineering mechanism that is independent of routing protocols and tables. Traditional routing protocols make decisions about packet routing at each hop independently: the next hop is chosen based on information in IP network layer header and analysis of a routing table. This process is done for each packet. In the case of MPLS technology, the packet header is analyzed once, when a packet enters the MPLS network. When a path to a destination is chosen (special protocols are used for this purpose), each packet gets a label that describes how to forward this packet through the network. At each hop, a packet is routed based on the value of incoming interface and label and forwarded to an outgoing interface with a new label value. The transition in label values defines the network path since the label is stored at each router (called Label Switched Router (LSR)). Since the mapping between labels is constant at each LSR, the complete path is determined by the initial label value. Many different headers can map to the same label, as long as those headers result in the same choice of next hop. When the packet reaches the output (egress) router, the label is removed, and the packet again becomes a regular IP packet and is again forwarded based on its IP routing information. The dedicated paths in the MPLS network (virtual circuits) are called Label Switched Paths (LSP). There are several algorithms, that can be used to form and restore LSPs.

The main QoS features provided by MPLS are listed in Table 3.1 (source – Cisco MPLS documentation [60]).

Table 3-1: MPLS QoS services and features

Service	QoS Function	Description
Packet classification	Committed access rate (CAR). Packets are classified at the edge of the network before labels are assigned.	Classifies packets according to input or output transmission rates. Allows you to set the MPLS experimental bits or the IP Precedence or DSCP bits (whichever is appropriate).
Congestion avoidance	Weighted Random Early Detection (WRED). Packet classes are differentiated based on drop probability.	Monitors network traffic to prevent congestion by dropping packets based on the IP Precedence or DSCP bits or the MPLS experimental field.
Congestion management	Class-based weighted fair queuing (CBWFQ). Packet classes are differentiated based on bandwidth and bounded delay.	An automated scheduling system that uses a queuing algorithm to ensure bandwidth allocation to different classes of network traffic.

MPLS technology provides significant traffic management flexibility: data can be transferred over any combination of the Data Link Layer technologies, support is offered for all Network Layer protocols, good scalability and increased effectiveness of links utilizations is provided.

Many people consider MPLS as an excellent solution to significantly improve the quality of VoIP communications. Service providers implement MPLS to prioritize VoIP packets, manage bandwidth and shape traffic to give VoIP traffic higher priority over corporate backbones. Other companies argue that MPLS is not an ideal solution, which can instantly solve all VoIP quality problems. For example, Avaya's white papers [61], [62] say: "Avaya measured the ability of MPLS services to support business quality VoIP communications. Avaya found that the performance and availability of MPLS services are not better than that of default Internet connections. These results suggest that an MPLS

service is not the panacea for VoIP. MPLS service is in fact essentially comparable to Internet service. Both provide good base connectivity, but by themselves neither can deliver the quality and availability required for business-quality voice communication." A white paper from Ditech Networks [63] published in January 2007, also says that "good" or "excellent" voice quality is possible only in VoIP networks where all impairment to voice quality are successfully removed. The existing approaches (including MPLS) complete only a portion of the traffic engineering and QoS picture for delivering VoIP; they do not provide end-to-end solutions and do not solve the problem completely. One more weakness of the technology is its inability to provide application-level routing intelligence, which is a fundamental component for voice delivery. For example, MPLS is not able to provide alternate routing on the call level to prevent latency, delay, packet loss and jitter for VoIP packets. Also, a selection of LSP does not will be able to handle its bandwidth requirements, which are increased by other LSPs traversing that same router and competing for the same resources. MPLS is a core network technology and it cannot prevent loss of quality caused by access networks.

MPLS is a very promising technology; many providers develop MPLS networks to achieve more efficient traffic engineering and quality of services management. There are some problems with the technology and time is required for a future improvement of IP QoS and MPLS. But many experts agree that MPLS can become an efficient core-network technology, providing a quality improvement of real-time communications.

In our dissertation we do not intend to prove that adaptive speech encoding will be better or worse than, for example, MPLS and IPv6. We investigate an alternative way of speech quality management, which can potentially be used together with other network-based QoS management algorithms and technologies.

Chapter 4

Computational Speech Quality Model for Variable VoIP Communications

Section 4.1 of this Chapter provides background information and describes previous research about wideband VoIP quality. Section 4.2 proposes a computational speech-quality model for wideband VoIP and describes how to compare the qualities of narrowband and wideband telephony, by extending the R-factor scale and by investigating how multiple parameters from the narrowband E-model change in the case of wideband communications. Section 4.3 presents a computational quality model for "non-standard" codecs (for any set of speech encoding parameters).

4.1 Background and Related Research

As mentioned in Section 2.2.2, the traditional E-model was designed for narrowband communications only. This Chapter proposes a methodology for extending the E-model extension to wideband telephony. Although several papers discuss the need for such a model and propose general approaches and rough estimates of some parameters to be included to the model, no research papers were found that propose complete methodologies for the development of the wideband E-model. There are several presentations discussing the need for this model and proposing general approaches and rough estimates of some parameters to be included to the model. Trond Ulseth (Telenor R&D) [64] proposed several general steps: (1) add a wideband advantage factor to the narrowband E-model, characterizing an increase in quality due to an increase of signal frequency range, and (2) analyze the effect of wideband communications on other parameters included in the E-model. But, no computational estimates and equations or additional details are provided. The white paper [65], published by Telchemy, estimates that the maximum wideband codec quality may result in an R-factor of 105, whereas the narrowband G.711 codec has an R-factor of 93, but this is

not supported by any facts or experiments. The Telchemy presentation [66] proposes to use an R-scale from 0 to 120 for the wideband telephony. Sebastian Möller *et al.* [67] investigated the equipment impairment factor I_e in the case of wideband telephony. The authors assume that the wideband direct transmission channel corresponds to a value of R equal to 110 and concluded that the R-range from 0 to 100 should be expanded to the range of 0 to 110 (the results from the narrowband E-model should be multiplied by 1.1) and that the impairment factor I_e of narrowband codecs on the extended R-scale should be increased by 15. A. Raake [29] uses different methodologies and extrapolations of available testing results for the R-scale extension. In the first approach, listening quality tests are used to estimate quality of narrowband and wideband samples depending on bandpass. Bandpass was changed from 2 kHz to 7 kHz, MOS scores of narrowband samples were measured, and. MOS scores in the narrowband range were converted to the R-factor. One extrapolation technique provided the R-factor equal to 112, and another or about 140, as the quality of the wideband 7-kHz speech. Linear and exponential extrapolations of test results published in [68] give similar results (Figure 4.1). Based on the same experiments, Amendment 1 to ITU-T Recommendation G.107 [2] proposes using an intermediate value of R=129 as the maximum achievable wideband codec quality on the extended R-scale.

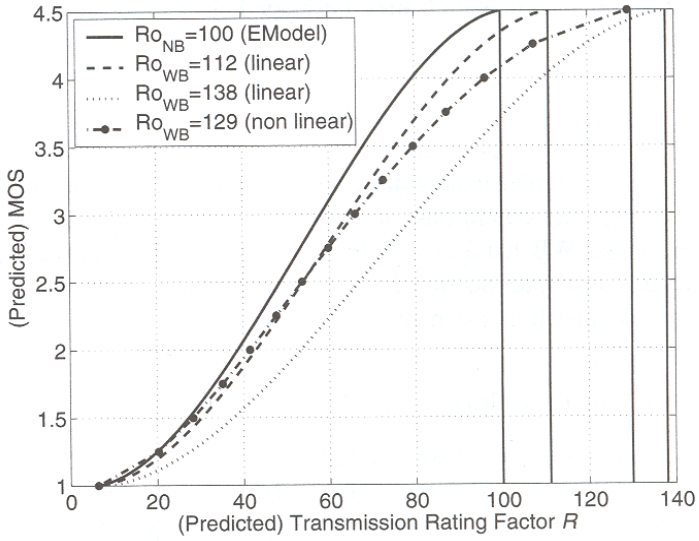


Figure 4-1: Variants of the R-scale extension

This dissertation proposes a different approach toward the wideband E-model. In Figure 4.1, the maximum achievable MOS is equal to 4.5 both for narrowband and wideband telephony. But multiple speech quality experiments (for example, [17, 68]) show that the maximum wideband quality on the 1-to-5 MOS scale is higher (approximately 4.8, although this value cannot be directly compared with MOS scores for narrowband codecs; see the discussion below). Even if a maximum wideband MOS is 4.5 is assumed, the absolute difference in quality between narrowband and wideband speech qualities with the same MOS is not clear. It is assumed here that, not only the R-scale, but also the MOS scale, must be extended to describe the effect of increase of quality due to the bandwidth extension. If the MOS scale from 1 to 5 is used to describe the quality of narrowband codecs and MOS = 4.41 is the best narrowband codec quality on this scale, the 1-to-5 MOS scale should be modified to also characterize wideband speech.

Before developing the model, several very important questions must be investigated and answered because, while wideband telephony is not new, it is has not been explored well enough. We know how to measure wideband voice quality: the new ITU-T P.862.2 standard [15], released in November 2005, proposed a mapping function between the WB-PESQ and MOS, but did not know how to compare the quality of narrowband and wideband speech. PESQ uses different approaches to measure narrowband and wideband speech quality, but it uses the same scale from 1 to 5 for the measurements. So, the results cannot be compared directly because there is no a single scale for the comparison. If the MOS of some narrowband sample is, for example, 4.0 and the MOS of some wideband sample is, for example, 3.9, it does not mean that the quality of the narrowband sample is better than the quality of the wideband sample because 4.0 score is measured with respect to the quality of direct (mouth to ear) narrowband speech and the 3.9 score is measured with respect to the quality of direct wideband speech (see Figure 4.2).

The use of wideband speech increases the listening quality and comfort due to the extension of the bandwidth in the low and high frequencies and the higher quantization rate [40], but we do not have tools or techniques to describe this quality difference quantitatively.

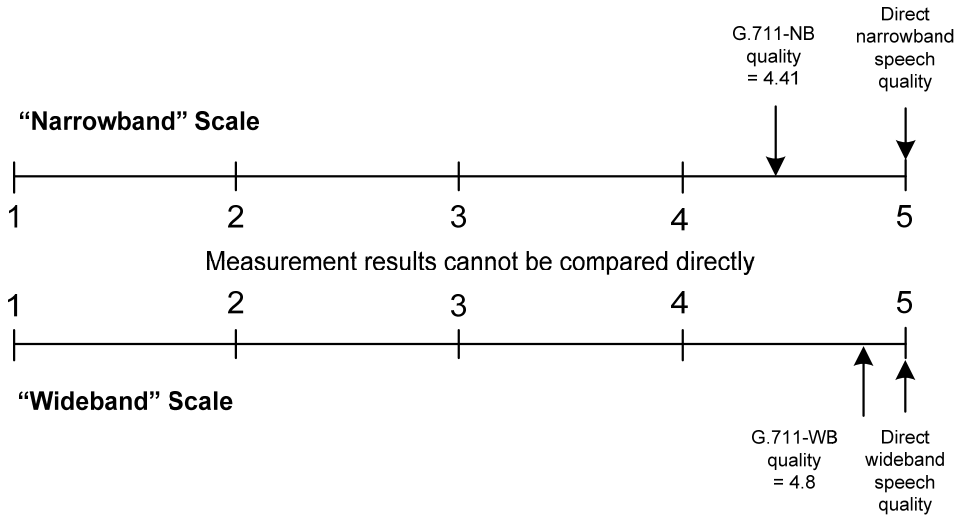


Figure 4-2: Narrowband and Wideband speech quality measurement scale

The main goals of this Chapter are:

- To develop a single MOS scale for narrowband and wideband VoIP and to establish a quantitative relationship between the new and the traditional scales
- To propose a single scale for the R-factor from the computational E-model, that will be valid both for narrowband and wideband VoIP.
- To propose the E-model for wideband communications by investigating how the narrowband E-model components (R-scale, $R_{0,}$ I_d, $I_{e\text{-}eff}$) change in the case of wideband speech.

Comparing wideband and narrowband voice qualities requires complicated human subjective testing. France Telecom's experimental results [68] were used for this purpose. The goal of their research project was to compare wideband PESQ values with narrowband values and to propose an extension of PESQ. To do that, the R&D Department of France Telecom did some experimental research to investigate the numerical relationships between narrowband and wideband voice qualities. A group of users was asked to estimate the quality of narrowband speech sampled with an 8-kHz rate (a typical human subjective test). Different narrowband and downsampled wideband codecs were used. The high-quality reference (MOS=5.0) in this case was an 8-kHz clear channel. Then, the users were asked to

evaluate the quality of the same speech samples encoded by the same codecs but with respect to the wideband reference. The high-quality reference (MOS = 5.0) in this case was a 16-kHz clear channel. Both experiments used the same MOS scale, from 1 to 5. Three different groups of 8 listeners listened to more than 100 sentences pronounced by 4 speakers (2 male and 2 female). While France Telecom's project did not cover the questions that are investigated and answered in this chapter, some of their human testing results can be used. These experiments were performed in the lab of this well-known telecommunications company using multiple codecs, a large pool of sentences, and many participating people. While these results are not perfectly accurate, because subjective testing cannot be perfect, these results are assumed to be accurate enough and they are used in the project. The results of the experiments are described in Table 4.1 and demonstrated on Figure 4.3.

Table 4-1: France Telecom's subjective testing results

Sample #	Narrowband MOS reference	Wideband MOS reference
1	1.18	1.28
2	1.53	1.35
3	1.53	1.38
4	1.85	1.85
5	2.28	2.1
6	2.5	2.45
7	2.7	2.48
8	2.78	2.55
9	3.2	2.98
10	3.4	3.04
11	3.5	3.15
12	3.52	3.15
13	3.55	3.2
14	3.62	3.2
15	3.75	3.2
16	3.75	3.33
17	3.98	3.49
18	3.98	3.5
19	4.06	3.58
20	4.07	3.63
21	4.41	3.88

The first graph in Figure 4.3 shows the reported MOS values for multiple codecs with respect to the narrowband reference. The second graph shows the Mean Opinion Scores for the same samples measured with respect to the wideband reference. The graphs demonstrate the intuitively clear result: when the real audible quality of the reference increases, the MOS scores of the same tested samples decreases.

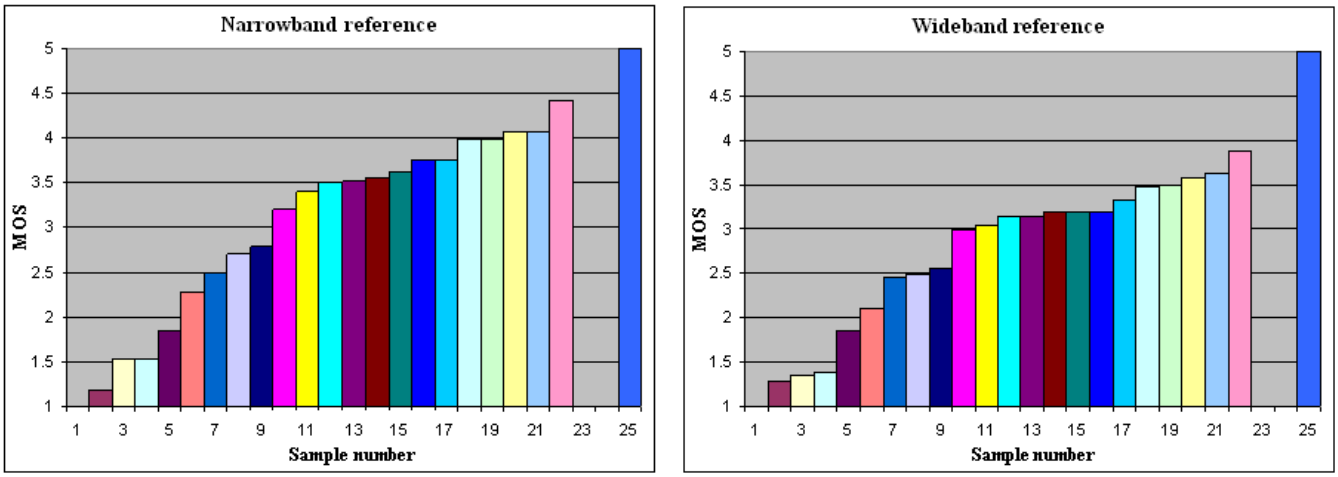


Figure 4-3: France Telecom's subjective testing results

4.2 Computational Quality Model for Wideband VoIP Communications

4.2.1 A single MOS scale for narrowband and wideband VoIP

The first goal is to propose a single scale for wideband and narrowband VoIP. If the MOS scale from 1 to 5 is used to describe the quality of narrowband codecs, how should this scale be modified to also characterize wideband speech? A metric is needed, that would allow the comparison of the qualities of wideband and narrowband speech directly. This new scale will be used in the future development of the wideband E-model.

The new scale will consist of two parts: the traditional narrowband MOS scale from 1 to 5 and the extension to characterize the quality of wideband VoIP (see Figure 4.4). On the narrowband scale, MOS equal to 5.0 corresponds to a direct (mouth-to-ear) analog

narrowband speech. The maximum speech quality, which can be achieved by narrowband codecs on this scale, is 4.41 (the G.711 codec). The reference point of the new "extended" scale (X) is direct analog wideband speech quality. So, the first step is to define the upper bound of the new scale with respect to the upper bound of the traditional narrowband 1-to-5 MOS scale by quantify the analog wideband speech quality with respect to the quality of analog narrowband voice signal.

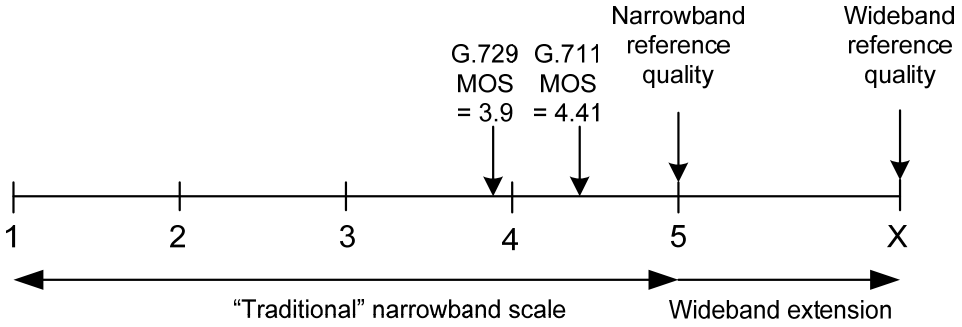


Figure 4-4: The extension of the traditional MOS scale

For this purpose, the results of the France Telecom's experiment are used. Figure 4.5 shows the relationship between sample qualities with respect to the narrowband reference (Y-scale) and the same speech samples qualities with respect to the wideband reference (X-scale). The graph demonstrates that, if the narrowband reference (which has MOS = 5.0) is projected onto the wideband reference, the previous narrowband reference will have MOS = 4.4. The trend line on the graph has an equation:

$$MOS_{NBref} = 1.175 \cdot MOS_{WBref} - 0.175 \qquad (4.1)$$

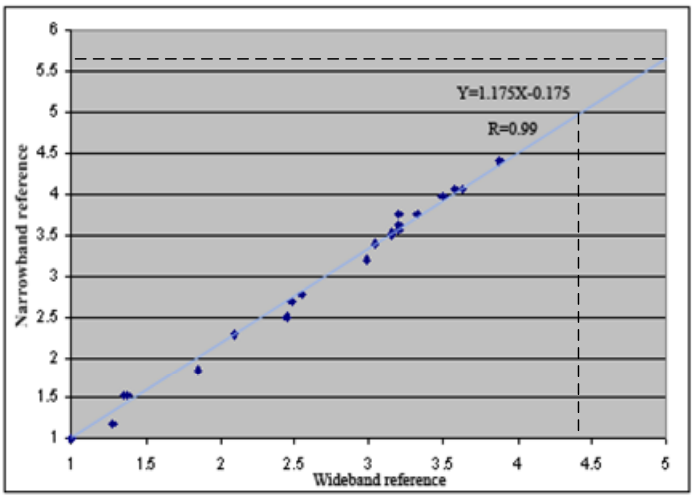


Figure 4-5: Narrowband MOS with respect to wideband MOS values

The coefficient of determination of this line is 0.99. MOS_{NBref} is the Mean Opinion Score with respect to the narrowband reference and it changes from 1.0 to 5.0. MOS_{WBref} is the Mean Opinion Score of the same speech sample with respect to the wideband reference. The results of this experiment demonstrate that instead of the traditional narrowband scale, the wideband MOS scale can be used from 1 to 5. In this case, the "narrowband" MOS scale will "shrink" so, instead of a scale from 1 to 5, it will become from 1 to 4.4 (see Figure 4.6).

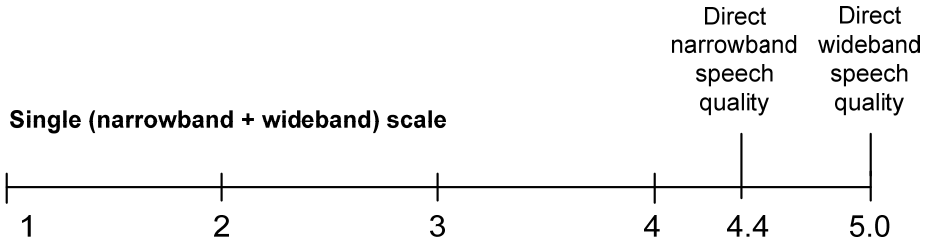


Figure 4-6: MOS scale with respect to the wideband reference

But, the goal is not to modify the narrowband 1-to-5 MOS scale because narrowband codecs and measurement techniques are used much more often than wideband codecs (at least, now). A better goal is to keep the previous narrowband MOS scale unchanged, and just extend it. Figure 4.5 and the trend-line Equation 4.1 both reveal that, if $MOS_{WBref} = 5.0$,

then MOS_{NBref} = 5.7. That is, if the highest narrowband quality is 5.0, the direct analog wideband speech has quality equal to 5.7 on the same scale (see Figures 4.7 and 4.8).

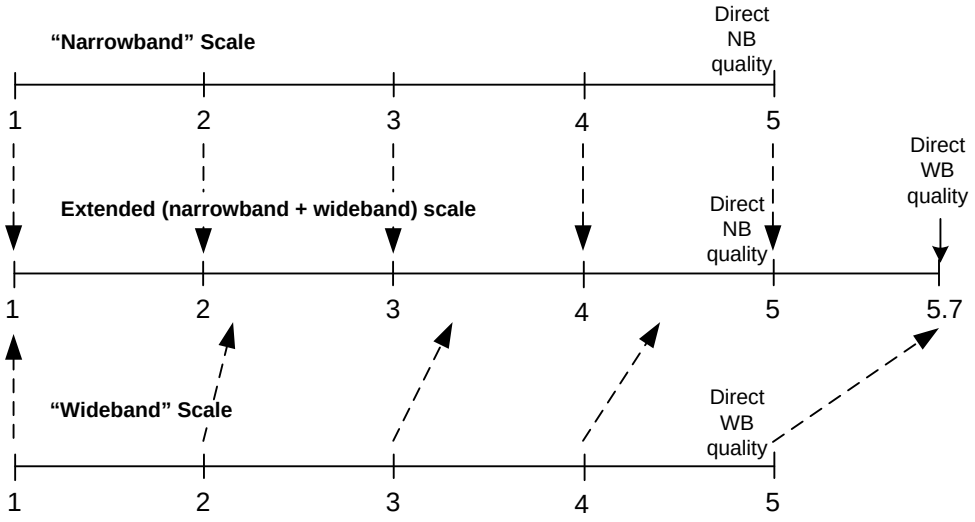


Figure 4-7: The extended MOS scale (1)

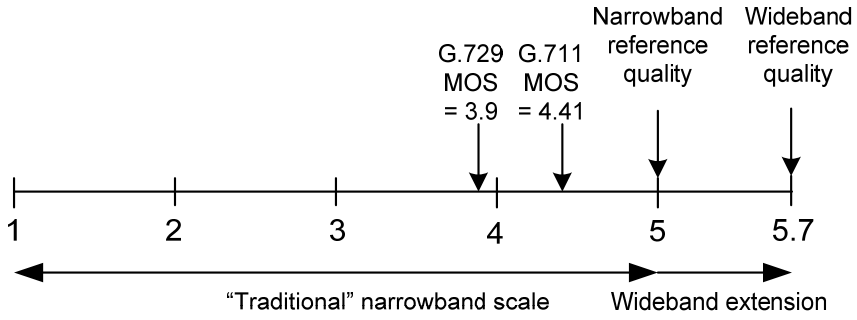


Figure 4-8: The extended MOS scale (2)

This conclusion is based entirely on the results of the France Telecom experiment, and no other assumptions. So, the traditional narrowband MOS scale remains the same: if the quality of some narrowband signal was, for example, 3.9, the quality of this signal on the extended 1-to-5.7 scale remains 3.9. If the quality of a wideband signal on the 1-to-5 scale was 3.9 (with respect to the direct wideband quality), the quality of this signal has to be converted to the extended scale using Equation 4.1.

This sub-section has defined the upper bound of the new scale, which allows comparison of the quality of the wideband and narrowband references. Now, the scale from 1 to 5.7 can be used, not only to measure both narrowband and wideband speech quality, but also to compare qualities directly. Here are several key points on this extended scale:

- MOS = 5.0 – the quality of direct analog narrowband speech
- MOS = 5.7 – the quality of direct analog wideband speech
- MOS = 4.41 – the maximum quality score, which can be achieved by narrowband codecs (the G.711-NB codec)
- MOS = 5.46 – the maximum quality score, which can be achieved by wideband codecs. The quality of the G.711-wideband corresponds to 4.8 on the 1-to-5 scale; the conversion Equation 4.1 is used to get this value on the extended scale.

Although it may look like that the difference in quality is not too big (5.7 versus 5.0), the difference between achievable narrowband and wideband qualities is significant. The maximum achievable narrowband quality has MOS equal to 4.41; the maximum achievable wideband quality has MOS equal to 5.46 (the difference is more than 1 MOS unit). The range of achievable toll-grade MOS scores has increased almost 3.5 times (from 4-to-4.4 interval to 4-to-5.46 interval). The range of achievable and relatively good MOS scores (3.6 to 4.4) has increased almost 2.5 times (3.6-to-5.46 interval) (see Figure 4.9).

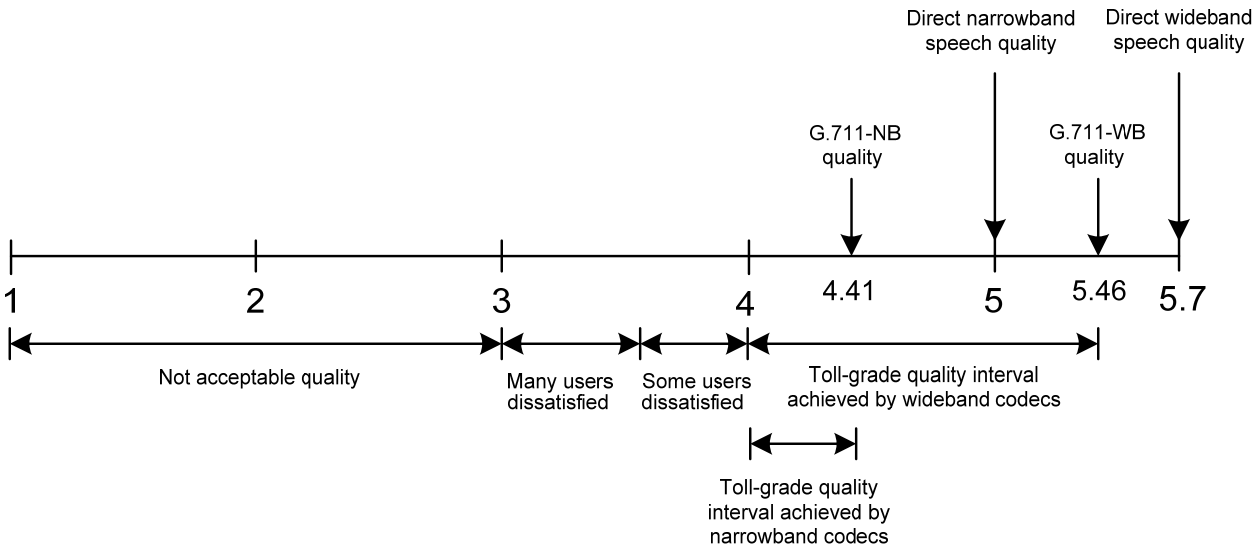


Figure 4-9: The extended MOS scale (3)

4.2.2 The R-factor scale for the wideband E-model

The traditional narrowband E-model uses the R-scale, which varies from 0 to 100, corresponding to MOS values from 1.0 to 4.5 (Equation 2.5 and Figure 2.4). The maximum achievable narrowband quality in this case is 4.41 (the G.711-NB codec, R = 93.2). The best wideband codec (the G.711-WB) has MOS equal to 4.8 on the 1-to-5 MOS scale (5.0 on this scale corresponds to a direct wideband speech quality) or 5.46 on the extended scale (see the previous section). We assume that the new model will have its extended range of R-values corresponding to MOS values from 1 to 5.5. The upper bound of this extended R-scale can be chosen arbitrarily (for example, it may correspond to MOS = 5.7 or even to MOS=6.0), but MOS = 5.5 is chosen as the upper bound, similar to the narrowband E-model (a little higher than the best wideband codec quality). So, the first step is to build a model to estimate the range of R values in the case of wideband communications.

What value of R corresponds to MOS = 5.5 on the extended scale? What is the value of R_0, which is the maximum achievable quality of wideband codecs on this scale? The answers to these questions are not evident for multiple reasons:

- The relationship between MOS and R-factor for narrowband codecs is not linear (see Equation 2.5 and Figure 2.4). So, the R-scale cannot be extended proportionally to the MOS scale. Increasing the MOS scale by 30% (from [1, 4.5] to [1, 5.5]) does not necessary imply that the R-scale will also increase by 30%, and will become [0, 130].
- It cannot be assumed that the relationship between the R-factor and a wideband speech quality measured on 1-to-5 MOS scale is the same as in the case of narrowband codecs (Equation 2.5). The reason is that the maximum achievable MOS for narrowband codecs is 4.5 (R = 100) compared to MOS = 4.8 for wideband codecs.
- If the R-scale is extended, it is not clear how to extrapolate MOS scores.
- There are multiple ways to choose the R-scale. Depending on a chosen scale, mapping functions between the R-scale and MOS will be different.

This dissertation proposes two variants of the R-scale for the wideband E-model. Either can be used in practice depending on the assumptions choosing the maximum R-factor. The first model assumes that the same 0-to-100 R-scale can be used for the wideband E-model. But this scale is different from the R-scale in the traditional narrowband E-model because the narrowband E-model has maximum MOS is 4.5 (R=100); in the wideband E-model the maximum MOS will be 5.5 (but R still will be equal to 100). Using the 0-to-100 R-scale is very beneficial not only because this scale is convenient, but also because it provides a metric for a statistical estimation of speech quality, similar to the narrowband E-model (see Equations 2.6, 2.7; Figure 2.5 and Table 2.2). The second model uses the extended R-scale from 0 to X, where X will be defined below. This subsection will explain the main principles to extrapolate the narrowband model and also use it for wideband speech.

First, it is necessary to define requirements to the wideband E-model:

- The new model has to be valid both for narrowband and wideband telephony, which means that the range of MOS scores has to be extended from [1, 4.5] to [1, 5.5]. A new function has to be proposed to establish the mapping between the new MOS scale and the chosen R-scale.

- The R-scale in the narrowband E-model provides quantitative and qualitative metrics for statistical estimation of speech quality. For example, when R exceeds 90 points, user satisfaction level is "Very satisfied" and more than 98% of users consider this quality level as good. If the R-factor is between 80 and 90, user satisfaction level is "Satisfied" and more than 90% of people consider this quality as good. The new model should have similar properties.
- The new model has to be consistent with the narrowband E-model. The extended MOS scale [1, 5.5] contains the range of values for narrowband codecs [1, 4.41]. So, R-scores corresponding to MOS values in these range has to be equal the R-score from the narrowband E-model (if we decide to extend the R-scale above 100) or an approach has be defined to convert the "new" R-factor (computed based on the new E-model) to the "old" R-factor (computed using the traditional narrowband E-model).

The first variant assumes that the same 0-to-100 R-scale can be used in the wideband E-model. The first graph in Figure 4.10 shows the R-to-MOS function for the traditional narrowband E-model (defined by Equation 2.5). The score 4.5 on this scale corresponds to R = 100. How will the shape of this curve change if the MOS scale is extended up to [1, 5.5] (the narrowband direct quality reference is replaced by the wideband reference)?

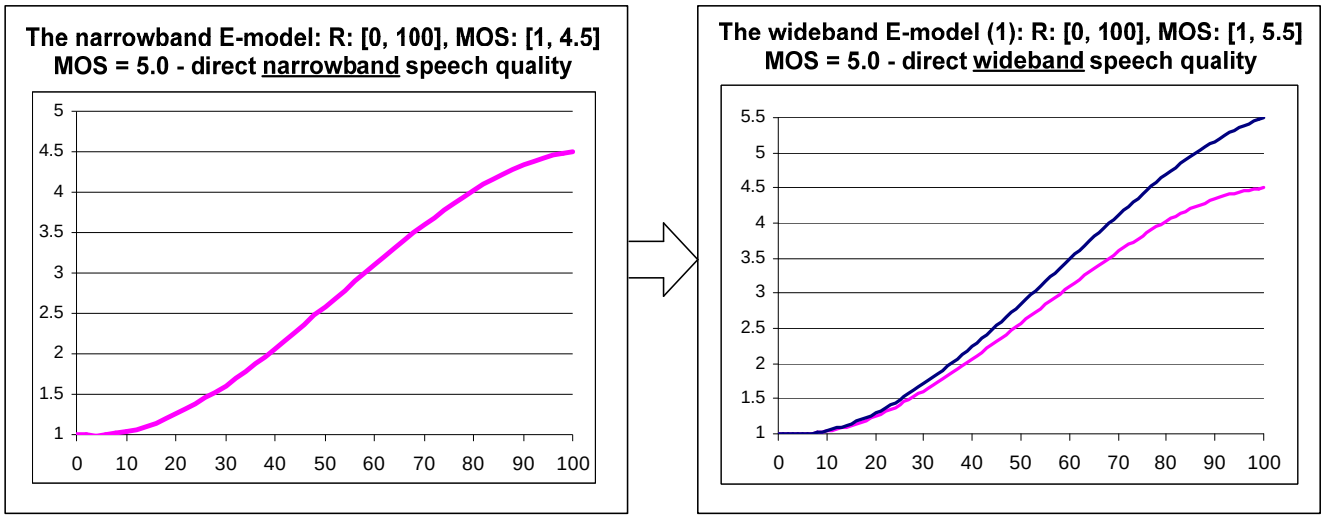


Figure 4-10: Methodology of the E-model extension

This curve will change. For example, in the traditional narrowband E-model an R-factor of 60 corresponds to MOS = 3.1. When the narrowband reference (MOS = 5.0) is substituted by the wideband reference (also MOS = 5.0), MOS = 3.1 will become MOS_1= (3.1+0.175)/1.175 = 2.79 (see Section 4.2.1). From Equation 2.5, if MOS = 2.79, then R = 54. So, if the narrowband reference is replaced by the wideband reference, this operation has to be applied to all values of R. The result is demonstrated on Figure 4.11.

This new curve, which defines qualities with respect to the wideband reference, is defined only on the [0, 77] interval. The reason is that the maximum quality of narrowband codecs is 4.41 (R=93.2). This quality with respect to the wideband reference is (4.41+0.175)/1.175=3.90 (R=77).

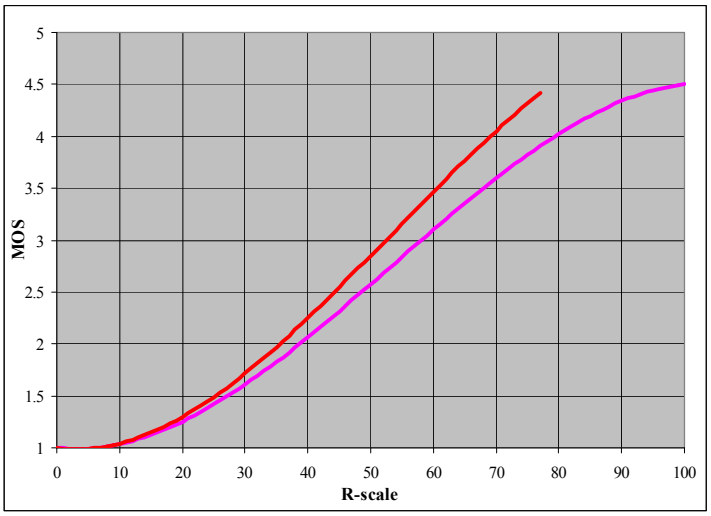


Figure 4-11: R-to-MOS conversion in the wideband E-model (1)

But, wideband telephony has to be defined over a [0,100] R-scale and has to achieve MOS = 5.5. How should the red curve on Figure 4.11 be extended? It is proposed to look for a new mapping function in the form similar to the Equation 2.5 from the narrowband E-model:

$$MOS_{NEW}=1+ \boldsymbol{\alpha}\cdot R + R(R-\boldsymbol{\beta})(100-R)\cdot\boldsymbol{\gamma} \qquad (4.2)$$

Now **α**, **β** and **γ** must be found. It is known that if R=100, then MOS=5.5. So, **α**=0.045. In the narrowband E-model **β** = 60**.** At an R-factor of 60, 50% of subscribers regard the call quality as "good". How is this factor going to change in the case of wideband telephony?

With better quality as a reference, user expectations about the speech quality level become higher; so **β** will also be higher. The linear equation, which establishes the mapping between the traditional [1, 4.5] MOS scale from the narrowband E-model and [1, 5.5] MOS scale from our wideband E-model is Y=1.3·X-0.3. In other words, if a narrowband MOS has a value X, the corresponding wideband MOS ("corresponding" means that the same percent of people will consider the service as good), should be calculated according to this equation. The middle (50% satisfaction) of the GoB curve has R-factor equal to 60 or 3.1 MOS (see Section 2.2.2). On the new scale, MOS=3.1 will be equivalent to MOS = 3.7, which corresponds to R = 72.3 (Equation 2.5). So, **β** = 72.3; the GoB curve shifts right by 12 units in the case of wideband VoIP (Figure 4.12) and is defined by Equation 4.3:

$$GoB = 100E\left(\frac{R-72.3}{11}\right)\% \tag{4.3}$$

E(x) is the Gaussian Error function defined by Equation 2.6. E(R=100) = 2.5 as in the narrowband E-model.

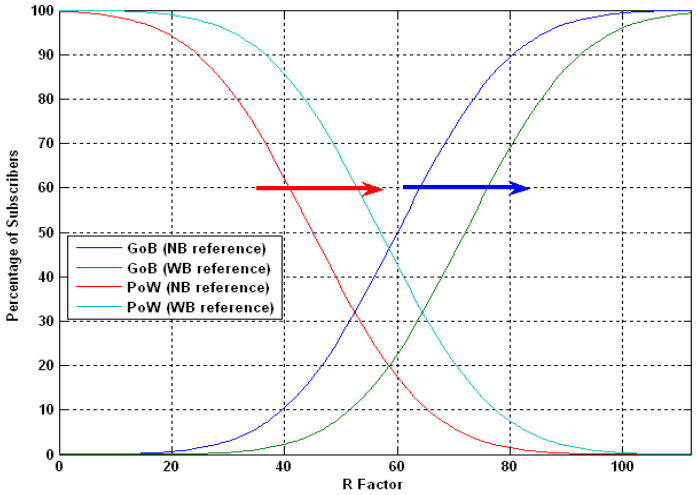


Figure 4-12: GoB curve for the NB and WB E-models (change)

γ was calculated numerically. The criterion used is that the new curve defined by Equation 4.2 has to coincide with the red curve on Figure 4.11. That is, **γ** must be found so that the

Mean Square Error between the red curve and the new curve is minimal. Modeling gives γ = 7.2. So,

$$MOS_{NEW}=1+ 0.045R + R(R - 72.3)(100 - R) \cdot 7.2 \cdot 10^{-6} \quad (4.4)$$

The graph of this function is shown on Figure 4.13.

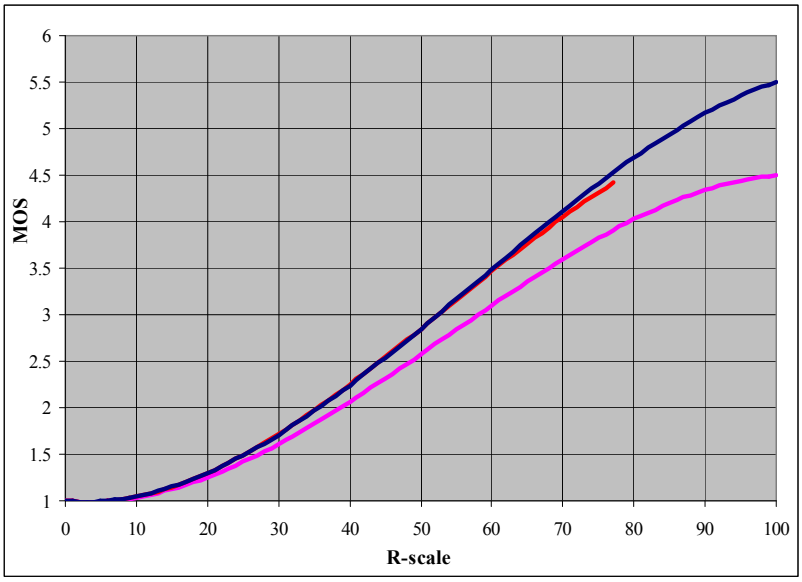


Figure 4-13: R-to-MOS conversion in the wideband E-model (2)

Intuitively, it would be more convenient to extend, not only the MOS scale, but also the R-scale. In other words, the narrowband curve can be extrapolated, extending both R and MOS scales to describe an increase in quality due to an increase of the signal frequency range. As Figure 4.1 demonstrates, different extrapolation techniques can be used, although they are not justified. It is not clear which extrapolation has to be used, because no rules or criteria are defined for the traditional 100 points R-scale extension. Which approach is more appropriate? A different criterion is proposed, based on the user satisfaction metric, discussed above. Starting with the linear MOS scale from 1 to 5.5, assume that the traditional R scale will be extended, but instead of [0, 100], it will become [0, X], X > 100. The transformation law is described by the equation: $R_1 = (X/100) \cdot R$. But what is the value of X? The middle (50% satisfaction) of the GoB curve has R-factor equal to 60. On the new

scale, it should translate to R=72.3 (Equation 2.5). So, X = 72.3 / 60 = 1.205 and the maximum value of the extended R-scale is 120.5.

An equation is needed to establish a relationship between the extended R and MOS scales. Again, this function has the form:

$$MOS_{NEW}=1+\boldsymbol{\alpha}\cdot R + R(R-72.3)(120.5-R)\cdot\boldsymbol{\gamma}$$

$$\boldsymbol{\alpha} = (5.5 - 1.0) / 120.5 = 0.037$$

$$\boldsymbol{\gamma} = 7.2 / 1.205^3 = 4.11$$

So, the relationship between the extended MOS and R-scales is defined by Equation 4.5 and is shown in Figure 4.14.

$$MOS=1+0.037R+R(R-72.3)(120.5-R)\cdot 4.11\cdot 10^{-6} \qquad (4.5)$$

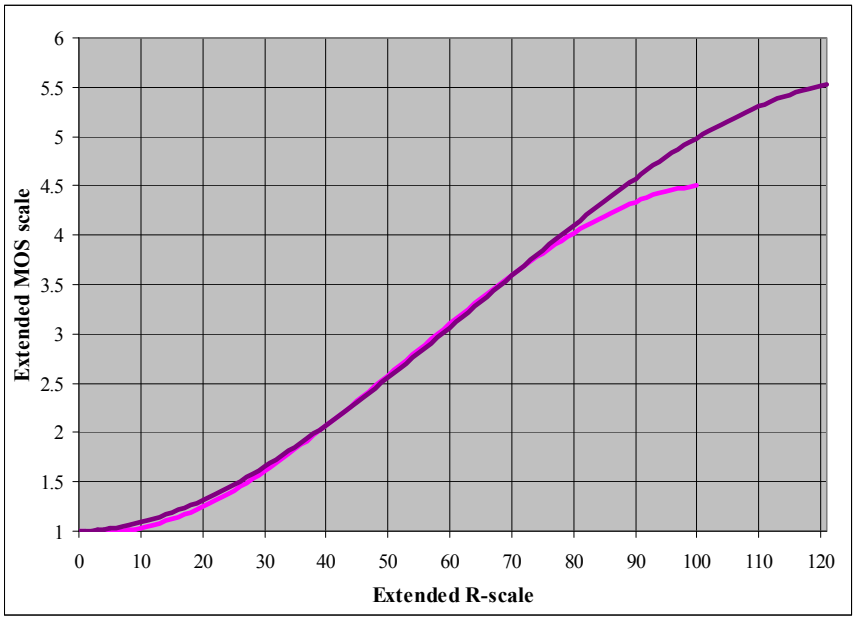


Figure 4-14: R-to-MOS conversion in the wideband E-model (3)

As it was mentioned in Section 4.2.1, the maximum achievable wideband quality on the extended scale is 5.46. The value of R in the wideband model that corresponds to this value of MOS is 118.2 (from Equation 4.5). Now, the narrowband and wideband computational

models can be "merged", using the equation $R = R_0 - I_d - I_{e\text{-}eff}$ with R_0=118.2 for both narrowband and wideband VoIP. This value is close to some estimates mentioned in the Section 4.1, but the methodology to explain these values has been provided.

4.2.3 Investigating equipment and network impairments for the wideband E-model

The previous two Sections proposed an approach to directly compare qualities of wideband and narrowband VoIP communications. They concluded that the extended R-scale can be used to describe the effect of increasing quality due to increase of bandwidth and sampling rate, and they defined the upper bound of this scale. This section extends work on the wideband E-model. The traditional narrowband E-model, expressed by Equation 2.1, includes three voice quality impairment factors: the delay impairment I_d, the equipment impairment I_e and the packet loss robustness factor *Bpl*. How do these parameters change in the case of wideband telephony?

1) Delay. The impairment of delay was described in Section 2.2.2. This type of voice quality impairment depends only on the absolute delay between a sender and a receiver. The effect of delay impairment in the case of wideband VoIP telephony is assumed to be the same as in the case of the narrowband VoIP. This assumption is confirmed by the fact that the quality of a wideband signal with a significant delay is still better then the quality of a narrowband signal with the same delay, but this difference in quality is described by the difference in the R_0 values (maximum achievable quality levels).

2) Effective equipment impairment $I_{e\text{-}eff}$. The description of this factor was also provided in Section 2.2.2. It is defined by Equation 4.6:

$$I_{e-eff} = I_e + (95 - I_e)\frac{Ppl}{Ppl + Bpl} \tag{4.6}$$

where $I_{e\text{-}eff}$ is the effective equipment impairment; I_e is the equipment impairment; Bpl is the packet loss robustness (the effectiveness of codec-specific packet-loss concealment

mechanism), and Ppl is the packet loss rate. It is necessary to investigate how this factor will change in the wideband E-model.

If there is no packet loss in the network (Ppl = 0), the E-model is expressed by Equation 4.7:

$$R = R_0 - I_d - I_e \tag{4.7}$$

The traditional narrowband E-model has $R_0 = 93.2$. The previous Section concluded that $R_0 = 118.2$ in the case of wideband VoIP. The narrowband and wideband computational models can be "merged" to the equation $R = R_0 - I_d - I_e$ with $R_0 = 118.2$ for both narrowband and wideband cases. But the value of I_e from the narrowband E-model (call it $I_{e\text{-}NB}$) will change. Assuming that I_d does not change, the value of $I_{e\text{-}NB}$ on the new extended R-scale will also increase by 118.2 - 93.2 = 25 units:

$$I_{e\text{-}NB} = I_{e\text{-}WB} + 25 \tag{4.8}$$

For example, the equipment impairment factor of the narrowband G.711 codec is 0 on the R-scale from 0 to 100. On the scale from 0 to 120.5, the same codec will have an equipment impairment factor equal to 25. The wideband G.711 codec will have I_e equal to 0 on the extended R-scale.

3) Packet loss robustness factor. The E-model includes one more codec parameter called the packet loss robustness, Bpl, which characterizes the "resistance" of codecs to packet loss (the effectiveness of a packet loss concealment algorithm). This is a codec-specific number, defined in Appendix I/G.113 [4]. Speech codecs with more efficient packet-loss concealment mechanisms have higher values of the Bpl factor. The ITU-T proposed a table with approximate Bpl values for several codecs (Table 2.1). But for wideband speech, using the extended R-scale and Equation 4.8, the effective equipment impairment factor in the wideband E-model will be:

$$I_{e-eff} = I_e + (120 - I_e)\frac{Ppl}{Ppl + Bpl} \tag{4.9}$$

It would be interesting to understand how the packet loss robustness factor is different for narrowband and wideband speech. It is intuitively clear that changing the sampling rate should not significantly affect this factor for the same type of codec because a wideband packet has the same time-duration as a narrowband packet, but a larger size in bytes. Loss-concealment algorithms compensate a packet duration gap, so packet size in bytes should not be very important. But, probably there is some effect because, for example, in the case of wideband speech it would be more difficult to restore (approximate) an original signal more accurately (see Figure 4.15). But, it is assumed here that the accuracy of lost-packet restoration is lower than the accuracy of the packet-loss concealment algorithm.

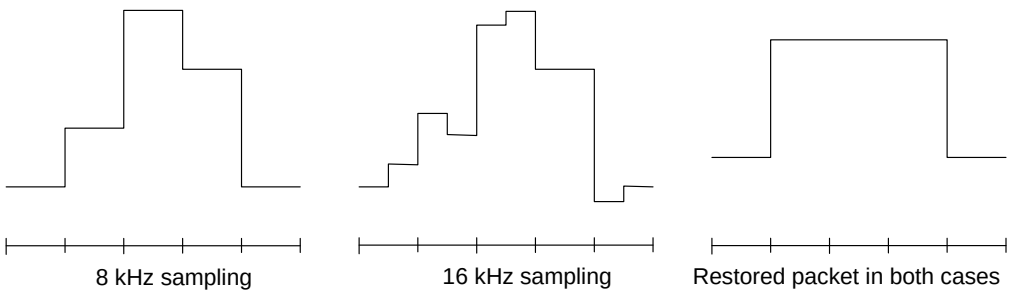


Figure 4-15: Packet loss concealment in the case of narrowband and wideband speech

An experiment with real speech sentences was performed to investigate the difference between packet-loss robustness of narrowband (Bpl_{NB}) and wideband speech codecs (Bpl_{WB}), to confirm the hypothesis for different kinds of experiments related to VoIP quality measurement [11]. This database contains "phonetically balanced" sentences, which include sounds (or sequences of sounds) that are difficult to understand over a telephone line. This experiment used 20 samples from the ITU database, where each sample is 8 seconds long and contains two short sentences consisting of approximately 40-60% of speech. An example of such a sentence is shown in Figure 4.16.

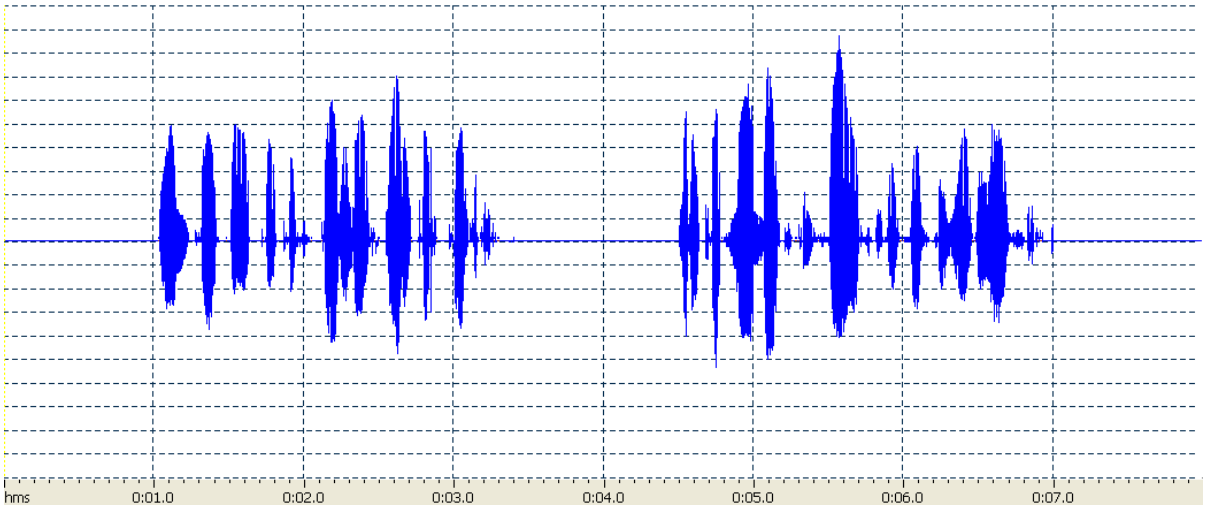


Figure 4-16: Example of a sentence from the ITU-T database

The speech samples were encoded using a codec, packetized, and sent through a virtual network with constant delay (jitter is not important in this section). The open-source software, called *Qofis*, written by Christian Hoene [69], was used to encode the sentences. This software was written to demonstrate different jitter buffer playout schemes and part of this C++ code was used to encode and decode sentences. An open-source ITU ANSI PESQ code was used to measure the resulting quality. PESQ can measure narrowband and wideband quality on 1.0-to-4.5 scale (different algorithms are used). The detailed scheme of the experiment is illustrated in Figure 4.17.

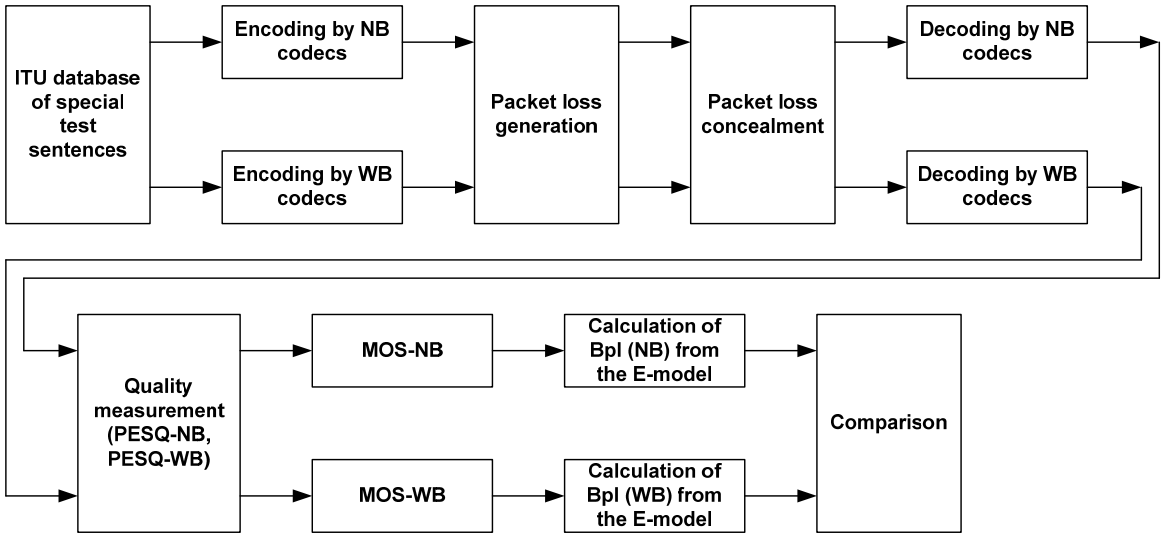


Figure 4-17: Experiment setup to compare Bpl values for narrowband and wideband speech samples

The experiment included several steps:

(1) Special samples were taken from the database (the experiment used 20 sentences). The original sentences were wideband 8-kHz samples, encoded by 16-bit PCM. Narrowband copies of these samples (4-kHz, 16-bit PCM) were created. Then, the narrowband sentences were encoded by a narrowband codec, and wideband sentences were encoded by wideband codec. The G.711 codec (narrowband and wideband) with packet-loss concealment was used in the experiment. Just one codec was used because the wideband versions of most other available codecs (for example, the G.723.1, the G. 729) are not developed yet or the open-source versions of these codecs are not available. Furthermore, an experiment with one codec should be enough to investigate the hypothesis.

(2) The encoded speech samples were then impaired by a random loss of packets. The loss pattern for the same narrowband and wideband samples was the same. Different loss rates (0-10%) were simulated. Because the positions of lost frames is also important and MOS scores of different speech samples can be much different for the same loss rate, 10

different loss patterns were used. The average MOS was calculated for a given sample and a given packet loss rate.

(3) The resulting speech was decoded with the same codec, and using a packet-loss concealment algorithm.

(4) To measure the quality of the decoded sentences, a PESQ open-source code was used. PESQ measures the speech quality of narrowband and wideband codecs on the 1-to-5 MOS scale, so, the results cannot be compared directly. MOS scores for narrowband codecs remain unchanged; MOS scores for wideband codecs are converted to MOS on the extended scale proposed in Section 4.2.1. Then, the narrowband and wideband MOS scores are converted to the extended R-scale and Equation 4.10 is used to calculate Bpl values:

$$Bpl = \left(\frac{120 - I_e}{118.2 - R - I_e} - 1 \right) Ppl \tag{4.10}$$

The tables below present the experimental results. Table 4.2 contains PESQ_MOS scores (MOS scores obtained using PESQ mechanism) for the narrowband G.711 codec and for the wideband G.711 codec with packet-loss concealment. Note that PESQ uses the same scale from 1 to 4.5 to estimate voice-quality of narrowband and wideband telephony. So, it is possible to have a PESQ-scaled quality of a wideband speech sample measured on this scale to be lower then the quality of a narrowband speech sample. The wideband PESQ_MOS scores should be converted to the extended scale [1.0, 5.7] proposed in the Section 4.2 in order to compare the qualities directly. Note that packet loss robustness is a codec specific value and does not depend on packet loss (at least theoretically). The theoretical value of the Bpl factor recommended by ITU for the narrowband G.711 codecs with packet-loss concealment is 25.1

Table 4-2: Experiment results for the G.711 codec with packet loss concealment

G.711-NB codec					
Packet loss rate	**1%**	**2%**	**3%**	**4%**	**5%**
MOS (mean)	4.21	4.01	3.84	3.68	3.54
90% confidence interval for the mean MOS	0.07	0.08	0.10	0.10	0.11
R-factor (mean)	85.4	79.6	75.4	71.8	68.8
R-factor (interval)	(81.93, 88.12)	(77.34, 82.79)	(73.13, 79.37)	(69.63, 75.03)	(66.52, 71.92)
Bpl-NB (mean)	11.14	12.00	13.03	13.75	14.46
Bpl-NB (interval)	(8.25, 14.43)	(9.98, 14.69)	(11.19, 17.60)	(12.12, 16.82)	(12.80, 16.32)

G.711-WB codec					
Packet loss rate	**1%**	**2%**	**3%**	**4%**	**5%**
MOS (mean)	4.18	4.02	3.84	3.68	3.58
90% confidence interval for the mean MOS	0.10	0.11	0.12	0.12	0.10
MOS (mean; extended scale)	5.13	4.93	4.69	4.48	4.35
90% confidence interval for the mean MOS (extended scale)	0.10	0.11	0.12	0.12	0.10
R-factor (mean)	104.2	98.8	93.1	88.2	85.5
R-factor (interval)	(102.15 108.11)	(94.50, 102.79)	(89.50, 96.80)	(85.02, 91.56)	(82.12, 81.54)
Bpl-WB (mean)	8.59	10.37	11.34	12	11.22
Bpl-WB (interval)	(6.35, 10.76)	(8.18, 13.58)	(9.54, 13.82)	(10.46, 13.98)	(9.56, 12.64)

$Bpl_{NB} - Bpl_{WB}$	2.55	1.63	1.69	1.75	3.24

Now, Bpl_{NB} can be compared to Bpl_{WB}, along with their confidence intervals. The results of the experiment show that the values of packet-loss robustness for narrowband and wideband samples are close to each other. There is some decrease of Bpl_{WB} with respect to $Bpl_{NB,}$ but this decrease is not very significant: the confidence intervals for Bpl-NB (mean) and Bpl-WB (mean) intersect, so it is concluded that the difference between these values is not statistically different from zero, and the same Bpl values for narrowband speech can be used for wideband speech. This result is intuitively clear: if the sampling rate is doubled, the size of packet measured in bytes also doubles. But, the packet size measured in milliseconds (the time gap) remains the same.

Observe the difference between these experimental results and the theoretical value for narrowband codecs: the theoretical value of Bpl for the G.711 codec is around 25 but the experiment's Bpl are lower and are not constant. The possible reasons for this result include:

(1) The representation of packet loss in the E-model is not very accurate. The accuracy of the E-model is about 80% and many people agree that high packet loss rates are not modeled well enough by the E-model.

(2) The PESQ measurement may be inaccurate: while the computational E-model and PESQ are not perfect tools to measure speech quality, (1) they have a high correlation coefficient which provides accurate results under the same testing conditions and (2) since the absolute value of the Bpl parameters is less important than their difference, any absolute errors in the algorithms are effectively canceled.

(3) More experiments are required. The work here was limited to 20 sentences and 10 loss patterns per given packet-loss rate per sentence. Greater accuracy requires using more sentences and loss patterns.

(4) The conditions of the experiments may be different from those used by ITU (the ratio of speech-to-silence frames can be different, etc)

(5) The ratio of speech-packets to silence-packets in these experiments may be different from the ratio used in the ITU experiments.

4.3 Computational quality model for arbitrary VoIP flow parameters (Extension of the E-model for non-standard codecs)

The traditional narrowband E-model contains several parameters defined only for a very limited number of codecs. For example, Ie and Bpl factors are defined for the G.711 codec with no compression and 10 ms packet size; for the G.729 codec with 8:1 compression and 10 ms packet size; and several others. This dissertation, assumes that we are not limited to existing codecs and that a "virtual" codec can have any settings in terms for sampling rate, packet size and compression ratio (but, with existing bit encoding schemes). What happens to the E-model parameters with, for example, a codec with 4:1 compression and 30 ms packet size is used? The answer requires a computational model for this "virtual" codec.

First, it is necessary choose one (or several) encoding schemes for the "virtual" codec. Two types of codecs are proposed for use: the G.711-type codecs (without compression) and the G.729-type codecs, which use CS-ACELP encoding. This encoding scheme is the most efficient based on bandwidth-to-quality ratio; although it is rather complex. According to subjective measurements, the quality of this codec has MOS = 3.9 under ideal conditions. It compresses 8-kHz 16-bit narrowband speech down to an 8 kbps stream. ACELP creates models of the human voice then predicts what the next sound will be. It encodes the difference between the actual sound and the predicted sound, and the difference is transmitted to the receiving end. Since the other end of the call also runs ACELP, the calculation of the difference allows for an acceptable recreation of human voice at the receiver. ACELP techniques create a less accurate representation of women's and children's voices, which are generally higher in pitch than male voices. With improvements in digital signal processors, this problem will probably be resolved in the near future. It is assumed here that a voice stream can be formed using ACELP or the G.711-type encoding scheme with any packet size, sampling rate and compression ratio.

This section proceeds to investigate how the E-model's effective equipment impairment factor $I_{e\text{-}eff}$ changes with changes in the characteristics of VoIP: packet size, sampling rate (signal frequency range), and compression. First, the situation without packet loss is analyzed. In the absence of packet loss, $I_{e\text{-}eff} = I_e$ (see the E-model equation).

I. No packet loss

An equation is needed for I_e as a function of packet duration, sampling rate, compression. The initial analysis examines the two-dimensional space that includes sampling rate (signal frequency range) and compression. The horizontal scale is a compression ratio: on the left side is uncompressed voice (narrowband and wideband), and on the right side is G.729-encoded speech. The vertical scale is sampling rate. So, the "corner" values of the rectangle in Figure 4.19 are 4 kHz and 8 kHz on the "signal frequency range" side; 1:1 and 8:1 on the "compression" range. I_e for these "corner" values is known: (a) non-compressed 8-kHz signal has $I_e = 25$ (the narrowband G.711 codec); (b) non-compressed 16-kHz signal has $I_e = 0$ (the wideband G.711 codec); (c) 8-kHz signal, 8:1 compression has $I_e = 10 + 25 = 35$ (the narrowband G.729 codec); (d) 16-kHz signal, 8:1 compression has $I_e = 10$. All values are provided for 10 ms voice packets. The wideband G.729 codec, also called G.729.1, is not available yet, but it should be available in the near future.

In the absence of subjective testing, the linear approximation is used to calculate the equipment impairment factor for any intermediate signal frequency range and compression. The approximation demonstrates that increasing the signal frequency range by 1 kHz decreases the equipment impairment factor Ie (improves quality) by 6.25 units on the R-scale (approximately 0.3 MOS).

The assumption about linear dependency between Ie and compression ratio is not so evident but it could be changed in the future if further evidence suggests a different relationship. In the case of linear approximation, increasing the compression decreases the speech quality by approximately 1.5 units if the G.729-type compression is used.

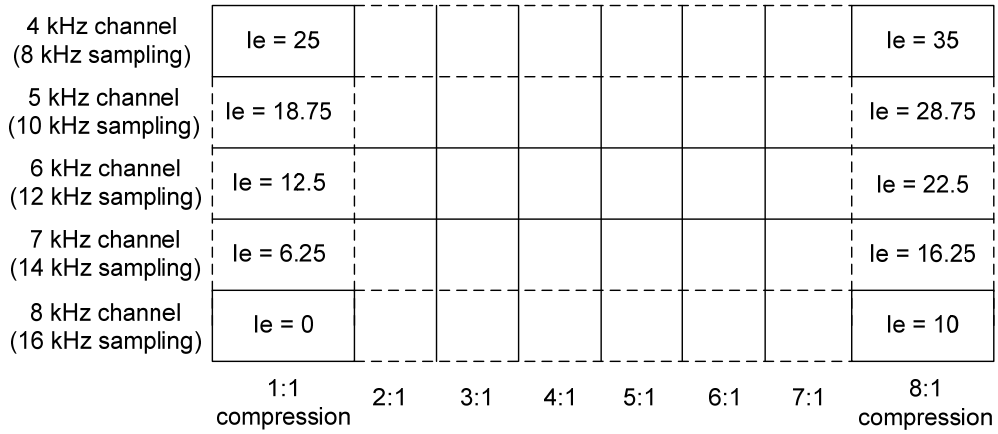


Figure 4-18: Equipment impairment as a function of signal frequency range and compression (1)

So, for X-kHz of signal frequency range and Y-to-1 compression, the equipment impairment factor I_e of this signal is calculated as:

$$I_e = 6.25(8 - X) + \frac{3}{2}(Y - 1) \tag{4.11}$$

The result is shown in Figure 4.19.

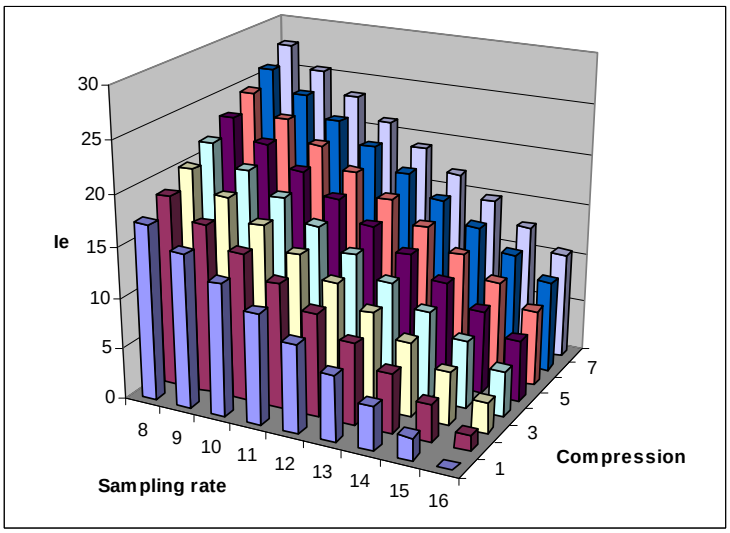


Figure 4-19: Equipment impairment as a function of signal frequency range and compression (2)

How does the equipment impairment factor depend on packet size (the third dimension)?

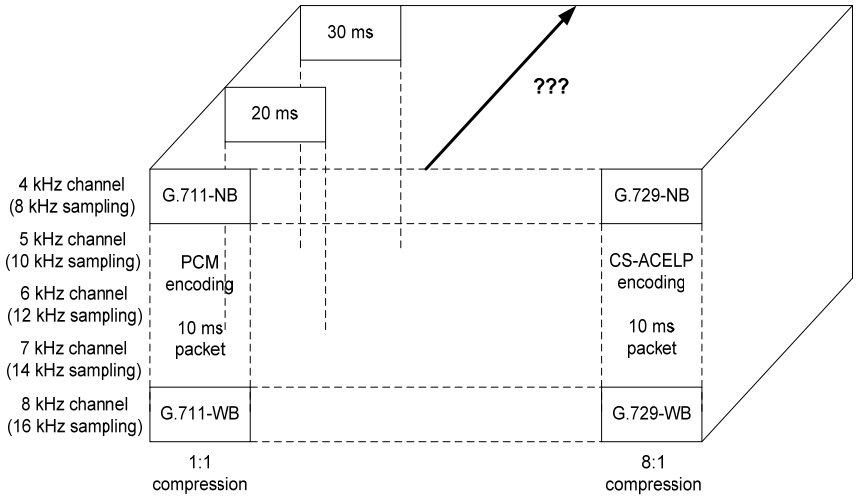


Figure 4-20: Equipment impairment as a function of packet size

The answer is not evident. Increasing the packet size leads to increased delay. If the delay is small (less then 150 ms one-way), the increased packet size does not decrease speech quality (see the function of delay impairment from the E-model). If the delay is larger than 150 ms, increased packet duration leads to decreased quality. So, if there is no packet loss in the network, the effect of increased packet size on delay is taken into account by the I_d factor. However, the situation is more complex when packet loss exists in the network.

II. In the presence of packet loss:

A change of sampling rate and / or compression leads to a change of the packet size in bytes. This change should not affect the packet loss robustness factor for the same type of codec because the packet duration is not changed. Loss concealment algorithms compensate the packet duration gap, so the packet size in bytes (sampling rate) is not very important. In the case of fixed packet-duration, this question was investigated and demonstrated in the previous section. Is it possible to propose an equation to model the effectiveness of packet loss concealment mechanisms as a function of packet duration?

The equipment impairment factor I_e does not take packet loss into account. Packet loss is included in the so-called effective equipment impairment factor $I_{e\text{-}eff}$ (see Section 4.2.3):

$$I_{e-eff} = I_e + (120 - I_e)\frac{Ppl}{Ppl + Bpl} \tag{4.12}$$

where Bpl is packet loss robustness and Ppl is packet loss rate (%).

Increasing the packet size in milliseconds (several smaller frames in one packet) leads to a greater number of bits lost in a sequence. This should result in decreased packet loss robustness because it is more difficult to restore one "long" packet than one "short" packet. The question of how the packet loss robustness factor Bpl depends on packet duration must be investigated.

Let a 10-ms VoIP packet be assumed, with packet loss robustness factor Bpl_0. this packet is increased by N times up to 10N ms, the loss robustness of this packet would Bpl_1. The relationship between Bpl_0 and Bpl_1 is estimated.

An increase of packet size can be considered from two points of view:

(1) As a bursty loss of "small" 10 ms packets. ITU has a special extension of the E-model to describe the effect of a bursty packet loss (Section 2.2.2 and Equation 2.4). This equation provides acceptable values only in the case of low packet loss rates less than 2%.

$$I_{e-eff} = I_e + (120 - I_e)\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \tag{4.13}$$

where BurstR is burst duration, which equals the number of packets lost in a sequence. So, if the packet size is increased by N times, the Equation 4.13 would look like:

$$I_{e-eff} = I_e + (120 - I_e)\frac{Ppl}{\frac{Ppl}{N} + Bpl_0} \tag{4.14}$$

From another point of view, the increase of packet size may also be considered as a new codec with packet loss robustness Bpl_1:

$$I_{e-eff} = I_e + (120 - I_e)\frac{Ppl}{Ppl + Bpl_1} \tag{4.15}$$

If this assumption is valid, Equation 4.12 becomes

$$(120 - I_e)\frac{Ppl}{\frac{Ppl}{N} + Bpl_0} = (120 - I_e)\frac{Ppl}{Ppl + Bpl_1} \Rightarrow \frac{Ppl}{N} + Bpl_0 = Ppl + Bpl_1 \Rightarrow$$

$$Bpl_1 = Bpl_0 - \left(1 - \frac{1}{N}\right)Ppl \Rightarrow Bpl_1 - Bpl_0 = \left(\frac{1}{N} - 1\right)Ppl \tag{4.16}$$

This theoretical result is derived from the theoretical equations 4.13 and 4.14. From one point of view, it confirms the assumption that packet loss robustness decreases with an increase of packet size. From another point of view, although Equation 4.16 includes packet loss value Ppl, it was stated that the packet loss robustness factor should not depend on packet loss. This can be explained by the fact that Equation 4.12 describes the effect of packet loss on speech quality: (1) it is just a mathematical equation (model, approximation), which does not have any physical sense; (2) it includes the sum of Bpl + Ppl, which have different units of measurement (packet loss is measured in percent, packet loss robustness does not have any unit of measurement); and (3) the E-model does not represent accurately the effect of a significant packet loss.

The theory demonstrates that packet loss robustness decreases when packet duration increases. Here are some rough estimates derived from the theoretical equations: for the narrowband G.711 codec, the packet loss robustness is Bpl_0=25.1 (table value), with packet size of 10 ms. the packet size is increased up to 20 ms, with low packet loss rate (for example, 2%), the packet loss robustness factor for the new packet size will decrease by (1-0.5) · 2=1 unit and will equal 24.1. The difference is very small. If the packet size is increased to 30 ms, the packet loss robustness factor for the new packet size will decrease by

(1-0.33) · 2=1.3 units and will equal 23.8. Again, the difference is small compared to the absolute value of Bpl.

This difference between Bpl_1 and Bpl_0 will be noticeable for codecs without packet loss concealment. But these codecs are not used in practice because they provide a low speech quality in the case of packet loss. The difference also becomes noticeable when a packet loss rate increases.

4.4 Conclusion and Future Work

This chapter has proposed a single scale to measure and directly compare the quality of 8-kHz wideband and narrowband telephony. The traditional 1-to-5 MOS scale can be extended up to 5.7 MOS. Based on this result, the R-scale from the E-model is also extended and the maximum quality of wideband codecs was found with respect to the maximum quality of narrowband codecs. This chapter analyzed how multiple parameters from the "VoIP-version" of the model will change in the wideband case and it provided a simple linear model of how the E-model parameters will change in the case of "non-standard" codecs.

Similar work can be extended to higher frequency ranges of signals (for example, up to 12 kHz). This chapter analyzed the simplified version of the E-model. The original model includes many other parameters (for example, effect of echo, noise, etc) and it is necessary to investigate how all these parameters will change in the case of wideband communications.

Chapter 5

Impact of Variable Speech Encoding on Quality of Voice-over-IP Communications

This Chapter investigates how varying of speech encoding parameters (packet size, compression, and signal frequency range) affects the quality of VoIP communications. Sections 5.1 and 5.2 provide a detailed description of goals, methodologies, simulation scheme, assumptions and scenarios analyzed in the project, and present an overview of the previous work in the area of sender-based adaptive speech quality management. Sections 5.3 – 5.6 study the impact of multiple parameters on VoIP quality under different scenarios and answer several other related questions.

5.1 Problem Statement and Related Research

5.1.1 Problem Statement

A packet-switched network does not provide reliable transport of real-time data: it does not guarantee available bandwidth, delay and loss bounds, which are critical for real-time voice traffic. Although TCP has a congestion control mechanism that is used in the event of packet loss, voice traffic uses the non-reliable UDP transport-layer protocol, and cannot react to changing network conditions. This may cause a significant degradation of communication quality.

The beginning of Chapter 1 discussed multiple network-based QoS mechanisms. This Chapter investigates an alternative approach: the concept of sender-based adaptive speech quality management. To manage speech quality during a communication session by changing speech encoding parameters, two main questions arise (Section 1.2): (1) How to estimate (or monitor) the quality of a communication session objectively and in real-time; (2) How to manage speech quality dynamically depending on specified criteria. The first

question was investigated in the previous Chapter, which proposed extending the computational E-model to wideband scenarios and "non-standard" codecs. This approach can be used to estimate the quality of codecs with any encoding parameters. The second question can be split into several "smaller" parts:

(a) How do the end-user parameters and variations of the speech encoding parameters affect VoIP quality under different network conditions? This must be known in order to choose the "correct" set of parameters that match a given network state, and to take a "correct action" to improve the average quality of a communication session.

(b) When and how a decision can be made to change the encoding parameters at the proper moment of time (including network state detection and quantitative criteria depending on speech quality management objectives)?

(c) How to make the system "adaptive" (how to establish the interaction between a sender and a receiver, how often to do it, and how to demonstrate effectiveness and stability of the system)?

The goal of this Chapter is to investigate the effect of multiple speech encoding parameters (packet size, compression / encoding, narrowband / wideband signal frequency range) on speech quality under different network conditions (that is to answer Question (a)). The remaining Questions (b) and (c) will be answered in the next chapter.

The E-model discussed in Chapters 2 and 4 presents quality as:

$$\text{Quality Factor} = \text{Maximum Quality Factor} - \text{Function [delay]} - \text{Function [packet loss rate, packet loss robustness]} \quad (5.1)$$

This equation does not include codec parameters that can be managed by the end-users (packet size, compression, signal frequency range) and does not answer the question of how the variation of these parameters affects speech quality. Any change of packet size or compression can increase or decrease the channel capacity requirements per call; increasing IP-rate will lead to increased quality, but the probability of quality degradation due to potential congestion also increases. There are various ways to trade between bandwidth and speech quality, but at this time there is no distinct answer or even acceptable mathematical model on how this should be achieved dynamically to improve or optimize, for example, the

average quality of a communication session. To manage speech quality in real-time, the effect of different encoding parameters must be known under various scenarios. This Chapter investigates this question.

End-users and VoIP providers often have different speech quality management objectives. Individual users usually want to get a good (at least toll-grade) level of quality. With wider deployment of wideband technologies, their expectations will increase even more (as discussed in Chapter 4). In most cases, individual users do not care about channel consumption per call because they do not pay for it. They want to get good communication quality and the number of kilobits-per-second occupying in their channel is not too important. The tradeoff between capacity-per-call and speech quality becomes important when users have to pay an additional fee to get a higher (premium) quality. In the case of IP-to-IP communications they have to invest money to buy, for example, DSL or some other broadband connection instead of dial-up. Although now, an increasing number of people have high-speed connections, so access network capacity is not a restriction in most situations. For such users, no additional investment is required. But, to get "better than average" quality under other scenarios, users may have to pay more. Multiple papers, research projects, and marketing surveys have investigated user behavior in these situations and user willingness-to-pay for quality of communications (for example, [88]). The goal of the network providers is not only to propose a reasonable quality of services to their customers but also to increase the economic effectiveness of their business, by decreasing expenses on bandwidth resources. So, this Chapter analyzes several different scenarios and explores quality management approaches under each scenario.

This dissertation will investigate two different scenarios. The first scenario addresses narrowband and wideband codecs. Although several wideband codecs have been developed and they provide a significant improvement in quality, narrowband codecs are still widely used, for two reasons: the quality of IP-to-PSTN communications is limited by 4-kHz of PSTN bandwidth. So wideband codecs do not provide increased quality for calls that partially use the PSTN; also, although wideband codecs provide higher communications quality, they require more bandwidth and bandwidth consumption is still important in many situations. This scenario assumes that speech quality is the most important parameter. This means that the network provider: (1) does not have the goal of maximizing the number of

calls in its network; (2) does not decrease speech quality by using "worse" codecs internally (that is the best narrowband G.711 codec is used in the narrowband scenario). Under these assumptions, the following questions are investigated and answered:

- How does VoIP communication quality depend on the proportion of voice and data traffic in the network?
- What is the effect of packet size variation on VoIP quality?
- What is the effect of compression on VoIP quality?
- What is the effect of simultaneous packet size and compression variation on VoIP quality?
- How can speech quality be managed adaptively using wideband codecs?

This scenario assumes that a given number of calls is managed by trying to find methods to improve average call quality under different network conditions (with different voice and background traffic loads). The number of calls is assumed constant in a given experiment and potential speech quality improvement can be achieved by managing the bandwidth required for a given call or a group of calls. The main goal of this part of the dissertation is to answer the following questions: What is the best way to manage speech quality in the given scenario? Does packet size or compression variation provide any positive effect on average communication quality? Which of these approaches is better (provides better resulting quality)?

In the second scenario we investigate the tradeoff between quality and performance. We do not use the assumption that the number of calls is fixed and the G.711 codec is used for speech encoding. Voice load is variable and we want to see how speech quality changes with increase of number of calls in the network and different codecs (Section 5.6). Based on our quantitative results, it is possible to develop various economic models and to see a tradeoff between required bandwidth resources (cost of communications) and quality.

Each Section includes a theoretical part and simulation results.

5.1.2 Related Research

Only a limited number of studies in the area of adaptive VoIP are available. Hiroyuki Oouchi *et al.* from NTT Laboratories, Japan, published a paper "Study on Appropriate Voice

Data Length of IP Packets for VoIP Network Adjustment" [71]. The paper evaluates suitable voice-data length in IP packets for the adjustment of VoIP network systems. The authors used a simple test network, generated background traffic, and evaluated the effect of changing the voice data length of IP packets and the packet loss rate in the simulated network. From the experiment, the authors found qualitative relationships between voice packet size and quality. They concluded that: (1) a VoIP system with long voice packets has high-transmission effectiveness but has a high deterioration in the voice-quality level in an inferior network; (2) a VoIP system with short voice packets is tolerant to packet losses and preserves voice quality. Based on these results, the authors concluded that "in most cases", a variable voice packet-length VoIP system would be useful to achieve both high-transmission effectiveness and stable voice quality. No adaptive algorithms were proposed in the paper and no detailed quantitative investigations were provided. Paper [72] also provided an experimental study of the effect of packet-size on speech quality. Based on subjective experiments, the authors proposed using the equation:

$$\text{Predicted MOS} = 4.3 - 0.7 \ln(\text{loss}) - 0.1 \ln(\text{size}) \qquad (5.2)$$

to describe the effects of packet size variation and packet loss, although it was not confirmed by testing in various scenarios and network conditions.

Boonchai Ngamwongwattana, in his dissertation [73] and in the paper "Variable rate VoIP based on packetization" [74], investigated the relationship between packet size and channel capacity requirements and used the end-to-end delay of VoIP communications as a quantitative optimization criterion. His theoretical studies and experiments concluded that there is a tradeoff between voice packet-size and total end-to-end delay. Small voice packet-size is preferred for minimal incurred delay but, because of a large IP-rate requirement, it has the potential to cause congestion, which could result in increasing end-to-end delay. Large voice packet size incurs additional delay due to packetization. He assumed that a choice of optimal packet size is possible to match available network capacity (to minimize end-to-end delay) and confirmed the hypothesis by theoretical investigations and simulation studies. This dissertation extends his work as follows:

(1) As demonstrated below, delay minimization does not necessary mean the optimization of quality.

(2) The relationships between multiple factors that affect speech quality are very complex and analyzed in detail in this project.

(3) This dissertation considers more complex systems with variable background traffic, efficient adaptive jitter buffers, and multiple scenarios.

We have already mentioned the GSM Adaptive Multi-Rate coder (AMR) [8, 17, 18, 47], which enables rate adaptation using variable encoding (changing speech compression). The latter part of Section 3.1.1 reviewed several papers, which proposed using this AMR codec in VoIP networks, while most other studies in this area focus on specific elements in developing an adaptive VoIP system. For example, Qiao *et al.* [75] develop a system combining the AMR codec (compression variation) and the DiffServ priority scheme. C. Mahlo *et al.* [76] and Jorg Widmer et al [77] study an adaptive VoIP system based on TCP-friendly flow control algorithms. Sender-based Error and Rate Control mechanisms for audio, such as by Bolot and Garcia [78] and Mohamed *et al.* [79] adapt to packet loss using RTCP feedback from the receiver. L. Roychoudhuri and E. Al-Shaer [80] propose using different audio codecs with different bit rates depending on the situation in the network. They predict network loss based on available bandwidth, also taking network loss as decision parameter.

5.2 Simulated Network

This chapter will analyze the two scenarios described in the previous Section. Both scenarios will provide theoretical studies and simulations results. This Section explains the assumptions and describes the simulated network structure used in the dissertation.

The quality of VoIP communications can be managed by changing codec settings on both the sender and receiver sides. There are only four end-user parameters that can be changed to affect speech quality (Figure 5.1):

- Voice payload size (milliseconds)
- Compression / encoding

- Signal frequency range (narrowband / wideband)
- Jitter buffer size (milliseconds)

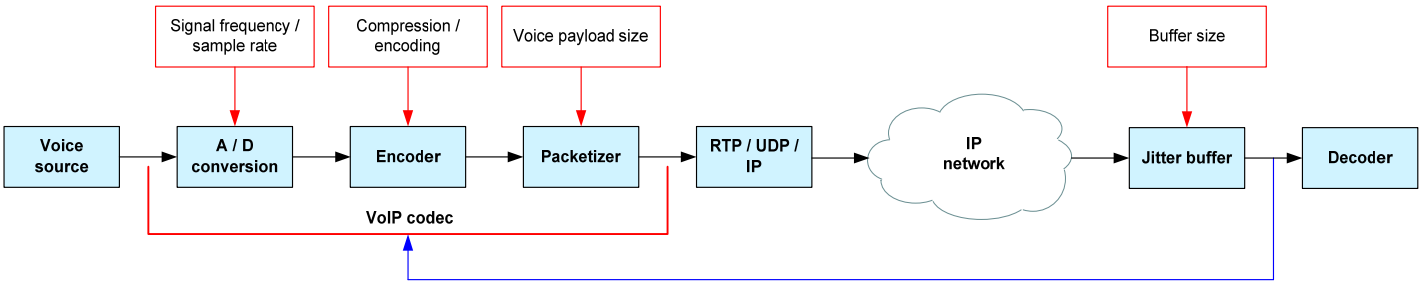


Figure 5-1: VoIP System Management

This dissertation considers a system consisting of three main components: a sender (a source of VoIP traffic), a receiver (a destination point of a VoIP stream) and the network. The sender can be represented by individual users or by VoIP providers (or PBXs). In the first case, the process of speech flow formation occurs on the local user's computer and a call typically goes through the traditional data network together with other types of traffic. The manipulation of codec settings in this case can happen only on the end-users' computers (Figure 5.2).

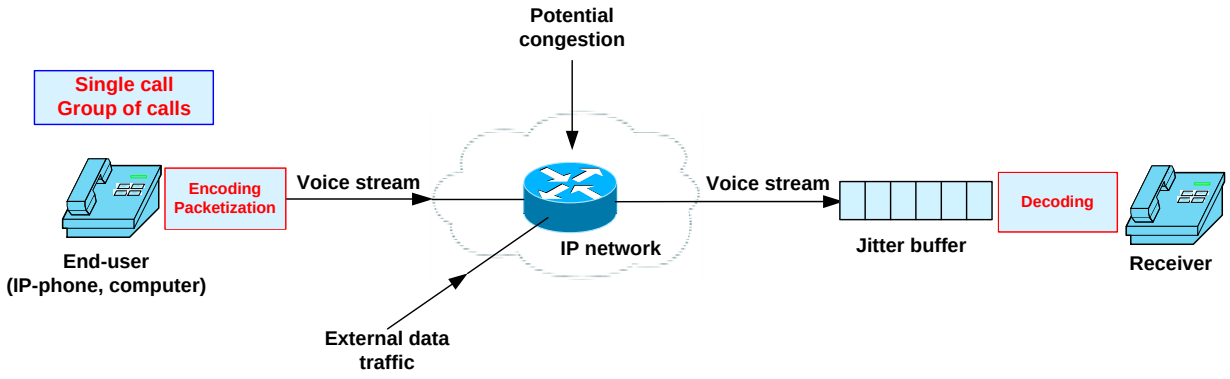


Figure 5-2: Single call management

In the second case, people may use VoIP technology using regular telephones, but making calls through a specially designed VoIP network (for example, making international calls using the services of a long-distance provider such as Qwest). In this case, individual calls or groups of calls can be managed by the VoIP provider (Figure 5.3).

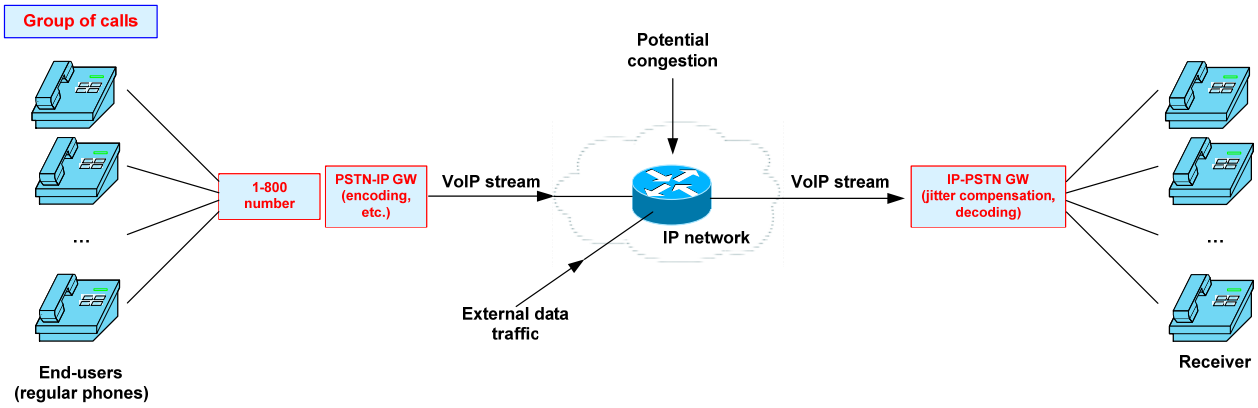


Figure 5-3: Group of calls management

This Chapter analyzes real-time speech quality management under both the single-call and group-of-calls scenarios. The detailed description of the simulated network is presented below (Figure 5.4). The simulation code was written and executed in Matlab.

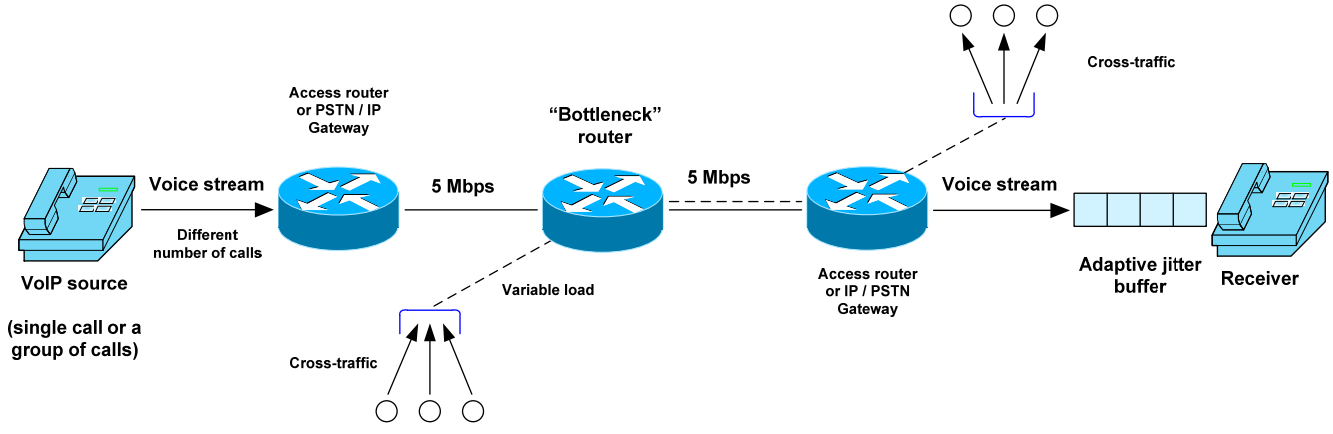


Figure 5-4: Simulated network

Parameters and assumptions:

- Voice source and call characteristics:
  - Speech codecs with variable parameters (packet size, compression, signal frequency range) are used
  - Simulated call duration – 120 seconds
  - Voice stream includes a single call or a group of calls; number of calls is variable from 0 – 100% link utilization
  - Silence suppression is not used (for simplicity)
  - All calls in the group use the same speech encoding algorithm; calls can be managed simultaneously; and all have the same behavior (the quality of all calls degrades equally)
  - There are different ways to restore lost packets. The detailed overview of multiple techniques is presented in [82]. This project assumes that codecs use internal packet loss concealment algorithms; the performance of these algorithms is defined by the Bpl factor from the E-model.
- Jitter buffer:
  - Uses the selected adaptive algorithm in Section 3.1
  - Calls consist of talkspurts of a fixed size (300 ms); and the adaptive jitter buffer can change its size only in the intervals between talkspurts. In our simulation, this period is also equal to 300 ms but, actually, this number is not important because we analyze only speech frames. Different papers use periods between 200 and 700 ms to describe talkspurt durations and there is no agreement about the "best" number. The ITU P.59 [89] recommendation specifies an artificial on/off model for generating human speech with the talkspurts and silence intervals of 227 ms and 596 ms. Simulation tools like OPNET use, by default, an on/off model with exponentially distributed on/off phases with durations of 352 and 650 ms, respectively. Jiang and Schulzrinne [90] reported mean spurts and gaps of 293 ms and 306 ms in experiments with the G.729 codec. The classical paper [91] of P. Brandy published in 1969 reported mean spurts and gaps of about 0.7 second. So, for simplicity, the simulations in this dissertation use fixed durations of talkspurts and silence periods both equal to 300 ms. Increasing this number means that the adaptive jitter buffer

- Minimum jitter buffer size is 30 milliseconds

- Connections and routing:
  - 5 Mbps links are used in the simulation (actually, link capacity is not important; portions of voice and data traffic in the network are important)
  - The network has a single place of congestion, which is simulated by an excessive number of calls or by bursty background traffic through one router
  - The bottleneck router uses FIFO queuing and the drop-tail mechanism in the case of overflow. The router has a fixed queue size (64 Kbyte), enough to keep packets for about 100 milliseconds
  - Network delay (transmission, propagation, network processing) is assumed fixed at 100 ms
- Background data traffic:

It is desirable to generate background traffic with characteristics similar to traffic patterns in the Internet. The problem of the Internet traffic modeling is very complex and Internet traffic measurement and modeling has been a subject of interest as long as there has been Internet traffic. Traditionally, data traffic in the Internet was described by a Poisson process, but studies revealed that Internet traffic exhibits properties of (1) self-similarity, (2) burstiness and (3) Long-Range Dependency (LRD) [86]. The self-similar process shows that short-time traffic behavior patterns are close to long-time patterns. LRD means that there is a statistically significant correlation across large time-scales [87], which is different from Poisson processes. Modeling self-similar traffic is not an easy task. In [83, 92], it is suggested to use multiple Pareto aggregation sources with a shape parameter $a < 2$ (a is a parameter in the Pareto distribution, called the Pareto index) so as to synthesize self-similar cross-traffic with behavior similar to the behavior of traffic in the real network. This dissertation uses this approach by generating Pareto On/Off traffic from 10 different sources. The traffic from each source consists of sending packets at a fixed rate only during the On periods, whereas the Off periods are idle. The aggregated traffic will have all the required characteristics. Network Simulator NS-2 [81] uses this model of background traffic generation.

It is important and surprising that the model does not separate the generated traffic into TCP and UDP flows. But the approach is good to model Internet traffic behavior, even in congested networks, ignoring nonlinearities arising from the interaction of multiple traffic sources because of network resource limitations and TCP's feedback congestion control algorithm [93]. One possible reason is that more than 90 percent of TCP sessions in the Internet are very short (1-2 seconds) and exchange less than 10 Kbytes of data [94].

- Average utilization is different in various experiments (0-100% link utilization)
- This Section assumes that we know the long-term average link utilization (that is we can, for example, generate data traffic to ensure an average of 50% link utilization during a 60-second call). This simulated utilization may be slightly different from a target value; the difference does not exceed 2% (for example, values between 48% and 52% are acceptable if we need 50%).
- Traffic is generated by multiple (ten) similar sources. It is Pareto-distributed with parameter a = 1.5. Such traffic produces Long Range Dependent traffic [83] with behavior similar to the behavior of traffic in the Internet. Packet sizes of "typical" Internet traffic are concentrated around three values: about 60% of the packets are 40 bytes, 25% are 550 bytes, and 15% are 1500 bytes [84, 85]. In terms of loads, small 40-Byte packets generate 6% of total traffic, 550-Byte packets generate 36% and 1500-Byte packets generate 58% of total traffic load.

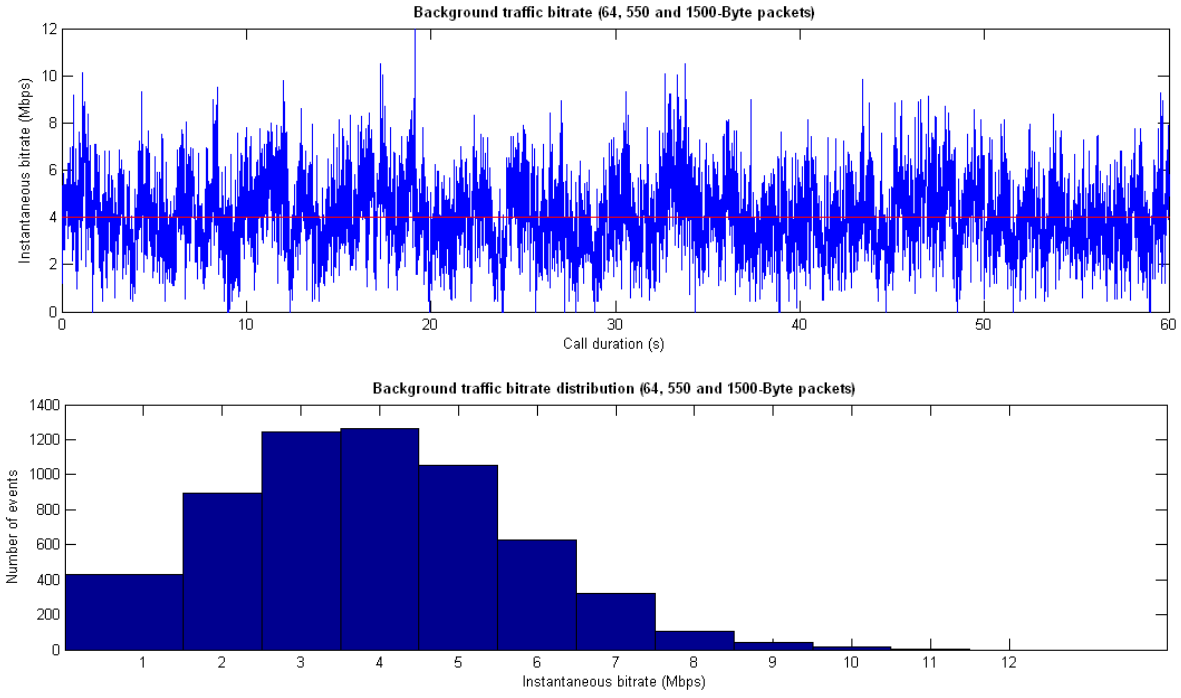
- General assumptions:
  - The network is not manageable (that is, we do not have access and cannot reconfigure network equipment and/or re-distribute traffic) and is not predictable (that is call quality degradation can occur at any moment of time, although the probability of this event may be different in different network conditions).
  - There is no speech decoding / encoding inside the network.

According to recent research [98], the nature of traffic in the Internet changes and assumptions used in this section may not be true in future. The proportion of peer-to-peer traffic in the network increased significantly during the last several years and exceeds 50% of the total traffic. This fact may change two assumptions: a) the packet-size distribution of the Internet traffic will change because most of peer-to-peer and streaming video applications use 550-byte packets; b) TCP session will exchange more data and will be longer.

If the duration of most of TCP session increases, the traffic generation model becomes more complex: it has to model the TCP back-off mechanism (a process of data packets retransmission in the case of packet loss). This question is not covered in the project and should be investigated in future.

If packet-size distribution is not concentrated around the three values (64, 550 and 1500 Bytes) but most of packets have 550-Byte size, traffic characteristics may also change. For example, Figure 5.5 shows background traffic patterns in two scenarios: a) three packet sizes are used; assumptions about their distribution are described above; b) only 550-byte packets are used. In this example, the average long-term link utilization is 4 Mbps, the analyzed time period is 60 seconds; instantaneous traffic bit-rates (per 10-millisecond interval) are calculated.

In both scenarios the amount of traffic in the network is the same. The Figure shows that in the absence (or with presence of a small number) of large 1500-Byte packets, the traffic behavior in the network becomes more stable (less bursty). The question about the effect of peer-to-peer and real-time video traffic on the traffic behavior in the Internet and on the traffic generation model should be investigated in more detail. This project uses the On/Off Pareto traffic distribution and packet sizes 64 bytes (60% of packets), 550 bytes (25% of packets) and 1500 bytes (15% of packets).



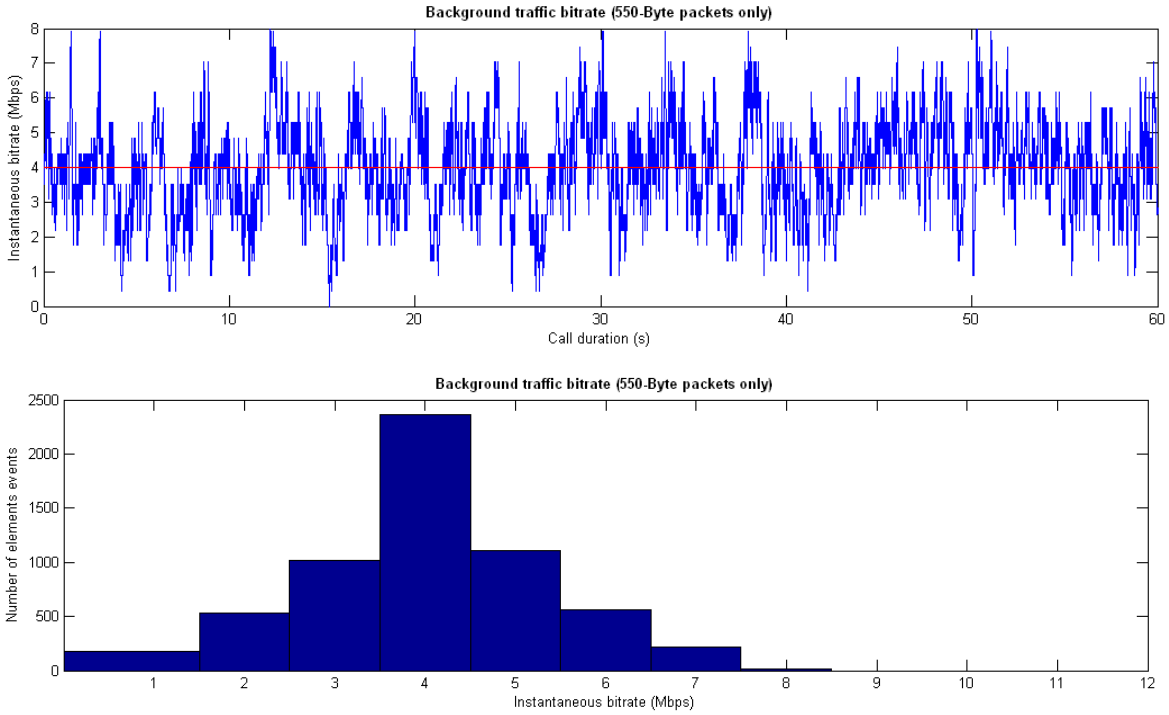


Figure 5-5: Background traffic generation

5.3 Investigation of VoIP Quality as a Function of the Proportion of Voice and Data Traffic in the Network

Before investigating the effect of various parameters on speech quality, it is important to demonstrate that speech quality is affected by not only link utilization, but also by the proportion of voice and data traffic in the network. A significant difference in VoIP quality is seen over the same network, with the same total (voice plus data) average traffic load, but with different voice-to-data traffic ratios. This hypothesis is rather evident: the presence of large data packets with bursty behavior creates some instability in the voice transmission process, which causes additional delay variation (jitter) and, as a result, higher delay and/or packet loss. So, this Section not only investigates the impact of jitter on VoIP quality but also

analyzes the reason for the jitter. This Section will check this hypothesis experimentally and will show quantitative differences in the resulting qualities.

The simulation design is based on the network scheme and all the assumptions described in the previous Section. It uses the narrowband G.711 codec with 64 kbps of audio bit rate. At 100 packets per second and 40 Bytes of IPv4 overhead per packet, the net bit rate is 96 kbps. The E-model defined by equations from Section 2.2.2 was used to estimate speech quality level on the receiver side.

Simulation parameters:

V – Portion of voice traffic in the network = Voice load (Mbps) / Link capacity (Mbps)

D – Portion of data traffic in the network = Average data load (Mbps) / Link capacity (Mbps)

U = V + D – Total link utilization (voice and data traffic; long term average)

$\mathbf{D_0}$ = D / U - The ratio of data traffic and total traffic

Speech quality is measured in MOS depending on U and D_0. U changes from 0.7 to 1.0 (from 70% to 100% average link utilization) in 0.1 unit steps. Speech quality degradation in the network with utilization lower than 70% was very rare and not significant. This number (70%) depends significantly on the assumptions and will be different with other adaptive jitter buffer algorithms, different approaches for background traffic generation, and assumptions about talkspurt duration. For example, preliminary simulation studies demonstrated that, in the case of a fixed 30-ms jitter buffer size, speech quality degradation was noticeable with 60% load. Changing the assumption of the 300-ms talkspurt duration also impacts results. For example, assuming 500-ms talkspurt duration, speech quality degradation will also be noticeable with a 60% load and will be more significant at higher loads than in the case of the 300 ms interval.

D_0 changes from 0 (no data; voice only network) to 1 (assume that there is a single voice call; the remaining traffic is data) in 0.2 unit steps. Given U and D_0, it is obvious that $D = U \cdot D_0$ and $V = U - D$. For example, if at U = 80% link utilization and D_0 = 60% of data traffic, the simulation requires 5 Mbps · 0.8 · (1-0.6) / 96 kbps = 17 simultaneous G.711 calls and 5 Mbps · 0.8 · 0.6 / 10 = 240 kbps of Pareto On/Off background traffic from each of the ten traffic generation sources (2.4 Mbps total). For each set of {U, D_0}, the simulation was executed 100 times. The results are presented in Table 5.1 and Figure 5.6. The Table contains

mean MOS scores for each set {U, D_0} and standard deviations for the scores.

Table 5-1: Speech quality depending of voice-to-data loads ratio

U - Total link utilization (average during a call)	D0 (Data traffic / Total traffic)	Mean MOS	Standard deviation of MOS scores
0.7	0	4.35	0.00
	0.2	4.35	0.00
	0.4	4.35	0.00
	0.6	4.35	0.00
	0.8	4.35	0.01
	1.0	4.23	0.05
0.8	0	4.35	0.00
	0.2	4.35	0.00
	0.4	4.35	0.00
	0.6	4.30	0.02
	0.8	4.16	0.07
	1.0	4.00	0.15
0.9	0	4.35	0.00
	0.2	4.35	0.00
	0.4	4.23	0.04
	0.6	4.05	0.13
	0.8	3.86	0.16
	1.0	3.52	0.17
1.0	0	4.35	0.00
	0.2	3.95	0.14
	0.4	3.72	0.17
	0.6	3.54	0.23
	0.8	3.42	0.24
	1.0	3.30	0.21

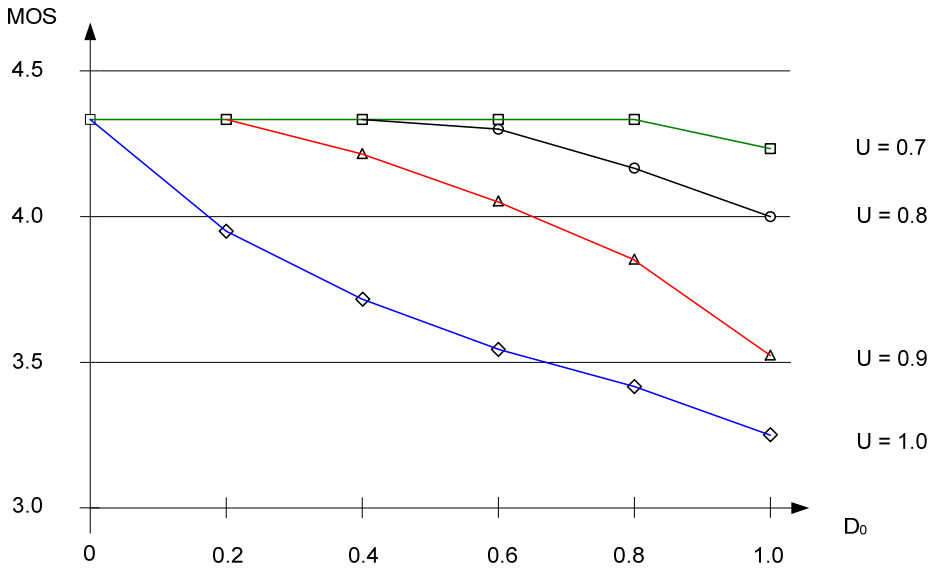


Figure 5-6: Speech quality depending of voice-to-data loads ratio

These results show that even if link utilization is constant, the behavior of voice traffic significantly depends on the presence of, and load of, the data traffic in the network. Even a relatively small volume of data traffic can cause a significant degradation of voice traffic. While this conclusion is intuitively clear, the simulation study gives numerical estimates. If the average link utilization lower than 70% ($U \leq 0.7$), voice quality degradation is rare and hardly noticeable. Local quality variations caused by burstiness of background traffic are compensated by the router queue and the adaptive jitter buffer. If average link utilization exceeds 70%, adaptive quality management mechanisms can be used and change of packet size and speech compression may potentially decrease the quality degradation effect. So, the next step is to investigate the impact of encoding parameters under different scenarios. It should also be remembered that, in "real networks", the average link utilization parameter is not known, so network state detection mechanisms must be proposed to estimate the impact of the network on speech quality. If $U < 0.7$, the signal frequency range can be changed to allow the use of wideband codecs (if wideband communication is possible). More channel resources will be used per call, but a voice quality will increase. This mechanism can only be used in IP-to-IP communication because the PSTN limits the call bandwidth to 4 kHz.

5.4 The Effect of Packet Size Variation on VoIP Quality

The rest of this Chapter investigates the effects of several encoding parameters on the quality of VoIP communication under the scenarios described in Section 5.1 and the assumptions described in Section 5.2. This research begins in this Section by investigating the effect of packet size on speech quality under the first scenario, which assumes that: (1) speech quality is the most important parameter and average call quality improvement is the main objective, (2) the number of calls in the network does not change (we do not have a goal to maximize a number of calls in the network), and (3) narrowband telephony is used and the initial chosen codec is the G.711 with 10 ms packet size. Will a change of packet size improve speech quality? This Section's theoretical and simulation studies will answer this question.

5.4.1 Theoretical Study

As it was demonstrated in the previous Section, the quality of VoIP communication depends, not only on the speech encoding parameters and network conditions ("bottleneck" link utilization), but also on the proportion of voice and data traffic in the network. So, all these parameters must be considered in future analysis. The impact of packet size on speech quality is very complex and the total effect of packet size variation is difficult to describe theoretically because many of the parameters affecting speech quality (delay, loss, jitter) are not independent; improving one parameter may cause a decline in another. Some effects of packet size on speech quality are very clear, others are less evident. The following theoretical study investigates different relationships to packet size variation but the total impact is investigated experimentally only.

Four main relationships are identified:

(1) The first relationship is very evident: increasing packet size leads to an increase of end-to-end delay. Quantitatively, the effect is described in the E-model. The end-to-end delay impairment factor I_d is calculated as

$$I_d = 0.024D+0.11(D-177.3)\cdot H(D-177.3) \qquad (5.3)$$

where the Heaviside function, $H(x) =0$ if $x<0$ and $H(x)=1$ if $x \geq 0$;

and end-to-end delay D obeys (Equation 2.3 and Figure 2.3).

I_d takes into account all delay components, including packetization delay. A change of packet size by ΔD, leads to a change of the delay impairment factor by $\Delta I_d = 0.024 \cdot \Delta D + 0.11 \cdot \Delta D \cdot H(D-177.3)$. If the end-to-end delay is not too significant (less than 177.3 ms), and voice packet size is increased, for example, from 10 ms to 40 ms, the decrease in quality is $\Delta I_d = 0.024 \cdot \Delta D = 0.024 \cdot (50-10) \approx 1$ unit on 0-to-100 R-scale or approximately 0.05 MOS. The result is evident: if the end-to-end delay is not too large, the direct impact of packet size increase is very small and not perceptually noticeable. But, if the end-to-end delay is significant (exceeds 177.3 ms), the effect of packet size variation is calculated as $\Delta I_d = 0.024 \cdot \Delta D + 0.11 \cdot \Delta D = 0.134 \cdot \Delta D$. So, if we packet size is increased from 10 ms to 40 ms in this case, the decrease in quality is $\Delta I_d = 0.134 \cdot \Delta D = 0.134 \cdot (50-10) \approx 5.5$ R-unit. This number is not too large and approximately corresponds to 0.2 units on the traditional MOS scale (see Figure 2.4). This change in quality is not very significant but is noticeable. Smaller packet size variations (for example, from 10 ms to 20 ms or to 30 ms) have even less significant impact. But, increasing packet size does not always lead to a decrease in quality because there are other factors to analyze.

(2) Increasing packet size leads to a decrease of the IP-rate per call. This may decrease congestion in the network and improve the quality of communication. This dependency is also evident and the question of the effectiveness of voice transmission was discussed in Section 3.1.2. The table below shows the channel capacity requirements for different codecs and different voice payload sizes (including RTP/UDP/IP overhead).

Table 5-2: IP-rates for different codecs and voice payload sizes

	G.711 (no compression)		**G.726 (2:1 compression)**		**G.729 (8:1 compression)**	
Packet size (ms)	**IP-rate (kbps)**	**IP-rate (%)**	**IP-rate (kbps)**	**IP-rate (%)**	**IP-rate (kbps)**	**IP-rate (%)**
1 ms	384 kbps	+ 300%	352 kbps	+ 550%	328 kbps	+ 820%
5 ms	128 kbps	+ 33%	96 kbps	+ 50%	72 kbps	+ 80%
10 ms	96 kbps	Reference	64 kbps	Reference	40 kbps	Reference
20 ms	80 kbps	- 17%	48 kbps	- 25%	24 kbps	- 40%
30 ms	74.6 kbps	- 22%	42.6 kbps	- 33%	18.6 kbps	- 54%
40 ms	72 kbps	- 25%	40 kbps	- 37%	16 kbps	- 60%
50 ms	70.4 kbps	- 27%	38.4 kbps	- 40%	14.4 kbps	-66%

The table shows that a change of packet size significantly effects channel requirements per call. This is especially noticeable in the case of codecs with compression. The table also demonstrates a slight difference in IP-rates between 30 ms and, for example, 50 ms; so an increase of packet size higher than 30 ms will not provide any noticeable benefits in capacity. (3) There are two, less evident but also important, effects of packet size variation on VoIP quality. Both factors are negative (an increase of packet size causes speech quality degradation, although not directly). A loss of one "large" packet has more significant negative effect on speech quality than a random loss of several "small" packets. In the E-model, the effect of random and bursty packet loss is described by the Equation:

$$I_{e-eff} = I_e + (95 - I_e)\frac{Ppl}{\frac{Ppl}{BurstR} + Bpl} \qquad (5.4)$$

where Ppl is the packet loss rate, Bpl is the packet loss robustness factor (codec characteristics), and BurstR is the so-called burst ratio, which is defined as the ratio of the average length of observed bursts in an arrival sequence to an average length of bursts expected under random loss. When packet loss is random (i.e., independent) BurstR = 1; and when packet loss is bursty (i.e., dependent) BurstR > 1. If packet size is increased from 10 ms to 30 ms, BurstR goes from 1 to 3. One simulation modeled several G.711 calls and changed voice payload size from 10 ms to 20 ms. Because of the decreased IP-rate per call, packet loss decreased from 8.5% to 5.9%. With a 10 ms packet size, Ppl = 8.5% and Bpl=25, the initial packet loss impairment factor in the voice stream with 10 ms packet size was $I_{10ms} = 95\frac{8.5}{25+8.5} = 24.1$. With a 20-ms packet, BurstR = 20ms/10ms = 2, Ppl = 5.9%, and Bpl = 25, $I_{20ms} = 95\frac{5.9}{25+3} = 20$ (the lower impairment is better). And, if bursty loss is ignored, $I_{10ms} = 95\frac{5.9}{25+5.9} = 18.1$. The difference is not too significant (~0.1 MOS) and will be even smaller with smaller packet loss rates, but it will also be larger for packets with larger packet size and for codecs with worse packet loss concealment algorithms (smaller Bpl

factors). The impact is not very significant individually, but together with other negative factors, it may also impact speech quality degradation.

(4) If there is also data traffic in the network, a change of packet size increases the data-to-voice traffic ratio, which decreases the IP-rate of voice traffic. As demonstrated in Section 5.3, this may cause an increase of "instability" in the network, which may result in increased jitter, delay or loss. It is possible that this factor may also affect the resulting speech quality, but it is not clear how significant the effect might be.

It is seen that increasing packet size causes different effects on the speech quality. Since, the total effect is difficult to discuss theoretically, it is investigated experimentally.

5.4.2 Simulation Results and Analysis

The design of this simulation also uses the scheme and the assumptions described in Section 5.2. The portions of voice (number of calls) and data traffic (Pareto On/Off background traffic) in the network was changed, and the average call quality was measured for different packet sizes. The initial packet size is 10 ms and total link utilization (voice plus data) was measured based on this packet size.

Simulation parameters:

V, the portion of voice traffic in the network, changes from 0 to 1 in 0.2 unit steps

D, the portion of data traffic in the network (average during a call)

U = V + D, the total link utilization, changes from 0.7 to 1.0 in 0.1 unit steps

PS, the packet size, changes from 10 ms to 50 ms in 10 ms steps

For each combination of {V, U, PS} 100 experiments were run. Standard deviations of individual MOS values do not exceed 0.15 MOS and the 90% confidence interval for the MOS means is about 0.04 MOS. Graphs of the simulation results are shown in Figure 5.7.

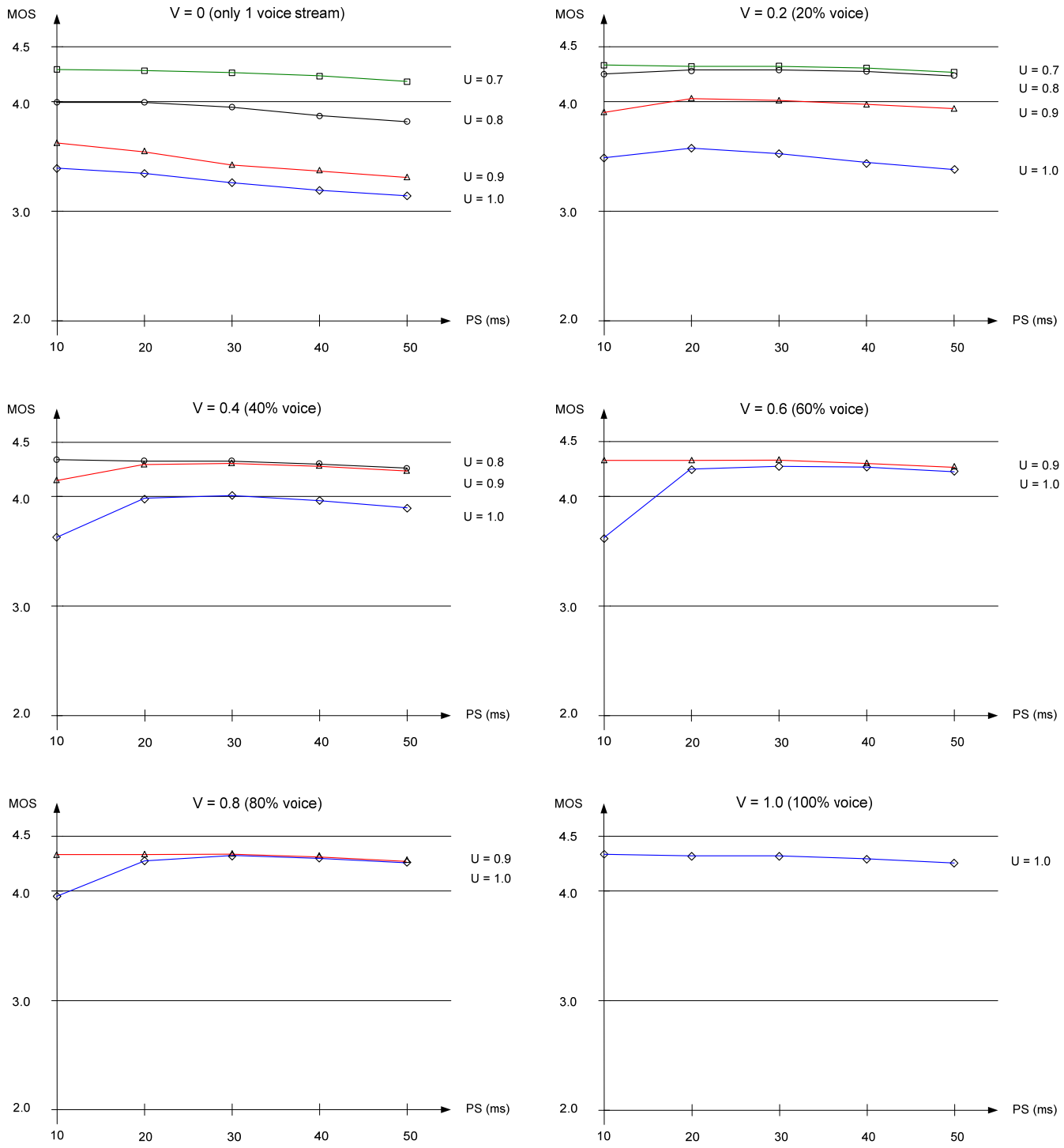


Figure 5-7: Effect of packet size variation on speech quality

The simulation leads to the following conclusions:

1. In the case of a single call or a relatively small number of calls, a change of packet size will not provide any noticeable improvement of communication quality. This result is evident: changes of voice characteristics do not impact the situation in the network significantly. Based on the results, the "lower bound" of V is defined when the adaptive packetization mechanism can be used (V is about 0.2 under the given scenario and assumptions).
2. With a higher voice load, a change of packet size improves quality despite the multiple negative effects discussed in the previous Section. Network capacity is seen to be the most critical factor affecting speech quality.
3. Discrete values 10 ms, 20 ms, and 30 ms must be used when choosing packet. "Optimal" choice of packet size is not possible and will not provide quality optimization because of variable and bursty background traffic. In most cases, a 20 ms packet size is enough to noticeably improve average call quality; 30 ms can provide even better quality under some scenarios.
4. Networks that are dominated by voice traffic are more stable. They can handle higher utilization (for example, 80% or even 90%) without noticeable quality degradation (again, this number will be different under different assumptions regarding the network structure, traffic generation, and jitter compensation mechanisms). VoIP-only networks can handle the 100% link utilization without noticeable quality degradation but we have to remember that even a voice network has data packets (for example, signaling packets, routing information, quality management packets), which may negatively impact speech quality. The summary of results is presented in Table 5.3.

Table 5-3: Impact of packet size on speech quality

	U = 0.7	**U = 0.8**	**U = 0.9**	**U = 1.0**
V = 0	No degradation	**No improvement**	**No improvement**	**No improvement**
V = 0.2	No degradation	**No improvement**	**No improvement**	**No improvement**
V = 0.4	No degradation	**Improved**	**Improved**	**Improved**
V = 0.6	No degradation	No degradation	**Improved**	**Improved**
V = 0.8		No degradation	No degradation	**Improved**
V = 1.0				No degradation

V, the portion of voice traffic in the network = Voice load (Mbps) / Link capacity (Mbps)
U, the total link utilization (voice and data traffic; long term average). U is the sum of the portions of voice and data traffic, so U > V.
These results are based on: using an efficient adaptive jitter buffer mechanism, a set of assumptions about background traffic behavior, a single place of potential congestion in the network, and the defined speech talkspurt duration. Different assumptions will provide different numbers in terms of impact of link utilization on speech quality but the general impact of packet size on voice quality will not change.

The next Section describes similar theoretical work and simulations on the effect of investigating impact of encoding / compression variation on the quality of VoIP communication.

5.5 The Effect of Compression Variation on VoIP Quality

Similar to the previous Section, speech compression can be used to decrease the bitrate-per-call and, as a result, the probability of voice quality degradation in case of congestion. This Section investigates two main issues: (1) To describe numerically the effect of compression variation on VoIP quality under the given assumptions and simulated network, and (2) To compare these results with the conclusions from the previous Section. These results should uncover some general rules of adaptive speech quality management to get the best possible quality under the given scenario. This Section starts with theoretical investigation.

5.5.1 Theoretical Study

Similar to the previous Section, a change of compression causes opposing effects on quality of VoIP communication. Again, the total effect, consisting of multiple factors, is difficult to describe theoretically. Here are some of relationships between codec compression and speech quality:

(1) Increasing compression generally leads to a decreased codec quality. This effect is described in the E-model by the codec-specific characteristic I_e, called the equipment

impairment [4, 12]. I_e describes the effectiveness of a given encoding algorithm and compression is one of speech encoding parameters. Better codecs have lower values of I_e: for example, the equipment impairment factor for the G.711 narrowband codec is 0 (with respect to narrowband communications; for wideband telephony, all parameters from the narrowband E-model should be recalculated using the methodology described in Chapter 4); I_e for the G.729 codec is 11. This means that the maximum quality, which can be achieved by the G.729 codec under ideal conditions, is noticeably less than that of the G.711 codec (see Equations 2.1 and 2.5).

(2) Increasing compression leads to decreased IP-rate per call. This may decrease congestion in the network (especially if a group of calls is managed) and improve quality of communication. The table below shows the bIP-rate requirements for different codecs compression ratios and 10-ms packet size (including RTP/UDP/IP overhead). Compressed voice streams are seen to use even less bandwidth than in the case of packet size variation.

Table 5-4: IP-rate requirements for different compression ratios

Compression (NB codecs)	IP-rate (kbps) No overhead	IP-rate (kbps) With overhead	IP-rate (%)
1 : 1	64 kbps	96 kbps	Reference
2 : 1	32 kbps	64 kbps	- 33%
3 : 1	21 kbps	53 kbps	- 44%
4 : 1	16 kbps	48 kbps	- 50%
6 : 1	11 kbps	43 kbps	- 56%
8 : 1	8 kbps	40 kbps	- 58%

(3) Increasing compression not only decreases codec quality (I_e increases), but also decreases the effectiveness of packet loss concealment algorithms. Since is more difficult to restore lost compressed packets, Bpl factors for codecs with compression are lower. For example, the Bpl factor for the G.711 codec is 25; the G.729 codec uses very significant compression and its lower Bpl is 19. This means that a loss, for example, of 1% of G.729 packets has more significant negative impact on speech quality than a loss of 1% of G.711 packets.

(4) As in the previous Section, if there is data traffic in the network, increasing payload compression increases the data-to-voice traffic ratio, which decreases the IP-rate of voice traffic. Section 5.3 discussed how this may cause an increase of "instability" in the network, which may result in increased jitter, delay or loss.

Again, a situation is seen in which a change of the encoding algorithm causes opposing effects on communication quality. Since the resulting net affect is difficult to describe theoretically, simulations are used to investigate the simultaneous effect all the factors.

5.5.2 Simulation Results and Analysis

This experiments compares the performances of three codecs: the G.711 codec (no compression), the G.726 codec (2:1 compression), and the G.729 codec (8:1 compression) under the same conditions as in the previous Section. The characteristics of these speech encoding algorithms for the 10-ms packet size are tabulated below:

Table 5-5: The G.711, the G.726 and the G.729 codecs characteristics

Codec	Compression	Audio bitrate	Total bitrate	Equipment impairment (Ie)	Packet loss robustness (Bpl)	Maximum MOS (with 150 ms delay)
G.711	1:1	64 kbps	96 kbps	0	25	4.3
G.726	2:1	32 kbps	64 kbps	7	23	4.1
G.729	8:1	8 kbps	40 kbps	11	19	3.9

The performance of these codecs are compared against the quality provided by the G.711 codec with a 30 ms packet size (the number of calls was not changed). As in the previous simulation, the change loads of voice and data traffic in the network are changed.

V, the portion of voice traffic in the network, changes from 0 to 1 in 0.2 unit steps

D, the portion of data traffic in the network (average during a call)

U = V + D, the total link utilization, changes from 0.7 to 1.0 in 0.1 unit steps

We run 100 experiments for each combination of {V, U, codec}. Standard deviations for individual MOS values do not exceed 0.15. Graphs of the average simulation results are shown in Figure 5.8.

The results show that, under the considered scenario (when a number of calls is not changed and quality improvement is our main objective), changing packetization provides better resulting speech quality than compression variation. When the portion of voice traffic is small, it is better not to change anything and keep minimal packet size. When the proportion of voice traffic in the network is more significant, changing packet size provides better quality despite the fact that the compressed speech uses less channel capacity.

Furthermore, the simulation demonstrates the effect of simultaneous compression and packetization variation on VoIP quality under the given scenario. Increasing the voice payload size of compressed speech requires even less resources per call than in the previous two situations, but the resulting voice quality becomes even worse. Increasing the voice payload size with no compression still provides better resulting quality.

Table 5.3 in Section 5.4 summarizes the results of this study. It shows that the variation of speech encoding parameters is not efficient in the case of a single call or a relatively small portion of voice traffic. If more voice streams can be managed simultaneously, increasing the voice payload size provides better resulting quality in congested networks. Voice-only networks are very stable and will not cause significant quality degradation even with load close to 100%. Increasing packet size, for example, up to 30 ms, can put even more calls through the network without noticeable quality degradation (the question about a tradeoff between a number of calls and speech quality will be discussed in the next Section).

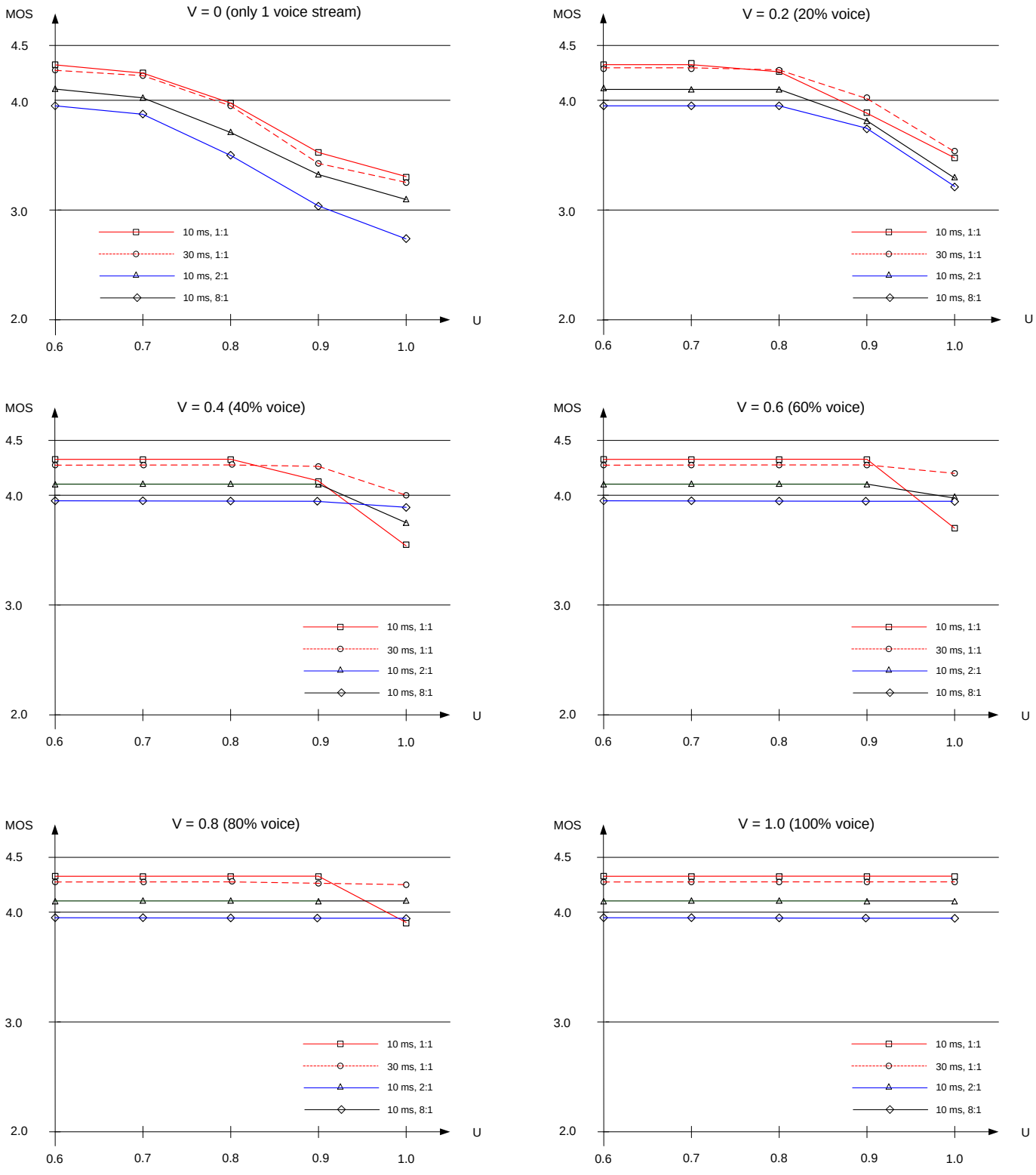


Figure 5-8: Effect of compression variation on speech quality

Is it possible to do anything with a single call or a relatively small number of calls? Here it is necessary to distinguish two scenarios: (1) IP-to-IP communications, and (2) IP-to-PSTN communications. In the case of IP-to-IP sessions, wideband encoding mechanisms can be used. For IP-to-PSTN call, regardless of how high the fidelity of a VoIP signal is, as soon as it passes into the PSTN, anything beyond 4 kHz of voice bandwidth (8 kHz sampling) is discarded. Furthermore, a 16 kHz signal that is downsampled to 8 kHz typically does not sound as good as one that started at 8 kHz from the start. This is a critical issue.

In the case of IP-to-IP communication, wideband codecs can be used. In the analyzed scenario, the narrowband G.711 codec can be replaced by the wideband G.722 codec, which uses 16-kHz sampling rate to encode a wideband 8-kHz signal. The codec uses 2:1 compression, which results in 64 kbps per audio stream with 10 ms packet size (the same as in the case of the narrowband G.711 codec). So, the narrowband G.711 can be simply replaced by the G.722 codec without increasing congestion in the network. Quality improvement under this scenario can achieve about 0.7-0.8 MOS, which is noticeable and significant (see the discussion in Section 4.2.1). All previous conclusions regarding packet size and compression variation are also valid for the wideband codec because it has the same IP-rate as the G.711 codec. Thus, the voice payload size of speech encoded by the G.722 codec can be managed to decrease the degradation effect from congestion. So, even in the cases of a single call or a relatively small portion of voice traffic, a noticeable quality degradation caused by the network be elevated to toll-grade level of quality under the given scenario.

5.6 Tradeoff between VoIP quality and Effectiveness of Communications

The first scenario assumed that there was enough network capacity to support a sufficient (required) number of the G.711 calls and that quality optimization was the main objective. Under these assumptions, different mechanisms of adaptive speech quality management were investigated for a congested network. But, providers usually have a different objective: they want, not only to provide a reasonable conversation quality, but also to improve the economic effectiveness of their business by increasing the number of simultaneous calls

through their network. Network quality parameters like delay, jitter and loss, which affect the number of calls that a network can support, are dependent on network design issues like what kind of network equipment is used and whether a communications network reaches across the street or across the country. The best way to determine the maximum number of simultaneous VoIP calls with a given average quality level is by simulating these calls on the network to determine how many calls can be supported and still maintain acceptable voice quality.

This Section demonstrates that an intelligent choice of speech encoding parameters is beneficial in this tradeoff between speech quality and number of simultaneous calls. Discussion and simulations are still based on the same network structure and assumptions that were described in Section 5.2. As in the previous scenario, background data traffic may exist in the network and its average rate is known (this assumption is made only in this Section, network state detection mechanisms will be investigated in the next Chapter).

Three network-types were investigated using three narrowband codecs: the G.711, G.726, and G.729. Similar simulations can be performed with wideband codecs or with a different set of narrowband codecs. The first network-type is the voice-only IP network. Previous results demonstrated that voice-only networks are very stable and can provide high communication quality even with traffic loads close to 100%. The absence of bursty background traffic eliminates congestion if the voice load does not exceed link capacity. So, calculating the amount of IP-rate per call determines the total number of calls that can within a given link (assumed 5 Mbps). The result is demonstrated in Figure 5.9.

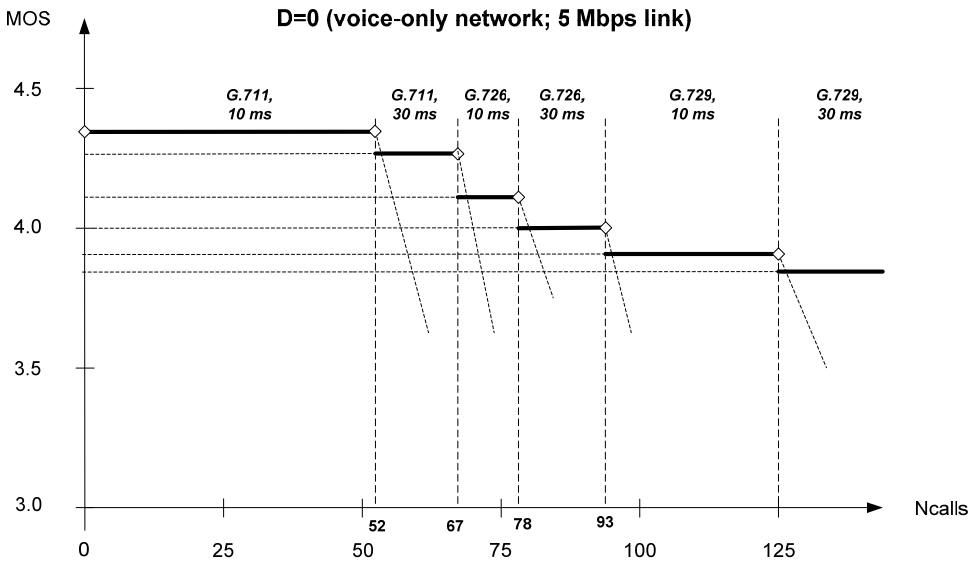


Figure 5-9: Tradeoff between a number of calls and quality in the voice-only network

The results show that, using the best existing narrowband G.711 codec, it is possible to send about 50 calls without noticeable quality degradation. If this number is increased by several percent, quality of all calls will degrade immediately and significantly. If this happens, the quality can be recovered by using a different codec (with higher compression) or by changing packet size. If we voice payload size is increased to 30 ms, more simultaneous calls (up to 67) can be transferred. But, because of increased packet size and its accompanying increased the end-to-end delay, the quality of these calls will be lower (see Section 5.4.1). If even more calls must be sent, codecs that consume even less capacity can be used. This can be achieved by payload compression because further increasing packet size does not give noticeable decrease of IP-rate. Using the G.726 codec with 2:1 compression and 10-ms packet size, the number of supported calls can be raised up to 78. The number of calls encoded by G.726 codec with 2:1 compression, but with 30 ms packet size, is even higher (93) and the quality level is still close to MOS = 4.0. The G.729 codec with 10-ms packet size provides a reasonable quality, which is a little less than the toll-grade level, but allows the number of simultaneous communication sessions to be go up to 125. This example demonstrates that, to improve effectiveness of speech transmission, both packet size and

compression can be managed choosing the most suitable encoding scheme in a given situation (when voice load in the network changes suddenly during a communication session or when it is necessary to send more calls through a link).

The second network type demonstrates that the network provide has much less flexibility if the network has relatively high background traffic. The negative effect of bursty background traffic is significant and even the best G.711 codec has quality lower than the toll-grade level. Figure 5.10 shows that codecs with higher compression or packet size will only help a little in such a network. For example, a voice-only network could use the G.729 codec and support 125 simultaneous calls with quality of about 3.9 MOS, when a20%-voice network carries a corresponding number of calls equal to 125 * 0.2 = 25 G.729 calls, the quality of these calls will be less than 3.0 because of the impact of the background traffic.

The third network type is the intermediate scenario, a network that carries 40% data load. Figure 5.11 shows that, using the G.711 codec, only about 20 simultaneous calls can be supported with the toll-grade quality level. With the G.729 codec we can send about 60 calls but with maximum quality level of about 3.9.

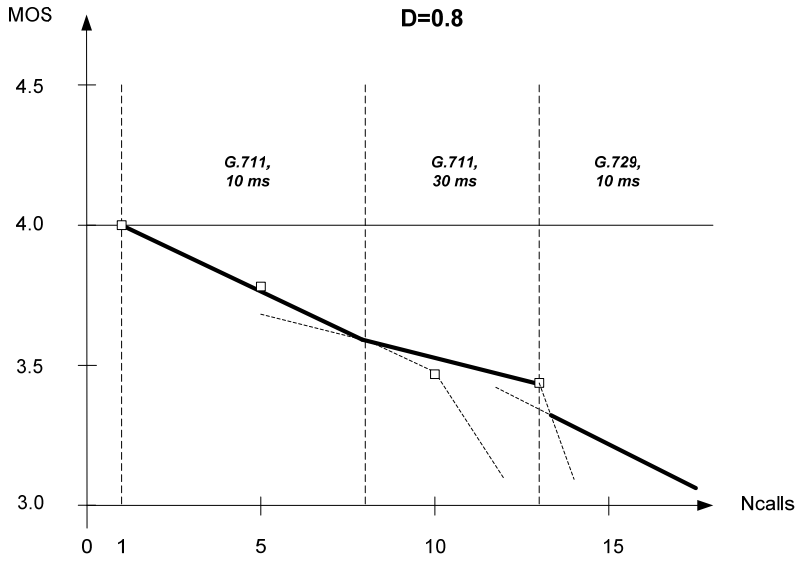


Figure 5-10: Tradeoff between a number of calls and quality in data network

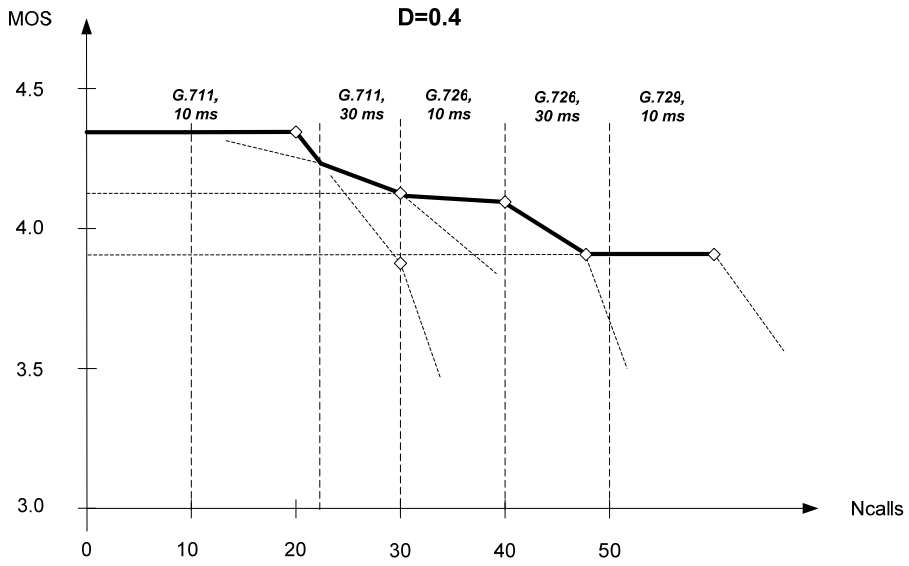


Figure 5-11: Tradeoff between a number of calls and quality with 40% of data traffic (1)

Similar simulations were run for other background traffic loads, in the range from 0 to 100%. Figure 5.12 shows how the maximum number of calls with toll-grade (or close to toll-grade) quality level changes with increasing of data traffic load.

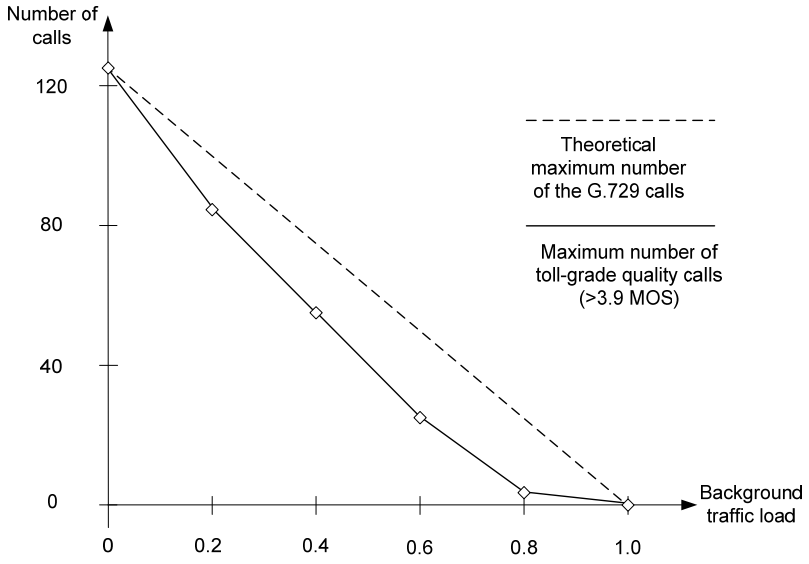


Figure 5-12: Tradeoff between a number of calls and quality with 40% of data traffic (2)

Based on the results, it is concluded that to improve the effectiveness of speech transmission both voice payload size and encoding mechanisms can be managed. This Section assumed that the average background traffic is known and that it is rather easy to choose the best codec depending on the number of calls in the network. If we this information is not known, some king of network state detection method would be needed to estimate the effect of the network on communication quality. This question is discussed in the next Chapter.

5.7 Summary

This Chapter investigated the effect of speech encoding parameters on quality of VoIP communications in two scenarios. The first scenario used we spoke about the narrowband G.711 codec and the goal was to find the best way to manage speech quality under different background traffic loads. Theoretical studies and experiments demonstrated that changing packet size provides better resulting quality than changing the encoding/compression despite the fact that a compressed voice stream uses less channel resources. The analysis extends to wideband telephony, where the narrowband G.711 codec can be replaced by the wideband G.722 codec, which uses the same amount of resources but provides noticeable improvement in quality. With a small number of calls or significant background traffic, changing the packet size or encoding will not improve communication quality. Wideband codecs can be used but they are problematic in the case of IP-to-PSTN communications.

Since the simulations are based on assumptions about network structure, end-user speech processing, background traffic generation, and speech characteristics (talkspurt duration), this Section uses a "good" scenario, based on an efficient adaptive jitter buffer mechanism and one place of potential congestion and On/Off Pareto background traffic in the network. Other jitter buffer algorithms or different models for background traffic generation will change numerical results, but the general conclusion will remain the same: managing voice payload size allows better quality level then using different codecs with higher compression.

The second scenario investigated the tradeoff between the number of calls in the network and the resulting average quality. Simulations demonstrated that a proper choice of

speech encoding mechanism will allow sending more calls through the network without significant quality degradation. This approach can be used if there is a variable volume of voice traffic in the network or if it is required to send more calls through a link. The results can be used to develop of various models to find optimal values between cost (bandwidth) and quality.

Both scenarios assumed that background traffic load in the network is known. In real networks this information is not available to the end-points, so mechanisms would be needed to estimate the effect of the network on communication quality. This problem, together with other questions, will be analyzed in the next Chapter.

Chapter 6

Adaptive Speech Quality Management

6.1 Introduction and Background

The previous two chapters of this dissertation investigated the effects of speech encoding parameters on the quality of VoIP communications. These chapters concluded, that if there is a moderate number of calls in the network (not less than 20% of voice traffic and no more than 100% of long-term average link utilization), the voice payload-size can be changed to improve the average communication quality in the case of degradation caused by bursty background traffic. If number of calls increases and the average bottleneck link utilization becomes close to 100% or even higher, it would be better to use a combination of packet size and compression variation. Previous chapter also investigated the question of how the traditional E-model changes in the case of wideband telephony and in the case of variable speech encoding. So, since communication is not limited to narrowband only, all principles and algorithms discussed in this chapter can also be applied to wideband telephony.

The goal of this chapter is to design an adaptive control mechanism that uses the previously discussed components of voice quality management. But the goal is not to propose some "optimal" adaptive quality management scheme because this question is very complex. It requires a separate research project to estimate the effectiveness of dynamic speech encoding under different scenarios, network conditions, and assumptions about network structure and background traffic behavior; and to investigate other questions and problems. This chapter has two main objectives: (1) to demonstrate that adaptive quality management can be used in real networks where we do not know the situation in the network (including the portion of background data traffic, etc.) at a given moment of time, and (2) to define conditions when these algorithms will provide noticeable quality improvement.

This chapter has three additional sections. Section 6.2 discusses the general approach toward adaptive quality management, main assumptions and scenarios. Section 6.3 presents the proposed adaptive quality management algorithms, including a detailed explanation and

analysis of each step of the algorithm and discussion of several other related questions. Section 6.4 describes the effectiveness and stability of the approach, discusses arising problems and analyzes the simulation results.

As we have already mentioned in the previous chapters, adaptive speech quality management in IP networks is a relatively recent idea, so there are only limited number of studies in this area. There is also a set of papers (reviewed below), which propose to adopt the Adaptive Multi-Rate (AMR) Codec to IP networks. As discussed in Chapter 3, the AMR codec was designed for GSM networks and makes decisions about bit rate adaptation based on noise level in the channel. But, different metrics and protocols are required in the case of data networks. Qiao et al [75] develop a system combining the AMR codec (8-bit rates between 4.75 kbps and 12.2 kbps with variable compression) and the DiffServ priority marking scheme. The decision metric used in their paper is a speech quality level measured in MOS using the PESQ algorithm (see Section 2.2). The decision to change the encoding mechanism is based on this parameter and feedback is sent to the source of voice traffic via RTCP reports every 5 seconds. The algorithm uses two thresholds to estimate the extent of quality degradation: 0.2 and 0.5 (based on the traditional 1-to-5 MOS scale). When the predicted PESQ MOS increases (or decreases) by 0.2, the bit rate of the encoder is set to the next higher (or lower) step. When predicted MOS decreases by 0.5, the bit rate of the encoder is reduced by half. This algorithm sounds very logical but it has several very critical drawbacks. First, it does not make any sense to use RTCP protocol to send feedback from a receiver to a sender. This protocol sends its messages every several seconds (usually every 5 seconds) and this interval is too large to provide any reasonable control. The second problem with the algorithm is that PESQ cannot be used on the receiver side. PESQ makes a decision about the quality of a given sample, comparing a received degraded signal with an original version of this signal. The PESQ algorithm is relatively accurate but an original signal is not available on the destination side (see the discussion in Chapter 2 about subjective and objective quality measurement). It is also not a very good idea to use the AMR codec in IP networks. This encoding algorithm is acceptable in wireless communications because it uses a very small amount of resources and provides a reasonable quality (which is generally lower than the toll-grade level). Taking into account the fact that mobile users do not expect a significant absolute quality of conversations, this codec is very good for use in wireless

networks. In wired networks, the situation is different: users do not have mobility and expect to get a level of quality that is not worse than the quality level of a regular telephone conversation.

There are several other related papers, which also propose using the AMR codec, but with a metric to estimate speech quality. J. Seo et al. [48] use network jitter as the indicator of network quality. Eight states are assigned (20 ms jitter increments) and each state corresponds to one of the eight bit rates of the AMR codec. The idea is interesting, the choice of the decision parameter may not be correct. There are two reasons: the adaptive jitter buffer compensates some of the delay variation and, in this situation, short-term quality behavior, which can be improved by the buffer, must be distinguished from long-term behavior. Also, the number of jitter increments mentioned in the paper is very questionable. It is difficult to image a situation in which jitter equals, for example, 100 milliseconds. The paper presented a reasonable idea, which can be potentially improved and further explored, but additional work is required, including a detailed experimental evaluation and confirmation of the assumptions.

Abreu-Sernandez et al. [95] uses packet loss instead of jitter as a decision parameter for AMR in VoIP. They choose five bit rates and rate adaptation is based on the following loss levels: less than 15%, 15-25%, 25-30%, 30-35%, more than 35%. This idea is also reasonable (although the choice of the numbers is also very questionable) because an increase of packet loss generally means a decrease of speech quality. However, packet loss depends significantly on the adaptive jitter buffer mechanism; few voice packets are lost in the network, more are lost because they overflow the jitter buffer on a receiver. If a very significant jitter buffer size is chosen, packet loss will be relatively low, but end-to-end delay will be significantly increased. Also, the authors do not define an interaction process between a receiver and a sender to send control information. A similar approach is investigated by Bolot and Garcia [78] and by Mohamed et al. [79]. In these papers, bit rate adaptation is also based on packet loss statistics using RTCP feedback from the receiver but, as already discussed, the RCTP mechanism is too slow to be used to send control information to a source of voice traffic.

B. Ngamwonwattana, in his dissertation [73], makes decision about codec rate adaptation based on moving average thresholds of delay and packet loss, and proposes sending control messages, not using RTCP, but "on demand".

L. Roychoudhuri and E. Al-Shaer [80] propose using, not the AMR codec, but a set of different audio codecs with different bit rates depending on the situation in the network. It is possible to choose codecs with quality better than the AMR, so toll-grade VoIP quality can be provided. This dissertation proposes to use this approach: to choose several codecs with different characteristics and quality levels. While the authors cited above also take network loss as decision parameter, more complex approach is proposed here.

6.2 General Overview of the Adaptive Voice Quality Management Process

6.2.1 Initial Information and Assumptions

Most of the assumptions used in this chapter are similar to those from the previous chapter. But there are still some significant differences. Monitoring a call or a group of calls on the receiver side, we do not know average link utilization and average volume of background traffic in the network. We know the number of simultaneously managed calls, but this information is not too important: quality depends on the proportion of data traffic in the network and, even if there are many calls, it cannot be said with high confidence whether the network is voice-only or even if voice traffic dominates in the network. For this reason, the conclusions from the previous chapter cannot be used directly. For example, one of the conclusions in Chapter 5 stated that packet size variation has a more significant positive effect on quality than codecs with a higher compression ratio if the average long-term link utilization does not exceed 90% (or even 100%). But this conclusion cannot be used in "real life" directly because the amount of background traffic through a bottleneck link is not known. Even if this statistics were known, the average long-term traffic load could not be measured because decisions are made in real-time based on instantaneous characteristics while background traffic is bursty and its utilization changes significantly and quickly.

The Adaptive Multi-Rate Coder (AMR) will not be used in this proposal because the quality of this codec is generally lower than the desired toll-grade quality level (see numerical values in Chapter 3). So, instead of using different AMR bitrates, this chapter's simulations will use three narrowband codecs: the G.711 (PCM encoding with no compression), the G.726 (ADPCM encoding with 2:1 compression), and the G.729 (the complex CS-ACELP encoding with 8:1 compression). While codec may have different voice payload sizes, the discrete values 10 ms, 20 ms, 30 ms will be used. So, nine different sets of encoding parameters are used. These assumptions are made because: 1) these codecs provide a relatively high level of quality (higher than or close to toll-grade); 2) their quantitative characteristics are known in terms of maximum encoded speech quality in the absence of packet loss and significant delay; and 3) the difference in channel capacity consumption in these codecs is significant: for example, the G.711 codec with 10 ms voice payload size requires 96 kbps per voice stream (64 kbps of audio bitrate and 32 kbps to send the RTP/UDP/IP overhead); the 30 ms G.729 codec needs 18.7 kbps channel (8 kbps audio rate and 10.7 kbps overhead bit rate). Selecting one of the nine sets of encoding parameters can be based on bitrate or codec quality. These codecs are chosen as examples; similar adaptive mechanisms can be used with a different set of narrowband codecs, with some set of wideband codecs or with a combination of narrowband and wideband codecs.

The simulation results presented in this chapter address narrowband telephony only. The algorithms can easily be extended to the wideband scenario based on the computational quality model developed in Chapter 4. Wideband encoding can be especially beneficial under two scenarios: (1) when just one or several calls are managed, the adaptive sender-based variation of speech encoding parameters will not provide any positive results (see the previous chapter). In this scenario it is better to use the best available codec; increasing the IP-rate per call will not make a situation in the network worse because the voice traffic is insignificant; (2) if the network has sufficient capacity to provide better-than-G.711 quality for many calls, the average communications quality can also be improved by using wideband codecs.

This chapter uses the same assumptions about network structure and background traffic generation as the previous chapter. The simulation from Chapter 5 was modified to

incorporate the adaptive mechanism. The source code of the Matlab simulation can be found in Appendix A.

When designing an adaptive quality-management algorithm, several related questions must be answered:

- Which factor (or factors) should be used to make a decision that a change of speech encoding parameters is required or not required at a given moment of time?
- How often should such a decision be made (per packet, per second, per talkspurt, etc.)?
- What algorithms can be used to manage quality adaptively (actions)?
- How should feedback from the receiver be sent to the sender side?

All these questions will be addressed below. It is important to remember that two adaptive mechanisms are going to work simultaneously: a) this proposed variable sender-based encoding mechanisms and b) the receiver's adaptive jitter buffer. The adaptive jitter buffer mechanism is used to improve short-term quality (its fast reaction does not change the encoding characteristics, but manages the delay-loss tradeoff). Sender-based management is designed to improve long-term voice flow characteristics (to choose the encoding scheme that best matches the given network conditions).

6.2.2 Scenarios

The dissertation develops and demonstrates the effectiveness of adaptive speech encoding by providing algorithms and analysis under the simplifying assumptions that number of calls, while different in various simulations, does not change within a considered session. The more general case of variable voice-load during a considered time interval (the problem in this situation is "to fit" a given number of calls to a channel maximizing their quality) is left for "Future Work".

6.2.3 Decision Metrics

Which parameters should be taken into account to make a decision about quality adaptation? One variant is to use, for example, the mean delay, the moving average delay, or loss

statistics. This approach has already been used in multiple papers mentioned above. It would not be very good to analyze these parameters separately: high packet loss definitely means a significant degradation in quality but low (or absence of) packet loss does not mean an absence of degradation because the adaptive jitter buffer size can be very significant and we can get high end-to-end delay instead of loss. So, it would be better to use these parameters together. In other words, one must measure quality, which depends on end-to-end delay, loss, and codec characteristics. As we mentioned before, this project does not analyze some "less evident" parameters affecting speech quality like echo, attenuation, noise in a channel, etc. The quality can be measured using the computational E-model. Chapter 3 had a long discussion about the accuracy of different quality measurement mechanisms, which concluded that the E-model is not a perfect tool to measure absolute quality level, but it is acceptable for measuring variations in quality. Since, goal of these algorithms is to achieve noticeable relative quality improvement, the E-model can be used to track changes in quality.

How can this E-model be used? The adaptive algorithms proposed here will measure and manage quality-per-talkspurt. Human communication consists of periods of active speech and periods of silence (Section 5.2 discusses the assumptions). The adaptive jitter buffer algorithm changes its buffer size in periods between talkspurts (during silence periods). So, it would be logical to analyze speech quality behavior, and to make decisions about adaptive quality management at the end of a talkspurt (at the end of an active speech period). The E-model can be used to measure the quality of each talkspurt (referred to as "instantaneous quality"). Assuming packets within a talkspurt have the same delay, network loss and jitter buffer loss can easily be accounted for. The difference between instantaneous quality levels in two consecutive talkspurts can be very significant because of the bursty nature of the background traffic. But, knowledge of instantaneous call quality is not enough to make a decision about changing the speech encoding parameters.

The E-model can also be used to measure the average call quality at a given moment of time during a conversation. This metric would be acceptable, but this dissertation tries a different approach, using a metric, that describes integral (total) speech quality. Integral quality that is calculated as a mean of instantaneous qualities is not a very good metric. First, these results sometimes are not mathematically correct. Imagine a situation in which 100

packets are divided into 10 equal talkspurts. Assume that 5 packets are lost in one of these talkspurts. Instantaneous quality of all talkspurts except one is maximal (for example, 4.4). The remaining talkspurt has 50% loss and has MOS = 1.6 (from the E-model). Average of all these instantaneous qualities is 4.1. From another point of view, out of the 100 packets, 5% are lost. If the E-model is applied to all these packets together, the MOS = 3.9, which is noticeably lower.

The second reason is that this model does not take into account the history of previous quality variations (frequent variations may result in a relatively high average quality but noticeably smaller real perceptual quality). As discussed in Chapter 3, quality levels at the end of a conversation have higher weights than quality levels at the beginning of a session. So, instead of using just a mean MOS metric, integral quality is calculated from the beginning of a call using the perceptual model of Rosenbluth [42] developed at AT&T. This model presents a call as a sequence of 8-second intervals. Quality within each interval is calculated as a mean of instantaneous qualities. Integral call quality is calculated as a weighed sum of the qualities of the longer intervals and reflects real human opinion. Perceptual quality is usually lower than the average computational quality if there are frequent variations of instantaneous quality levels. This model was confirmed by multiple subjective experiments performed by AT&T. There is also a patented dynamic quality of service monitoring method [96], which is based on this algorithm.

6.2.4 Control mechanism

In sender-based control, observations about the network and resulting speech quality must be reported back to the sender. Utilizing RTCP is a common approach: packet loss and delay variation statistics are included in RTCP reports. These packets are sent periodically, usually every 5 seconds. But, obviously, this type of control is very slow to respond to the network. If the control mechanism must make decisions more frequently, it is necessary to use a different scheme rather than RTCP. But more frequent periodic call control may introduce additional traffic in the network. The proposal assumes that the adaptive quality management mechanism sends control messages, not periodically, but “on demand”, that is, when a change of sender parameters is required. As the simulation results will demonstrate,

a decision to vary the encoding parameters is made less frequently than per talkspurt. Also, if a group of calls is managed, just one message can be sent to deliver feedback from the receiver, but not to control every call independently. This approach will not create a significant amount of additional control traffic in the network.

6.3 Design of Adaptive Quality Management Algorithm

This section describes the proposed adaptive quality management mechanism. Each step is explained in detail and an example is analyzed. The following section provides the results of a simulation study.

Step 1: Collect statistics of packet delays before the jitter buffer. If there are multiple calls, is assumed that the quality degradation pattern of all calls to be similar (this assumption is confirmed by the simulations). So, just one call is chosen for analysis.

Step 2: Calculate packet playout time. This parameter is calculated continuously but, since the jitter buffer is adjusted only in pauses between talkspurts, the end-to-end delay is constant within a talkspurt.

```
mean_delay(i)=0.875*mean_delay(i-1)+0.125*ins_delay(i);
mean_variance(i)=0.875*mean_variance(i-1)+0.125*abs(mean_delay(i-1)-
ins_delay(i));
playout_time(i)=mean_delay(i)+4*mean_variance(i);
```

The adaptive jitter buffer tries to improve short-term quality, while the goal of adaptive sender-based encoding is to improve long-term voice behavior.

Step 3: Calculate the quality of a talkspurt based on the E-model. This calculation includes: (1) calculating packet loss within a talkspurt (network and jitter buffer loss); (2) measuring end-to-end delay, which is constant within a talkspurt. This "quality per talkspurt" is referred as "instantaneous quality" and denoted it as Q_I. The difference between instantaneous

quality levels in two consecutive frames can be very significant because of the bursty nature of the background traffic. But, knowledge of this instantaneous call quality is not enough to make a decision about adaptation of speech encoding parameters.

Step 4: Calculate the maximum achievable quality level for a given codec under the given network conditions. This calculation is based on zero packet loss and minimum network delay (the sum of transmission delay, propagation delay and minimum queuing delay). Using this minimum network delay, calculate the maximum achievable quality under the given set of encoding characteristics:

$$MOS = f(R);\ R=R_0-I_e-I_d(\text{min. network delay + min. jitter buffer delay})$$

The result looks something like this:

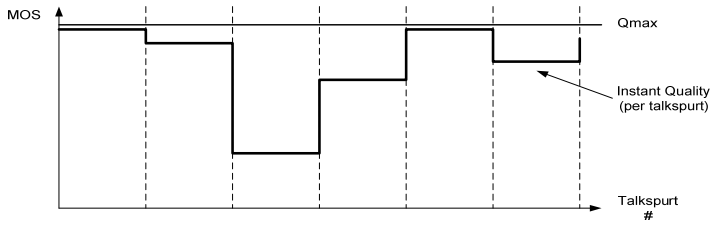


Figure 6-1: Instantaneous speech quality measurement

Step 5: Continuously calculate the integral voice quality level based on the AT&T model [42]:

$$W_i = \max\left[1, 1 + (0.038 + 1.3 \cdot L_i^{0.68}) \cdot (4.3 - MOS_i)^{\left\{0.96+0.61 \cdot L_i^{1.2}\right\}}\right] \quad (6.1)$$

$$MOS_I = \frac{\sum_i W_i \cdot MOS_i}{\sum_i W_i} \quad (6.2)$$

where W_i is he relative weight of passage *i*; MOS_i is the mean MOS rating of the degradation period *i*; L*i* is the location of period *i* (ranging form 0 to 1 with 0 being the beginning and 1 the end of a call), and MOS_I is the integral MOS.

The following is an example of the statistics collected in one of the simulations. The first picture shows the packet delay and the jitter buffer size calculated in Step 2. If packet delay (blue line) is higher than jitter buffer size (red line), the packet is lost (discarded) and restored using the codec's packet loss concealment mechanism. The second figure demonstrates instantaneous (per talkspurt) speech quality (Step 3) and integral (perceptual) quality (Step 5). The third picture shows the background traffic rate averaged over 10 ms intervals, and also the average long-term traffic rate.

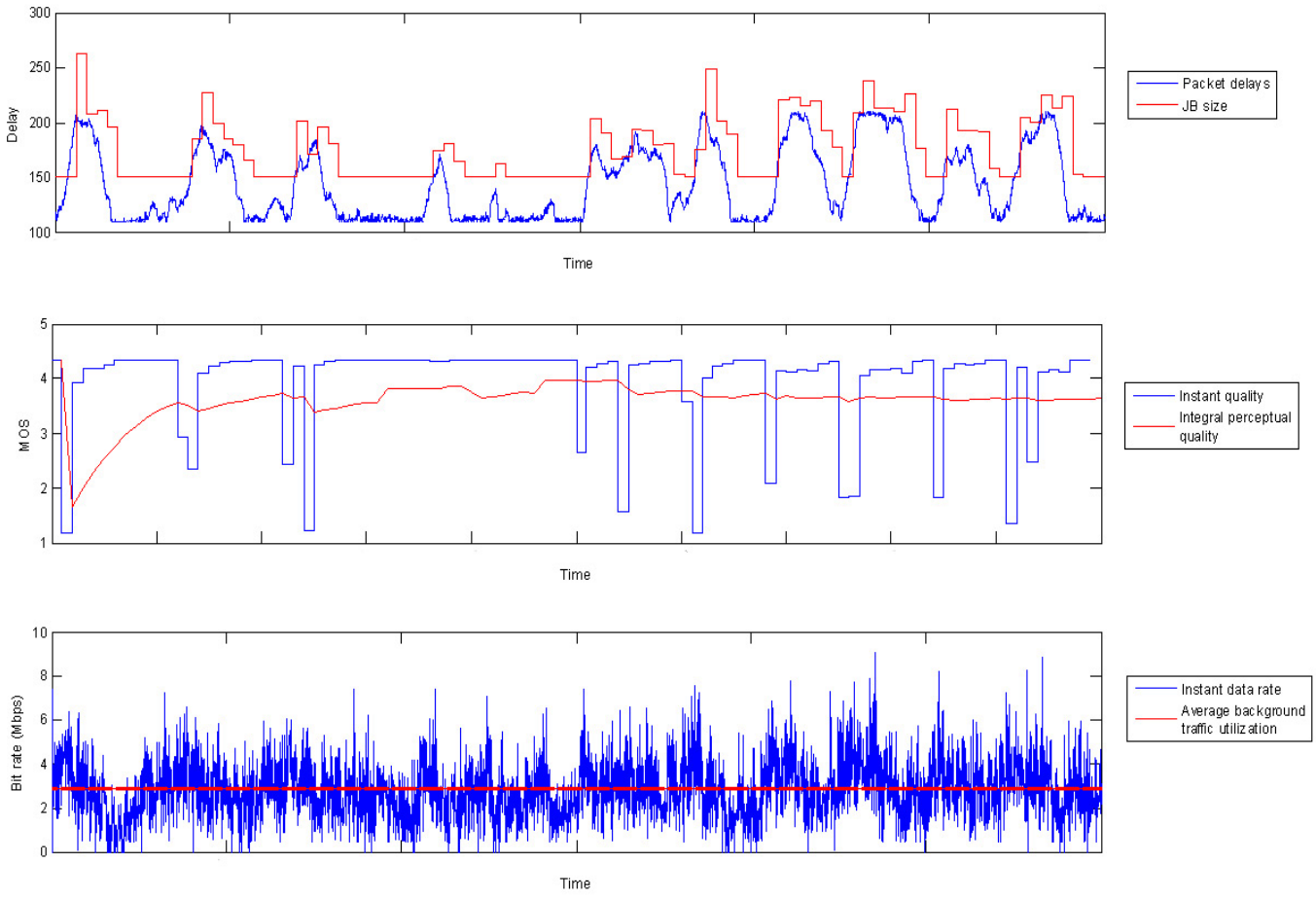


Figure 6-2: Example of statistical data collected in a simulation

Step 6: Decision

The proposed quality management scheme uses three parameters: 1) instantaneous quality level – Q_I; 2) integral perceptual quality – Q_T; and 3) maximum quality level achievable with under the given set of speech encoding parameters – Q_M. If the number of managed calls (which is assumed to be knows) is not significant (for example, one or two), voice flow is concluded to have an insignificant effect on the network, and it would be better to use the best available codec.

Similar to [75], changing audible quality is based on two thresholds: 0.25 MOS and 0.5 MOS. These thresholds are used only to describe integral quality variation and these numbers are not chosen arbitrarily. A change in quality of 0.2-0.25 MOS is not too significant, but is noticeable by some people; smaller changes in quality are noticeable only by a relatively small percentage of listeners [97]. A change in quality of about 0.5 MOS is very significant and is noticeable by almost everybody. If the best narrowband G.711 codec is used and its quality decreases by 0.5 MOS, the resulting quality will be noticeably lower than the toll-grade quality level. If the G.729 codec is used and its quality degrades by 0.5 MOS, the resulting level of speech quality (about 3.4-3.5 MOS) is considered to be low by most people.

The situation with thresholds for instantaneous quality level is a little more complex. Talkspurt quality can decrease for two reasons: 1) high jitter buffer size, which results in high end-to-end delay and, usually, not significant loss, or 2) packet loss (usually not in the network but on a receiver size caused by significant delay variation and insufficient jitter buffer size). The effect of delay is generally lower: for example, with 150 ms of network and packetization delay and an additional 80 or 100 ms of jitter buffer size, the decrease in quality is about 0.3-0.4 MOS. But, if a bursty loss of packets causes a loss of, for example, only 3 out of 30 packets in a given talkspurt, the decrease in the instantaneous quality equals to 0.9 MOS. If 5 packets out of 30 are discarded, the resulting quality (for the G.711 codec) will be only 2.65 MOS (with a maximum level of 4.3-4.4 MOS). As in the case of integral quality, two thresholds are used: 0.3 and 1.0. If the difference between maximum and instantaneous quality levels does not exceed 1.0 on the MOS scale, the observed packet loss is reasonable.

This model does not use quality adaptation mechanisms during the first several seconds of conversation because the perceptual quality model is very sensitive to quality variations in the beginning of a call. This period is set to 6 seconds (10 talkspurts plus 10 silence intervals between the talkspurts).

The details of the algorithm follow. Consider the differences between two parameters: 1) between the maximum and integral qualities (Q_M-Q_T), and 2) between the maximum and instantaneous quality levels ($Q_M - Q_I$). The first difference quantifies total quality variation; the second one describes instantaneous quality changes.

Condition 1:

- if $Q_M - Q_T > 0.5$ // Low or unacceptable level of quality. Something has to be done immediately
  - o if $Q_M - Q_I > 1.0$ // Instantaneous quality level is also very low. Not too much can be done in this situation. Switch to the G.729 codec with 30 ms packet size (the worst codec using the minimum IP-rate)

  => *Action: switch to the G.729 codec, 30 ms packet size*
  - o if $0.3 < Q_M - Q_I < 1.0$

  => *Action: keep current settings expecting that adaptive jitter buffer will compensate this degradation*
  - o if $Q_M - Q_I < 0.3$ // Total quality is very low, but instantaneous quality level is close to maximum. Quality degradation is not seen, so start slowly to improve the codec quality by deceasing packet size

  => *Action: decrease packet size by 10 ms if a current size is higher*

Condition 2:

- if $0.2 < Q_M - Q_T < 0.5$ // Degradation of quality is noticeable, try to improve the situation
  - o if $Q_M - Q_I > 1.0$ // Have a long bursty loss of packets. The network is significantly congested.

  => *Action: use codec with higher compression: if current codec is g.711, switch to g.726; if current codec is g.726, switch to g.729*
  - o if $0.3< Q_M - Q_I < 1.0$ // Also, the situation is not good and bursty packet loss is observed

=> Action: increase packet size by 10 ms or change codec if current packet size is 30 ms (maximum)

- o if $Q_M - Q_I < 0.3$ // Instantaneous quality is good; expect integral quality improvement

=> Action: decrease packet size by 10 ms if a current size is higher

Condition 3:

- if $Q_M - Q_T < 0.2$ // Degradation of quality is not significant but might be noticeable, try to improve the situation
  - o if $Q_M - Q_I > 1.0$ // Significant quality degradation. Total quality is good but one more bursty loss can significantly drop overall quality. Try to avoid.

=> Action: increase packet size by 10 ms up to 30 ms

- o if $0.3 < Q_M - Q_I < 1.0$ // Assume that this decrease of quality is temporal and due to single loss or increase of end-to-end delay

=> Action: keep current settings

- o if $Q_M - Q_I < 0.3$ // Everything is fine: both total and instantaneous qualities are high.

=> Action: decrease packet size up to minimum or switch to a better codec

The algorithm is summarized in the Table 6.1.

Table 6-1: Decision matrix

Q(M)-Q(T) / Q(M)-Q(I)	*<= 0.2*	*0.2 < … < 0.5*	*>= 0.5*
<= 0.3	- if current packet size is higher than 10 ms, decrease it - if current packet size is 10 ms but a used codec is not G.711, switch to a better codec	- decrease packet size up to 10 ms	- if current packet size is lower than 30 ms, increase it
0.3 < … < 1.0	- keep current settings	- if current packet size is lower than 30 ms, increase it	- if a used codec is not G.729, switch to a codec with higher compression
>= 1.0	- if current packet size is lower than 30 ms, increase it	- if a used codec is not G.729, switch to a codec with higher compression	- switch to the G.729 codec with 30 ms packet size

Step 7: Changes the speech encoding.

The action defined above cannot be executed immediately. The collected information about instantaneous and integral quality levels has to be transmitted to the sender. The transmission delay can be significant in congested networks. Assume that three consecutive talkspurts on the sender side (TS1, TS2, TS3) are separated by periods of silence (S1, S2). According to assumptions in Section 5.2, the durations of the active speech and silence period are 300 milliseconds. Assume that the receiver gets TS1 and makes a decision to send some control information to the sender. The period of time between the departure of the TS1 talkspurt and the arrival of the feedback from the receiver is equal to a round-trip delay (RTT). In congested networks this RTT might be significant and longer than the period of silence between the TS1 and TS2 talkspurts. In this case, the decision about quality adaptation will not be applied to the second talkspurt (TS2); it would be applied to TS2 only if the RTT is less than the S1 duration (300 ms). So, the receiver would not see the result of the requested changes of speech encoding parameters until the TS3 talkspurt, about one second later. The minimum reaction time of the algorithm is 300 ms (when RTT $\leq$ 300 ms). If the assumptions about speech/silence duration are different, these numbers will change respectively.

This fact has to be taken into account in the adaptation scheme. So, a restriction is added that, if a receiver analyzes a talkspurt (for example, TS1) and sends a control message to the sender to change speech encoding parameters, the next control message cannot be sent after the next consecutive talkspurt.(TS2); but only after analyzing of TS3, if it is required.

One more restriction is added. If a decision is made about several consecutive improvements of speech encoding parameters (for example, to decrease the voice payload size from 30 ms to 20 ms and then to 10 ms or to replace a given codec by a better codec) these changes should not be made too quickly because each change causes noticeable increase of IP-rate per call and thus, a higher probability of degradation due to congestion. The preliminary experiments showed that the system is more stable if the receiver waits for four talkspurts (about 2 seconds) between decisions (see the Figure below).

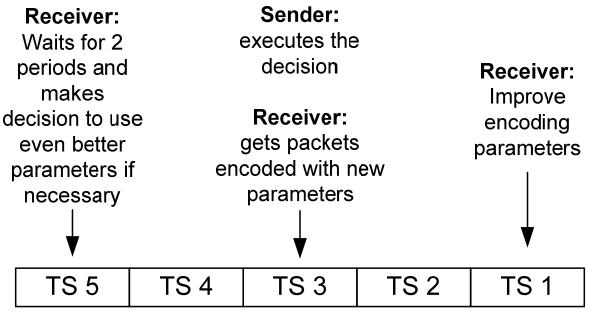


Figure 6-3: Elements of adaptive control mechanism

6.4 Simulation Results and Example

An example is given presenting the simulation study of the algorithm's performance. The example compares speech quality under the scenario in which the adaptive scheme is not used against quality under the scenario in which the adaptive speech quality management algorithm is implemented. This example considers a rather congested network with 40% of voice and 50% of data traffic. Both scenarios have the same background traffic pattern.

Figure 6.4 shows statistics without using the adaptive quality management scheme. This picture shows that, because of significant network congestion, the call periodically suffers from significant delay and loss (especially in the fist half on the analyzed period). The average MOS during the considered period is 3.55, which is a relatively low score (noticeably lower than the toll-grade level). There are intervals of time during perceptual (real) quality becomes even lower (close to 3.0), which is generally unacceptable.

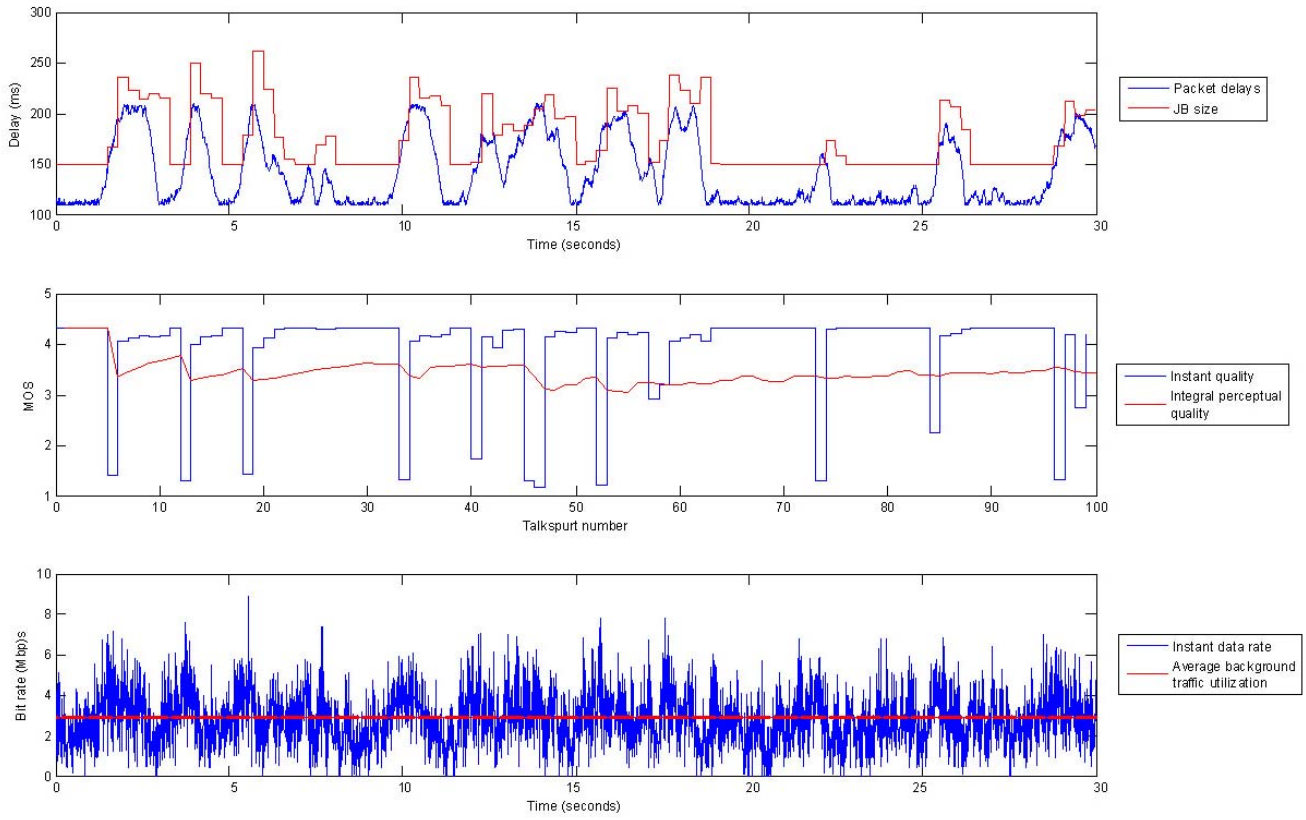


Figure 6-4: Speech quality without the adaptive algorithm

Figure 6.5 shows the result when exactly the same background traffic pattern is managed by the proposed adaptive quality management mechanism.

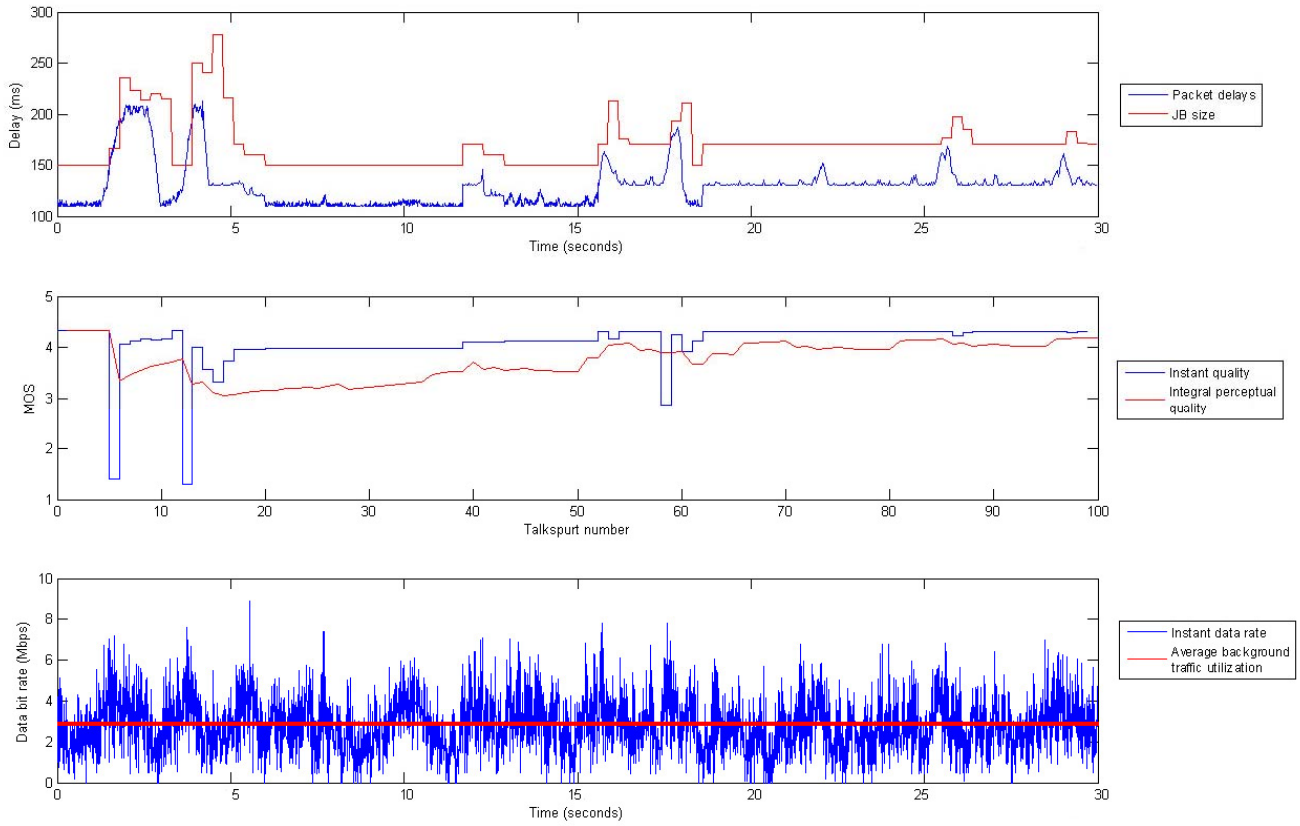


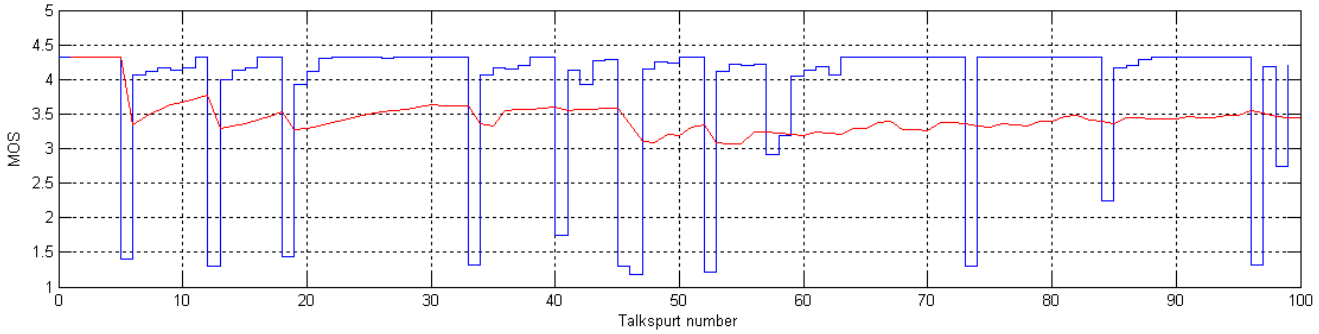
Figure 6-5: Speech quality with the adaptive algorithm

A noticeable quality improvement is seen. The quality characteristics of both calls are tabulated below.

	Mean delay (ms)	90% delay (ms)	Packet loss (%)	MOS perceptual
Without adaptive encoding	181.37	223.63	6.4	3.55
With adaptive scheme	169.99	204.54	2.3	3.86

Now, all decisions of the algorithm are analyzed step by step. The Figures below show instantaneous and perceptual quality levels in both scenarios.

Without adaptive encoding



With adaptive scheme

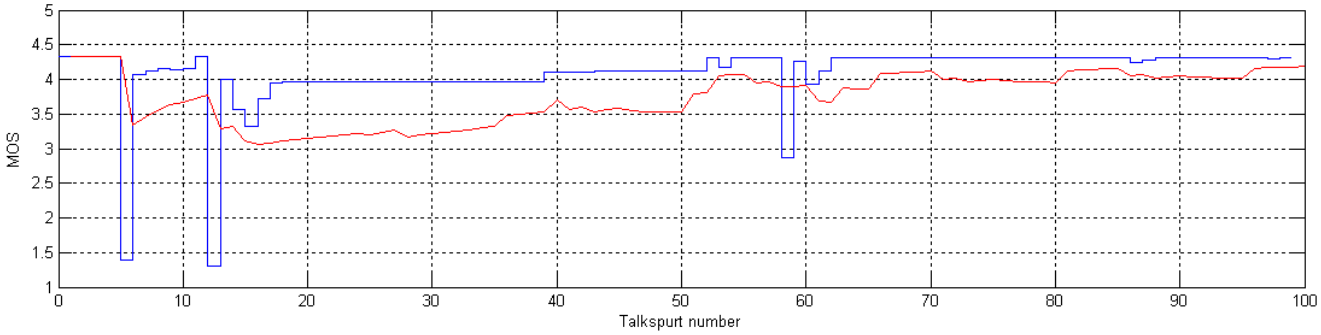


Figure 6-6: Speech quality comparison

Analysis:

Talkspurt #	Comments
1 - 10	Collect statistics. This dissertation suggests not changing any speech encoding parameters during this initial interval because quality measurement metrics are very sensitive to quality variations. In future work, quality behavior should be analyzed in this period in greater detail. If there are too many calls in the network, some response has to be sent from the receiver to the sender immediately based on analyzing this initial period.
11-12	The difference between max quality (MOS=4.3) and integral perceptual quality exceeds 0.5 MOS because of the bursty loss period during the initial 1-10 interval. But the instantaneous quality level is close to

	maximum and some improvement of the integral quality was seen. The encoding parameters are not changed, expecting even further improvement.
13	A very significant quality degradation occurred (a long bursty loss of voice packets). Instantaneous quality became almost minimal. Integral quality also became very low. Something has to be done immediately because, if one more bursty loss occurs after this, the overall call quality will be even more significantly affected and more difficult to recover. The algorithm switches to the G.729 codec (8:1 compression, 30 ms packet size). These new parameters will be adopted by the sender two talkspurts later (in TS 15).
15-16	The new encoding parameters are used for talkspurt #15, but there is no an immediate improvement of the situation. Instantaneous quality levels for these intervals are even lower as in the first scenario because a poorer codec is in use and there are still some "remnants" of congestion in the network.
17 - 20	Finally, a slow improvement of the integral quality is seen, so the algorithm "decides" to decrease the voice payload size from 30 ms to 20 ms and then to 10 ms. Integral quality is still low but it is already seen that the decision to change speech encoding parameters was correct: we avoided the next period of quality degradation during TS 18, was avoided (see the upper picture).
21-37	The situation in the network became stable because compressed codecs were used and several periods of speech quality degradation were avoided. No encoding parameters were changed during this period of time. Integral perceptual quality was relatively low, so the algorithm is waiting for its improvement.
38	The difference between the maximum quality of the G.729 codec and integral quality is less than 0.5 MOS and the instantaneous quality level is rather high (more than 4.0). So, the algorithm decides to use a better codec, expecting to improve integral quality even further. It switches to

	the G.726 codec with 30 ms packet size.
39-42	Significant degradations in quality are not seen. Notice that, even with these better parameters, the bursty loss in TS 41 was avoided. Packet size is slowly decreased up to 10 ms.
43-50	Do nothing. Wait until the integral quality level becomes close to the maximum quality achievable by the given codec (about 4.1 MOS).
35-39	Still use the G.726 codec. Because there was no more degradation packet size was decreased 30 ms to 20 ms and, then to 10 ms.
51-52	Switch to the better G.711 codec. These new parameters will be adopted by the sender in two talkspurts (is TS 53).
53-58	Do nothing: monitor slow improvement of integral quality.
59	Another degradation in quality is seen. Because a 30 ms packet size was used instead of 10 ms, the degradation is not as significant as in the first scenario (see TS 57-58 on the first picture). But this decrease in quality is an indicator that network is moderately congested, so anticipating a potentially serious drop in quality, the algorithm switches to the lower quality G.726 codec with higher compression.
60-…	Similar to previous steps…

In this simulated example, managing the long-term voice flow behavior caused an increase in quality equal to 0.3 MOS (comparing perceptual qualities). This improvement is noticeable. Toll-grade quality (MOS = 4.0) was not achieved, but the resulting quality (MOS = 3.86) is relatively close to it (at least compared to the unmanaged quality with MOS = 3.55). This example illustrates main principles of adaptive speech quality management. The example also demonstrates that reasonable threshold values were chosen to ensure the system's stability.

To prove the effectiveness of the algorithm, its performance must be analyzed under different scenarios and network conditions. This is very difficult because simulation results depend, not only on the proportion of voice and data traffic in the network, but also on the background traffic pattern (the same proportion of the background traffic in the network produces different call qualities depending on the positions of the data bursts). Also, it takes

very significant time and computational resources to simulate voice and data traffic in a 5 Mbps channel and to repeat this simulation hundreds of times for many sets of voice and data utilizations.

As demonstrated in Chapter 5, adaptive quality mechanisms can be efficient with at least 20% voice traffic in the network. With, for example, 20 or 30 percent of voice packets, quality degradation becomes noticeable when total link utilization exceeds 70%. But. this does not mean that instantaneous quality degradations will not be seen when the utilization is, for example, 60%. If the utilization is lower than 70%, quality degradations are possible but they are rare and the resulting average quality is close to the maximum level. With more voice packets in the network (for example 80%), speech quality degradation can be noticeable if total link utilization is close or higher than 90% (see Chapter 5). The range of parameters, where adaptive speech quality management can potentially be beneficial, is shown on Figure 6.7.

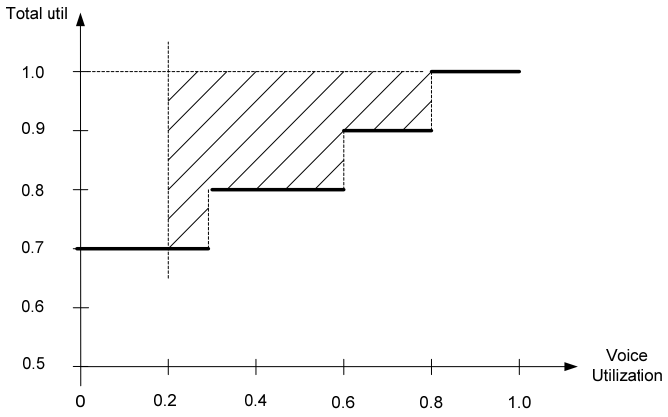


Figure 6-7: Speech quality comparison

It must be remembered that these numbers depend significantly on the effectiveness of the adaptive jitter buffer and on other assumptions; the shaded area will be larger if a less efficient jitter buffer scheme is used or because of other reasons. It is also possible to have an excessive number of calls in the network and greater than 100% total link utilization (measured assuming G.711 encoding), but this scenario is not analyzed in this dissertation, and is included it in the "Future work" Section.

Several pairs of voice and data loads were analyzed by running 20 simulations for each pair. For each pair, the voice qualities were compared without and with adaptation. Table 6.2 presents the algorithm's performance results.

Table 6-2: Simulation results

Voice load	Data load	Total link utilization	Number of VoIP calls	Mean MOS without adaptation	# experiments with quality improvement > 0.2	# experiments with quality improvement < 0.2	# experiments with quality decrease < 0.2	# experiments with quality decrease > 0.2
0.3	0.5	0.8	15	4.15	3	12	3	2
0.5	0.3	0.8	26	4.22	0	14	4	1
0.3	0.6	0.9	15	3.98	9	7	2	2
0.5	0.4	0.9	26	4.04	6	10	3	1
0.7	0.2	0.9	36	4.21	0	13	7	0
0.3	0.7	1.0	15	3.51	14	4	2	0
0.5	0.5	1.0	26	3.63	13	5	2	0
0.7	0.3	1.0	36	3.82	12	5	2	1
0.8	0.2	1.0	41	4.15	3	12	5	0

Simulation study was used to analyze efficiency of the proposed adaptive speech quality management algorithm. The difference in qualities measured with and without adaptive encoding, was calculated. The measurements were performed for different values of the total link utilization in the range from 80% to 100%. Different proportions of voice and data traffic in the network were analyzed and speech quality was calculated as the average of 20 simulation runs for each pair.

Table 6.2 presents the algorithm's performance results. For example, consider the simulation with 50% voice and 40% data in a channel. There were 20 experiments. Four of these 20 experiments got negative results: some quality management decisions were not effective and the resulting quality was even lower than the initial level. But 16 cases out of 20 got improvement in quality. In 6 cases of these 16, it was very noticeable and higher than 0.2 MOS. In other cases, it was in the range between 0 and 0.2 MOS. In some other scenarios, for example with 50% voice and 30% data in a channel, there are no improvements higher than 0.2 MOS. This is explained by the fact that the maximum quality

achievable by narrowband codecs is about 4.4. Again, all these numbers depend on multiple parameters and assumptions about traffic behavior in the Internet, used adaptive jitter buffer mechanism, proportion of voice and data traffic in the network, etc.

Looking at all these results, we can make several important conclusions about the algorithm's performance:

- The number of positive "guesses" of the algorithm is noticeably higher than the number of negative results. Future work on development of more efficient mechanism can potentially make this situation even better.
- The current version of the algorithm is especially efficient in highly congested networks (we get a noticeable improvement in quality in approximately 70% of experiments). In some situations, the improvement of quality was very significant (~ 0.5 MOS).
- The current version of the algorithm is especially efficient when there is a significant portion of data traffic in the network.
- Our adaptive quality management scheme is not always efficient. Situations occur when the adaptive scheme provides a lower quality than without the mechanism. This happens for two main reasons: 1) since future quality degradation is expected, compressed lower quality speech is used; but nothing happens. This is a relatively frequent situation when there is a significant portion of voice traffic in the network but data traffic still has some negative impact; 2) the algorithm makes a wrong decision using good codecs and gets a noticeable drop in quality. Apparently, it is possible to improve the "intelligence" of the algorithm to decrease the number of such failures.

This Chapter has demonstrated that adaptive quality management can be used in real networks where the situation in the network at a given moment of time, the proportion of background data traffic and many other factors are not known.

6.5 Potential Markets for the Technology

Several potential areas can be seen for adaptive speech encoding.

1. VoIP services providers. Making international call using a calling card, people usually dial some 1-800 number. Then, their call is converted to IP packets on a provider's facility and sent over the Internet. If there is an excessive number of calls, it is possible to manage encoding of the group of calls simultaneously and to find some balance between a number of calls in the network and their quality. Instead of dropping calls or suffering from a jitter or delay (and echo as a result...), it is possible to change a type of encoding dynamically.

2. Video conferencing can be used for demanding applications such as playing music together over networks, and other distributed multimedia. This application is very demanding with respect to the end-to-end delay and especially to audio quality. CD-quality audio requires a significant channel capacity (in addition to a video stream) and a probability of congestion in the network (and, as a result, delay and packet loss) becomes more significant than in the case of a traditional VoIP call or of standard video conferencing formats with higher latencies but efficient compression. There are algorithms to change a bit-rate of a video stream dynamically. The same thing can be potentially done with audio. For example, it would better to deliver a 22-kHz stream instead of a 44-kHz stream or to change a packet size from 10 ms to 20 ms (and thus to decrease IP-rate) than to get a noticeable delay and/or loss of packets.

3. Skype uses multiple bit-rates. A bit-rate is chosen in the beginning of a conversation depending on some criteria (the technology is proprietary) and is not changed during a session. Speaking about a single call, adaptive encoding will likely not provide any noticeable improvement in quality. For example, the G.711 call has 96 kbps IP-rate (64 kbps of audio and 32 kbps to send RTP/UDP/IP overhead). The G.729 call requires 40 kbps (8 kbps audio). The difference is 50 kbps and it is noticeable in the case of a dial-up access but not noticeable if a user has a high-speed connection. The situation with high-quality audio is different: CD audio quality requires noticeably more bandwidth (44 kHz sampling * 8 bits = ~ 350 kbps). Requirements to available channel capacity will be even higher in the case of high-quality multi-channel audio. Dynamic changes in quality will be more effective if an

access network is congested by some other applications. The algorithms can be implemented on the end-user software or on a provider's hardware.

6.6 Summary

In this Chapter, an adaptive control mechanism was designed to improve the average quality of VoIP communication. In this scheme, the receiver makes a control decision based on two parameters: 1) the computational instantaneous quality level, which is calculated per talkspurt using the E-model and 2) perceptual metrics, which estimate the integral speech quality by taking into account the fact that a decrease of communication quality depends, not only on the presence of packet delay or loss in the network, but also on the position of a quality degradation period in the call.

The algorithm works together with the adaptive jitter buffer mechanism. The adaptive jitter buffer is used to manage short-term quality; the sender-based adaptation technique tries to choose encoding parameters to improve a long-term quality by decreasing network congestion and, as a result, significant instantaneous changes in quality.

The algorithm uses four threshold parameters (two for instantaneous and two for integral quality level). Depending on these parameters, different mechanisms are used by a sender to manage speech quality. The last Section of this chapter analyzed an example and conducted a simulation study to estimate the algorithm’s performance.

Chapter 7

Conclusion and Future Work

7.1 Dissertation Summary and Contribution

This dissertation investigated the important problem of real-time voice-over-IP communications; specifically, how the mismatch between the VoIP service and the network may result in a significant degradation of voice quality. The packet-switched Internet does not provide reliable transport of real-time data; it does not usually guarantee the bandwidth, delay, and loss bounds that are important for real-time voice traffic. Current approaches like MPLS, IPv6, DiffServ, and other network-based QoS management schemes, try to resolve this problem but it still exists and is important. As an alternative approach, this project investigated the concept of sender-based adaptive speech quality management. Based on the analysis of different parameters and characteristics that affect speech quality, it is proposed to improve the average quality of VoIP communications by managing voice encoding parameters dynamically. The choice of optimal end-user parameters under given network conditions enhances the quality of VoIP because it changes the configuration of a VoIP system (a long-term behavior of a voice stream) so the system better matches the current state of the network.

The contributions of this dissertation are threefold. First, several important questions in the area of wideband telephony were investigated. A computational tool called the E-model was developed to measure speech quality level based on multiple codecs and network characteristics. This model is not especially good for measuring the absolute quality level, but is good for comparing the qualities of two speech samples, which is what is needed to track real-time changes in quality. This dissertation developed a similar model for wideband communications and proposed an approach to extend the traditional MOS scale and the E-model's R-scale to describe improvements in quality because of the higher signal frequency range used in the wideband telephony. Analysis described how multiple parameters from the

model change in the wideband case and a simple linear model for "non-standard" codecs is provided. So, we are no longer limited to narrowband codecs when we measure speech quality in real-time.

Second, this dissertation studies the impact of multiple speech encoding parameters on the quality of VoIP telephony. Theory and simulations showed that, if long-term total link utilization does not exceed 90-100%, changing packet size provides better resulting quality than changing encoding/compression despite the fact that a compressed voice stream uses less resources. It was also shown that, if the number of calls in the network increases, it is better to use a combination of packet size and compression variation together. The proper choice of speech encoding mechanism allows sending more calls through the network with less noticeable quality degradation. Furthermore, for a small number of calls in the presence of significant background traffic load, changing the packet size or encoding was shown to not improve communication quality. Wideband codecs can be used in this situation, except in the case of IP-to-PSTN communications.

Third, this dissertation included the design of an adaptive control mechanism to improve the average quality of VoIP communications. In this scheme, the receiver makes a control decision based on two parameters: 1) the computed instantaneous quality level, which is calculated per talkspurt using the E-model and 2) a perceptual metric, which estimates a weighted average speech quality taking into account the fact that decreased communications quality depends not only on the presence of packet delay or loss in the network but also on the position of the quality degradation in call interval. This algorithm works in conjunction with an adaptive jitter buffer mechanism such that the jitter buffer manages the short-term quality and the sender-based adaptation technique chooses encoding parameters to improve the long-term quality. While simulations showed positive results in most cases, additional work has to be done in this area, as discussed in the next section.

7.2 Future Research

There are several directions for our future research. First, it would be interesting to continue the theoretical work in the area of wideband telephony. This dissertation described how the

traditional narrowband 1-to-5 MOS scale could be extended to wideband communications. But, it would be useful to get more experimental results and to address some of the questions raised in the dissertation in more detail.

The studies in this dissertation are based on simulations, which were designed to thoroughly investigate multiple approaches to background traffic generation, adaptive jitter buffer management, etc. But, many assumptions were made that, while reasonable, they affected the final results. For example, since no papers were found that compared the efficiencies of multiple existing adaptive jitter buffer mechanisms, several were implemented, and coefficients were adjusted to chose the best scheme. Deploying poorer adaptive mechanisms should give a more noticeable quality degradation. Other assumptions were made regarding talk-spurt duration, background traffic pattern, and queuing. All these assumptions are reasonable and confirmed by multiple research studies, but it would be extremely useful to verify everything in real-world experiments.

This dissertation discussed multiple network-, sender-, and receiver-based quality management mechanisms. These multiple approaches were developed to improve the quality of real-time communications, but it is not clear which of these mechanisms is more efficient, and under which conditions. It necessary to compare multiple QoS management approaches, although a very significant research effort would be required. It would also be interesting to see how network-based QoS management technologies work with this adaptive algorithm.

An adaptive sender-based quality management mechanism was selected to demonstrate that adaptive speech quality can be managed in real networks even when the algorithm does not know the situation inside the network (for example, the portion of background data traffic) at any given moment of time. But, this question is very complex and requires a separate research project to estimate the effectiveness of dynamic speech encoding under different scenarios and conditions. It is also necessary to investigate each of the following scenarios in detail: a) IP-to-IP communication in which wideband telephony and the wideband E-model may be used, b) IP-to-PSTN communications that is limited to narrowband codecs only (the approach started here might be extended to a greater variety of codecs); and c) solving the problem of quality optimization for any random number of calls in the network (maximizing the quality of multiple simultaneous calls).

Bibliography

[1] ITU-T G.107, “The E-model, a computational model for use in transmission planning”, 2000

[2] ITU-T G.107 Amendment 1, “Provisional impairment factor framework for wideband speech transmission”, 2006

[3] ITU-T G.108.1, “Guidance for assessing conversational speech transmission quality effects not covered by the E-model”, 2000

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

[5] ITU-T G.114, “One-way transmission time”, 2003

[6] ITU-T G.711, “Pulse code modulation (PCM) of voice frequencies”, 1988

[7] ITU-T G.722, “7 kHz audio-coding within 64 kbit/s”, 1988

[8] ITU-T G.722.2, “G.722.2: Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB)”, 2003

[9] ITU-T G.726, “40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)”, 1990

[10] ITU-T G.729, “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)”, 1996

[11] ITU-T P.800, “Methods for subjective determination of transmission quality”, 1996

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

[13] ITU-T P.861, “Objective Quality measurement of telephone-band (300-3400 Hz) speech codecs”, 1996

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

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

[16] ITU-T P.862.3, “Application guide for objective quality measurement based on Recommendations P.862, P.862.1 and P. 862.2”, 2005

[17] 3GPP TR 26.975 v.6.0.0, “Performance characterization of the Adaptive Multi-Rate (AMR) speech codec”, 2004

[18] 3GPP TR 26.976 v.6.0.0, “AMR-WB Speech Codec Performance Characterization”, 2004

[19] RFC 3267, “Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codec”, 2002

[20] RFC 3031, “Multiprotocol Label Switching Architecture”, 2002

[21] RFC 2474, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, 1998

[22] RFC 2475, “An Architecture for Differentiated Service”, 1998

[23] RFC 2205, “Resource ReSerVation Protocol (RSVP)”, 1997

[24] RFC 1633, “Integrated Services in the Internet Architecture: an Overview”, 1994

[25] RFC 2309, “Recommendations on Queue Management and Congestion Avoidance in the Internet”, 1998

[26] 3GPP TS 26.071 v 7.0.0, “AMR Speech Codec, General Description”, 2007

[27] [Book] O. Hersent, J. Petit, D. Gurle, “IP Telephony. Packet-based multimedia communications system”, 2000, ISBN: 0201619105

[28] [Book] S. Walters, “The New Telephony, Technology convergence. Industry collision”, 2002, ISBN: 0130358142

[29] [Book] A. Raake, “Speech quality of VoIP. Assessment and Prediction”, 2006, ISBN: 978-0-470-03060-8

[30] [Book] O. Hersent, J. Petit, D. Gurle, “Beyond VoIP protocols. Understanding voice technology and networking techniques for IP telephony”, 2005, ISBN 0-470-02362-7

[31] [Book] S. Moller, “Assessment and Prediction of Speech Quality in Telecommunications”, Kluwer Academic Publ., US-Boston MA, 2000, ISBN 0-7923-7894-6

[32] [Book] D. Minoli, “Voice over MPLS. Planning and Designing Networks”, 2003, ISBN 0-07-140615-8

[33] R. Reynolds, A. Rix, “Quality VoIP – an engineering challenge”, BT Technology Journal, 2001

[34] R. Beuran, M. Ivanovichi, “User-Perceived Quality Assessment for VoIP Applications”, Technical Report, 2004

[35] A.W. Rix, J.G. Beerends, M.P. Hollier and A.P. Hekstra, “Perceptual Evaluation of Speech Quality (PESQ) – a new method for speech quality assessment of telephone networks and codecs”, IEEE ICASSP, 2001

[36] S. Pennock, “Accuracy of the perceptual evaluation of speech quality (PESQ) algorithm,” In Proc. Measurement of Speech and Audio Quality in Networks, Prague, Czech Republic, May 2002

[37] R. Cole and J. Rosenbluth, “Voice over IP performance monitoring,” Journal on Computer Communications Review, 2001

[38] A. Takahashi, H. Yoshino, “Perceptual QoS assessment technologies for VoIP”, IEEE Communications Magazine, July 2004

[39] J. Rodman, “The effect of bandwidth on speech intelligibility”, Polycom, 2006

[40] R. A. Thompson, "The Ugly Duckling - An Essay on Network Integration", Tutorial at IEEE Comsoc HPSR Workshop, Poznan, Poland, June 2006

[41] ITU-T SG12 D.139: “France Telecom study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences”, 2000

[42] J. H. Rosenbluth, “Testing the Quality of Connections having Time Varying Impairment”, Committee contribution T1A1.7/98-031, 1998

[43] A. D. Clark, “Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality”, 2003

[44] L. Gros, N. Chateau, “Instantaneous and Overall Judgements for Time-Varying Speech Quality: Assessments and Relationships, Acta Acustica, Volume 87, Number 3, May/June 2001, pp. 367-377(11)

[45] X. Tan, S. Wanstedt, G. Heikkila, “Experiments and modeling of perceived speech quality of long samples”, Ericsson Research, 2001

[46] M. Hollier, “An experimental investigation of the accumulation of perceived error in time-varying speech distortions”, ITU Study group 12 - Contribution 21, COM 12-21-E

[47] R. Salami *et al,* “The Adaptive Multirate Wideband Speech Codec (AMR-WB)”, IEEE Transactions on Speech and Audio Processing, vol. 10, no. 8, pp. 620-636, 2002

[48] J. Seo, S. Woo, K. Bae, “A study on the application of an AMR speech codec to VoIP”, 2001

[49] J. Matta, C.Pepin, K. Lashkari, R. Jain, “A source and channel arte adaptation algorithm for AMR in VoIP using the E-model”, NOSSDAV, pp. 92-99, 2003

[50] Y. Huang, J. Korhonen, Y. Wang, “Optimization of source and channel coding for voice over IP”, ICME, 2005

[51] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks”, in Proc. IEEE INFOCOM, 1994

[52] S. B. Moon, J. Kurose, and D. Towsley, “Packet audio playout delay adjustment: performance bounds and algorithms”, ACM/Springer Multimedia Systems, vol. 5, Jan. 1998

[53] M. Narbutt, L.Murphy, “A New VoIP Adaptive Playout Algorithm”, IEEE Trans. on Broadcasting, 2004

[54] M. Narbutt, L. Murphy, “VoIP Playout Buffer Adjustment using Adaptive Estimation of Network Delays”, Proceedings of the 18-th International Teletraffic Congress – ITC-18, Berlin, Germany, September 2003

[55] L. Atzori, M. Lobina, “Speech playout buffering based on simplified version of the ITU-T E-model,” IEEE Signal Process, vol. 11, no. 3, pp. 382-385, March 2004

[56] A. Shallwani, P. Kabal. “An adaptive playout algorithm with delay spike detection for real-time VoIP”, 2003

[57] S. Huang, P. Chang, E. Wu, “Adaptive voice smoothing with optimal E-model method for VoIP services “, 2006

[58] RFC 2460, “Internet Protocol, Version 6 (IPv6) Specification”, 1998

[59] “Internet Protocol version 6 (IPv6). Conformance and Performance Testing”, White paper, Ixia, 2004

[60] Cisco Systems, “Multiprotocol Label Switching Documentation”, http://www.cisco.com/univercd/home/home.htm

[61] “Performance of Virtualized MPLS Internet Infrastructure in Delivering VoIP Services “, Avaya, 2005

[62] Avaya, “Comparing MPLS and Internet Links for Delivering VoIP Services”, 2005

[63] Ditech Networks, “Voice Quality beyond IP QoS”, White Paper, January 2007

[64] Trond Ulseth, “A path toward common quality assessment of narrowband and wideband voice”, Telenor R&D, ETSI Workshop on Wideband Speech Quality in Terminals and Networks: Accessing and Prediction, 2004

[65] Alan Clark, “Tech Note: Voice Quality Measurement”, Telchemy, 2005

[66] Alan Clark, “Demystifying QoS”, Telchemy, 2006, www.telchemy.com/ Conferences /2006/Telchemy_ITexpo_2006_demystifying_QoS.pdf

[67] Sebastian Möller, Alexander Raake, Vincent Barriac, Catherine Quinquis, “Deriving Equipment Impairment Factors for Wideband Speech Codecs”, ETSI Workshop on Wideband Speech Quality in Terminals and Networks: Accessing and Prediction, 2004

[68] Vincent Barriac, Jean-Yves Le Saout, Catherine Lockwood, “Discussion on unified objective methodologies for the comparison of voice quality of narrowband and wideband scenarios”, France Telecom, R&D Division, 2004

[69] C. Hoene, S. Wietholter, “Simulating Playout Schedulers for VoIP-Software”, 2004

[70] Avaya, “Performance of Virtualized MPLS Internet Infrastructure in Delivering VoIP Services”, 2005

[71] Hiroyuki Oouchi, Tsuyoshi Takenaga, Hajime Sugawara and Masao Masugi, “Study on Appropriate Voice Data Length of IP Packets for VoIP Network Adjustment”, NTT Network Service Systems Laboratories, 2004

[72] L. Yamamoto, J. Beerends, “Impact of network performance parameters on the end-to-end perceived speech quality”, Expert ATM Traffic Symposium, Greece, 1997

[73] B. Ngamwongwattana, “Sync & Sense Enabled Adaptive Packetization VoIP”, PhD Dissertation, University of Pittsburgh, 2007

[74] B. Ngamwongwattana, R. Thompson, “Achieving adaptive rate Voice over IP through optimizing packetization”, ITERA conference, Las Vegas, 2006

[75] Z. Qiao, L. Sun, N. Heilemann and E. Ifeachor, “A New Method for VoIP Quality of Service Control Use Combined Adaptive Sender Rate and Priority Marking”, IEEE International Conference on Communications, 2004

[76] C. Mahlo, C. Hoene, A. Rostami, A. Wolisz, “Adaptive Coding and Packet Rates for TCP-Friendly VoIP Flows”, in Proc. of International Symposium on Telecommunications, Iran, 2005

[77] Jorg Widmer, Catherine Boutremans, Jean-Yves Le Boudec, "End-to-end Congestion Control for TCP-friendly Flows with Variable Packet Size", ACM SIGCOMM Computer Communications Review , Volume 34, Number 2: April 2004

[78] J. Bolot and A. Vega-Garcia, "Control Mechanisms for Packet Audio in the Internet", Proceedings IEEE Infocom, San Francisco, CA, pp 232-239, 1996

[79] S. Mohamed, F. Cervantes-Perez and H. Afifi, "Integrating Network Measurements and Speech Quality Subjective Scores for Control Purposes." IEEE Infocom, 2001

[80] L. Roychoudhuri, E. Al-Shaer, "Adaptive Rate Control for Real-time Packet Audio Based on Loss Prediction", Globecom, 2004

[81] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/

[82] Wenyu Jiang and Henning Schulzrinne, "Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss", *NOSSDAV'02,* Miami Beach, Florida, USA, 2002

[83] M. S. Taqqu, W. Willinger, and R. Sherman, "Proof of a Fundamental Result in Self-Similar Traffic Modeling," ACM Computer Communications Review, pp. 5 – 23, April 1997

[84] K. Claffy, G. J. Miller, K. Thompson, "The nature of the beast: recent traffic measurement from an Internet backbone," Proceedings of INET '98, Geneva, 1998

[85] K. Thompson, G. J. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," IEEE Network, 1997

[86] V. J. Ribeiro, M. Coates, R. H. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation", in Proc. of ITC Specialist Seminar on IP Traffic Measurement, September 2000

[87] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-range dependence: Ten years of Internet traffic modeling", IEEE Internet Computing, 2004

[88] Ericsson Consumer & Enterprise Lab, "Voice quality consumer trial", 2006, http://www.ericsson.com/technology/tech_articles/amr_files/presentation_voice_uality.pdf

[89] ITU-T P.59, "Telephone Transmission Quality Objective Measuring Apparatus: Artificial Conversational Speech," 1996

[90] W. Jiang and H. Schulzrinne, "Analysis of On-Off Patterns in VoIP and their Effect on Voice Traffic Aggregation," IEEE International Conference on Computer Communications and Networks (ICCCN), 2000

[91] P. T. Brady, “A Model for Generating ON-OFF Speech Patterns in Two-Way Conversations,” Bell Systems Technical Journal, vol. 48, pp. 2445–2472, 1969

[92] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, “Self-similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level”, Proceedings of the ACM/SIGCOMM'95, Cambridge, MA, 1995

[93] K. Park, G. Kim, M. Crovella, “On the relationship between file sizes, transport protocols, and self-similar network traffic”, Proc. IEEE International Conference on Network Protocols, 1996

[94] C. Williamson, “Internet traffic measurement”, Internet Computing, IEEE, 2001

[95] V. Abreu-Sernandez, C. Garcia-Mateo, “Adaptive multi-rate speech codec for VoIP transmission”, Electronic Letters, Volume 36, Number 23, November 2000

[96] Patent 7075981, Alan Clark, “Dynamic quality of service monitor”, Telchemy Inc., 2006

[97] J. Conti, “Avoiding the downhill”, Communication Engineer magazine, June-July 2004

[98] Laird Popkin (Pando Network), Doug Pasko (Verizon), “P4P: ISPs and P2P”, http://www.nanog.org/mtg-0802/presentations/PopkinPasko_Presentation.pdf

Appendix A: Simulation Source Code

```
%=========================================================================%

results=zeros();                    % Final arrays initialization
report=zeros();
t=1; m=1; tt=1;                     % Counters

U = 0.4;                             % Portion of voice traffic
D = 0.4;                            % Portion of data traffic

%=========================================================================%

% General parameters

sfrange = 4000;           % Signal frequency range in Hz (4000-8000 range)
cduration = 30000;        % Calls duration (ms)
pr_delay = 100;           % Network delay (ms)
Capacity = 5000/8;        % Outgoing link capacity in kilobytes per second
source_number = 10;              % Number of sources of data traffic
burst_rate=2*(Capacity*8*D)/source_number;
MaxQsize = Capacity*0.1*1000;    % Router queue size in bytes, 100 ms of
                                 queuing delay
IMPORT = 1;                      % 1 - import data traffic pattern flom
file
ALGORITHM = 1;                   % 0 - do not use; 1 - use
jitter_buffer=1;                 % 0 - fixed; 1 - variable (existing)
Qsize=0;                         % Router queue size initialization
Lost=0;                          % Initialization of a lost packet
indicator
count=0;
min_JB=40;                       % minimum JB size
dmin=0;
Talkspurt = 300;                 % Talkspurt duration (active speech)
Qmatrix=zeros();
TS_change=1;                     % last talkspurt when encoding parameters
                                 were changes; initialization
TS_array=zeros();
data_loss=0;                     % lost data traffic
change=0; W = 0; Product = 0;

%=========================================================================%

% Modeling of data traffic

if (IMPORT==0)          % generate traffic, not to use a previuos pattern

    psize1 = 40;        % 40 Bytes packets
    psize2 = 550;       % 550 Bytes packets
    psize3 = 1500;      % 1500 Bytes packets

    pport1 = 0.6;       % 60% of packets
```

```
    pport2 = 0.25;      % 25% of packets
    pport3 = 0.15;      % 15% of packets

    Pareto_a = 1.5;     % parameter of pareto distribution
    Pareto_b = 50 ;     % minimum burst length in milliseconds

    RU=0;               % average link utilizaton (calculated)

dtraffic=zeros();            % data traffic array initialization

while (abs(RU-D))>0.03       % difference between average and real
                                 utilization

      dtraffic=zeros();            % data traffic array initialization
      tnumber = 0;                 % sequence number in the mixed data
                                      stream
      i=1;

      for i=1:1:source_number
         timer = 0;                  % starting time of data traffic
                                        generation
         pnumber = 0;                % packet sequence number in a given
                                        data stream
         state=randsrc(1,1,[0,1;0.5,0.5]);  % initial state (1-burst; 0-
                                               idle)

         while (timer < cduration)
            if state==1
                 % generate traffic
                  burst_length=min(Pareto_b/(rand()^Pareto_a),1000);
                                                    % burst length in
                                                    ms
                 if (burst_length>cduration-timer)
                       burst_length=cduration-timer;
                 end
                 end_burst=timer+burst_length;
                                                     % end of burst
                                                    time
                 while (timer < end_burst) && (timer < cduration)
                     gp_size = randsrc(1,1,[psize1, psize2, psize3; pport1,
pport2, pport3]);
                     gp_duration = gp_size*8/burst_rate;
                     pnumber=pnumber+1;
                     tnumber=tnumber+1;
                     dtraffic (tnumber, 1) = timer;
                     dtraffic (tnumber, 2) = gp_size;
                     dtraffic (tnumber, 3) = 0;
                     timer=timer+gp_duration;
                  end
                    state=0;
               else
                   % adjust timer
                   idle_length=min(Pareto_b/rand()^Pareto_a,1000);
                                                % idle state length in
                                                ms
                      if (idle_length>cduration-timer)
```

```
                        idle_length=cduration-timer;
                    end
                end_idle=timer+idle_length;
                timer=timer+idle_length;
                state=1;
            end
        end
    end

    RU = sum(dtraffic(:,2))/1000/Capacity/(cduration/1000)

end

dtraffic=sortrows(dtraffic,1);
end

%=========================================================================%

% Initial encoding parameters and number of simulated calls

vpduration=10;
IE=0;
BPL=25;
vpcompression=1;
vpsize = (sfrange*2*vpduration/1000)/vpcompression+40; % Voice packet size
                                                        (bytes) + overheads
ncalls=
floor(8*Capacity*U/(((sfrange*2*10/1000)/vpcompression+40)*8*0.1));
%ncalls = 30;                   % Arbitrary number of calls

%=========================================================================%

% Modeling of VoIP traffic.

stime=sort(rand(ncalls, 1)* vpduration);    % Calls are not synchronized.
if (IMPORT==0) stime_tmp=stime;             % tmp array
else stime=stime_tmp;
end

TS = cduration/Talkspurt;
vtraffic_t=zeros();
inc_traffic_t=zeros();
out_traffic_t=zeros();
vouttraffic_t=zeros();
dest_delay_t=zeros();

for ts_number=1:1:TS
    Qmax=93.2-IE-3.6;
    NP = round(Talkspurt/vpduration);       % Number of packets per call
    vtraffic = zeros();                     % Voice traffic array

    for i=1:1:ncalls
        for j =1:1:NP
            vtraffic(j+(i-1)*NP, 1)= stime(i,1)+(j-1)*vpduration;
                                            % Packet generation time
```

```
            vtraffic(j+(i-1)*NP, 2)= vpsize;
                                                % Voice packet size
            vtraffic(j+(i-1)*NP, 3)= i;
                                                % Call number
        end
    end

stime=stime + Talkspurt;
vtraffic=sortrows(vtraffic,1);

if (ts_number==1)vtraffic_t=cat(vtraffic_t, vtraffic);
else vtraffic_t=cat(1, vtraffic_t, vtraffic); end

%=========================================================================%

%Mixed voice and data traffic

if (IMPORT==0)tmp_traffic=dtraffic; % tmp array
else dtraffic=tmp_traffic;
end
low=find(dtraffic(:,1)>=(ts_number-1)*Talkspurt,1,'first');
high=find(dtraffic(:,1)<ts_number*Talkspurt,1,'last');
inc_traffic=zeros();
inc_traffic=sortrows(cat(1,vtraffic,dtraffic(low:high,:)),1);
                                                % Total incoming traffic

if (ts_number==1)inc_traffic_t=cat(inc_traffic_t, inc_traffic);
else inc_traffic_t=cat(1, inc_traffic_t, inc_traffic); end

%=========================================================================%

% Router queue simulation

out_traffic=zeros();       % Initialization of a received packets array

if (ts_number==1 && count==0) SchDep = 0;  end

for count=1:1:size(inc_traffic,1)

   if (SchDep-inc_traffic(count,1) > (MaxQsize-inc_traffic(count,2)) /
Capacity)
       out_traffic (count,1)=inc_traffic(count,1);     % sender time
       out_traffic (count,2)=0;                        % packet is lost
       out_traffic (count,3)=0;                        % packet is lost
       out_traffic (count,4)=0;                        % packet is lost
       Lost=1;
   elseif (inc_traffic(count,1) >= SchDep)
       out_traffic (count,1)=inc_traffic(count,1);     % sender time
       out_traffic (count,2)=0;                        % waiting time
       out_traffic (count,3)=inc_traffic(count,2)/Capacity; % transm time
       out_traffic (count,4)=inc_traffic(count,1)+
inc_traffic(count,2)/Capacity;
                                                       % departure time
       SchDep=inc_traffic(count,1);
```

```
        Lost=0;
    else
        out_traffic (count,1)=inc_traffic(count,1);     % sender time
        out_traffic (count,2)=SchDep-inc_traffic(count,1);   % waiting time
        out_traffic (count,3)=inc_traffic(count,2)/Capacity; % transm time
        out_traffic (count,4)=SchDep+inc_traffic(count,2)/Capacity ;
        Lost=0;
    end

    out_traffic (count,5)=inc_traffic(count,2);   % packet size
    out_traffic (count,6)=inc_traffic(count,3);   % call number

    if (Lost==0)
         SchDep=SchDep+inc_traffic(count,2)/Capacity;
    end

end

if (ts_number==1)out_traffic_t=cat(out_traffic_t, out_traffic);
else out_traffic_t=cat(1, out_traffic_t, out_traffic); end

%=========================================================================%

% Output VoIP stream

vouttraffic=zeros(); % Array of received voice packets only

for K=1:1:1  % monitor one voice stream, several or all of them
K = 1; tt=1;
    for count=1:1:size(out_traffic,1)
        if (out_traffic(count,6)== K && out_traffic(count,6)>0)
            vouttraffic (tt,1)= out_traffic(count,1);  % sender time
            vouttraffic (tt,2)= out_traffic(count,2);  % waiting time
            vouttraffic (tt,3)= out_traffic(count,3);  % transm time
            if (out_traffic(count,4)>0)
                vouttraffic (tt,4)= out_traffic(count,4); % arrival time
without propagation and packetization
                vouttraffic (tt,5)= out_traffic(count,4) + pr_delay +
vpduration;  % arrival time
                vouttraffic (tt,6)= out_traffic(count,4) + pr_delay +
vpduration - out_traffic(count,1); % network delay (queuing + propagation)
            else
                vouttraffic (tt,4)= 0;
                vouttraffic (tt,5)= 0;
                vouttraffic (tt,6)= 500;
            end
            vouttraffic (tt,7)= out_traffic(count,6);
            tt=tt+1;
        end

        if (out_traffic(count,3)== 0 && out_traffic(count,6)==0)
            data_loss=data_loss+out_traffic(count,5);
        end

    end
```

```
    if (ts_number==1 && K==1) vouttraffic_t=cat(vouttraffic_t,
vouttraffic);
    else vouttraffic_t=cat(1, vouttraffic_t, vouttraffic); end

%=========================================================================%

%Jitter buffer simulation

dmin=pr_delay+vpduration;

dest_delay=zeros(); % delays of packets on a destination side (incl JB)

if (jitter_buffer==0)   % Fixed jitter buffer scenario (if required)
    counter=0; %counter
    for i = 1:1:size(vouttraffic,1)
        if vouttraffic(i,6)-dmin<=min_JB
            dest_delay(i,1)=vouttraffic(i,6); % delay before JB
            dest_delay(i,2)=dmin+min_JB;      % arrival time (incl JB)
            dest_delay(i,3)= 1;               % 1-arrived; 0-lost (incl
JB)
        else counter=counter+1;
            dest_delay(i,1)=vouttraffic(i,6);
            dest_delay(i,2)= dmin+min_JB;
            dest_delay(i,3)= 0;
        end
    dest_delay(i,4)= vouttraffic(i,7);
    end
end

% if (ts_number==1 && K==1) dest_delay_t=cat(dest_delay_t, dest_delay);
% else dest_delay_t=cat(1, dest_delay_t, dest_delay); end
% dest_delay_t=sortrows(dest_delay_t,4);

if (ts_number==1)
    mean_delay(K) = vouttraffic(1,6);        % cimulative delay
    mean_variance(K) = 0;     % cumulative variance
    counter(K)=0;
    JB_size(K)=min_JB;
end

if (jitter_buffer==1)      % variable jitter buffer; instant adaptation

    for i = 1:1:size(vouttraffic,1)

        ins_delay(i)=vouttraffic(i,6);

        if (ins_delay(i)< 500)
        mean_variance(K)=0.875*mean_variance(K)+0.125*abs(mean_delay(K)-
ins_delay(i));
        mean_delay(K)=0.875*mean_delay(K)+0.125*ins_delay(i);
        playout_time(K)=mean_delay(K)+4*mean_variance(K);
        end
```

```
        if vouttraffic(i,6)>dmin+JB_size(K)
                dest_delay(i,1)=vouttraffic(i,6);   % delay before JB
                dest_delay(i,2)=dmin+JB_size(K)+vouttraffic(i,3);
                dest_delay(i,3)=0;                  % 0-lost packet
                counter(K)=counter(K)+1;
        else
                dest_delay(i,1)=vouttraffic(i,6);   % delay before JB
                dest_delay(i,2)=dmin+JB_size(K)+vouttraffic(i,3);       %
delay including JB
                dest_delay(i,3)=1;                  % 1 - arrived packet
        end

        dest_delay(i,4)=mean_delay(K);
        dest_delay(i,5)=mean_variance(K);
        dest_delay(i,6)=playout_time(K);
        dest_delay(i,7)=vouttraffic(i,7);

    end

        JB_size(K)=max(playout_time(K), dmin+min_JB)-dmin;
end

if (ts_number==1 && K==1) dest_delay_t=cat(dest_delay_t, dest_delay);
else dest_delay_t=cat(1, dest_delay_t, dest_delay); end
dest_delay_t=sortrows(dest_delay_t,7);

JB_loss(K)=counter(K)/round(cduration/vpduration);
format('bank');

%=========================================================================%



% Computational quality model

% Instant quality (per talkspurt)
Qmatrix(ts_number,1)=ts_number;     % measurement period number
Qmatrix(ts_number,2)=dest_delay(1,2);    % delay during a measurement
period
Qmatrix(ts_number,3)=1-sum(dest_delay(:,3))/size(dest_delay,1);   % packet
loss rate per talkspurt
Qmatrix(ts_number,4)=0.024*Qmatrix(ts_number,2)+0.11*(max(Qmatrix(ts_numbe
r,2)-177.3,0));  % delay impairment
Qmatrix(ts_number,5)=IE+(95-
IE)*Qmatrix(ts_number,3)*100/((Qmatrix(ts_number,3)*100/(vpduration/10))+B
PL); % loss impairment
Qmatrix(ts_number,6)= 93.2-Qmatrix(ts_number,4)-Qmatrix(ts_number,5);
R=max(Qmatrix(ts_number,6), 6.5);
Qmatrix(ts_number,7)=1+0.035*R+R*(R-60)*(100-R)*7*10^(-6); % computational
quality

% Integral quality (average)
Delay=0.024*mean(Qmatrix(:,2))+0.11*(max(mean(Qmatrix(:,2))-177.3,0));  %
delay impairment
```

```
L=IE+(95-
IE)*mean(Qmatrix(:,3))*100/(mean((Qmatrix(:,3))*100/(vpduration/10))+BPL);
 % loss impairment
Q=93.2-Delay-L; R=max(Q, 6.5);
Qmatrix(ts_number,8)=1+0.035*R+R*(R-60)*(100-R)*7*10^(-6);

%=========================================================================%

% Perceptual quality model

% Model 1: integral perceptual model (AT&T)

if (rem(ts_number,15)<=10) sample = floor(ts_number/15);
else sample=ceil(ts_number/15);
end

W=zeros(); ave=zeros(); location=zeros(); Product=zeros();

if (sample==0 || sample==1)

Delay=0.024*mean(Qmatrix(1:ts_number,2))+0.11*(max(mean(Qmatrix(1:ts_numbe
r,2))-177.3,0));  % delay impairment
    L=IE+(95-
IE)*mean(Qmatrix(1:ts_number,3))*100/(mean(Qmatrix(1:ts_number,3))*100/(vp
duration/10)+BPL); % loss impairment
    Q=93.2-Delay-L; R=max(Q, 6.5);
    Qmatrix(ts_number,9)=1+0.035*R+R*(R-60)*(100-R)*7*10^(-6);
else

    for i=1:1:sample
            Delay=0.024*mean(Qmatrix(max(ts_number-i*15+1,1):ts_number-(i-
1)*15,2))+0.11*(max(mean(Qmatrix(max(ts_number-i*15+1,1):ts_number-(i-
1)*15,2))-177.3,0));  % delay impairment
            L=IE+(95-IE)*mean(Qmatrix(max(ts_number-i*15+1,1):ts_number-
(i-1)*15,3))*100/(mean(Qmatrix(max(ts_number-i*15+1,1):ts_number-(i-
1)*15,3))*100/(vpduration/10)+BPL); % loss impairment
            Q=93.2-Delay-L; R=max(Q, 6.5);
            ave(i)=1+0.035*R+R*(R-60)*(100-R)*7*10^(-6);
        if (i<sample)
            location(i)=1-15*(i-0.5)/ts_number;
        else
            location(i)=0.5*(1-15*(i-1)/ts_number);
        end

        W(i)=max(1,1+(0.038+1.3*location(i)^0.68)*(4.33-
ave(i))^(0.96+0.61*location(i)^1.2));
        Product(i)=W(i)*ave(i);

    end

Qmatrix(ts_number,9)=sum(Product)/sum(W);   % perceptual quality

end
end
```

```
if (ALGORITHM > 0)

    threshold1=0.2;
    threshold2=0.5;
    threshold3=0.3;
    threshold4=1.0;

    MOSmax=max(4.0, 1+0.035*Qmax+Qmax*(Qmax-60)*(100-Qmax)*7*10^(-6));

    Qinst=Qmatrix(ts_number,7);
    Qtotal=Qmatrix(ts_number,9);

        TS_array(ts_number,1)=ts_number;
        TS_array(ts_number,2)=1+0.035*Qmax+Qmax*(Qmax-60)*(100-
Qmax)*7*10^(-6);
        TS_array(ts_number,3)=MOSmax-Qtotal;
        TS_array(ts_number,4)=MOSmax-Qinst;


    if (ts_number >10 && change==0)

        if MOSmax-Qtotal<=threshold1

            if MOSmax-Qinst <=threshold3

                % better codec or decrease packet size
                TS_array(ts_number,5)=11;

                if vpduration > 10
                    vpduration_new = vpduration - 10; IE_new=IE;
BPL_new=BPL; vpcompression_new=vpcompression;
                    sprintf('TS number = %.0f; Action = 11; Change PS
from %.0f to %.0f; Codec IE = %.0f\', ts_number, vpduration,
vpduration_new, IE_new)
                elseif (IE==7)
                    vpduration_new=30; IE_new=0; BPL_new=25;
vpcompression_new=1;
                    sprintf('TS number = %.0f; Action = 11; Change codec
from G.726 to G.711, 30 ms packet size', ts_number)
                elseif (IE==11)
                    vpduration_new=30; IE_new=7; BPL_new=23;
vpcompression_new=2;
                    sprintf('TS number = %.0f; Action = 11; Change codec
from G.729 to G.726, 30 ms packet size', ts_number)

                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;

                end

            elseif MOSmax-Qinst <=threshold4

                % do nothing; expect that this is a temporary degradation
```

```
                TS_array(ts_number,5)=12;

            else

                % increase packet size
                TS_array(ts_number,5)=13;

                if vpduration < 30
                    vpduration_new = vpduration + 10; IE_new=IE;
BPL_new=BPL; vpcompression_new=vpcomression;
                    sprintf('TS number = %.0f; Action = 13; Change PS
from %.0f to %.0f. Codec IE = %.0f.', ts_number, vpduration,
vpduration_new, IE_new)
                end

                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;

            end

        elseif MOSmax-Qtotal<=threshold2

            if MOSmax-Qinst <=threshold3

                % do nothing; expect further improvement
                TS_array(ts_number,5)=21;

                if (IE==7 && vpduration==10)
                    vpduration_new = 30; IE_new=0; BPL_new=25;
vpcompression_new=1;
                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
                elseif (IE==11 && vpduration==10)
                    vpduration_new = 30; IE_new=7; BPL_new=23;
vpcompression_new=2;
                    sprintf('Change codec to G.726, 30 ms packet size. TS
number = %.0f', ts_number)
                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
                elseif (IE==11 && vpduration>10)
                    vpduration_new = vpduration - 10; IE_new=IE;
BPL_new=BPL; vpcompression_new=vpcompression;
               change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
                elseif (IE==7 && vpduration>10)
                    vpduration_new = vpduration - 10; IE_new=IE;
BPL_new=BPL; vpcompression_new=vpcompression;
                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
                end
```

```
            elseif MOSmax-Qinst <=threshold4

                % increase packet size
                TS_array(ts_number,5)=22;

                if vpduration < 30
                    vpduration_new = vpduration + 10; IE_new=IE;
BPL_new=BPL; vpcompression_new=vpcompression;
                    sprintf('TS number = %.0f; Action = 22; Change PS
from %.0f to %.0f. Codec IE = %.0f.', ts_number, vpduration,
vpduration_new, IE_new)

                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
                end


            else

                % change codec
                TS_array(ts_number,5)=23;

                if (IE==0)
                    vpduration_new=10; IE_new=7; BPL_new=23;
vpcompression_new=2;
                    sprintf('TS number = %.0f; Action = 23; Change codec
to G.726, 10 ms packet size', ts_number)
                end
                if (IE==7)
                    vpduration_new=10; IE_new=11; BPL_new=19;
vpcompression_new=8;
                    sprintf('TS number = %.0f; Action = 23; Change codec
to G.729, 10 ms packet size', ts_number)
                end

                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
            end

        else

            if MOSmax-Qinst <=threshold3

                % increase packet size up to maximum
                TS_array(ts_number,5)=31;

                if (vpduration == 30 || vpduration==20)
                    vpduration_new=max(vpduration-10, 10); IE_new=IE;
BPL_new=BPL; vpcompression_new=vpcompression;
```

```
                    sprintf('TS number = %.0f; Action = 31; Change PS
from %.0f to 30 ms. Codec IE = %.0f. ', ts_number, vpduration, IE_new)

                change=1;
                TS_change=ts_number;
                TS_array(ts_number,6)=1;
                end

            else

                % change codec
                    TS_array(ts_number,5)=33;
                    if (vpduration <30 || IE<11)
                    vpduration_new=30; IE_new=11; BPL_new=19;
vpcompression_new=8;
                    sprintf('S number = %.0f; Action = 33; Change codec to
G.729, 30 ms packet size', ts_number)

                    change=1;
                    TS_change=ts_number;
                    TS_array(ts_number,6)=1;
                    end
            end
        end
    end

end


if (change==1 && ts_number==TS_change+1)
    vpduration=vpduration_new;
    vpcompression=vpcompression_new;
    BPL=BPL_new; IE=IE_new;
    vpsize = (sfrange*2*vpduration_new/1000)/vpcompression_new+40;
    change=0;
end

end

%=========================================================================%

% Statistics generation and analysis

Mean_delay=mean(Qmatrix(:,2));
Pr_delay=prctile(Qmatrix(:,2),90);
Mean_loss=mean(Qmatrix(:,3));
Mean_MOS=Qmatrix(TS,8);
Percept_MOS=mean(Qmatrix(:,9));

Delay=0.024*Pr_delay+0.11*(max(Pr_delay-177.3,0));  % delay impairment
R=max(Qmax-Delay-L, 6.5);
Pr_MOS=1+0.035*R+R*(R-60)*(100-R)*7*10^(-6);

tr=zeros(cduration/10,2);
```

```
for X=1:1:cduration/10
tr(X,1)=X;
end

for P=1:1:size(dtraffic,1)
N=floor(dtraffic(P,1)/10)+1;
tr(N,2)=tr(N,2)+dtraffic(P,2)*8/10/1000;
end


format('short');
sprintf('Mean delay %.2f; \t VoIP loss %.3f;\t Data loss %.3f;\t Mean
MOS %.3f;\t Percept MOS %.3f;\n',  Mean_delay, Mean_loss,
data_loss/(Capacity*RU*cduration), Mean_MOS, Percept_MOS)

subplot(3,1,1); plot(dest_delay_t(:,8), dest_delay_t(:,1)); hold on;
plot(dest_delay_t(:,8), dest_delay_t(:,2), 'r'); hold off;
subplot(3,1,2); stairs(0:size(Qmatrix(:,7))-1,Qmatrix(:,7)); hold on;
plot(Qmatrix(:,9), 'r'); hold off;
subplot(3,1,3); plot(tr(:,1)*10, tr(:,2), 'b', tr(:,1)*10,
Capacity*8*D/1000, 'r');

results(t,1)=U;
results(t,2)=D;
results(t,3)=Mean_delay;
results(t,4)=Pr_delay;
results(t,5)=Mean_loss;
results(t,6)=Mean_MOS;
results(t,7)=Pr_MOS;
results(t,8)=Percept_MOS;
t=t+1;
```