Parametric Prediction Model for Perceived Voice Quality in Secure VoIP

Martin Andersson
Master of Science Thesis in Information Coding

Parametric Prediction Model for Perceived Voice Quality in Secure VoIP

Martin Andersson

LiTH-ISY-EX--16/4940--SE

Supervisor:  Johan Uppman
Sectra Communications
Jonthan Jogenfors
ISY, Linköping University

Examiner:  Jan-Åke Larsson
ISY, Linköping University

Division of Information Coding
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Copyright © 2016 Martin Andersson
Intentionally left empty!
Abstract

More and more sensitive information is communicated digitally and with that comes the demand for security and privacy on the services being used. An accurate QoS metric for these services are of interest both for the customer and the service provider. This thesis has investigated the impact of different parameters on the perceived voice quality for encrypted VoIP using a PESQ score as reference value. Based on this investigation a parametric prediction model has been developed which outputs a R-value, comparable to that of the widely used E-model from ITU. This thesis can further be seen as a template for how to construct models of other equipments or codecs than those evaluated here since they effect the result but are hard to parametrise.

The results of the investigation are consistent with previous studies regarding the impact of packet loss, the impact of jitter is shown to be significant over 40 ms. The results from three different packetizers are presented which illustrates the need to take such aspects into consideration when constructing a model to predict voice quality. The model derived from the investigation performs well with no mean error and a standard deviation of the error of a mere 1.45 R-value units when validated in conditions to be expected in GSM networks. When validated against an emulated 3G network the standard deviation is even lower.
Acknowledgments

First and foremost I would like to thank Sectra Communications for giving me the opportunity to work within an interesting and highly relevant field given today’s need for secure communication. The work environment has been stimulating and everyone I have been in contact with have been most helpful. I would like to especially thank my supervisor, Johan Uppman, for his interest and help throughout the project. From helping me with all practical matters to providing me with valuable feedback during the project, thank you.

For their comments and helpful advice, I would like to extend my gratitude to my examiner Jan-Åke Larsson and my supervisor Jonathan Jogenfors at Linköping University.

Many thanks to my course mates who have made the time at the University a pleasure, from long hours at campus to all the fun we have had.

Finally, I would like to thank my family and my girlfriend for their continuous support in all matters. Without them this would not have been possible.

Linköping, June 2016
Martin Andersson
Contents

Notation xiii

1 Introduction 1
  1.1 Motivation ........................................ 1
  1.2 Purpose ........................................ 2
  1.3 Problem Formulation .............................. 2
  1.4 Delimitations ..................................... 2
  1.5 Thesis Outline ................................... 3

2 Voice over IP 5
  2.1 VoIP Protocol Architecture ....................... 5
    2.1.1 Transport Control Protocol .................. 6
    2.1.2 User Datagram Protocol ....................... 6
    2.1.3 Datagram Transport Layer Security ............ 6
    2.1.4 Real-time Transport Protocol ................. 6
    2.1.5 RTP Control Protocol ......................... 6
    2.1.6 Session Initiation Protocol ................... 7
  2.2 VoIP System Structure ............................ 8
    2.2.1 Voice-codecs ................................ 8
    2.2.2 Packetizer .................................. 9
    2.2.3 Jitter buffer ................................ 9
  2.3 Quality of Service Metrics for VoIP .............. 10
    2.3.1 E-model .................................... 10
    2.3.2 Perceptual Evaluation of Speech Quality ...... 11
    2.3.3 Mean Opinion Score ........................... 11
    2.3.4 Conversion Formulas between Metrics .......... 11
  2.4 Prediction Models for VoIP QoS ................... 12
    2.4.1 Regression Based ................................ 13
    2.4.2 Neural Networks ............................... 13

3 Secure Voice over IP 15
  3.1 Cryptography .................................... 15
  3.2 Encryption ...................................... 16
### Contents

3.2.1 Mode of Operation ........................................... 16  
3.3 Secure Communication Interoperability Protocol ............... 18  
3.3.1 Cryptographic Synchronisation ............................... 18  
3.3.2 Packetization .................................................. 19

4 Related Work 21  
4.1 VoIP Quality Prediction ......................................... 21  
4.2 Encrypted VoIP Performance Analysis .......................... 22

5 Method 23  
5.1 Test Setup ....................................................... 23  
5.1.1 Virtual Machines .............................................. 23  
5.1.2 Overview ...................................................... 24  
5.2 Workflow to Perform a Measurement ............................ 26  
5.3 Initial Calibration and Setup Validation ......................... 26  
5.3.1 Network Characteristics ..................................... 26  
5.3.2 Call Duration .................................................. 26  
5.3.3 R-value Baseline .............................................. 27  
5.4 Test Runs .......................................................... 27  
5.4.1 Collected Metrics .............................................. 27  
5.4.2 Test Configurations .......................................... 28  
5.5 Model Development ............................................... 29

6 Results 31  
6.1 Setup Validation and Calibration Results ....................... 31  
6.1.1 Network Characteristics ..................................... 31  
6.1.2 Test Call Duration ............................................ 31  
6.1.3 R-value Baseline .............................................. 32  
6.1.4 Voice Dependency ............................................ 33  
6.2 Test Runs .......................................................... 33  
6.2.1 Loss .............................................................. 33  
6.2.2 Jitter .............................................................. 34  
6.2.3 Delay ............................................................. 34  
6.2.4 Packetizer ....................................................... 34  
6.3 Additivity ........................................................... 35  
6.4 Derived Model ..................................................... 35  
6.5 Validation .......................................................... 36  
6.6 Figures ............................................................... 37

7 Discussion 45  
7.1 Setup Validation and Calibration ................................ 45  
7.1.1 Difference between Configuration and Results .............. 45  
7.2 Loss ................................................................. 46  
7.3 Jitter ............................................................... 46  
7.4 Delay ............................................................... 46  
7.5 Packetizer .......................................................... 47  
7.6 Additivity .......................................................... 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.7 Validation</td>
<td>48</td>
</tr>
<tr>
<td>7.8 Method Criticism</td>
<td>49</td>
</tr>
<tr>
<td>7.8.1 PESQ as Reference Value</td>
<td>49</td>
</tr>
<tr>
<td>8 Conclusion</td>
<td>51</td>
</tr>
<tr>
<td>8.1 Further work</td>
<td>53</td>
</tr>
<tr>
<td>A Network Emulator Configurations</td>
<td>57</td>
</tr>
<tr>
<td>Bibliography</td>
<td>61</td>
</tr>
</tbody>
</table>
## Notation

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBC</td>
<td>Cipher Block Chaining (Mode of operation)</td>
</tr>
<tr>
<td>CTR</td>
<td>Counter (Mode of operation)</td>
</tr>
<tr>
<td>CS-ACELP</td>
<td>Conjugate Structure Algebraic Code-Excited Linear Prediction</td>
</tr>
<tr>
<td>DTLS</td>
<td>Datagram Transport Layer Security</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hashed Message Authentication Code</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IPsec</td>
<td>Internet Protocol Security</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>IV</td>
<td>Initialisation Vector</td>
</tr>
<tr>
<td>kbps</td>
<td>kilobits per second</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MELPe</td>
<td>Enhanced Mixed-Excitation Linear Prediction</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>PESQ</td>
<td>Perceptual Evaluation of Speech Quality</td>
</tr>
<tr>
<td>POLQA</td>
<td>Perceptual Objective Listening Quality Assessment</td>
</tr>
<tr>
<td>QoE</td>
<td>Quality of Experience</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RTP</td>
<td>Real-time Transport Protocol</td>
</tr>
<tr>
<td>RTT</td>
<td>Round-Trip Time</td>
</tr>
<tr>
<td>RTCP</td>
<td>RTP Control Protocol</td>
</tr>
<tr>
<td>SCIP</td>
<td>Secure Communications Interoperability Protocol</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SRTP</td>
<td>Secure Real-time Transport Protocol</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>VQT</td>
<td>Voice Quality Testing</td>
</tr>
</tbody>
</table>
The following chapter gives an introduction to the topic of the thesis, a motivation of its relevance and states the questions to be answered. The chapter concludes with an outline of the thesis.

1.1 Motivation

Today’s modern society is highly dependent on several communication services. On a daily basis most of us retrieve information online, correspond in different forms of text messages and communicate through speech over the cellular network. The globalisation has led to Voice over IP (VoIP) becoming increasingly more popular with Skype as one of the most widely used applications. All of these services either come with a promised Quality of Service (QoS), e.g. the speed of a broadband or coverage for cellphones, or an expected Quality of Experience (QoE), e.g. a text message should not take more than a few seconds to reach the receiver and VoIP calls should be comprehensible. (QoS is often used as a wider concept which includes QoE and that is how it is used in this thesis.) From a business perspective it is therefore crucial to be able to show the user what level of QoS they can expect. Measuring or predicting a QoS metric can however pose a major challenge, as is the case for VoIP.

More and more sensitive information is communicated digitally and with that comes the demand for security and privacy on the services being used. Encryption provides privacy but introduces complexities to the system, such as measuring and predicting a QoS metric. When producing a QoS metric for encrypted signals a parametric model must be used since no access is given to the original nor the resulting signal, at least not in cleartext.
An accurate QoS metric is not only of interest for customers, it is at least as useful for the service provider. The metric can be used to efficiently evaluate design choices and can even be used to configure the service when in use, in order to maximise the QoS provided.

Previous work within this area can be divided into two parts. The first part relates to predicting VoIP quality for a transmission when no encryption is used. The second part have analysed the performance of encrypted VoIP traffic mainly regarding different protocols used. To the best of my knowledge this thesis is the first work done where the two parts are combined to predict VoIP quality for an encrypted transmission.

1.2 Purpose

The purpose of this thesis is to investigate the impact of different parameters on the perceived audio quality for encrypted VoIP. The goal is to develop a parametric model, based on the investigation, able to predict a QoS metric for encrypted VoIP.

1.3 Problem Formulation

The following questions constitute the thesis’ problem formulation:

- How can the perceived audio quality of an encrypted VoIP call be predicted based on link statistics and information about used encryption schemes and codecs?
  - Which available parameters affect the perceived audio quality and to what extent?
  - How can the prediction above be related to widely applied QoS metrics in industry?

1.4 Delimitations

To fully validate predicted perception of a voice call, subjective testing should be undertaken. However, since subjective testing is highly resource demanding, both in terms of time and manpower, this work is limited to objective validation methods.

Another delimitation is that the network, over which the quality will be predicted, only will be simulated. Performing real tests in different kinds of networks would be of great interest but would require additional resources such as hardware and accounts for an encrypted VoIP service.

VoIP consists of a collection of technologies and protocols and evaluating them all is simply not feasible. The work of this thesis is hence limited to the configu-
ration described in Chapter 2 and Chapter 3 with focus on the generic aspects of encrypted VoIP systems.

1.5 Thesis Outline

Relevant background information about VoIP, applicable to both cleartext and encrypted applications, is introduced in Chapter 2 in order to enable the reader to comprehend the investigation and the discussion to follow. Aspects of encrypted VoIP traffic, in particular those affecting QoS when the network introduces impairments, are given in Chapter 3. Previous work within the field is discussed in Chapter 4 together with how it relates to this thesis. In Chapter 5 the work undertaken is described and the results thereof are presented in Chapter 6, followed by a discussion in Chapter 7. The thesis is concluded in Chapter 8 by answering the problem formulation and discussing further work.
This chapter provides general background information about VoIP relevant for the rest of the thesis. Protocols, codecs and different QoS metrics are presented here while aspects regarding securing VoIP traffic are discussed in Chapter 3.

2.1 VoIP Protocol Architecture

VoIP consists of a collection of technologies and protocols that together enables communication services over the Internet rather than over the publicly switched telephone network as for conventional voice calls.

The relevant protocols are depicted in Figure 2.1 and described in the sections to follow. The network layer and below are of less interest in this thesis and are therefore not described in any detail. The key feature to regard is that the IP protocol adds a header of 20 bytes [1] and that the link layer adds additional overhead, 18 bytes for a MAC header if Ethernet is used [2].

![Figure 2.1: Protocols relevant in the VoIP architecture and mapping to their respective layers in the Internet protocol suite.](image-url)
2.1.1 Transport Control Protocol

The Transport Control Protocol (TCP) [3] was originally specified in 1981 and is one of the core protocols upon which Internet communication relies. It is connection-oriented and designed to provide reliability and flow-control resulting in that the data is accurately delivered and in-order, assuming the two entities have a functioning network connection between them. The reliability comes at the price of additional overhead and increased latency since a connection must be established and maintained.

2.1.2 User Datagram Protocol

The User Datagram Protocol (UDP) [4] and TCP are the two main protocols for the transport layer in the Internet protocol suite. Being connectionless, UDP differs fundamentally from TCP in what it provides. Data may be delivered out-of-order, duplicated or even lost but since no connection is required to be established or maintained the latency is greatly reduced compared to TCP. Hence UDP is suitable for real-time data where low latency is of greater importance than complete delivery. Voice communication is such an example since minimal delay is more desirable than completely accurate recreation of the audio signal, as long as it is comprehensible.

2.1.3 Datagram Transport Layer Security

The Datagram Transport Layer Security (DTLS) protocol is specified by the Internet Engineering Task Force (IETF) in [5]. It is designed to provide integrity, authentication and confidentiality for the UDP protocol.

The use of the DTLS protocol is optional for VoIP traffic. The purpose of it is to create a Virtual Private Network (VPN) between the two clients. Using a VPN connection prevents the traffic from being classified as VoIP traffic en route and thereby potentially sniffed or blocked by various parties. On the other hand, the prioritisation of VoIP traffic that some network equipment provides can not be utilised if VPN is enabled. Note that a VPN can be created using other protocols as well, such as IPsec [6] or TLS [7]. The main point is that it is possible, and sometimes necessary, to hide the VoIP traffic and doing so adds an overhead.

2.1.4 Real-time Transport Protocol

When transmitting real-time data over a network, the Real-time Transport Protocol (RTP) [8] is typically used. The protocol provides end-to-end network functions for streaming media and can run both on TCP and UDP but is in practice only used with UDP since it is more suitable for real-time data. RTP itself does not provide any mechanisms for QoS, that is instead handled by its sister protocol RTCP, see Section 2.1.5.

2.1.5 RTP Control Protocol

The RTP Control Protocol (RTCP) is defined together with RTP in [8]. RTCP is used to monitor the delivery of data by providing transmission statistics, and by
doing so QoS, while all the payload data is carried by RTP. The RTCP packets of interest for this thesis is the sender and receiver reports and in particular their fields presented below. For the full specification of RTCP please consult [8].

- **fraction lost**: The fraction of RTP data packets lost since the last RTCP report.
- **cumulative number of packets lost**: The total number of RTP data packets that have been lost during the session.
- **interarrival jitter**: A metric for the statistical variance of the interarrival time for RTP data packets. As defined in Definition 2.1, the interarrival jitter $J$ is the mean deviation of the difference $D$ in packet spacing. The jitter value $J$ is sampled every time a RTCP report is issued.

**Definition 2.1 (Interarrival jitter).** Let $S_i$ be the RTP timestamp for packet $i$ and $R_i$ be the arrival time in corresponding time units for packet $i$. The difference $D$ in packet spacing between packet $i$ and $j$ can then be expressed as

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

The interarrival jitter $J$ is calculated continuously in order of arrival (not necessarily in sequence) as RTP data packet $i$ is received. The difference $D$ between that and the previous packet yields the interarrival jitter as

$$J(i) = J(i-1) + \frac{|D(i-1, i)| - J(i-1)}{16}$$

- **timestamp**: The time when the report was sent. Used together with the timestamp of a returned packet, a round trip time (RTT) can be calculated. The interval between RTCP reports is implementation specific but the standard provides the following recommendations [8]:
  - RTCP packets should constitute 5% of the session bandwidth.
  - If a fixed minimal interval is used, 5 seconds is recommended .
  - If a reduced minimum interval is used then it should be set to 360 divided by the session bandwidth in kbps. For bandwidths greater then 72 kbps this results in intervals smaller than 5 seconds.

### 2.1.6 Session Initiation Protocol

The Session Initiation Protocol (SIP) [9] is used to establish, modify and terminate multimedia sessions, VoIP being one of the main applications. For the purpose of this thesis no further details of SIP is warranted since focus is on payload handling and QoS rather than session handling, the interested reader is referred to [9].
2.2 VoIP System Structure

At a high level a VoIP system consists of three main parts - a sender, an IP network and a receiver, all depicted in Figure 2.2. A voice-codec in the sender digitalizes and compresses the received voice stream into speech frames, see Section 2.2.1. To not congest the network, several speech frames are then packetized to form the payload of a packet (e.g. RTP packet) and headers required by the network are added, see Section 2.2.2. The network may then introduce different impairments such as packet loss, delay and jitter before the packet is delivered to the receiver. The packets are there stripped of their headers and the speech frames are extracted by the depacketizer. A buffer is used to counteract the jitter introduced by the network at the cost of additional delay. Finally the speech frames are decoded and outputted, potentially with the use of packet loss concealment to compensate for lost packets. [10]

![Figure 2.2: Conceptual diagram of a VoIP system.](image)

2.2.1 Voice-codecs

There exists a variety of voice-codecs, whose functions are to convert analog voice signals to digitally encoded versions. The codecs vary in sound quality, bandwidth required, computational complexity and so on. For VoIP over the cellular network and with somewhat limited hardware resources the resulting sound quality is consequently restricted. Two low-rate codecs, G.729d and MELPe, are presented as two suitable candidates for this scenario.

**G.729d**

The International Telecommunication Union (ITU) has specified a voice-codec using Conjugate Structure Algebraic Code-Excited Linear Prediction (CS-ACELP) in Recommendation G.729 [11]. The original codec requires 8 kbps but in Annex D a version using 6.4 kbps is specified, this codec is referred to as G.729d and it is
the main codec considered in this thesis. The codec operates on 10 ms frames and extracts prediction parameters so that the decoder can recreate the current frame based on the previous one. The implication of this is that losing one speech frame affects consecutive frames as well. To compensate the codec uses packet loss concealment which tries to maintain the characteristics of the signal while gradually decreasing its energy when a frame is lost. How to detect missing frames is not given by the standard but dependent on the implementation.

Enhanced Mixed-Excitation Linear Prediction

If even higher constraints are given on the bandwidth (e.g. satellite communication) then Enhanced Mixed-Excitation Linear Prediction (MELPe) might be a better choice than G.729d. It was originally a United States Department of Defense standard which was later adopted by NATO, under the name STANAG-4591 [12]. This codec is also based on a predictive model and exists in versions with bitrates of 2.4 kbps, 1.2 kbps or 0.6 kbps with payload intervals of 22.5 ms, 67.5 ms or 90 ms respectively.

2.2.2 Packetizer

As discussed above, the packetizer is responsible for packaging the encoded speech frames in a manner resulting in high throughput over the network while minimizing the delay. The implementation of the packetizer is application specific to meet the demands of the particular situation and therefore hard to evaluate in a general perspective. For example speech frames can be duplicated in consecutive network packets to increase the redundancy and hence the resistance against network impairments. The packetizer itself will add some header used for synchronisation and then the following headers are added before the packet is sent over the network.

- 12 bytes RTP header [8]
- If a VPN solution is used, as discussed in Section 2.1, the DTLS header adds roughly 20 bytes depending on the configuration [5]
- 8 bytes UDP header [4]
- 20 bytes IP header [1]
- Link layer header, for Ethernet a MAC header of 18 bytes is added [2]

This adds up to 78 bytes of headers per packet, which should be compared to 8 bytes per frame of 10 ms speech for the G.729d voice-codec. The packetizer therefore affects both the final actual bandwidth and sets a lower limit on the delay of the entire system depending on how many speech frames that are enclosed in each network packet.

2.2.3 Jitter buffer

The transmission over the network will most likely cause the packets to arrive with different timing to when they were sent, they may even arrive out-of-order
when UDP is used. The jitter buffer compensates for this by buffering the incoming packets and then sending them onwards in a continuous and ordered stream. A small buffer reduces the delay introduced but it can lead to packet loss if the buffer is not large enough to absorb the transport delay variation. Adaptive buffers may be used which estimates the current scenario and adopts the size of the buffer accordingly, given a set of optimisation rules. Better performance tend to be achieved than for static buffers if the size is appropriately controlled. As for the packetizer, the jitter buffer is typically application specific and thus hard to evaluate generally.

2.3 Quality of Service Metrics for VoIP

For VoIP applications a metric for the QoS is usually equivalent to measuring or estimating the speech quality. Obtaining an easily interpreted metric is highly useful since it allows for comparisons between different systems or configurations. Speech quality measurements can be either subjective or objective. The subjective ones are substantially more time consuming but benefits from the fact that they represent the very metric to be measured. Objective methods can either be intrusive or not depending on if they require the original signal. Non-intrusive methods can be further divided into parametric or signal-based methods, where the latter demands access to the signal transmitted. For secure VoIP, neither the reference signal nor the signal itself is available for measurements a parametric method has to be used.

2.3.1 E-model

The E-model defined by ITU [13] is intended for planning purposes but is widely used for speech quality measurements as well. It is an extensive parametric model but it still requires subjective testing to find some parameters if the conditions of interest have not been previously scored by subjective panels. The complexity of it makes it accurate but less suited for real-time monitoring or control purposes. Its main equation, stated in Equation (2.1), yields a scalar rating value $R$ of the conversational quality in the interval $[0,100]$ ranging from poor to excellent. The fundamental principle that the E-model is based on is that psychological factors on the psychological scale are additive. Note that this does not imply that factors such as delay and loss are uncorrelated, only that their effects on the psychological scale are additive.

$$R = R_o - I_s - I_d - I_e + A$$ (2.1)

$R_o$: The basic signal-to-noise ratio.

$I_s$: The signal-to-noise impairment factor.

$I_d$: The delay impairment factor, includes three different delay metrics.
$I_e$: The effective equipment impairment factor, includes effects of low bit-rate codecs and randomly distributed packet loss.

$A$: The advantage factor accounts for compensation of impairment factors if the user is given other benefits such as mobility or coverage in remote locations.

### 2.3.2 Perceptual Evaluation of Speech Quality

Perceptual Evaluation of Speech Quality (PESQ) [14] is an objective and intrusive industry standard for speech quality measurement which is applied worldwide. It uses a perceptual model of the end user to predict how the end user would perceive the degraded signal. Before the comparison can be made the degraded signal has to be aligned in time with the original. Unfortunately the alignment results in that impairments due to delay are not accounted for. The output of the model is a value in the interval $[-0.5, 4.5]$ with a higher score being the better result.

The original model, developed in 2001, was intended for narrowband networks and codecs (30 - 3100 Hz) but in 2007 an extension was released for wideband systems (50 - 7000 Hz) [15]. The successor to PESQ, named Perceptual Objective Listening Quality Assessment (POLQA) [16], was released in 2014 and handles signals in the range of 50 - 14000 Hz.

### 2.3.3 Mean Opinion Score

To benchmark objective methods, subjective ones as the Mean Opinion Score (MOS) are crucial. Procedures for conducting the subjective evaluations of the transmission quality has been defined by ITU [17]. Subjects listen to a transmission and rate it on a scale from 1 to 5 ranging from bad to excellent quality. The mean of the scores then yields the MOS value.

### 2.3.4 Conversion Formulas between Metrics

Formulas exists to convert between the different metrics described above. Since MOS is what they try to estimate that is also how comparisons between metrics are done, by conversion to MOS. The notation for MOS, defined in [18] and given in Table 2.1, is used to distinguish between what the score relates to and how it was obtained. For example, delay is a factor that only affects conversational quality and not listening.

<table>
<thead>
<tr>
<th></th>
<th>Listening-only</th>
<th>Conversational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subjective</td>
<td>MOS$_{LQS}$</td>
<td>MOS$_{CQS}$</td>
</tr>
<tr>
<td>Objective</td>
<td>MOS$_{LQO}$</td>
<td>MOS$_{CQO}$</td>
</tr>
<tr>
<td>Estimated</td>
<td>MOS$_{LQE}$</td>
<td>MOS$_{CQE}$</td>
</tr>
</tbody>
</table>

*Table 2.1: Notation for different MOS values.*
PESQ to MOS\textsubscript{LQO}

A MOS\textsubscript{LQO} value can be obtained from a PESQ score by Equation (2.2) [19] and the mapping is depicted in Figure 2.3. The inversion is given in Equation (2.3).

\[
MOS_{LQO} = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.495PESQ+4.6607}} \quad (2.2)
\]

\[
PESQ = \frac{4.6607 - \ln \left( \frac{4.999-MOS_{LQO}}{MOS_{LQO}-0.999} \right)}{1.4945} \quad (2.3)
\]

\textbf{Figure 2.3: Mapping PESQ score to MOS\textsubscript{LQO} as per Equation (2.2).}

R-value to MOS\textsubscript{CQE}

From a R-value the corresponding MOS value can be obtained per Equation (2.4) [13], the mapping is depicted in Figure 2.4. A highly complex inversion function is also given in [13], presented in Equation (2.5) is instead a 3\textsuperscript{rd} order polynomial fitting of the inversion function[10].

\[
MOS_{CQE} = \begin{cases} 
1 & \text{for } R \leq 0 \\
1 + 0.035R + R(R - 60)(100 - R)7 \cdot 10^{-6} & \text{for } 0 < R < 100 \\
4.5 & \text{for } R \geq 100 
\end{cases} \quad (2.4)
\]

\[
R = 3.062MOS_{CQE}^3 - 25.314MOS_{CQE}^2 + 87.060MOS_{CQE} - 57.336 \quad (2.5)
\]

\textbf{2.4 Prediction Models for VoIP QoS}

Two different types of prediction models were considered for the work of this thesis, regression based and neural networks. The regression based approach
was chosen since it is the more commonly used and the easier one to interpret and thereby easier to compare to other studies. Furthermore, Sun[10] implemented both approaches but achieved better performance with the regression based model. This is understandable since what effects the QoS is rather well known and can hence be well modelled without the need of the hidden relationships obtained in neural networks. In general, machine learning techniques also need more data to be effective [20] but with the advantage that they can find relationships not taken into account originally.

### 2.4.1 Regression Based

A regression based model is the most commonly used in the literature [10, 21]. In essence it fits a function $f$ with a pre-defined form given the data points $X$ and their values $Y$ and outputs the parameters of the function $\beta$, or more formally $E(Y|X) = f(X, \beta)$ [22, pp. 358-360]. The benefit from using this method is that the relation between input and output is made clear. However, as for all parameter based methods, impairments not captured by the parameters are not accounted for [23].

### 2.4.2 Neural Networks

Neural networks is a machine learning approach which tries to artificially emulate a brain. The brain is made up of billions of simple structures, neurons, which are interconnected to each other and handles relatively simple inputs and outputs. The neurons are modelled by what is called perceptrons which can be thought of as simple functions. The perceptrons are trained on labelled data in order to form this intricate network and make the model able to predict the output given a certain input. Note that the network consists of multiple layers and that there may be hidden parameters within this network which is never exposed to the user, a conceptual diagram of this is depicted in Figure 2.5. The hidden parameters means that the model can perform well even if not all data is measurable or known. The drawback with the complex network is that it is hard to understand the relationships between input and output. [24, Ch. 5]
Figure 2.5: The basic concept of a neural network. Input perceptrons to the left, hidden ones in the middle and the output to the right.
The following chapter together with Chapter 2 forms the background for the rest of the thesis regarding secure VoIP. General aspects of VoIP were discussed in the previous chapter and here information related to privacy and security of VoIP traffic will be presented.

3.1 Cryptography

Cryptography is an extensive subject which can be used for a variety of applications and purposes. For the scope of this thesis a general understanding is sufficient and how the addition of cryptography to VoIP affects the quality of service. The reader is assumed to be familiar with key concepts within cryptography, together with being outside the scope of this thesis, formal definitions and security notions will therefore not be given for cryptographic primitives. If a more thorough background or more details are desired then the interested reader is referred to the ample amount of literature on the subject such as [25].

The purpose of cryptography is usually divided into the four parts below. Note that a system does not have to fulfil all of the below to be considered a cryptographic system. [25, p. 9]

- **Confidentiality** – An adversary cannot see which messages that are transmitted.
- **Integrity** – An adversary cannot tamper with the message without the receiver detecting this.
- **Authentication** – A recipient can determine the sender of a message meaning that an adversary cannot impersonate an authentic sender.
Secure Voice over IP

• Non-repudiation – The sender should not be able to deny that it sent the message.

Confidentiality is provided through encryption which has an impact on communication requirements and will therefore be further discussed in the next section. The other aspects does not impact the communication of the payload data as much and are therefore only mentioned briefly.

A Hashed Message Authentication Code (HMAC) is a cryptographic structure based on a keyed-hash function which can be used for integrity and authentication of a message. The strength of it depends on the underlying hash function, such as MD5 or preferably the more secure SHA-1 or SHA-2. [26]

Non-repudiation is achieved through digital signatures and certificates. Digital signatures are used in the same sense as hand-written ones and digital certificates is a way of proving that your signature belongs to you. Most commonly RSA or ElGamal [25, pp. 244-250] are the signature schemes used together with X.509 signatures [25, pp. 270-272].

3.2 Encryption

Encryption is in essence a mean of hiding the content of a message through the use of an algorithm and a key. How the keys to be used for encryption and decryption are distributed is one aspect that can differ between encryption schemes. The easiest, both in terms of understanding and computational complexity, is the symmetric case where both parties have the same secret key. The secret key has to be agreed upon and for this asymmetric-key cryptography can be used. It allows two parties to agree on a key that only they will know even if the communication is intercepted by an adversary. Due to the higher computational complexity, asymmetric schemes are typically just used for key agreement and then the actual payload is encrypted with a symmetric scheme.

Symmetric schemes can either operate on plaintext of arbitrary length, stream ciphers, or on plaintext of fixed length, block ciphers. Stream ciphers may seem attractive for real-time streaming media but due to the many vulnerabilities in the most popular stream cipher RC4 [27, 28, 29, 30, 31], a secure block cipher should be used instead.

The most widely used and currently most secure block cipher is the Advanced Encryption Standard (AES) [32]. It operates on blocks of 128 bits and can be used with keys of 128, 192 or 256 bits. With the use of padding and certain modes of operation, see Section 3.2.1, block ciphers can be made to operate on plaintext of arbitrary length as well.

3.2.1 Mode of Operation

A mode of operation defines how to apply a cipher’s single-block operation on data larger than one block. The mode should hide patterns in the plaintext
and the same plaintext should give different ciphertexts when independently encrypted with the same key multiple times.

**CTR Mode of Operation**

A rather basic mode of operation is the Counter (CTR) mode which allows for access of a random block of data. A counter function is used to produce a sequence which must not repeat itself for any block encrypted under the given key, the counter is then encrypted using the cipher $CIPH$ under the key $K$ and exclusive-ORed with the plaintext to generate the ciphertext \[33\]. The operation for decryption follows trivially due to the commutativity of the exclusive-OR operation. A schematic overview of a block cipher in CTR mode is depicted in Figure 3.1. CTR mode is suitable for streaming data over connection-less transmissions since blocks do not depend on information in other blocks. The loss of one packet will hence not impact the decryption of other packets as long as the counter is kept in synchronisation.

![CTR Mode Schematic](image)

**Figure 3.1:** Schematic overview of encryption and decryption in CTR mode \[33, Fig. 5\].

**CBC Mode of Operation**

Another common mode of operation is Cipher Block Chaining (CBC). The ciphertext from a block is the encryption with cipher $CIPH$ under key $K$ of the plaintext exclusive-ORed with the ciphertext of the previous block \[33\]. The first block is exclusive-ORed with an Initialisation Vector (IV) which must be random but not necessarily secret. The purpose of the IV is to ensure that the same plaintext does not result in the same ciphertext when encrypted multiples times. The en-
The dependency on the previous block makes CBC less suited for communication with unreliable delivery since the loss of a packet will impact the decryption of subsequent data.

![Diagram of encryption and decryption in CBC mode](image)

**Figure 3.2:** Schematic overview of encryption and decryption in CBC mode [33, Fig. 2].

### 3.3 Secure Communication Interoperability Protocol

To allow different organisations worldwide to communicate securely the Secure Communication Interoperability Protocol (SCIP) is used. The protocol provides a secure overlay on a variety of digital networks, for instance GSM, 3G, IP and satellite. The SCIP Signaling Plan [34] defines the application layer communication used to establish a secure end-to-end session between two parties. Considerations for using SCIP with RTP are given in [35].

#### 3.3.1 Cryptographic Synchronisation

One interesting aspect of SCIP within the scope of this thesis is how cryptographic synchronisation is maintained during a session. SCIP uses AES in CTR mode for encryption. As counter function a 128 bit State Vector is defined which includes a counter segment of 42 bits [36]. The encoder and decoder must use the same State Vector for operations on the same block and this is assured through cryptographic synchronisation. Information about the State Vector is transmitted in each packet over the network with the aid of Sync Management Frames [34]. As can be seen in Figure 3.3 the least-significant bits (Short Term Component) are always part of the frame while the most-significant bits (Long Term Component) are split over three consecutive Sync Management Frames. If synchronisation is
lost and cannot be restored with the information at hand then a cryptographic resynchronisation may be initiated so that the communication may continue [34].

Figure 3.3: Mapping of State Vector to Sync Management Frame [37, Fig. 2.3-1].

3.3.2 Packetization

Each network packet consists of a superframe and the structure of such a frame depends on the voice-codec used. The superframe for the G.729D codec will be presented here and the interested reader can find the MELPe superframe in [38].

The G.729D superframe, as illustrated in Figure 3.4, begins with a Sync Management frame. After that follows one to eight Encrypted Speech Frames, each consisting of the lowest 8-bits of the counter and four encrypted G.729D frames [37]. Consequently each superframe contains between 40 and 320 ms of speech which can be compared to cleartext VoIP which typically packetize voice in 20 ms intervals [39].

Considering the additional headers discussed in Section 2.2.2 as well, this results in that 30 – 74% of the VoIP traffic is headers which must be considered as a substantial part.
Figure 3.4: Superframe in the SCIP protocol when using the G.729D voice-codec [37, Fig. 2.0-1].
The following chapter presents previous work within the area which can be divided into two parts. The first part relates to prediction of VoIP quality for a transmission when no encryption is used. The second part presents analyses done on the performance of encrypted VoIP traffic mainly regarding different protocols. To the best of my knowledge this thesis is the first work done where the two parts are combined to predict VoIP quality for an encrypted transmission.

4.1 VoIP Quality Prediction

The E-model develop by ITU is extensive but also cumbersome to use in new scenarios because subjective testing is required for derivation of a subset of the system parameters [40]. Hence, previous work [10, 41] has aimed at simplifying the model by removing this requirement and base the estimation only on directly observable parameters such as packet loss, delay and jitter.

Other studies have extended the E-model to handle wideband signals [42], in addition to narrowband, or by taking burst properties of the packet loss into account [43]. It was shown that the difference in quality of service between random and burst packet loss increased with the fraction of lost packets. Cole and Rosenbluth [21] reached the same conclusion independently. Intuitively the difference should be just as prominent when encryption is used due to the need for cryptographic synchronisation and the difficulty to recreate lost packets.

Alavi and Nikmehr [42] along with Sun and Ifeachor [44] found that the voice-codec affected the QoS metric for VoIP. Especially how well it handled packet loss had an impact on the performance. This indicates that the voice-codec used should be considered as a potential parameter to the system.
Common for the studies mentioned in this section is that they assume the additive property of the impairments factors in the E-model to hold. It is a necessary assumption for the E-model but only Ding and Goubran [41] attempts to validate it while Sun [10] mentions it as a limitation of the work. It should be noted that the document defining the E-model [13] points out that the additivity property of the model “has not been checked to a satisfactory extent”.

Another thing in common among the studies in this section is that they use PESQ as a reference value, as recommended in [45]. When given access to the original signal this metric is efficient, no subjective testing is needed, while still yielding accurate results. Following common practice, PESQ is also the reference value used in this work.

4.2 Encrypted VoIP Performance Analysis

Studies have been performed to evaluate the performance of VoIP traffic in relation to encryption and different protocols used. All of their results show that encryption has an impact on performance and voice quality but the details differ.

Guillen and Chacon [46], as well as Talha and Barry [47], found that the jitter increased for encrypted VoIP traffic. The reported increase was from 1 to 15 ms and from 19 to 30 ms, respectively. In conflict with that, the results of Radmand and Talevski [48] showed a decrease in jitter when encryption was applied. The relevance of their test system for this thesis could however be questioned since they report jitter of less than 0.5 ms in all cases and jitter is greater than that in a cellular network.

The impact of encryption on the delay was by Talha and Barry [47] found to be just 0.1 ms. Given that typical voice latency in commercial cellular networks is 200-300 ms [39], this has to be considered as negligible.

Barbieri et al. [49] found that the effective bandwidth can be heavily reduced when encryption is used due to the additional overhead. This implies that QoS for encrypted VoIP traffic can not be predicted with conventional models for cleartext VoIP.

Both [47] and [48] stresses the need of future work to identify factors that affect voice quality in an encrypted VoIP scenario.

It is worth noting that the studies presented here have been conducted in a LAN or WAN setting while the work to be carried out in this project will focus on transmission using a (simulated) cellular network. This will clearly influence aspects such as the relation between transmission delay and computational delay.
The practical work of this thesis consisted of three main parts. First, the test environment was setup in order to perform and measure VoIP calls under controlled circumstances. Second, data was gathered by performing calls, largely via an automated procedure. Finally the data was analysed to reach a conclusion. How this work was done is presented in this chapter while the chapters to follow will present and analyse the findings.

Some details regarding the implementation of the VoIP clients and related protocols and codecs can not be disclosed due to security considerations and confidentiality.

5.1 Test Setup

A test environment was needed which resembled real life configurations as closely as possible while still providing control of the parameters. To increase efficiency together with reducing the number of error sources the test procedure had to be automated to a large extent. That included placing calls between two VoIP clients, send audio and compare it to the received audio while controlling the characteristics of the network and gathering relevant data.

5.1.1 Virtual Machines

At first a configuration where the two VoIP clients was run from within virtual machines was considered. Less hardware would be needed and more control over the machines would be possible with the use of virtual machines. However, initial testing revealed significant problems with the audio and an abnormally high CPU load for the virtual machines. For virtual machines the CPU load was ten
times higher than for physical machines. Only running one client in a virtual machine and the other client on the host computer also resulted in corrupted audio. The setup was tested both with the client described in Section 5.1.2 and an open-source alternative, Jitsi [50], with the same result. Audio playback and recording was functional whenever the VoIP clients were not running in the virtual machine so the problem could be deduced to originate from the combination of a VoIP client and the virtual environment, possibly related to the soundcard and its drivers. Additional hardware was then made available so that the calls used to gather statistics could be made between two VoIP clients running on physical machines. No further investigation was made as to what caused the problems for the virtual machines since it is outside the scope of this thesis.

5.1.2 Overview

An overview of the test setup is depicted in Figure 5.1 and the different entities are described below.

![Figure 5.1: Schematic overview of the test setup.](image)

**Client A & B**

Two Linux computers constituted the two clients, A and B. They were running a VoIP client for Linux developed by Sectra Communications for testing purposes. Note that this VoIP client is intended solely for internal use and is not one of their commercial products. Configurations concerning cryptographic aspects could not be altered due to the required security clearance but three different packetizers (referred to as A, B and C) were available.
5.1 Test Setup

**SIP**

A SIP server was used to initiate the VoIP calls between the two clients. Access was granted to the SIP server used internally at Sectra Communications and accounts for the clients were created.

**DB**

An SQL database was used to store the results from the different tests.

**VQT**

A VQuad Probe [51] was used for Voice Quality Testing (VQT). It sent reference audio over the call and compared the received signal with the reference in order to compute a PESQ score. The audio was carried to and from the clients with the use of 3.5 mm audio cables and the device was also connected to the LAN to make the data accessible to the Test Controller.

Six different audio snippets, roughly 5 seconds long each, was used as reference signals and then repeated throughout the call. Each of them was a short sentence meant to capture the range of different sounds in natural speech. Half of them was read by females and half by males to remove any gender biases. Each snippet is individually given a PESQ score and entered into the database.

**Network Emulator**

To emulate different network conditions an iTrinegy Enterprise network emulator [52] was used. The server can impair network traffic in a variety of ways, packet loss and latency are the two main parameters for this work. Jitter is set implicitly by specifying the latency as a range rather than a fixed value.

During an emulation the network conditions can be set to change at specific times and thereby the conditions can be set for individual calls. Unfortunately these changes can not be triggered via an interface so to utilise this feature the calls have to be synchronised in time with the network emulator which is handled by the Test Controller.

**Test Controller**

The test controller initiated all events such as call setup and termination, VQT and network emulation. Information about the connection was gathered from the clients and the PESQ scores were gathered from the VQT device by the Test Controller and then inserted into the database.

Synchronisation between the clients, the network emulator and the VQT device was only done via time. As a consequence the VQT device might be in the middle of receiving an audio snippet when the call was terminated which leads to a low PESQ score for the last snippet. Furthermore, the different snippets gets different PESQ scores under the same circumstances, see Section 6.1.4 for details. To remove any biases based on the different audio snippets, the same number of entries per audio snippet per call was used. This adjustment was done after removing the last VQT entry in the database as argued for above.
LAN

All entities were directly connected to the LAN, except client B which was connected through the network emulator.

5.2 Workflow to Perform a Measurement

The steps to perform a test run will be briefly described in this section. After connecting all entities as described above the VQT device and network emulator were booted and configured. With the network emulator running and allowing traffic to client B the two clients could be started so that they could register with the SIP server. At this point the script run by the test controller could be started.

The script in the test controller loads the correct configuration into the network emulator and starts the scripts run locally on the VQT device which sends and compares the audio snippets. Thereafter multiple calls are performed by the test controller and after each call the relevant data is fetched and stored in the database for later access.

When all calls have been performed the test controller stops the network emulator and the VQT device and closes all sockets.

5.3 Initial Calibration and Setup Validation

A few measurements had to be taken initially to validate the setup and determine parameters for future test cases.

5.3.1 Network Characteristics

At first the impact of the LAN was investigated by not introducing any impairments in the network emulator and measure the effects on a 30 minutes call. As suspected the impairments introduced by the LAN were negligible in comparison to the impairments to be introduced by the network emulator later on, for details please see Section 6.1.1. With this fact established it was possible to move on since the basic test setup had been verified.

5.3.2 Call Duration

A sufficiently long call duration had to be determined to give reliable results of future tests while minimising the amount of time required. Calls with different durations were performed under conditions meant to resemble a realistic, although bad, situation in a cellular network. Using degraded conditions should increase the variance of the call quality since impairments can occur at more or less sensitive places. The future tests should thereby give reliable results even in these situations. To emulate a degraded GSM connection the network emulator was configured to a random packet loss of 5 percent, the latency was set to be chosen uniformly at random between 270 and 490 ms and 3 percent of the pack-
ets were moved 1 step out of order. These values were chosen based on in-house expertise at Sectra Communications.

### 5.3.3 R-value Baseline

The first step towards predicting the voice quality was to get a baseline for the R-value under ideal conditions. This value can be seen as the combined effects of $R_0$, the basic signal-to-noise ratio, and $I_s$, the signal-to-noise impairment factor, from the E-model which were introduced in Section 2.3.1. Factors that effect the baseline value are mainly the codecs and equipment used. The data from when the impact of the LAN was investigated was also used for this baseline estimation.

### 5.4 Test Runs

Once the initial calibration had been done a variety of tests were performed to gather as much data as possible under the time constraint of this thesis.

The different metrics that were collected and how the tests were configured are described in the sections to follow.

#### 5.4.1 Collected Metrics

For each call the following data was collected and stored in the database for analysis.

- Test run ID.
- Call ID.
- Packetizer codec.
- Call duration.
- Percentage of packets lost.
- Average jitter for the call.
- Average RTT for the call.
- During a call, for each voice snippet the following was stored:
  - PESQ score
  - Voice ID

**Burst Metric**

Previous studies [21, 43] recommend using a burst metric since it has been shown to influence the quality of the call. However, the recommended intervals for RTCP reports, see Section 2.1.5, means that at least tens of RTP packets will be sent between each RTCP report, if not more. The implication is that a reliable burst metric can not be obtained since the RTP packet resolution is too low. Close to a per-packet resolution would be required to get a useful burst metric and for
that the VoIP clients would have to be modified to supply this metric. Doing so would reduce the generality of the model to be developed and it would have a negative impact on the performance of the client. Although interesting, a burst metric has therefore not been considered for this thesis.

### 5.4.2 Test Configurations

Several tests were performed to gather the necessary data for further analysis. In essence there are three metrics of interest: packet loss, jitter and RTT. The network emulator was configured to test these metrics and combinations thereof. Below is a list of the different tests with a brief explanation of what the test was meant to investigate. Some of them were planned already from the start while others were added to investigate certain aspects further. For details of the configurations please see Table A.1. When the time allowed for it, 3 calls were performed with the same network configuration to reduce uncertainties from the measurements.

- **Loss** and **LossHigh** - Increasing packet loss under otherwise perfect conditions. Used to investigate the effects of packet loss.

- **Latency** - Increasing latency under otherwise perfect conditions. The impairments due to increased latency can not be measured directly with a PESQ score as reference since the PESQ algorithm compensates for delay. This test is therefore used to investigate if increased latency causes any side-effects.

- **Jitter** - Increasing jitter under otherwise perfect conditions. Note that to achieve high jitter the maximum RTT also had to be increased. Used to investigate the effects of jitter.

- **JitterMoveOnly** - Increasing percentage of packets moved out of order under otherwise perfect conditions. Used to investigate the effects of latency and out of order on jitter and packet loss.

- **JitterLatencyOnly** - Increasing jitter only due to latency under otherwise perfect conditions. Used to investigate the effects of latency and out of order on jitter and packet loss.

- **LossPacketizer** - Increasing packet loss under otherwise perfect conditions, with packetizer B and C respectively. Used to investigate the effects of the packetizer when packet loss occurs.

- **JitterPacketizer** - Increasing jitter under otherwise perfect conditions, with packetizer B and C respectively. Used to investigate the effects of the packetizer for different jitter values.

- **JitterGSM** - Increasing jitter under conditions to be expected in a GSM network. Used to investigate the additivity property of the effects of jitter.

- **LossGSM** - Increasing loss under conditions to be expected in a GSM network. Used to investigate the additivity property of the effects of packet
5.5 Model Development

loss.

- **RandomGSM** - Values chosen at random meant to resemble conditions in a GSM network. Used to investigate how well the model performs.
- **Random3G** - Values chosen at random meant to resemble conditions in a 3G network. Used to investigate how well the model performs.

The values in Table A.1 were chosen based on in-house expertise at Sectra Communications and with support from the literature. In [53], 3G and 4G networks in the USA were studied and they found that most networks have a RTT of 120-700 ms and less than 1 % of the packets are lost. No studies have been found that measure jitter in cellular networks, nevertheless [54] simulated jitter and used values in the range of 0-60 ms. The reason the tests were mainly configured to resemble GSM conditions is that 3G and 4G performs significantly better and thereby the full range is covered when GSM networks are investigated. If more time had been allocated for the thesis then it would have been of interest to evaluate 3G and 4G network conditions in more detail in order to increase the model’s performance under those conditions. Without a model to confirm nor describe a relationship between packet loss and jitter the values were drawn from a Gaussian distribution for the validation of the model, i.e. the RandomGSM and Random3G test.

The bandwidth of real systems is assumed to be high enough to not throttle the communication. Another way to see it is that parameters of the protocol are chosen so that communication is assumed to be possible without congestion. Hence, no impairments were introduced with respect to the bandwidth.

5.5 Model Development

To model the data the Python library SciPy was used and in particular the `optimize.curve_fit()` function. It uses a non-linear least squares method to fit a function to given data. The form of the function can be specified, making it possible to fit virtually any kind of function to the data. Visual inspection of the data and trial and error was the approach taken to decide the correct form of the function to be used.
The results gained from the measurements and analysis undertaken, as described in Chapter 5, are presented in this chapter. Note that the R-values presented here are all converted from the PESQ scores outputted from the VQT device, according to the equations in Section 2.3.4.

In the figures, most of them given at the end of the chapter, the PESQ score is given on the right-hand y-axis. Note that this value is converted from the corresponding value on the R-value axis to the left and then rounded. The values are plotted against the R-value axis and the PESQ score axis should therefore only be seen as a reference.

6.1 Setup Validation and Calibration Results

The results from the initial measurements taken to calibrate and validate the test setup are presented in this section.

6.1.1 Network Characteristics

The initial check that the LAN did not introduce any significant impairments on the connection proved successful, as expected. During a 30 minutes call not a single packet was lost out of the 44 000 packets sent. The measured RTT and jitter are given in Table 6.1.

6.1.2 Test Call Duration

The setup to determine the required call duration resulted in an actual packet loss of 5 percent, a round-trip time of 800 ms and jitter of 40 ms on average for the different calls. The R-value for the different calls are depicted in Figure 6.1 and
the related statistics are presented in Table 6.2. We can see that there is not much to gain from increasing the call duration from 15 to 20 minutes. Furthermore, the 15 minutes calls have a lower variance than the 10 minutes one. Hence, 15 minutes was chosen as the duration for future tests.

![Figure 6.1](Image)

*Figure 6.1: Measured voice quality for calls with different durations.*

<table>
<thead>
<tr>
<th>Call duration (min)</th>
<th>3 min</th>
<th>5 min</th>
<th>10 min</th>
<th>15 min</th>
<th>20 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (R-value)</td>
<td>58.87</td>
<td>59.18</td>
<td>59.17</td>
<td>59.05</td>
<td>59.16</td>
</tr>
<tr>
<td>Std. (R-value)</td>
<td>1.99</td>
<td>1.01</td>
<td>0.80</td>
<td>0.43</td>
<td>0.42</td>
</tr>
</tbody>
</table>

### 6.1.3 R-value Baseline

When no impairments were introduced by the network emulator a R-value of 63.30 was obtained. The data was gathered from the same call as when investigating the network characteristics in Section 6.1.1.
6.1.4 Voice Dependency

As mentioned in Section 5.1.2, the different audio snippets sent by the VQT device gets different PESQ scores under the same conditions. To give the reader an understanding of how much they differ the results for the call described in Section 6.1.1 with ideal conditions are given in Table 6.3. Those values can be compared with the data in Table 6.4 which is taken from one of the 20 minutes call presented in Section 6.1.2 when GSM conditions applied. Note that the standard deviation for the overall call presented in the tables is the mean of the standard deviations for the different snippets since that better represents the differences within a call due to network factors rather than different snippets.

**Table 6.3: R-value for the different audio snippets in a call under perfect network conditions.**

<table>
<thead>
<tr>
<th></th>
<th>Overall</th>
<th>Male1</th>
<th>Male2</th>
<th>Male3</th>
<th>Female1</th>
<th>Female2</th>
<th>Female3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>63.30</td>
<td>62.84</td>
<td>70.17</td>
<td>64.90</td>
<td>53.44</td>
<td>65.51</td>
<td>62.92</td>
</tr>
<tr>
<td><strong>Std.</strong></td>
<td>1.697</td>
<td>1.456</td>
<td>1.497</td>
<td>2.205</td>
<td>1.857</td>
<td>1.943</td>
<td>1.226</td>
</tr>
</tbody>
</table>

**Table 6.4: R-value for the different audio snippets in a call under network conditions resembling GSM.**

<table>
<thead>
<tr>
<th></th>
<th>Overall</th>
<th>Male1</th>
<th>Male2</th>
<th>Male3</th>
<th>Female1</th>
<th>Female2</th>
<th>Female3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>59.46</td>
<td>56.28</td>
<td>68.23</td>
<td>60.48</td>
<td>49.02</td>
<td>62.32</td>
<td>60.44</td>
</tr>
<tr>
<td><strong>Std.</strong></td>
<td>5.037</td>
<td>4.654</td>
<td>5.331</td>
<td>5.075</td>
<td>4.916</td>
<td>5.888</td>
<td>4.360</td>
</tr>
</tbody>
</table>

6.2 Test Runs

The measurements obtained to investigate the impact of different kinds of impairments are presented here.

6.2.1 Loss

The investigation of the impact of packet loss, \( l \), yielded the results given in Figure 6.2. The equation for the fitted red line is stated in Equation (6.1).

\[
f_{\text{Loss}} = R_0 + b \log(1 + cl),
\]

where \( R_0 = 64.3 \),
\[
b = 17.8,
\]
\[
c = -0.0335
\]
6.2.2 Jitter

The measurements with increasing jitter, $j$, under otherwise perfect conditions gave interesting results, seen in Figure 6.3. There is a clear difference in how jitter impacts the R-value below and over 40 ms. The fitted red line is therefore constructed as two linear sections as per Equation (6.2).

$$f_{jitter} = (R_0 + bj)(1 - H(j - e)) + (c + dj)H(j - e),$$

where $R_0 = 63.6$,

$$b = -0.0162,$$

$$c = 122,$$

$$d = -1.42$$

$$e = 41.5,$$

and \( H(x) = 0, \quad \text{if } x < 0 \)

\( H(x) = 1, \quad \text{if } x \geq 0 \)

Another aspect worth mentioning is that even though the network emulator did not drop any packets, the clients reported an increasing packet loss from 0 up to 2.8 % as the jitter increased. Jitter was implicitly set in the network emulator by specifying minimum and maximum latency together with a percentage of packets to be moved out of order. Further investigation showed that the packets moved out-of-order caused the reported packet loss. However, packets moved out-of-order did not decrease the measured voice quality, that decrease was only due to the latency.

6.2.3 Delay

As mentioned before, the obtained PESQ scores cannot be used to estimate the effects of delay since the PESQ algorithm compensates for delays. Increasing the latency did not cause any side-effects either since all calls from the Latency run scored an R-value within the interval $[63.3, 63.8]$ and both packet loss and jitter were negligible.

6.2.4 Packetizer

It is clear from Figure 6.4 and 6.6 that the packetizer has an impact on the performance and robustness of the system. The figures depict the measured R-value under packet loss for packetizer B and C and the model of packet loss based on packetizer A (blue line). The reason that no measurements are taken for higher packet losses, as for packetizer A in Figure 6.2, is that the call setup failed in those cases.

From Figure 6.5, it may at first seem like packetizer B performs better than packetizer A but that is not the complete truth. Yes, for jitter values above 50 ms packetizer B shows better results but note that the measured data points in Figure 6.3 and Figure 6.5 are obtained under similar network conditions, see Table A.1 for
details. Hence, packetizer B give rise to higher jitter values and actually performs worse under the same network conditions if we look carefully at the different R-value axes in the figures. Packetizer C shows similar results for packet loss and jitter but with even worse performance than packetizer B, see Figure 6.6 and 6.7.

Another aspect of interest is that the packet loss for the packetizers differed when only jitter was introduced by the network emulator. For when the greatest jitter was introduced on the network, packetizer A had a packet loss of 2.75 %, packetizer B 1.70 % and packetizer C 0.40 %.

### 6.3 Additivity

The additivity of the different impairments was examined by altering packet loss and jitter, respectively, under conditions resembling those in a GSM network.

When packet loss was examined the configuration, stated as LossGSM in Table A.1, resulted in RTT in the interval [815, 830] ms and jitter in the interval [41.0, 42.5] ms. The result is depicted in Figure 6.8 together with the expected R-value if only packet loss was present, as given by Equation (6.1).

The test run when jitter was investigated had RTT in the range [750-1030] ms and packet losses in the interval [2.4, 5.2] %. Figure 6.9 shows the result together with the expected R-value if only jitter was present, as given by Equation (6.2).

For both packet loss and jitter, the other metric increased together with the investigated one even though the configuration of the network emulator remained the same for them.

### 6.4 Derived Model

The additive property of the model holds well enough in order to proceed and create a joint model of packet loss and jitter, this conclusion is further discussed in Section 7.6. Combining $f_{\text{loss}}$ and $f_{\text{jitter}}$ is trivial under the additivity assumption and leads to Equation (6.3), where the two different $R_0$ have been combined to $R_0 = 64$.\(^1\) If it is desired to account for the impairments due to delay as well then please see the discussion in Section 7.4.

\[
R = R_0 + 17.8 \log(1 - 0.0335x) - 0.0162x(1 - H(x - 41.5)) + (122 - R_0 - 1.42x)H(x - 41.5) \tag{6.3}
\]

\(^1\) $R_0$ is further updated in Section 6.5 to $R_0 = 65.39$. 

6.5 Validation

To validate the derived model 230 calls were performed with network parameters set randomly to resemble a bad GSM connection, for details see RandomGSM in Table A.1. In Figure 6.10 the result is plotted with respect to R-value and packet loss, Figure 6.11 similarly shows the R-value related to jitter. Combining the impairments as per Equation (6.3) yields Figure 6.12 where the measured values are plotted against the predicted ones.

The mean error of the calculated R-value compared to the measured one was $-1.39$ and the standard deviation of the error was $1.45$. The difference for each call is depicted in Figure 6.13.

The model was also validated against conditions to be expected in a 3G network. For that 90 calls were performed with the network configuration described under Random3G in Table A.1. As for the GSM conditions the results are given in Figure 6.14 with respect to packet loss, Figure 6.15 shows the relation to jitter, Figure 6.16 compares the calculated and measured R-value and in Figure 6.17 the difference between the two is plotted. The mean error was $-1.39$ R-value units for 3G as well while the standard deviation of the error was $0.79$.

Since both validations show a mean error of $-1.39$ Equation (6.3) can be updated with $R_0 = 65.39$ and the reasoning behind this is given in Section 7.7.
6.6 Figures

Packet Loss and Jitter

Figure 6.2: Measurements of R-value under random packet loss conditions and the fitted model (red line).

Figure 6.3: Measurements of R-value under increasing jitter conditions and the fitted model (red line).
Figure 6.4: Measurements of R-value under random packet loss conditions for packetizer B and the model calculated for packetizer A (blue line) together with a curve fitted to the obtained data points (black dashed line) for comparison.

Figure 6.5: Measurements of R-value under increasing jitter conditions for packetizer B and the model calculated for packetizer A (blue line) together with a curve fitted to the obtained data points (black dashed line) for comparison.
**Figure 6.6:** Measurements of R-value under random packet loss conditions for packetizer C and the model calculated for packetizer A (blue line) together with a curve fitted to the obtained data points (black dashed line) for comparison.

**Figure 6.7:** Measurements of R-value under increasing jitter conditions for packetizer C and the model calculated for packetizer A (blue line) together with a curve fitted to the obtained data points (black dashed line) for comparison.
Figure 6.8: Measurements of R-value with increasing packet loss, under conditions to be expected in a GSM network. Plotted together with the calculated R-value based solely on packet loss (blue line) and a curve fitted to the obtained data points (black dashed line) for comparison.

Figure 6.9: Measurements of R-value with increasing jitter, under conditions to be expected in a GSM network. Plotted together with the calculated R-value based solely on jitter (blue line) and a curve fitted to the obtained data points (black dashed line) for comparison.
6.6 Figures

Packet Loss and Jitter under GSM Conditions

**Figure 6.10:** Measurements of R-value under random GSM conditions plotted against packet loss together with the calculated R-value based solely on packet loss (blue line).

**Figure 6.11:** Measurements of R-value under random GSM conditions plotted against jitter together with the calculated R-value based solely on jitter (blue line).
**Validation under GSM Conditions**

![Graph](image)

**Figure 6.12:** Measurements of R-value under random GSM conditions compared with the calculated R-value based on packet loss and jitter.

![Graph](image)

**Figure 6.13:** Difference between the measured and calculated R-value under random GSM conditions, see Figure 6.12 for absolute values.
Figure 6.14: Measurements of R-value under random 3G conditions plotted against packet loss together with the calculated R-value based solely on packet loss (blue line).

Figure 6.15: Measurements of R-value under random 3G conditions plotted against jitter together with the calculated R-value based solely on jitter (blue line).
Validation under 3G Conditions

Figure 6.16: Measurements of R-value under random 3G conditions compared with the calculated R-value based on packet loss and jitter. The scale on the y-axis is set to be equal to the one in Figure 6.12 for easier comparison.

Figure 6.17: Difference between the measured and calculated R-value under random 3G conditions, see Figure 6.16 for absolute values.
This chapter analyses the results presented in the previous chapter and discusses the method which was described in Chapter 5.

7.1 Setup Validation and Calibration

Since the setup is kept consistent during each call one would expect the variance within a call to be low. However, both where packet loss occurs and the possible presence of burst errors impacts the final result as previously shown by Sun [10] and Clark [43]. The rather high deviations within a call in a GSM network, stated in Table 6.4, can therefore be attributed to these aspects.

Even though the investigation of call durations showed that 15 minutes was a good choice to mitigate the impact of packet loss location, multiple calls were performed with the same configuration. Doing so allowed the uncertainty in the measurements to be clearly visible in the graphs and thereby give a better understanding of the results, in addition to reducing the variance.

7.1.1 Difference between Configuration and Results

There is a small difference between the configuration of the network emulator and the measured results. This difference is most apparent in Figure 6.4 and 6.6 depicting packet loss for different packetizers. The small clusters of data points with the same network configuration should be vertically aligned with the respective packet loss if the measured packet loss were exactly the same as the configured one.

The reason for this slight divergence is probably that the measured data is only for VoIP traffic while the configuration applies to all packets sent on the network.
Given the satisfying results from the setup validation it is highly unlikely that the difference is due to impairments introduced by the LAN.

### 7.2 Loss

The impact of packet loss, as seen in Figure 6.2, followed a logarithmic function which is consistent with previous studies [10, 21, 41]. The logarithmic behaviour was therefore expected and the contribution by this work is the estimation of the parameters for this setup and of course confirming the expected behaviour.

### 7.3 Jitter

Packet loss and delay has been rather extensively researched before but only one study [41] have modelled the impact of jitter explicitly. What impact to expect from jitter was hence largely unknown. It was therefore interesting to find the clear distinction, seen in Figure 6.3, on how jitter under and above roughly 40 ms effected the measured quality.

The difference in impact of the jitter most likely depends on that an internal buffer does not handle jitter values above 40 ms well. Why the limit is around 40 ms is harder to tell. The Encrypted Speech Frame is the smallest unit of speech sent together in the SCIP standard, as described in Section 3.3.2. It consists of 40 ms of speech and if the implementation of the internal tool at Sectra Communications has a similar structure then it might be a reason as to why the difference appears around 40 ms.

### 7.4 Delay

Impairments due to delay have not been included in the derived model in this thesis since the PESQ algorithm, which has been used as reference, does not account for those impairments. The recommendation is therefore to use the E-model’s estimate of impairment due to delay, $I_d$, instead. For delays, $d$, less than 600 ms Equation (7.1) can be used which is a simplification of the E-model to only depend on the measured delay [44]. With the assumption of additivity, including the impairments due to delay in the model is trivial. Note that the delays mentioned in this section refers to one-way delays and not RTT.

\[
I_d = -2.468 \cdot 10^{-14} d^6 + 5.062 \cdot 10^{-11} d^5 - 3.903 \cdot 10^{-8} d^4 \\
+ 1.344 \cdot 10^{-5} d^3 + 0.001802 d^3 - 0.1698
\]  

(7.1)
7.5 Packetizer

There are clear distinctions between how the packetizers handle impairments due to packet loss and jitter, as seen in the figures in Section 6.6. If different packetizers encapsulate different amounts of voice data in a packet then it follows that their sensitivity to network impairments will vary. The results infer that packetizer A adds some sort of redundancy since it handles packet loss significantly better than the other two.

Interestingly, the different packetizers have an impact on the measured jitter values. When the network emulator was configured to give the highest jitter, the measured jitter was 53 ms, 82 ms and 90 ms for the respective packetizers. As described in Chapter 2, the data from the packetizer is sent over the network in RTP packets. The measured jitter comes from information in the RTCP protocol which relates to RTP packets, the data within should therefore not effect the jitter. However, the rate at which the packetizer sends packets over the network is of relevance and is linked to the different packet losses mentioned in Section 6.2.4, when only jitter was introduced on the network. A packetizer that expects to receive packets often will have a smaller acceptable delay before a packet is considered lost. High jitter will therefore manifest itself as packet loss rather than jitter since the jitter value is only calculated against received packets. The conclusion that can be drawn is that packetizer A outputs packets more often than packetizer B and C. The disadvantage of frequent packets is the larger overhead since the packets will contain less voice data per packet.

7.6 Additivity

As in previous studies and in the E-model, the different impairments have been assumed to be additive on the psychological scale. From Figure 6.8 where the additivity of packet loss was investigated no conclusion can be drawn that would contradict the additivity assumption. The measured R-value is naturally lower than the calculated line, which is based solely on packet loss, since jitter is also present. Based on the jitter measured for the calls the offset should go from 0.67 to 2.35 R-value units as the jitter increases. The increase of the offset is as expected but in the interval [1.36, 3.04] meaning that the absolute offset is 0.69 units larger than what the model predicts. The larger absolute offset could be because packet loss and jitter are not additive but the difference is not significant enough to draw any conclusions.

Additivity can not be rejected from the results in Figure 6.9 either, where jitter was investigated, since the obtained data points follow the model rather well. The clear offset seen when the additivity of packet loss was investigated, is not present here however. Given that the packet loss ranged from 2.4 % to 5.2 % for these calls an offset of 1.5 to 3.4 units on the R-value scale would be expected. One possible explanation to why this offset is not present could be that the introduced packet loss masks some of the impairments due to jitter, in which case
additivity would not hold. It is however not possible to draw that conclusion definitively since the difference is not large enough to not be ruled out as measurement variance.

To conclude, the additive property of the impairments holds well enough to base the model on this assumption.

### 7.7 Validation

From Figure 6.10 and 6.11 where packet loss and jitter are depicted in conditions resembling GSM it appears as if jitter is a more dominant factor than packet loss. However, this is probably due to the ranges of packet loss and jitter since the impact of packet loss is not expected to differ as much as jitter in the presented intervals, seen by the blue lines representing the values calculated from the model. The situation in 3G networks, Figure 6.14 and 6.15, does not show any clear signs that one impairment would dominate the other.

Given the spread between measured and calculated values when only considering one impairment, it is satisfying to see that in Figure 6.12 the combined results follow the measured values well. This further strengthens the conclusion that the additivity of the impairments hold well enough to use an additive model.

It would of course have been valuable to validate the model in real GSM and 3G networks but unfortunately that was not possible due to lack of hardware. Instead the parameters of the network emulator was set at random from Gaussian distributions and a large number of validation calls were performed. By doing so the validation became as detached from the creation of the model as possible under the given circumstances. That both the validation for GSM and 3G networks resulted in the same mean error of $-1.39$ R-value units therefore implies that there might be a masking factor involved. The reasoning behind this is that impairments from packet loss and jitter may mask each other when both are present simultaneously. Taking this masking factor into account and combining it with $R_0$ from Equation (6.3) leads to an updated $R_0$ value of 65.39, as previously mentioned in Section 6.5.

It is not surprising that the validation for 3G networks resulted in a lower standard deviation of the error than for GSM networks. The lower ranges of impairments, jitter in particular, means that the resulting R-values are in a narrower interval which naturally leads to a lower standard deviation. The standard deviation of the difference between calculated and measured R-value is 1.45 for GSM networks and 0.79 R-value units for 3G networks. That is to be compared with a standard deviation within a call of 1.70, from Table 6.3, when the conditions are ideal and 5.04 R-value units, from Table 6.4, when the network resembles GSM. From this comparison it becomes clear that higher accuracy can not be expected given the variations within a call under consistent circumstances.
7.8 Method Criticism

The initial desire was to make the model as general as possible. However, after studying the litterature it could be deduced that it would not be entirely possible since it showed that the jitter-buffer [21] and the codecs used [42] impacts the final result. Furthermore, the results from this work showed in Section 6.2.4 that the packetizer also impacts the outcome. The model produced is therefore limited to the client and configuration used internally as Sectra Communications but the method can easily be applied to other configurations.

In a cellular network where one or both parties may be moving the network conditions can change rapidly during a call. In this work the network conditions have been the same during a call due to restrictions in data gathering. As discussed in Section 5.4.1 it would however be of interest to model burst properties of the network as well, if possible.

7.8.1 PESQ as Reference Value

As previously mentioned in the thesis, the PESQ algorithm does not take delay into account, unfortunately. Severe delay will clearly effect the conversational quality of a call but unfortunately no fully objective metric can reflect this. Subjective testing has not been feasible within the constraints of this thesis so a metric that can be calculated has been a necessity.

The literature also mentions the possibility to use other performance metrics than quality, such as intelligibility [45]. For example using speech recognition to capture the sound, send it as text and then use a synthesised voice for playback would probably score high for intelligibility but low in perceived quality. In our opinion however, one of the great benefits of voice communication is the possibility to perceive the subtle differences in intonation and speech rhythm which would not be captured using such a metric. Moreover, there is no reliable way to measure intelligibility objectively, at least to the best of our knowledge, which makes it infeasible for the work performed here.
Conclusion

This thesis has investigated the impact of different factors on the perceived voice quality in encrypted VoIP communication. From that investigation a parametric prediction model has been developed and evaluated. The questions constituting the thesis’ problem formulation, originally stated in Section 1.3, are hereby answered.

- How can the perceived audio quality of an encrypted VoIP call be predicted based on link statistics and information about used encryption schemes and codecs?

  - Which available parameters affect the perceived audio quality and to what extent?

Packet loss was shown to have a negative impact on the perceived voice quality in Section 6.2.1. It could be confirmed that the logarithmic impact of packet loss found in previous studies [10, 21, 41] also applied to this setup.

Jitter has not been thoroughly investigated before so it was interesting to find that the impact differed significantly below and above 40 ms, shown in Section 6.2.2. Under 40 ms jitter had a small impact on the QoS, while above 40 ms it reduced the quality rapidly. It is believed that the distinction at 40 ms depends on buffer sizes in the system together with how voice data is packetized to be sent over the network.

Because PESQ was used as a reference value the impact of delay could not be investigated since the PESQ algorithm compensates for delays before performing the comparison. Obviously severe delay affects the conversational quality of a voice call, as further discussed in Section 7.4. The suggestion is to use a simplification of the E-model only depen-
dent on the measurable delay, presented in a previous study, to account for the impact of the delay as well.

From the literature presented in Chapter 4 it can be stated that both voice-codec and encryption scheme affect the performance of VoIP systems. One difference between voice-codecs is the bandwidth they require which naturally impacts how well they can capture and recreate the audio. The impact of encryption have only been studied in LAN settings and the found differences were small enough to reasonably not have an impact when the encryption is used in a cellular network. It would have been interesting to evaluate these aspects but due to restrictions in the setup and the security classification of the parts closely related to the encryption, this was unfortunately not possible.

One of the bigger contributions by this thesis is the investigation of three different packetizers which was shown, in Section 6.2.4, to have a significant impact on how packet loss and jitter affected the quality. The results infer that there are two main aspects of the packetizers that affect the final quality. The use of redundancy clearly leads to a better quality in the presence of packet loss. How much voice data each packet contains is also related to the performance of the packetizer. Small amounts of voice data per packet will lead to smaller buffers which results in that substantial jitter on the network will manifest itself as packet loss since the packets will be dropped. Furthermore, less voice data per packet yields a larger overhead and does therefore have to be considered in relation to the available bandwidth in order to not congest the network. Large amounts of voice data per packet will, on the other hand, increase the end-to-end delay and enlarge the impact of packet loss.

For the configuration used in this thesis the measured difference in quality did not differ more than 5 R-value units (0.17 on the PESQ scale) when emulating a 3G network. From the authors’ experience it is questionable whether a human can make a distinction at that level with certainty. However, when emulating a GSM network, a rather bad one admittedly, the difference was as large as 23 R-value units (0.86 on the PESQ scale) which undoubtedly is significant.

- How can the prediction above be related to widely applied QoS metrics in industry?

The prediction model developed for this thesis outputs a R-value, same metric as the widely used E-model. With the conversion formulas stated in Section 2.3 this value can be converted to other industry standards such as MOS and PESQ. To facilitate the comparison of the results from this thesis the PESQ score is also given as a reference in the figures in Section 6.6.
8.1 Further work

It would be interesting to continue the work started here and investigate packetizers further. Preferably in a setting where the specific features of the packetizers can be discussed so that it hopefully can lead to a model based on these features rather than models specific to each packetizer. Aspects suggested to consider include milliseconds of voice per network packet, buffer sizes and the use of redundancy since the results in this work implies that those are the main parameters. The same goes for voice-codecs where aspects such as if the codec is predictive or not and the presence of packet loss concealment could be considered as potential parameters together with the required bandwidth. These investigations will presumably not be straight forward since, probably, the additive property of the different parameters will not hold. However, if a successful model could be derived then it will mitigate the process of predicting VoIP quality substantially since the need to model each new configuration would vanish.

It is my belief that different cryptographic primitives mainly affect the computational complexity at the clients and the computational delay that it leads to is negligible in cellular networks. On the other hand how cryptographic synchronisation is maintained may matter from a quality perspective. In particular, how well the system copes with packet loss should be related to this aspect. A study of this could hopefully bring some more definitive answers.
Appendix
A

Network Emulator Configurations

Listed on the following pages are the different configurations of the network emulator used to gather the necessary data for the analysis, as described in Section 5.4. Note that the delay listed here is the one-way delay and not the RTT.

If a setting is not mentioned for a test then it was set to not impair the network, e.g. for the Loss test no delays were introduced. Values that remain the same for consecutive calls are only stated when they are changed, e.g. for the LossGSM test the minimum delay is 270 ms for all calls.
<table>
<thead>
<tr>
<th>Call setup</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Loss</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>2</td>
<td>2.5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td><strong>LossHigh</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Latency</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>45</td>
<td>95</td>
<td>145</td>
<td>195</td>
<td>245</td>
<td>295</td>
<td>345</td>
<td>395</td>
<td>445</td>
<td>495</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>55</td>
<td>105</td>
<td>155</td>
<td>205</td>
<td>255</td>
<td>305</td>
<td>355</td>
<td>405</td>
<td>455</td>
<td>505</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Jitter</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>0</td>
<td>25</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>350</td>
<td>400</td>
<td>450</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.39</td>
<td>0.78</td>
<td>1.17</td>
<td>1.56</td>
<td>1.94</td>
<td>2.33</td>
<td>2.72</td>
<td>3.11</td>
<td>3.50</td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JitterLatencyOnly</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>350</td>
<td>400</td>
<td>450</td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JitterMoveOnly</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>0</td>
<td>0.39</td>
<td>0.78</td>
<td>1.17</td>
<td>1.56</td>
<td>1.94</td>
<td>2.33</td>
<td>2.72</td>
<td>3.11</td>
<td>3.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JitterPacketizer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>50</td>
<td>100</td>
<td>150</td>
<td>200</td>
<td>250</td>
<td>300</td>
<td>350</td>
<td>400</td>
<td>450</td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>0</td>
<td>0.39</td>
<td>0.78</td>
<td>1.17</td>
<td>1.56</td>
<td>1.94</td>
<td>2.33</td>
<td>2.72</td>
<td>3.11</td>
<td>3.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A.1: Continued from previous page.  
Network configurations for the different test runs.

<table>
<thead>
<tr>
<th>Call setup</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LossPacketizer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>(10)</td>
<td>(12)</td>
<td>(14)</td>
<td>(16)</td>
<td>(18)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LossGSM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>270</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>490</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JitterGSM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>380</td>
<td>350</td>
<td>320</td>
<td>290</td>
<td>260</td>
<td>230</td>
<td>200</td>
<td>170</td>
<td>140</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>380</td>
<td>410</td>
<td>440</td>
<td>470</td>
<td>500</td>
<td>530</td>
<td>560</td>
<td>590</td>
<td>620</td>
<td>650</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>0</td>
<td>0.39</td>
<td>0.78</td>
<td>1.17</td>
<td>1.56</td>
<td>1.94</td>
<td>2.33</td>
<td>2.72</td>
<td>3.11</td>
<td>3.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RandomGSM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>Normally distributed, N(2.5, 1.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>Normally distributed, N(200, 100)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>Normally distributed, N(400, 100)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>Normally distributed, N(2.5, 1.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A total of 230 calls were performed.
Table A.1: Continued from previous page.
Network configurations for the different test runs.

<table>
<thead>
<tr>
<th>Call setup</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random3G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random loss (%)</td>
<td>Normally distributed, N(2.0, 1.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay min (ms)</td>
<td>Normally distributed, N(60, 30)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay max (ms)</td>
<td>Normally distributed, N(140, 50)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (%)</td>
<td>Normally distributed, N(2.0, 1.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move (Min/Max)</td>
<td>1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


