Universität Zürich[UZH]

Quality-of-Experience Measurement Setup

Manuel Rösch
Zurich, Switzerland
Student ID: 10-918-332

Supervisor: ChristosTsiaras, Dr. Thomas Bocek
Date of Submission: August 7, 2014

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland

BACHELOR THESIS — Communication Systems Group, Prof. Dr. Burkhard Stiller

Bachelor Thesis
Communication Systems Group
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

The end-user's overall satisfaction is important information needed by Service Providers (SP) to improve their services. However, this so-called Quality-of-Experience (QoE) cannot be easily measured like technical variables can (*e.g.,* bandwidth and latency, in Internet-based services). QoE can only be estimated through mathematical models, or it can be measured indirectly through an experimental setup. The latest demands a controlled environment were end-user's feedback is collected and evaluated.

This thesis goal is to create a Voice over Internet Protocol (VoIP) QoE measurement setup. For this purpose a customized VoIP application based on Web Real-Time Communications (WebRTC) technology is implemented. The measured data from these experiments are used to define the necessary parameters of the Deterministic QoE model (DQX) [39], which is made to predict end-user's satisfaction in form of Mean Opinion Score (MOS). Furthermore, in this thesis the DQX results are compared with the results of two other QoE models: (a) the IQX Hypothesis [7] and (b) the E-Model [18] which is proposed by the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T).

The results of this work are: (a) the selection of all the parameters needed in the DQX model for the VoIP scenario and (b) a proposal for a correction of DQX that enable better results in the multiple network parameters consideration case.

Zusammenfassung

Durch Informationen über die Kundenzufriedenheit kann ein Serviceanbieter seinen Dienst stets verbessern und wettbewerbsfähig bleiben. Jedoch ist diese sogenannte Quality-of-Experience (QoE), im Gegensatz zu technischen Variabeln (z.B. Bandbreite oder Latenz in einem internetbasierten Service), nicht direkt messbar. Es ist lediglich möglich, diese entweder mit mathematischen Modellen vorherzusagen oder sie indirekt, anhand von Experimenten, zu bestimmen. Bei solchen Experimenten werden Kundenfeedbacks in einer kontrollierten Umgebung gesammelt und ausgewertet.

In dieser Arbeit geht es um den Aufbau eines solchen Experiments und dessen Verwendung für die Durchführung von Testanrufen mit einer selbstentwickelten, auf Web Real-Time Communications (WebRTC) basierenden Voice over Internet Protocol (VoIP) Applikation. Die in diesen Anrufen gesammelten Daten werden dann zur Definition der notwendigen Parameter des Deterministischen QoE Modells (DQX) [39] verwendet. Dieses Modell wurde entwickelt, um Benutzerzufriedenheit in Form von Mean Opinion Score (MOS) zu prognostizieren und es wird in dieser Arbeit mit zwei anderen solchen Modellen verglichen: (a) mit der IQX Hypothesis [7] und (b) mit dem E-Model [18], welches die offizielle Empfehlung der Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) ist.

Die Resultate dieser Arbeit sind: (a) die Selektion von allen notwendigen Parameter des DQX Models in einem VoIP Szenario (b) eine Formelkorrektur am DQX Models, welche bei der Berechung mit mehreren Netzwerkvariabeln zu besseren Resultaten führt.

Acknowledgments

I would like to thank Prof. Dr. Burkhard Stiller and the Communication Systems Group (CSG) for providing this interesting bachelor thesis and supporting its realization.

Special thanks go to my supervisor Christos Tsiaras for his great advices and patient support. Additionally, I would like to thank Andri Lareida for his help during the evaluation process and Dr. Thomas Bocek for his co-supervision.

I would also like to show my gratitude to all the people who took the time to participate my experiment. Especially, I would like to thank my former teacher Igo Schaller who provided me the opportunity to perform my experiment in his lessons.

Table of Content

1 Introduction

Quality-of-Service (QoS) is defined by the threshold of technical variables such as, latency, packet loss, and bandwidth. These variables are well known for different technologies and services and they can be easily measured. Furthermore, such variables are often used for marketing purposes. *E.g.*, Mobile Network Operators (MNOs) and Internet Service Providers (ISPs) advertise high bandwidth performance. However, QoS variables are not explicitly linked with the end-user's satisfaction. It is naive to conclude that the end-user's Quality-of-Experience (QoE) can be increased by simply adjusting one QoS variable, because the relationship between QoS variables and the end-user's experience depends on the Type of Service (ToS). A typical example is large latency which has higher negative effect on Voice over Internet Protocol (VoIP) services than on video streaming.

1.1 Motivation

With a determined relation between QoS and QoE, SPs can evaluate their service in an effective manner and hopefully find and improve existing bottlenecks so that end-users can benefit from better services. SPs would also benefit from maximizing end-user's satisfaction because that would increase their credibility and eventually their revenues. Thus, it is essential that a relationship between the easily measurable QoS variables and the hardly accessible QoE can be clarified.

1.2 Description of Work

This thesis is focused on defining the relationship between QoS variables and the QoE in the VoIP scenario. The 4-step QoE formalization road-map is:

(1) Create an experimental setup that allows the emulation of various network connection performance. This setup supports on-demand adjustment of four QoS variables (jitter, latency, packet loss and bandwidth). In the development process a Web Real-Time Communications (WebRTC) VoIP client which can be used to collect user feedback from experimental VoIP calls in different network scenarios, was developed.

(2) Perform various test calls in predefined experimental setups and collect QoE-related feedback from end-users.

(3) Use the collected feedback to evaluate the Deterministic QoE model (DQX), and to determine through non-linear regression this model's importance factor *m* for different variables in the VoIP scenario.

(4) Compare DQX to two other QoE-predicting models.

1.3 Thesis Outline

The remainder of this thesis is structured as follows. The related work is discussed in the following chapter were the DQX model [39], the E-model [18] of the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), the IQX hypothesis [7], and the Mean Opinion Score (MOS) [14] that is used for measuring QoE are discussed. Next, the creation and utilization of the experimental setup and the experiments are briefly presented. The following chapter evaluates and compares the DQX model. Furthermore, a new equation for multiple variables QoE prediction and further calibrations of the DQX model are discussed. Finally, the closing chapter that summarizes this work, draws conclusions as well as critical thoughts and presents future work.

2 Related Work

This chapter gives an overview of three models that are used for predicting the user experience, as well as QoE measurement techniques and best practices.

2.1 QoE Measurement

The measurement of the user experience in the VoIP scenario is a challenging task, since there are some details concerning the experimental setup and QoE measurement process, that need to be defined. The key details are: (i) the location where the experiment should take place, (ii) the people that will participate in the experiments, (iii) how they can be encouraged to have a balanced and natural conversation, and (iv) how QoE can be measured. The ITU provides recommendation regarding these issues. Such guidelines are important to get accurate and comparable results. The ITU recommendations were used for the experimental setup of this thesis.

2.1.1 Experiment Setup and QoE Measuring Process

A first recommendation concerns the room and its environment. The participants of the experiment should be seated in two separate rooms with a volume of more than $20m^3$. There should also be a sufficient sound attenuation of the outside noise environment so that the noise level in the rooms is as low as in hospitals and libraries. It is also stressed that the rooms should not look like an experimental room but they should be favorably decorated and seem like natural environment to the participant [14].

Besides facilities, people (subjects) that form the sample of the experiment are an essential and crucial part of the process. They can be categorized into three categories: experts, experienced, and untrained/naive subjects. While expert and experienced subjects are familiar with QoE measurement and can therefore provide more elaborated feedback, the untrained subjects represent the vast majority of people who know almost nothing about the user experience collection and evaluation. Depending on the purpose of the experiment, a different category of subjects must be chosen. Often, the focus is on the untrained subjects since they represent accurately the opinion of the end-user. That is the reason why this was also the target category is in this thesis [14][17].

The next fundamental aspect to define is the type of test that will be performed. There are two kind of tests: conversation-opinion tests and listening-opinion tests. In the first kind of test two subjects are actively participating in a conversation. In the second kind of test there is only one participant who passively listens to some audio samples. The advantage of a conversation-opinion test is that it is similar to a real conversation and all its characteristics. However, the content of the discussion between subjects demands concentration. Therefore, subjects are less focused on their actual task of QoE evaluation. For that reason, in some cases it is necessary to use a listening-opinion test to have the people's full attention. In this thesis conversation-opinion tests are used and therefore the focus of the rest of the thesis will be on such tests [14].

Conversation-opinion tests lead to an additional issue concerning the conversation. It is challenging to make two possibly complete strangers having a naturally and balanced dialogue. Small talk can be difficult and it does not work between any two dialog partner. Hence, it needs special methods to stimulate a conversation. ITU recommends conversational tasks to solve this issue. It should be a task that fits to the experiment and the cultural factors of the subjects. Due to limited time it should also be a task that is quickly explained and easily learned. Moreover the task should not be too demanding so that it is still possible to focus on the experiment while performing it. The aim is that the task leads to a purposeful semi-structured conversation that is not too open but also not too structured so that there is still room left to develop a balanced opinion of the channel. Ideally the task stimulates a conversation with equal participation of both parties that is close to a real conversation. In the recommendation it is also mentioned that the conversation should have a natural beginning and a natural ending. That means all the talks should be initialized and terminated by the subjects [14][17].

To find such a task that meets all this requirements is complicated. Therefore, the ITU has provided reference tasks. One task is to let the subject order postcards by preference or importance. Another task is to let the subjects be part of a typical role play of everyday life. There are predefined scenarios like railway inquiries, the rental of a car, or booking holidays. Every scenario has fixed roles and predefined goals which have to be achieved. Other task proposals are games that can be played by speaking. The famous game Battleship is one of these games. There is also a suggestion of a data exchanging game where the aim is to exchange or synchronize data as fast as possible [17].

2.1.2 Mean Opinion Score (MOS)

To capture the user experience in the QoE experiments, a five-point opinion scale, that is suggested by ITU, was used. This opinion scale is used in many QoE related researches and therefore it is a good choice to use it so that the results are comparable to the results of others. In this scale there are scores from one to five and each score has a certain meaning. The recommendation assigns following English words to each score [14]:

Table 1: MOS Levels of End-to End Perceived Quality

Score	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

If the experiment takes place in a different language, equivalent wording should be used. To clarify the meaning of each score, an illustration was created that comprises additional explanation for each score (cf. Figure 1). This Illustration was also used to explain MOS to the subjects at the beginning of each experiment [14].

In the experiment for every scenario a rating from one to five is collected. Then, the mean for each scenario is calculated out of the collected data which results the MOS [14].

The MOS is manly used for capturing the subjects' opinion on the performance of a telephone transmission system. It is used either for scenarios where a conversation take place or for scenarios where subjects only listen to spoken material. From this difference comes the first subdivision of MOS: (a) Listening Quality (LQ), and (b) Conversational Quality (CQ) [13].

Figure 1: Explanation of the MOS as Presented to the Subjects

Besides this subdivision the MOS can also be categorized according to the method used for producing the score: (i) Subjective (S) which means that the MOS is produced by human subject, (ii) Objective (O) which means that the MOS was predicted by perceptual models, and (iii) Estimated (E) which means that the MOS was produced by parametric models. Table 2 provides an overview of the entire MOS terminology [13].

Table 2: MOS Terminology

	Listening-only	**Conversational**
Subjective	MOS-LQS	MOS-CQS
Objective	MOS-LQO	MOS-CQO
Estimated	MOS-LQE	MOS-CQE

The Subjective MOS (MOSs) is according to [6] the most accurate measurement method followed by Objective MOS (MOSo) and the least accurate is the Estimated MOS (MOSe). This thesis makes use of MOS-CQS and MOS-CQE. The MOS-CQS is used during the performed experiments and in the analysis part there is the DQX model which takes typical service variables and the calculated output corresponds to the MOS-CQE [6][13].

2.2 E-Model

The E-Model is a transmission planning tool that can be used to predict QoE for a typical telephone user in a complete end-to-end conversational scenario. The model takes a wide range of transmission variables into account and it can be used to assess the voice quality of wired and wireless scenarios, based on circuit-switched and packet-switched technology [10].

The output of the model is in contrast to the other models that are presented in this chapter not in the form of a MOS but it uses the Transmission Rating Factor, *R* as output. However *R* can be transformed into MOS and therefore is also possible to compare it to the other models [10][18].

The E-Model uses mathematical algorithms that are based on the analysis of a large number of subjective tests with a wide range of transmission variables and these algorithms can transform transmission variables into so called “impairment factors”. There are according to [10] five impairment factors used to calculate *R* values:

Ro: This factor represents the basic signal-to-noise ratio what means that the received speech level is compared to the circuit and acoustic noise

Is: This term take impairments into account that more or less simultaneously with the voice signal. Examples are too loud speech level (non-optimum OLR), Non-Optimum Sidetone (STMR), Quantization Noise (qdu)

Id: This factor represents all impairments that are caused by delay and echo effects

Ie,eff: This term is called “effective equipment impairment factor” and represents the impairments that are caused by the used codec and packet-loss

A: Is the advantage factor and considers the advantage of service access. *E.g.*, a user that is connected in a hard-to-reach region expects a lower quality and therefore it tolerates more impairment.

Out of the previous mentioned impairment factors, Equation 1 is used for the *R* value calculation [18]:

$$R = Ro - Is - Id - Ie, eff + A \qquad (1)$$

All the impairment factors are calculated through algorithms that takes several transmission variables as input. An overview over all the variables that are used for the calculation is illustrated in Figure 2 were a reference connection of the E-Model is shown. The ITU-T recommendation G.107 [18] has more detailed calculations of each impairment factor.

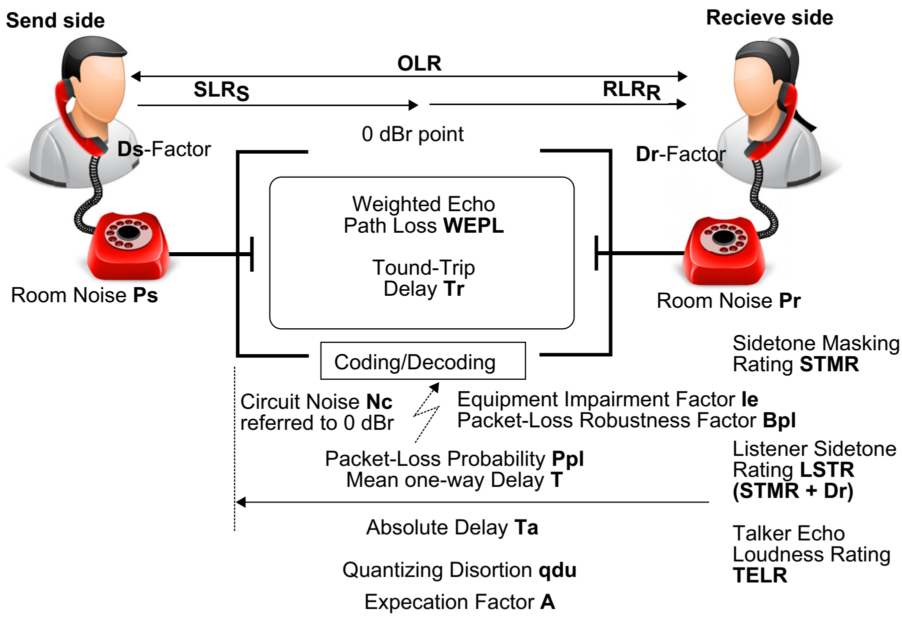

Figure 2: Reference Connection of the E-Model [10]

2.3 IQX Hypothesis

The IQX hypothesis [7] proposes a generic equation which predicts QoE for specific QoS variables. While other equations in this field are based on a logarithmic relationship between QoS and QoE such as the well-known ITU-T equation in [12], or the equation in

[23], the IQX hypothesis uses an exponential approach.This approach is proven to be more accurate through non-linear regression and comparison of the correlation coefficient [7].

The exponential equation is derived from the idea that the change of QoE depends on the current level of QoE, given the same amount of change of the QoS value. When a linear dependence on the QoE level is assumed, the relationship can be written as a differential equation and this equation can be resolved to the final equation of this Hypothesis (cf. Equation 2) [7].

$$\frac{\partial \mathrm{QoE}}{\partial \mathrm{Qos}} \sim -(\mathrm{QoE} - \gamma) \rightarrow \mathrm{QoE} = \alpha \cdot e^{-(\beta \cdot \mathrm{QoS})} + \gamma \tag{2}$$

In contrast to this exponential approach, logarithmic equations can be expressed through the differential Equation 3 and they are based on the idea that the change of QoE depends on the negative reciprocal QoS value [7].

$$\frac{\partial \mathrm{QoE}}{\partial \mathrm{Qos}} \sim -\frac{1}{\mathrm{QoS}} \tag{3}$$

As seen above, the equation of the IQX hypothesis (cf. Equation 2) contains the three parameters *α,β* and *γ* which must be determined by experimental sessions where non-linear regression is applied to collected MOS. Such a regression was made in a VoIP scenario for packet loss and the resulting equation, including the found values for α,β and γ, has the following form [7]:

$$\mathrm{QoE} = 3,010 \cdot e^{(-4,473 \cdot p_{\mathrm{loss}})} + 1,065 \tag{4}$$

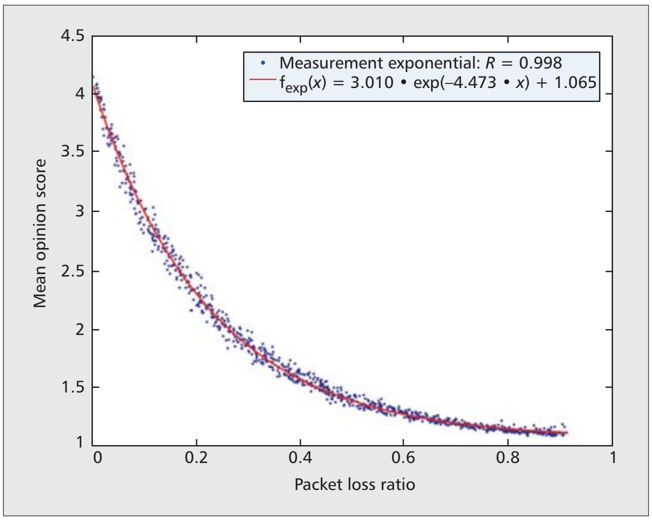

Figure 3: The IQX Hypothesis Applied to Collected MOS [7]

In the legend of the plot in Figure 3 can be seen that the *R* value is high for the packet loss scenario. This high *R* value supports the hypothesis that QoE and QoS are linked in an exponential way. Besides this scenario, other similar high correlations where achieved in other scenarios with packet reordering, jitter, bandwidth variations and different response times. It turned out that the correlation is in several scenarios higher with this exponential than with a logarithmic approach [7].

2.4 DQX-Model

The DQX model [39] uses, like the IQX Hypothesis, an exponential approach to link QoS variables and QoE. The difference between the two models is that DQX is deterministic and it proposes a way to calculate the QoE with multiple QoS variables as an input. This thesis has its main focus on this model and the whole analysis and evaluation part is referred to it [39].

For every service, there are diverse technical and non-technical variables which affect QoE. The model distinguishes between two types of such variables: There are Increasing Variables (IV) which increase the user satisfaction with their growth and there are Decreasing Variables (DV) which do the opposite. For all this variables there is a certain value where the user is satisfied with the service. These values are called Expected Variable Value (eV^2) and they are either defined in the Service Level Agreement (SLA) between the service provider and the customer, or by service-specific constraints. The eV^2 in the DQX formalization is the x_0 value and the end-user satisfaction that is reached at this point is defined as e_0 [39].

The idea of the model is, that there is a minimum user satisfaction called μ and a maximum user satisfaction called M. The IV start in the point μ and go through the x_0 point towards M. For DV it is vice versa. They start with the user satisfaction M and decrease through x_0 point towards μ. For a better understanding, Figure 4 illustrates an exemplary plot of an IV and DV of the DQX model [39].

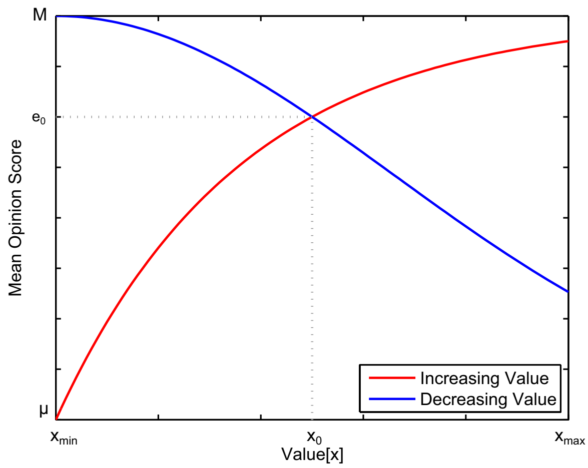

Figure 4: MOS Evolution for IV and DV in the DQX Model [39]

Since the shape of the graph differs for each service, technology, and user-base, the model introduces an influence factor m. As illustrated in Figure 5 this m value leads to different shaped graphs. For small values the graph gets flat and for high values of m it gets steep [39].

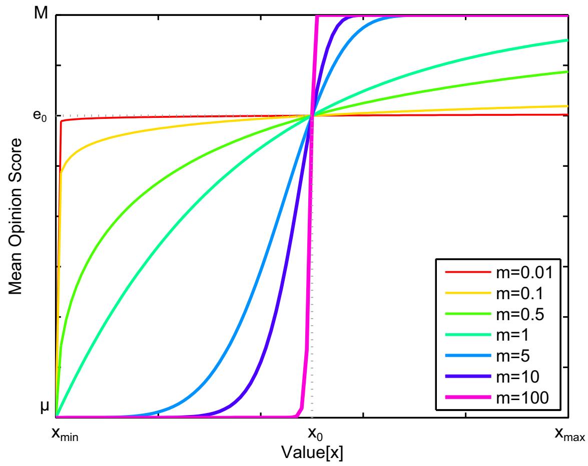

Figure 5: Plot of the DQX Model for Different m Values [39]

The equation for IV is [39]:

$$e_i(x) = h \cdot (1 - e^{-(\lambda \cdot x^m)}) + \mu \tag{5}$$

The parameter *h* stands for difference between the maximum and the minimum QoE score ($h = M - \mu$), *m* is the influence factor and *λ* is an exponent that is defined through the point in x_0. Equation 6 shows how *λ* can be derived from Equation 5 [39].

$$e_i(x_0) = e_0 \Leftrightarrow \lambda = x_0^{-m} \ln\left(\frac{h}{h - e_0 + \mu}\right) \tag{6}$$

For the decreasing variable there is an analogous Equation 7 where the factor *λ* is defined equally [39]:

$$e_d(x) = h \cdot e^{-\lambda \cdot x^m} + \mu, \lambda = x_0^{-m} \ln\left(\frac{h}{e_0 - \mu}\right) \tag{7}$$

The mathematical proof of all the equations above can be found in the paper [39].

Furthermore, the model proposes two distinct *m* values for the graph depending on the position relative to the x_0 value. The *m* value used for the graph above x_0 is called m^+ and the *m* value used for the graph below x_0 is called m^-. There is an equation to directly define these values using one point above and one below x_0 and the point x_0 itself. But since this equations are not used in the analysis they will not be presented here [39].

As mentioned at the beginning of this section, the model also introduces an equation for multiple variables. The equation uses the single variables equations (Equation 5 and Equation 7) and combines them. In the original equation for multiple variables there was a contradiction and during this thesis it could be corrected. The new DQX equation is [39]:

$$E(x) = \mu + h \cdot \prod_{k=1}^{N} \left[\frac{e_{i \vee d}(x_k) - \mu}{h}\right]^{w_k} \tag{8}$$

The thought behind this equation is that the QoE reduction for all the QOS variables is multiplied and weighted with the exponent w_k depending on the relevance of the particular variable [39].

Because the model has no defined rating as an outcome, the maximum value *M* and the minimum value *μ* must be defined in advance and the output will be according to that. In the case of MOS, the rating used in this thesis, the maximum value is *M*=5 and the minimum is *μ*=1. The difference *h* between the parameters is therefore 4. Inserting these parameters into the previously mentioned equations (cf. Equation 5, Equation 7 and Equation 8) the following equation for the DQX-Model using MOS results [39]:

Increasing Variable:

$$e_i(x) = 4 \cdot (1 - e^{-(\lambda \cdot x^m)}) + 1, \lambda = x_0^{-m} \ln(4) \tag{9}$$

Decreasing Variable:

$$e_d(x) = 4 \cdot e^{-\lambda \cdot x^m} + 1, \lambda = x_0^{-m} \ln\left(\frac{4}{3}\right) \tag{10}$$

Multiple Variables:

$$E(x) = 1 + 4 \cdot \prod_{k=1}^{N} \left[\frac{e_{i \vee d}(x_k) - 1}{4}\right]^{w_k} \tag{11}$$

3 Experimental Design

Before the QoE measurement experiments could start the decision which technology to use for implementing the VoIP QoE evaluation setup has been taken. Basically there were two options: a traditional approach with the Session Initiation Protocol (SIP) or an approach with WebRTC. Since the latter is a new, promising technology that will probably overtake a big part of the communication market, it was clear that the VoIP service will be based on WebRTC.

3.1 WebRTC

WebRTC is an official World Wide Web Consortium (W3C) draft standard for real-time communication between browsers [43]. The goal of WebRTC is plugin-free low-cost communication in real-time between any browser. And with communication is not only meant audio and video communication but also the direct exchange of data. So with the help of WebRTC it is possible to create a Peer-to-Peer (P2P) connection from browser to browser and send audio, video and data over it [40].

Since it is a draft standard, so far the technology is only supported by the newer versions of the Firefox, Chrome and Opera browser and except for the Chrome browser the mobile version is not supported. Since this three mentioned browsers have a market share of around 65% worldwide [2][46] many people can be reached with the WebRTC technology. However, two commonly used browsers, Safari and Internet Explorer do not support it yet [40].

3.1.1 Architecture and API

WebRTC offers two Application Programming Interfaces (APIs). The first API that is in JavaScript and edited by W3C is created for web developers. The other API is for browser developer and written in C++. A creator of a web application needs the Web API and this Web API can then access through the C++ API all the functions that are provided by WebRTC [45].

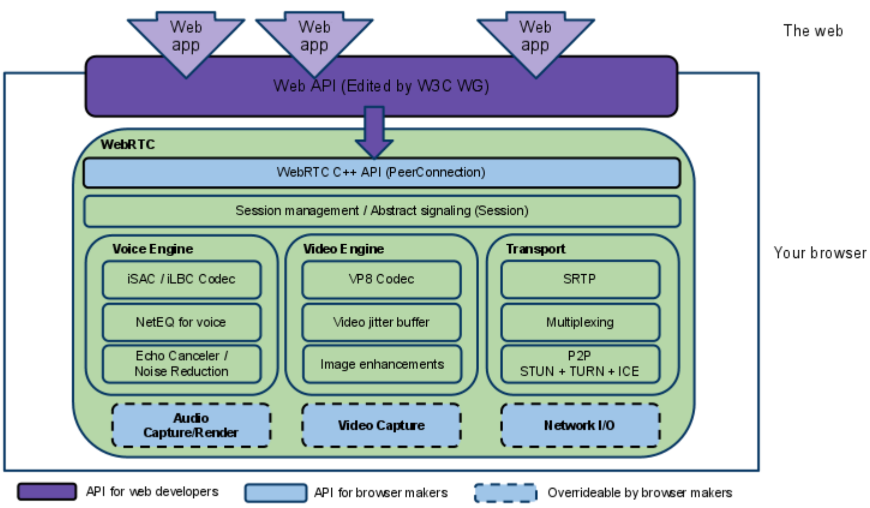

Figure 6: WebRTC Architecture Overview [45]

Due to the fact that during this thesis only the JavaScript Web API was used, the C++ API will not be explained any further at this point. The WebRTC JavaScript Web API offers basically three functions: Firstly the MediaStream (aka getUserMedia) API can be used to access the device's internal or external microphone and camera. Secondly, the RTCPeerConnection API is used for creating and maintaining a P2P connection, and is thought to exchange audio and video. Thirdly, if data exchange is needed, the RTCDataChannel API provides this functionality [40][43].

3.1.2 Signaling and Communication

As mentioned before, the communication happens directly between two browsers through a P2P connection. To set up that connection the first step is that every peer gets its public Internet Protocol (IP) address. This is achieved by accessing Session Traversal Utilities for NAT (STUN) servers. These servers are simple with low energy and hardware demands and therefore many companies such as Google and Mozilla run a public available STUN server. With the access the peer gets its public IP and the next step is now the signaling [40][43].

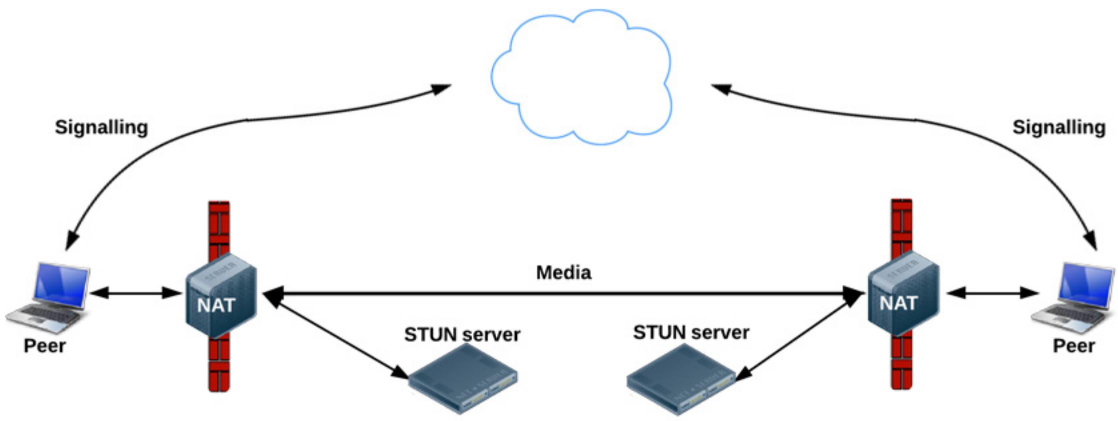

Figure 7: WebRTC Communication Scheme [40]

The signaling is indicated by a cloud in the illustration above. That is because it is left open to the developer how to implement it as long as it fulfills it purpose and the necessary information to establish a P2P connection is exchanged [43]. The VoIP application that was originated during this thesis uses a node.js server to accomplish the signaling.

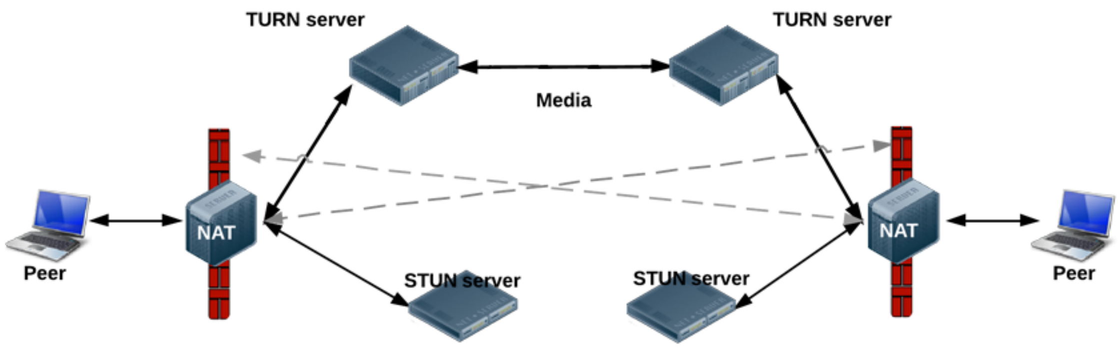

Figure 8: WebRTC Communication Fallback Mechanism [40]

A P2P connection is not always possible since unsupported network topologies and firewalls prevent a direct exchange of media and data. Thus, Traversal Using Relays around NAT (TURN) servers have been introduced as a fall-back method in case that a P2P connection is not possible (cf. Figure 8). TURN server act as an intermediate link so that it is not a P2P connection anymore but both peers send and receive data through a server. This has the disadvantage that it causes traffic on a server and that means costs to the provider. In around 85% of the calls a direct connection is possible and the TURN server is not used [40]. The decision which connection type to use is made automatically by the Interactive Connectivity Establishment (ICE) peer connection framework. So the developer does not need to care about it and a P2P connection is established whenever possible [40][43].

3.2 The QoEssenger

The QoEssenger is the messenger that was developed during this thesis for the purpose of experimental VoIP calls. The QoEssenger is based on the WebRTC technology and thus it works as in-browser web application that communicates through a P2P connection. It is written in HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript and for saving the ratings there are Hypertext Preprocessor (PHP) scripts. Moreover, the JavaScript library jQuery is used for User Interface (UI) effects.

Figure 9: Interface of the QoEssenger

The key feature of the QoEssenger is, that it looks familiar and pleasant to the experiment participants. That is necessary because the less people feel like being in an experimental environment; the more accurate are the answers they give. Because of that, the messenger has a user-friendly interface with additional functionality like a chat, smiles, ring tones and video-chat support (cf. Figure 9).

An important functionality of the QoEssenger is the rating feature, which is used for collecting user feedback. The rating is inspired by the star rating system that is used on many Internet pages for movie ratings, on-line shop products or hotels. After every call that is made with the QoEssenger the MOS rating field, as shown in Figure 10, appears. The MOS is represented by five stars and when the subjects holds his mouse pointer over the stars he gets a short explanation what this rating means. Like this the subjects feels familiar while using the rating and he has the possibility to reread the meaning of the rating whenever he wants.

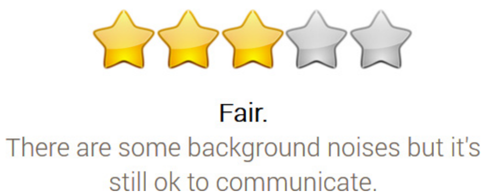

Figure 10: Rating System of the QoEssenger

3.3 Experimental Setup

The experimental setup considers three constraints. (1) An architecture that manages the entire experiments process with the minimum human-overhead. (2) Hardware decisions that affect in a minimum way the results, and (3) the experiment process which should also not interfere with the results.

3.3.1 Architecture and Software

There are two different ways how to emulate a network. There are either an expensive hardware solutions where a device is bought that is specially made for emulating different network topologies (e.g., Apposite [2] or GigaNet Systems [8]), or a software solution is used which fulfills the same task as the hardware (e.g., WANsim [9] or WANem [38]). After the evaluation of the previously mentioned options the decision to use the software solution WANem has been taken. This software is convincing because it is based on the well-accepted open source network tool for Linux called NetEm [26] and there is also the possibility to build a UI on top of it. Like this it is possible to create a comfortable control panel that meets exactly the demands of the experiment.

The architecture using WANem is illustrated in Figure 11: There are four computers connected through in a local Local Area Network (LAN) through a switch. Two computers run the QoEssenger, one computer run a web server, a signaling server as well as a data base and the last computer runs the WANem tool that can emulate the network. Now the WANem works the following: The routing table of the two computers that run the QoEssenger is modified in such a manner, that all the packets are routed to the other peer through the computer that runs WANem. On this computer, the network emulation happens. So for example if the packet loss is set to 50%, the WANem computer will drop every second packet that is routed through.

.

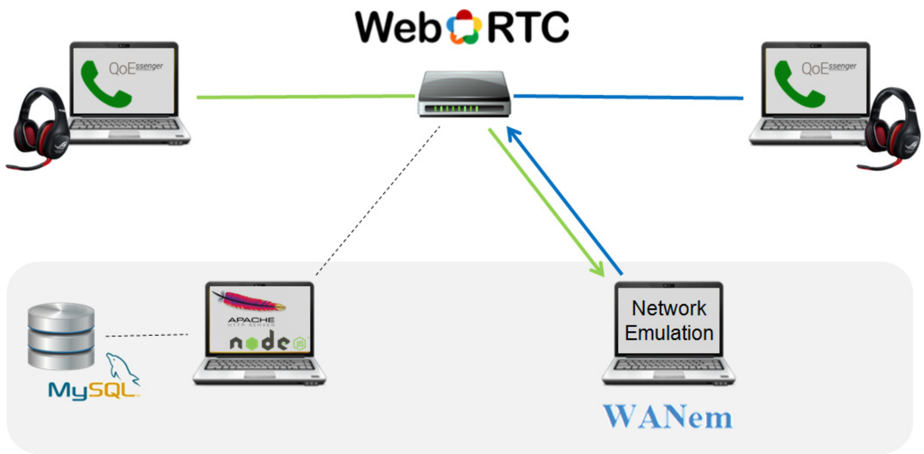

Figure 11: Architecture of the Experimental Setup

Such architecture with LAN cables and a switch is necessary to guarantee a controlled network environment without the interferences that happen in a Wireless LAN (WLAN) network. WANem was running on a virtual machine on the computer with the web server and data base. This computer runs as Operating System (OS) a Windows 8.1 and the tool used for virtualization is a freeware called VirtualBox.

Besides this virtualization tool a lot of other software is used in this architecture. As a web server the popular open-source server Apache is used, as signaling server there is a node.js server and the database which stores the user ratings is a MySQL database. On the two computers that are used for running the VoIP application have the most recent version of the OS Ubuntu (14.04) installed and as browser the latest version of Firefox (30.0) is used to run the QoEssenger. The reason for that is because Firefox allows an automated start of the calls and that is not possible with Chrome or other Browsers [5]. Other software that was used is XAMPP [1], a tool to install, start and maintain Apache and MySQL.

The software used for the purposed of this thesis is considered a standard solution [1] and because of that its usage is not explained any further. The only exception is the node.js server which is a relatively young project and one of the possible ways how to do the signaling in WebRTC. There is an interesting concept behind node.js as it is possible to write with only a few lines of codes powerful server-side networking applications and furthermore it is extremely fast and low-cost. That is the reason why big companies like Microsoft, Yahoo or Wallmart use this technology [20].

More information about all the before mentioned software like the used version number and the official website can be found in Appendix C.

3.3.2 Hardware

The hardware decisions were less complex than the software decisions because the whole project needs only three computers, one switch and three LAN cables. The first approach was using the existing infrastructure of the mobile systems lab of the University Zurich. However, it turned out that this approach was not sufficient since in an acceptable experimental setup subjects are in two separate rooms. Furthermore, for experiments outside the University premises it would help to have a portable experimental setup. For this reason, the decision fell to three laptops: a HP EliteBook 8440p and two Lenovo Thinkpads (T133 and T345).

Besides the computers there was also the decision about which audio input device to use. There is the possibility to use the laptops internal microphone and speakers, a standard low-cost headset, or a high-level gaming and entertainment headset. The experiment was made with high-level headset for the following reasons. Firstly, the user rating should not be a rating about the hardware but a rating about the connection quality. A high-quality headset with a good microphone and earphones eliminates any quality issues caused by the hardware. Secondly, the head-set used in the experiments had complete over-ear coverage what leads to a noise isolation for high frequency human and environmental noise. This means that the subjects are less influenced and can focus better on the phone call.

3.3.3 Planned Procedure

Besides the hardware and software-related decisions there were also decisions to make concerning the procedure of the experiment. Firstly, the number and the duration of the test calls should be defined. Since it is an experiment about the human experience it should not be longer than 30 minutes. Otherwise people become probably annoyed and/or bored and the answers are influenced by these emotions. Having a fixed interview length the decision about the number and the duration of the test calls was a trade-off between the number of measurements and the accuracy. Thus, if the test calls are longer, less can be performed within this fix time period. After performing different durations, the final decision about the procedure of the interview was like this:

0-5 min Introduction, explanation of the experiment and rating system

5-25 min 16 Test Calls, around 45 seconds calling time + 15 seconds voting time each

25-30 min Question and Answers about the calling experience

As a next step, a further planning of the test calls was necessary. It is assumed that people are not able to have a free and balanced 45 second conversation on their own since it does not seem to be enough time to develop a proper conversation, especially between strangers. Thus, some playful approaches were evaluated to support the conversation. The first approach was a word guessing game called Taboo. In this game, a word has to be described without using that word and some defined related words. However, 45 seconds are not long enough to play that game and the conversation is not that balanced. Another idea was a conversation game where two people developing a story by alternately saying one sentence that uses the last word of the previous sentence. This game has good a speaking balance but it requires a lot of concentration and thinking power so that it is difficult for the subjects to focus on the actual call quality evaluation. Finally, the following method was used to support the conversation in the test calls: At the beginning of the experiment each participant gets around 300 easy general knowledge questions [22][34] and the subjects have two ask and answer them alternately. This approach leads to a fluent and balanced conversation without distracting the subjects from their evaluation task.

Figure 12: Picture of the Experimental Setup with two Subjects

The Figure 12 shows exemplary the experimental setup with the hardware used in this experiment and two subjects that took part in the experiment. It is to notice that the person on the right will leave the room for the experiment. The supervisor of the experiment sits at the computer on the left.

3.4 Network Emulation with WANem

WANem is the network emulation tool that was used in the experimental setup of this thesis. WANem is an open source project from the Performance Engineering Research Centre, TATA Consultancy Services in Mumbai India and has won the Free and Open Source Software (FOSS) India Award 2008. A computer that runs the tool is needed, and all the packets must be routed through this computer. Different variables like latency or packet loss can be adjusted either by using the provided UI or by executing PHP scripts. The latter technique was used in this thesis to build a customized UI [38].

Moreover, the tool does not have to be installed in a computer, since it is based on a live Linux OS called KOPPNIX [24] that can be booted directly from a (Compact Disc Read-Only Memory) CD-ROM with all its functionality [38].

3.4.1 Architecture

The architecture of WANem is illustrated in Figure 13 and operate like this: Since WANem runs on a Linux based OS, it can make use of one of the popular network emulation tools for Linux called NetEm [26] which is an accurate tool for network emulation [21]. Building on this, there is a modified version of the tool PHPNetEmGUI. As the name already indicated, this tool can be used for running and controlling the NetEm tool via PHP access. Such a tool that uses PHP must have a PHP-capable web server in the background. Therefore an Apache server runs in the background. On top of all that there is the control panel that was used for running and managing the experiments. This control panel makes use of the PHP scripts that are provided by the PHPNetEmGUI [24][38].

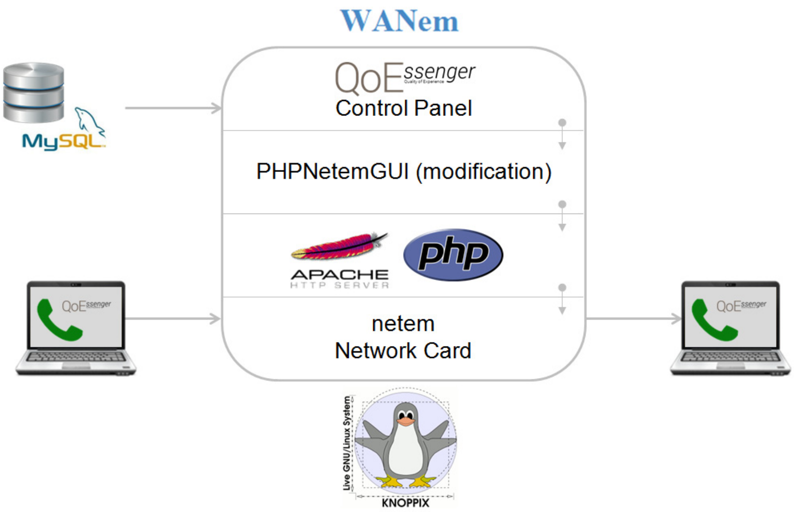

Figure 13: Architecture of WANem

3.4.2 Core Application: NetEm

NetEm is the core of the QoE experiments setup and therefore it was necessary to examine this tool thoroughly, since for accurate results a precise emulation is needed. A helpful and complete evaluation of this tool was found at [21]. In this paper an empirical study was made for evaluating NetEm. The conclusion is that the tool works fairly accurate for the purposes of this work, because there are only accuracy troubles using low values such as 1ms delay. Since such low values are not tested in this thesis the use of NetEm is suitable [21].

Nevertheless, the accuracy of NetEm was tested after the implementation of the measurement setup with the help of standard Linux network monitoring. For packet loss and delay the tools mtr [25] and ping were used, and for the bandwidth the tools iperf [41] and nload [37]. The outcome of this testing confirmed NetEm as an accurate tool suitable for the purposes of this thesis.

3.5 Control Panel

A central part of this thesis is the control panel. It is made to facilitate the experiment procedure through the automation of different network emulation scenarios as well as the collection, storage and visualization of the user ratings. The control panel is built on top of the WANem tool and it uses its PHP scripts to control the network emulation.

The UI of the control panel consists of four areas: a timer, network settings, database information and a results overview. Each part has a certain function during the experiments:

The timer area is used to set the calling time, the voting time and the number of runs. There is also a play button in it that is used to begin the experiment session. After the experiment is started, there is a countdown in this area which shows how much time is left and if the subjects are in calling or in voting mode. This area also shows the estimated that is left for the entire experiment session.

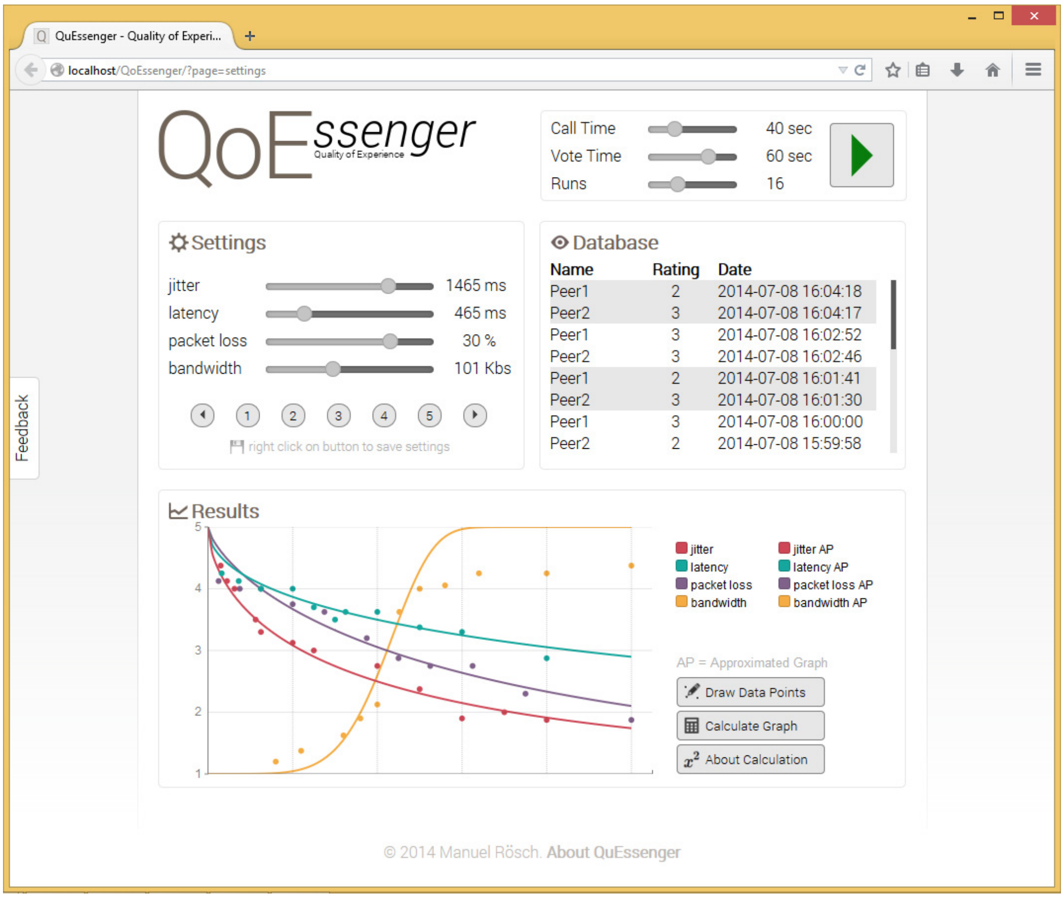

Figure 14: Interface of the Control Panel

The network settings are in the middle on the left side. In this area different network scenarios can be set. These scenarios can be saved to the numbered button. A right click on such a button saves the settings and a left click loads it. This network settings area interacts with the timer area: As soon as the play button in the timer area is pressed the settings are automatically loaded in increasing order after every calling period. So if the button with the number 15 is selected and then play is clicked, the control panel will automatically go through all saved networks scenarios starting from the 15th scenario.

The database area is in the middle on the right and it shows the last database entries. This section is necessary to control the rating process. It can be seen if there are troubles with rating, or finish the voting time earlier when it is seen that both participants of the experiment have provided their ratings.

Last but not least, there is a results area on the bottom of the screen. This area draws the collected data points in a grid. Moreover, it is possible to provide basic analysis possibilities, such as change the input parameters, or perform a least square approximation to the DQX model. These functions are mostly useful between experiments. Thanks to this area it is possible to find more variable values that should be tested, or export preliminary assumptions, that allow to adjust the input parameters properly. Finally, it can be used to demonstrates to the subjects what the experiment is all about.

4 Evaluation and Results

The following evaluation is based on the MOS of 34 subjects which produced in total more than 500 end-user's opinion score ratings in an overall calling time of approximately 6 hours. 80% of ratings were collected in a single variable scenario where only one variable was adjusted. The rest of the ratings were mixed variable scenarios where multiple variable were adjusted.

4.1 About the Evaluation

All the calculation and creation of the plots is fully automated using MATLAB and its statistic toolbox [32]. The MATLAB-scripts as well as all the implemented tools that were used for the evaluation can be found on the attached CD-ROM. Furthermore, in the Appendix C of this thesis there is a step-by-step manual describing how the automated evaluation can be executed. During this evaluation, the following steps are performed: (i) First of all the MOS and the standard deviation is calculated using the raw data of the single variable scenarios. (ii) After that, the DQX-Model is fitted several times through the found data points to get the required values for influence factors *m*. (iii) 3D-graphs and an overview table is created out of the data from the experiments with multiple variables.

4.1.1 Evaluation Software: MATLAB

For processing and accumulating the data, the MATLAB tool `grpstats` was used. This tool allows grouping data by manually defined criteria and calculating standard statistics like the mean or the standard deviation. That way, the MOS and the standard deviation were calculated for each network scenario. MATLAB uses Equation 12 for calculating the mean $\overline{X}$ and the standard deviation S [31][33].

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i,\ S = \left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{\frac{1}{2}},\ n = \text{Number of elements in the sample} \quad (12)$$

To fit the DQX model through the previously calculated data point, a MATLAB tool called `fit` were used. This tool can perform non-linear regressions using the least-square method. The idea of this method is that the square of the distance between the data points and the resulting graph is minimized. This optimization is achieved by several iterations in which the free parameters are always adjusted minimally and the effect of its adjustment is analyzed and accordingly the adjustment of the next iteration is determined. These iterations continue until a specified convergence criteria are reached. So it is a heuristic trial and error method that uses computational power to solve the equation. In a mathematical expression the least square analysis can be formulated like this [28][29][30]:

$$\min S \mid S = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (13)$$

The output of such a fit are the respective values of the free parameters and moreover there is information about the Goodness of Fit (GOF). For this thesis the adjusted R^2 value is used to determine the GOF. This value shows on a scale from 0 to 1 how good the graph fits through the data points. A high value, close to 1, means that the graph fits well through the points, a low value means that it does not. Adjusted R^2 is defined as the ratio of the Sum of Squares of the Regression (SSR) and the Total Sum of Squares (SST) (cf. Equation 14) [27].

$$\text{Adjusted } R^2 = 1 - \frac{SSE(n-1)}{SST(v)} \;\mid\; SSE = \sum_{i=1}^{n} w_i \hat{y}_i - \bar{y}^2 \;\mid\; SST = \sum_{i=1}^{n} w_i y_i - \bar{y}^2 \quad (14)$$

4.1.2 Variables and Expected Variable Values x_0

The DQX model can be used with different technical and non-technical variables. In this work, the focus is on four technical variables: jitter, latency, packet loss and bandwidth. The DQX model needs for every technical variable an Expected Variable Value (eV^2) x_0 [39]. Since the WebRTC technology that was used in this thesis is relatively young, there not yet rich literature that can be used to find possible x_0 values. Thus, the following references were used for this evaluation:

Latency (x_0 = 150ms): This variable is technology and codec independent. Constant latency cannot be corrected like packet loss and it does not influence the audio quality. It is simply a delay from the speaker's mouth to the receiver's ear. The disturbance through this delay depends on the conversation and also on the habits of the user. Therefore, many different values can be found in literature. For this thesis the ITU-T recommendation G.114 [15] and G.1010 [11] is used. ITU states, that for almost all applications a latency till 150ms is satisfying to the user [11][15].

Jitter (x_0= 100ms): For jitter there are not many references in literature. Thus, a reference for WebRTC, which uses the relatively new Opus codec, could not be found. As a reference value a recommendation of Cisco, one of the world's leading network solutions provider, was used. Cisco recommends that jitter should not exceed 100ms to keep the user satisfied [4].

Packet Loss (x_0= 5%): On the official page of the Opus codec, which is used in WebRTC, can be found a document that presents the codec's robustness towards packet loss. Through advanced error correcting mechanisms it is possible to remain the user satisfied until a packet loss value of around 5% [35].

Bandwidth (x_0= 64 kbit/s): The default bandwidth for WebRTC is according to one of the project members of WebRTC 64 kbit/s [44]. This value was approved by test measurements made with the QoEssenger and standard Linux network monitoring tools.

4.2 Single Variables

This section presents results on test scenarios where only one variable was tested in an isolated test environment. This means that all the other variables where in a status where they do not have any influence on the connection. *E.g.*, in a test with 5% packet loss, the latency and jitter is set to 0ms and the bandwidth is unlimited. Most of the test scenarios were such single variable scenarios.

Each variable is separately discussed under three aspects:

Analysis A: In this part, a plot is shown which contains all the collected MOS as well as the standard deviation of the MOS for each tested scenario. Furthermore, there is a plot of the DQX model fitted through the collected MOS points as well as the found m^+ and m^- values. If available, there are also plots from other approaches like the IQX hypothesis and the E-Model.

Analysis B: This analysis is focused on the *m* values of the DQX model and their behavior. For this analysis, the DQX model was fitted through every neighboring pair of the collected MOS points separately and the results are plotted as a scatter plot. To visualize the development of the *m* values, a linear regression was made and included in this analysis.

Analysis C: This part of the analysis shows a bubble chart showing the distribution of the opinion score votes. The x-axis shows the tested scenario and the y-axis shows which opinion score was voted. The size of the bubble represents the number of people who voted for the particular score in the particular scenario, under the rule that the bigger the bubble the more votes are collected. So this part is about the subjects' individual votes and opinions. Especially in this part there will be presented some of the participant's statements collected during the interviews after the experiments.

4.2.1 Latency

Latency was tested in a range from 65ms to 1600ms. As x_0 value 150ms was used. The resulting *m* values from the fit are for m^+ 0.4 and for m^- 0.32. Comparing to the other variables this difference of 0.08 between m^+ and m^- is by far the lowest and it can be considered as quasi constant for values above and below x_0.

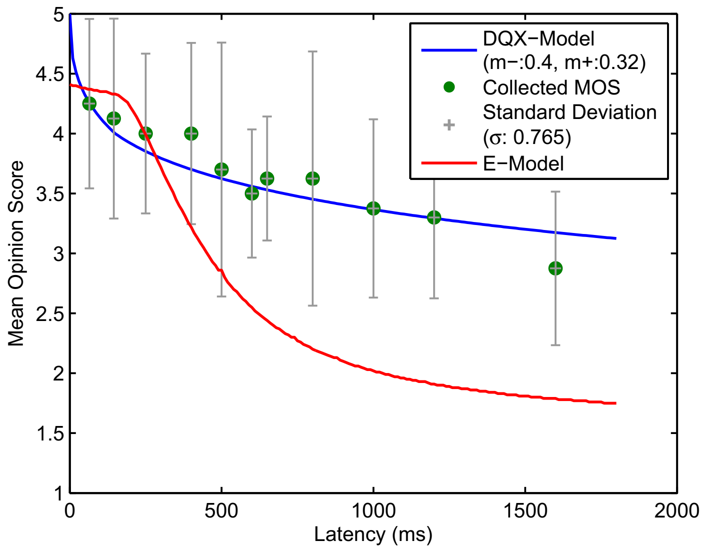

Figure 15: DQX-Model Fit and Comparison for Latency (A)

As shown in Figure 15 there is an agreeable fit between the collected MOS and the DQX model. The GOF expressed with the R^2 value is 0.75 for points below x_0 and -0.65 for points above. This are, however, the lowest correlation values of all variables. This relatively low correlation can be explained through the characteristics of the variable latency. There is a big variance how people experience latency and it is also depending on the type of conversation. That is the reason why different values for x_0 can be found in literature. This indeterminacy is also reflected by the standard deviation that is with 0.765 the highest compared to the others.

It is noticeable that the E-Model proposes most of the time lower MOS values than the fitted DQX model. So for example for a latency of 1600ms the MOS for the E-Model is 1.79 and for the DQX-Model 2.875. The latter value is probably too high for such a high latency value. The reason for such high values could be the following: Latency is something that is not directly annoying like a bad audio quality. It is something that gets more annoying the longer and faster a conversation becomes. Latency is not that disturbing in a short conversation with small talk characteristics. The conversations of the experiments had exactly these characteristics.

Since the values seem rather high, some extra experiments were performed with longer experimental calls in which only latency was tested. For these calls, three different conversational tasks proposed by the ITU-T where tested: a travel office role play, a random number verification task and a contacts exchange task [17]. The results of these extra tests where unexpected. The subjects rated still high. For a test scenario with 1500ms latency the MOS was still 3.17. Therefore it is not only the duration and the type of conversational task that is responsible for the high outcome. More likely it is a cultural phenomenon. As stated in [17] MOS can vary due to cultural differences. Except four subjects, all subjects spoke Swiss German which is a rather slow language and therefore latency probably disturbs less. Supported is this hypothesis by a test call between a Russian and an Italian participant held in English that seems to be faster and more interactive than most of the native Swiss German speaker's conversations. However, a proof for this phenomenon could not be found in literature and since the sample was not large enough it is only a hypothesis.

Thus, it is shown that E-model is not suitable for every scenario. The MOS depends on the service and the respective users. Therefore a model, such DQX, is demanded that can be calibrated to predict QoE accurately in diverse scenarios.

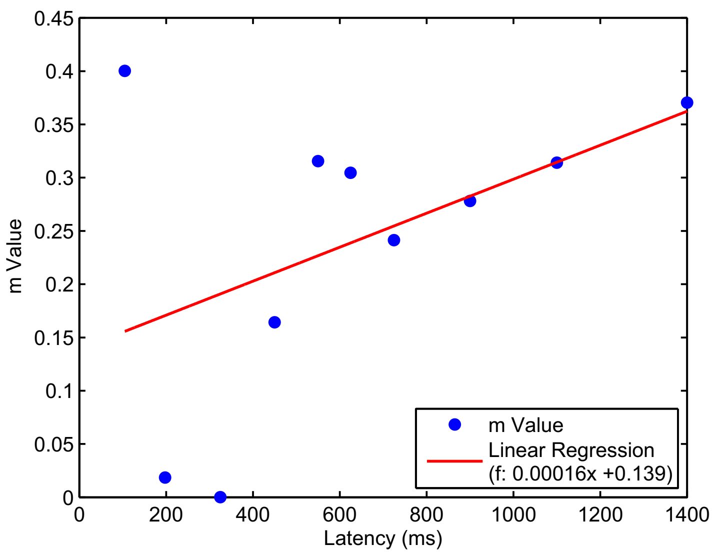

Figure 16: Development of the m Values for Latency (B)

Figure 16 illustrates the development of the *m* values. In contrast to the m^+ and m^- values, the *m* values by pairs is no longer constant. Here, after six irregular changes, the *m* value increases constantly with modest growth. Compared to the other variables, this growth is relatively low and almost negligible and considering the large standard deviation *m* could still be constant.

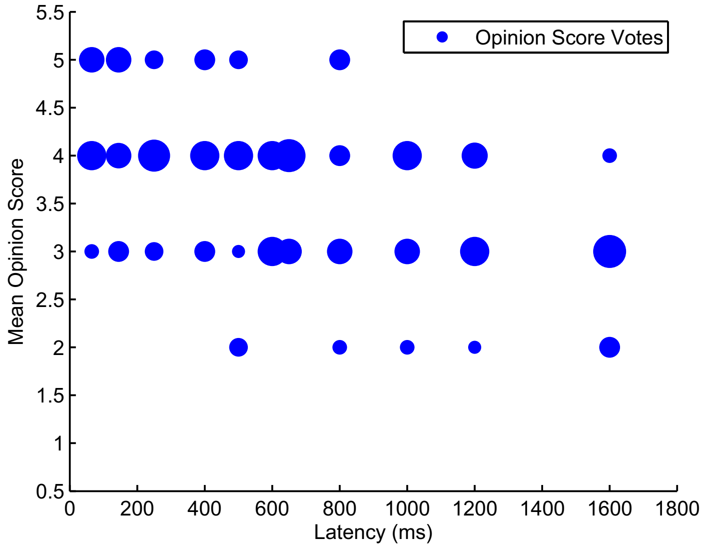

Figure 17: Distribution of the Collected Votes for Latency (C)

The bubble chart in Figure 17 shows again the rather too high votes and the indeterminacy of the votes. There are a lot of medium sized bubbles what means that the votes are widely distributed. In the discussions about the test calls the subjects gave further explanation for this vote distribution. There were people who could, for whatever reason, not notice any latency at all. Others explained that they associate the delays not with a bad connection but with a temporarily slow responding of the conversation partner and therefore they did not consider it for the voting. Last but no least, the subjects‘ experience with Skype did also influence the votes. Four subjects mentioned that the started to compare the conversation with the Skype conversations they had and that they use Skype usually only to make long distance calls from an Internet cafe with a slow and unreliable connection. Like this, the expectations were low and the votes high. It can be assumed that not only this four subjects are biased by Skype. Since most of the people still use their phone or mobile phone to make calls, it is highly likely that lots of them switch their perspective as soon as they sit in front of a computer and wear a headset.

4.2.2 Packet Loss

Packet loss was tested in a range from 1% to 40%. As x_0 value 5% was used. The resulting *m* values from the fit are 0.085 for m^+ and 0.73 for m^-. The difference between these parameters is relatively high and therefore *m* does not seem to be constant for values above and below x_0.

As can be seen in Figure 18 the DQX model fits well through the collected MOS. This is also reflected in the relatively high R^2 value of 0.95 for values below x_0 and 0.85 for the ones above. The standard deviation of 0.703 is rather low and only the variable bandwidth got a lower one. This means that the subjects voted consistently.

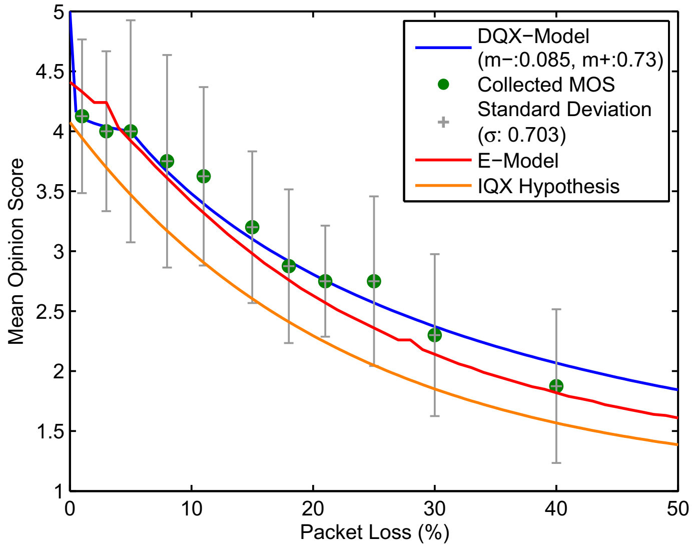

Figure 18: DQX-Model Fit and Comparison for Packet Loss (A)

Compared to the E-Model and the IQX Hypothesis, the fitted DQX model is located above the two. This can be explained with the tested codec. In the IQX Hypothesis it is the Internet Low Bitrate Codec (iLBC) [7] and for the E-Model it is the G.711 Codec [19]. The codec in the experiments of this thesis was Opus. This codec has advanced error correction mechanisms and probably therefore the result is higher [35]. It is also noticeable that the most advanced version of G.711 is used in the E-Model and therefore this graphs is relatively close to the DQX model. Interesting is also the shape of the curves. For all three models, the shape is more or less the same.

Like for the variable bandwidth, the plotted *m* values in Figure 19 are modestly growing with an increasing packet loss value. It is a larger growth than for bandwidth so that it is likely for packet loss that *m* does not remain constant with the growth of packet loss.

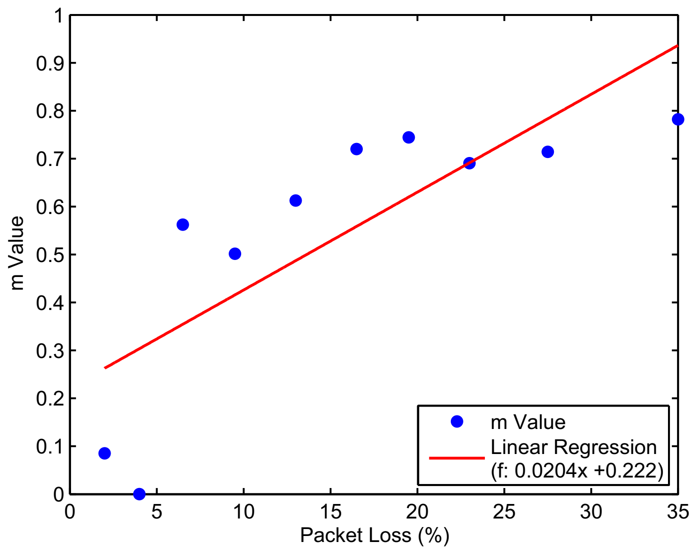

Figure 19: Development of the m Values for Packet Loss (B)

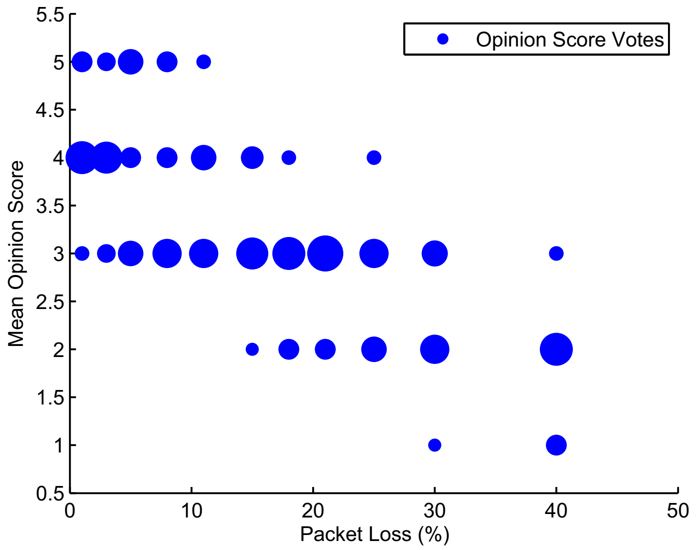

Figure 20: Distribution of Collected Votes for Packet Loss (C)

The bubble chart in Figure 20 shows what the standard deviation already has indicated: The distribution of the votes is not that heterogeneous so that there are for most of the scenarios a big bubble representing a big collection of equal votes. Such a distribution can be explained by the fact that the effects of packet loss are easier to rate. According to subjects' statments in the interviews it is easier to vote in a packet loss scenario where every thing is normal but a reduced audio quality.

4.2.3 Jitter

Jitter was tested in a range from 60ms to 1600ms. As x_0 value 100ms was used. The resulting *m* values from the fit are 1.1 for m^+ and 0.59 for m^- which is a high difference and therefore a constancy of *m* cannot be assumed.

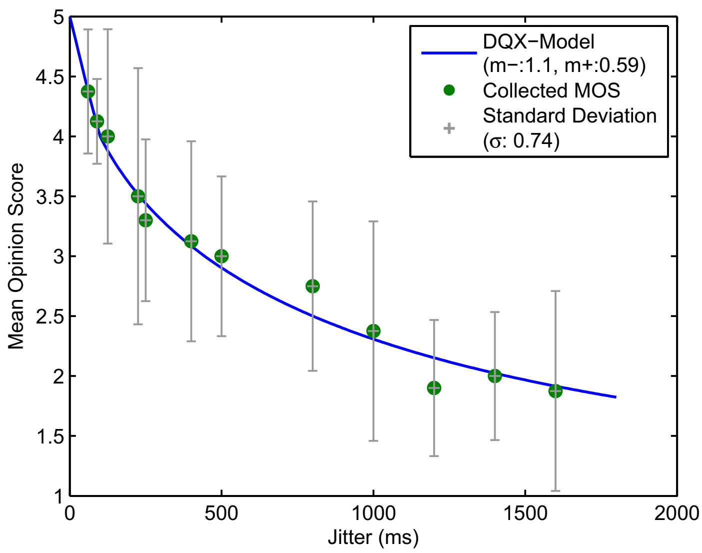

Figure 21: DQX-Model Fit and Comparison for Jitter (A)

In Figure 21 there is illustrated the fit of the DQX model through the collected MOS. The GOF is rather high. The R^2 values for m^+ values and m^- values is 0.96. This is the highest goodness fit of all four variables. However, since the standard deviation is rather high with 0.74 this GOF must be considered with caution and so it would be wrong to state that the DQX model works best with jitter. The explanation for the high standard deviation of jitter is probably the same as for latency. Since jitter basically results in latency and packet loss due to jitter buffers which drop packets [42], the same issues apply here as well. So there might be some troubles for the subjects to experience jitter due to conversational and cultural difficulties.

As shown in Figure 22 the *m* values produces an almost constant linear regression. This is unique for this variable since for all others it is either increasing or decreasing. Even though the regression indicates constancy, the *m* values are not. The are oscillating with a distance of 0.1 around the linear regression line. There are also two *m* values at the beginning which are far away from the regression line, one above and one below. These outliers can be explained easily by the fact that the DQX model starts with the unrealistic MOS of 5 and this causes deviations in the calculation of first *m* values. This issue is not only for jitter but can also be found in the plots of other variables. A method to prevent that will be presented later in this thesis.

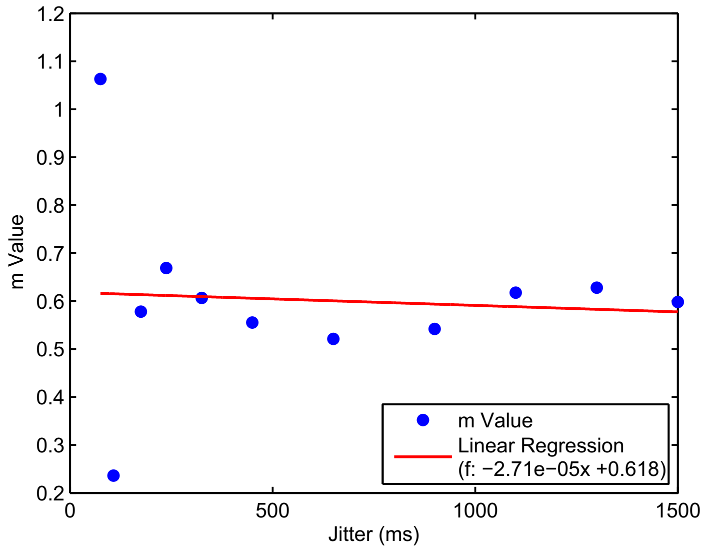

Figure 22: Development of the m Values for Jitter (B)

The bubble chart in Figure 23 visualizes once again the distribution of the votes. Even though there are several big bubbles indicating uniform votes, there are also scenarios in which the subjects had divided opinions about the call experience. The reason for this could be, as mentioned before, that the effects of jitter is like a mix between packet loss and latency. And comparing this chart to the bubble charts of packet loss and latency, it looks like a mix of both of them. This hypothesis is also supported by the standard deviations. The middle between the standard deviation of latency and packet loss is 0.734 and the standard deviation of jitter is 0.740 what is almost the same.

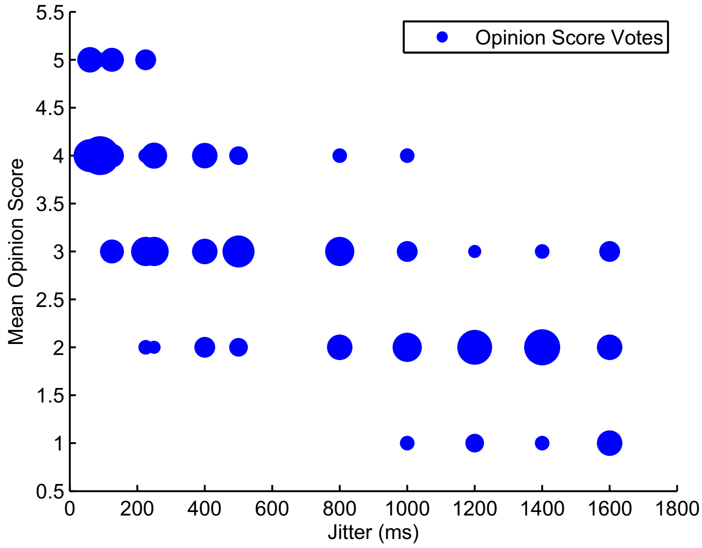

Figure 23: Distribution of the Collected Votes for Jitter (C)

4.2.4 Bandwidth

Bandwidth was tested in a range from 20 kbit/s to 125 kbit/s. As x_0 value 64 kbit/s was used. The resulting *m* values from the fit are 4.5 for m^- and 0.47 for m^+ which is the highest difference of all variables.

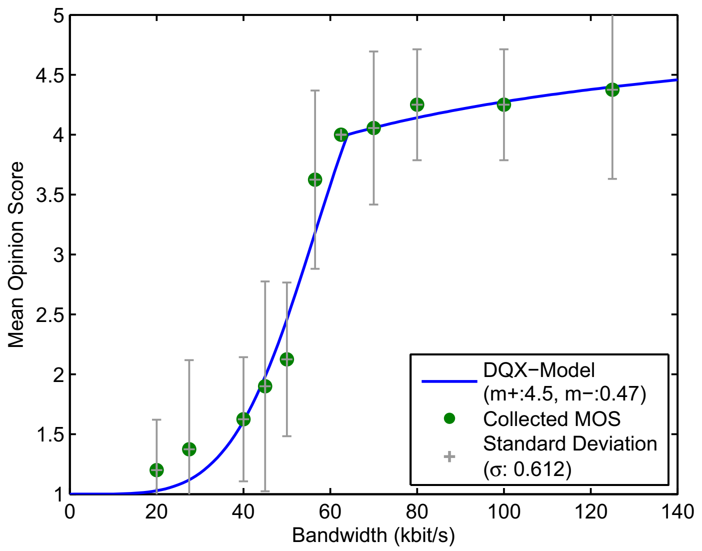

Figure 24: DQX-Model Fit and Comparison for Bandwidth (A)

The previously mentioned *m* values of the variable bandwidth can be easily interpreted by analyze the Figure 24. The *m* value is high for values below x_0 what means the MOS growth rapidly in this area. Once the required bandwidth of the codec is reached in x_0 the MOS stays almost the same and for that reason there is a small *m* that leads to a flat curve.

Moreover, it can be seen in Figure 24 that the curve of the DQX model fits well through the collected MOS. This is also represented by a high R^2 value which is 0.94 for values below

x_0 and 0.75 for values above. Moreover it is to notice that for bandwidth there is the lowest standard deviation, what means that the rating of the subject where rather consistent. This consistency can be explained with the shape of the graph. Between 40 kbit/s and 60 kbit/s it is steep, above and below it is almost flat. In this flat areas the voting is rather conclusive because the connection is either bad and no conversation is possible or it is perfect because the requirements of the codec are fulfilled so the connection is like in an uninfluenced scenario. This hypothesis is supported by the length of the gray standard deviation lines in the plot. In the area where the graph is steep, they are large and in the flat areas they are small.

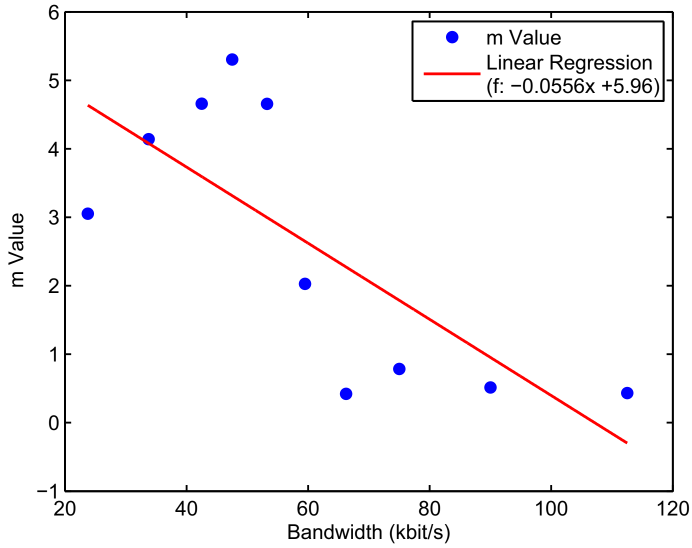

Figure 25: Development of the m Values for Bandwidth (B)

Interesting is the development of the *m* values for bandwidth. As plotted in Figure 25 the linear regression line for bandwidth is decreasing. So at this point it can be assumed that the linear regression for *m* values is increasing or constant for DVs like latency, jitter and packet loss, and decreasing for IVs like bandwidth. However by having a closer look to the data points it can be easily seen that they are not really linear. It is more as they would increase linearly till a bandwidth of 50 kbit/s then decrease linearly till 70 kbit/s and after that the remain constant.

Another way to interpret the plot is that the *m* values actually increases on two levels, one above and one below x_0. On that way, the previously mentioned assumption of the increasing and decreasing regression lines is wrong. What is really the case must be shown in further measurements.

As mentioned before, for bandwidth there was the lowest standard deviation of all the values. This is also reflected by the Figure 26 as there are several big bubbles which indicate a high level of consistency concerning the opinion scores. The bandwidth scenarios were also mentioned most often in the discussion with the subjects about fatal communication issues. This assessment of the subjects is also confirmed by the collected data. In the bubble graph there are several big bubbles for a score of 1. This is for no other variable the case.

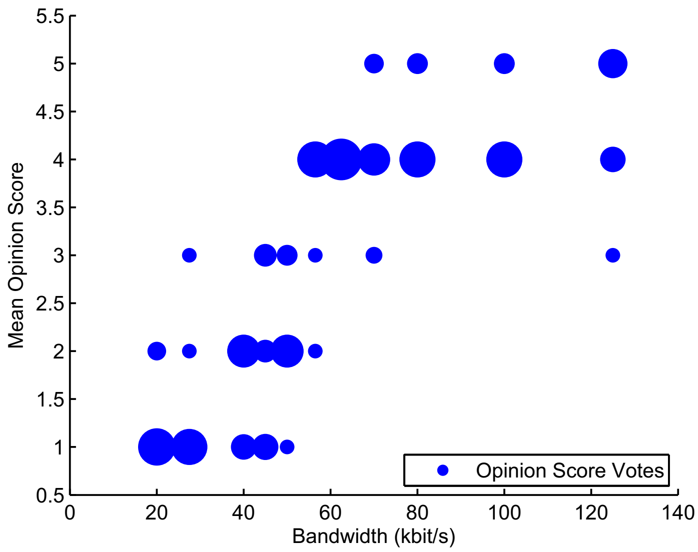

Figure 26: Distribution of the Collected Votes for Bandwidth (C)

4.2.5 Comparison

In this section, all the previous results are summarized in Table 3. The gradient of *m* value is related to the linear regression of the *m* values and represents its slope.

Table 3: Single Variables Comparison

	latency	packet loss	jitter	bandwidth
m^+ R^2	0,40 -0,65	0,09 0,85	1,06 0,96	4,53 0,75
m^- R^2	0,32 0,75	0,73 0,95	0,59 0,96	0,47 0,94
Standard Deviation	0,76	0,70	0,74	0,61
Gradient of m	$1{,}6 * 10^{-4}$	$2{,}04 * 10^{-2}$	$-2{,}71 * 10^{-5}$	$-5{,}56 * 10^{-2}$

4.3 Multiple Variables

This part evaluates DQX model for the scenarios where multiple variables where tested. Since the main focus was on the single variable tests, there are not so many data points for these mixed scenarios. However it was enough to find a contradiction in the initial equation of the DQX model, suggest a new one and make some first evaluations of the new equation.

4.3.1 The original and the new Equation

The initial equation of the DQX model for mixed variables was:

$$E(x) = 1 + 4 \cdot \prod_{k=1}^{N} [e_{i \vee d}(x_k) - 1]^{w_k} \qquad (15)$$

The idea behind this equation is that IVs can compensate DVs. But this leads to a contradiction. In case where the packet loss is set to 15% and the bandwidth is limited to 500 kbit/s the calculated MOS with the original model would be this:

$$E(x) = 1 + e_i(500)^{\frac{1}{2}} \cdot e_d(15)^{\frac{1}{2}} = 4,8955 \tag{16}$$

The calculation with the single variable packet loss set to 15% results to the following MOS:

$$e_d(15) = h \cdot e^{-x_0^{-m} \ln\left(\frac{4}{3}\right) \cdot 15^m} + 1 = 4 \cdot e^{-5^{(-0,773)} \ln\left(\frac{4}{3}\right) \cdot 15^{0,773}} + 1 = 3,1014 \tag{17}$$

The contradiction is that even though the bandwidth is limited in Equation 16 the result it is higher than the result of Equation 17 where the bandwidth is unlimited. To prevent such contradictory results a new equation was introduced during this thesis. The idea of this equation is that for every variable it is calculated how many percents are left over from the maximum rating after applying the particular variable. All this percentages are then weighted due to its influence and multiplied. The resulting percentage is then applied to the rating scale (cf. Equation 18).

$$E(x) = 1 + 4 \cdot \prod_{k=1}^{N} \left[\frac{e_{i \vee d}(x_k) - 1}{4}\right]^{w_k} \tag{18}$$

If this equation is used to the scenario above the following values for mixed variables can be calculated (15% packet loss and 500 kbit/s):

$$E(x) = 1 + 4 \cdot \left[\frac{e_i(500) - 1}{4}\right]^1 \cdot \left[\frac{e_d(15) - 1}{4}\right]^1 = 3,0452 \tag{19}$$

And for the single variable the resulting MOS would stays the same as above: 3,1014

As expected, the contradiction does not exist any longer. The MOS from the single variable scenario in Equation 17 is higher than the one in the mixed variable scenario (cf. Equation 19) as it should be. And the small difference between the two scenarios seems to reflect the reality realistically, since a bandwidth of 500 kbit/s means for audio only VoIP calls almost the same as unlimited bandwidth.

4.3.2 Comparison of the Collected and Calculated MOS

Having a new equation for mixed variables, the next step is a comparison between the collected mixed variable MOS and the calculated ones. In Figure 27 there are 3D-plots of such mixed variable scenarios using the parameters found during the analysis of the single variable. Since a lack in the number of data points for mixed variables it makes at this point no sense to adjust the weights of the mixed variables Equation 18. For that reason all the weight are set equally to 1. For visibility reasons, the collected MOS are drawn as a big black bullet which should ideally be cut by the graph in half.

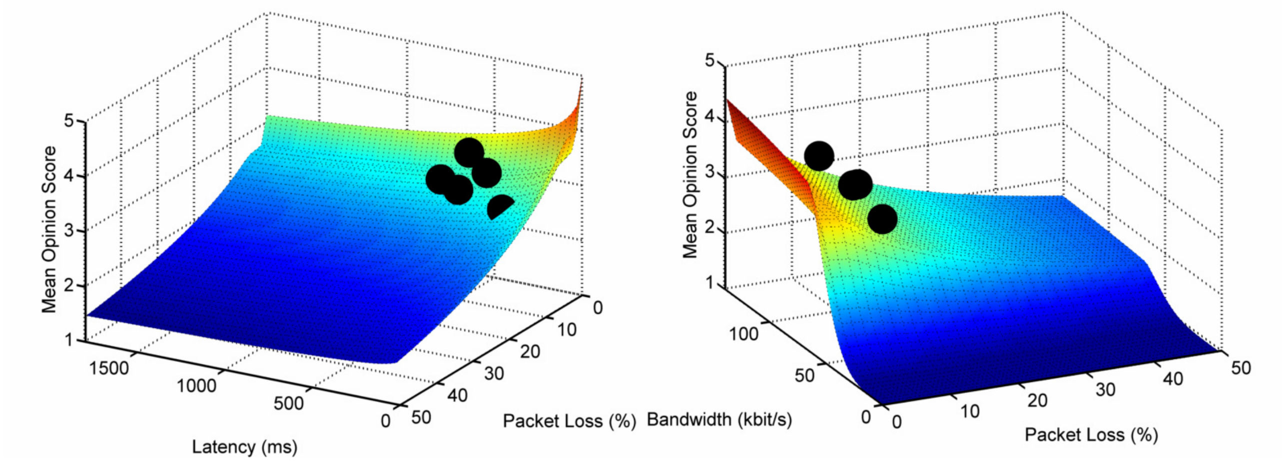

Figure 27: 3D-Plots of the DQX-Model for Multiple Variables

Since the distance between the calculated and the collected MOS is sometimes hardly visible in the 3D-plots the results are also presented in Table 4:

Table 4: Collected MOS for Mixed Variables Compared to the Calculated MOS

Latency	Packet Loss	Jitter	Band-width	MOS collected	Standard Deviation	MOS DQX	MOS difference
600	10	0	0	3,13	0,64	2,59	0,54
500	7	0	0	3,56	0,73	2,82	0,74
500	10	0	0	3,00	0,67	2,63	0,37
500	10	0	60	3,25	0,46	2,05	1,20
400	0	0	75	3,38	0,74	3,09	0,28
400	7	0	0	3,25	0,71	2,87	0,38
400	20	0	75	2,50	0,93	1,95	0,55
250	10	0	0	2,80	0,63	2,77	0,03
0	7	0	64	3,88	0,64	3,08	0,80
0	7	0	98	3,88	0,64	3,26	0,62
0	10	0	60	3,25	0,46	2,60	0,65
0	12	0	98	3,25	0,71	2,89	0,36
0	0	300	63	3,13	0,83	2,64	0,48
0	12	400	0	2,63	0,74	2,21	0,42

Comparing the results in the Table 4 it can be remarked that the Equation 18 creates promising results since the differences between calculated and collected MOS is small. The mean of all the MOS differences is 0.53 what is small for an unadjusted calculation where all the weights are 1. However there are not enough data points to make any significant statement. Considering the high standard deviation of the measurements, another thorough verification should be done in future work.

4.4 Further Calibration

During this thesis the idea of two further calibration of the DQX model for the VoIP scenario came up. The first one is an adjusted MOS scale. The idea comes from the E-Model where under normal conditions a MOS of 4.41 is expected [16]. Similar findings were made during the experiments. With all the subjects an uninfluenced scenario was performed to get the maximum possible MOS. The result of this was a MOS of 4.432 what is close to the one from the E-Model. The assumption is now that there is no MOS higher than 4.432 possible and therefore the new scale is from 1 to 4.432. By applying the new scale to the DQX model as it is done before with the MOS scale from 1 to 5 in Section 2.4, the following result:

$$e_i(x) = 3,432 \cdot (1 - e^{-(\lambda \cdot x^m)}) + 1, \lambda = x_0^{-m} \ln\left(\frac{3,432}{3,432 - 3}\right) \quad (20)$$

$$e_d(x) = 3,432 \cdot e^{-\lambda \cdot x^m} + 1, \lambda = x_0^{-m} \ln\left(\frac{3,432}{3}\right) \quad (21)$$

$$E(x) = 1 + 3,432 \cdot \prod_{k=1}^{N} \left[\frac{e_{i \vee d}(x_k) - 1}{3,432}\right]^{w_k} \quad (22)$$

The second calibration which can be done to the DQX model is an adjustment of the x_0 parameter. Such an adaptation of the parameter x_0 is a contradiction to the idea of the DQX model because this parameter should be determined before the experiments according to the SLA. However, one could argue that often the x_0 values are not exactly defined and vary between literature. Therefore as a second further calibration the x_0 values can be determined like the m values through a non-linear least square regression. The results are summarized in the following table.

Table 5: Original x_0 Values Compared to the Adjusted x_0 Values

	Latency	**Packet Loss**	**Jitter**	**Bandwidth**
Original x_0 values	150	5	100	64
Adjusted x_0 values	271,8	4,7681	96,423	62,976

The Table 5 shows that for all the values except latency the calculated x_0 values are close to the original x_0 values that were selected in a deterministic way. However, for latency the new x_0 values will lead to more accurate fits and results.

When these two further calibrations are applied to the single variables they change their appearance and their GOF. Figure 28 shows the difference between the deterministic plot on the left side and the adjusted plot on the right side. The plot on the right starts at 4.432 and goes through the x_0 point at 271.8 ms. With these adjusted values the GOF increased. It is now for the adjusted graph on the right an R^2 value of 0.98 above x_0 and 0.89 below what are both higher values as in the original plot on the left.

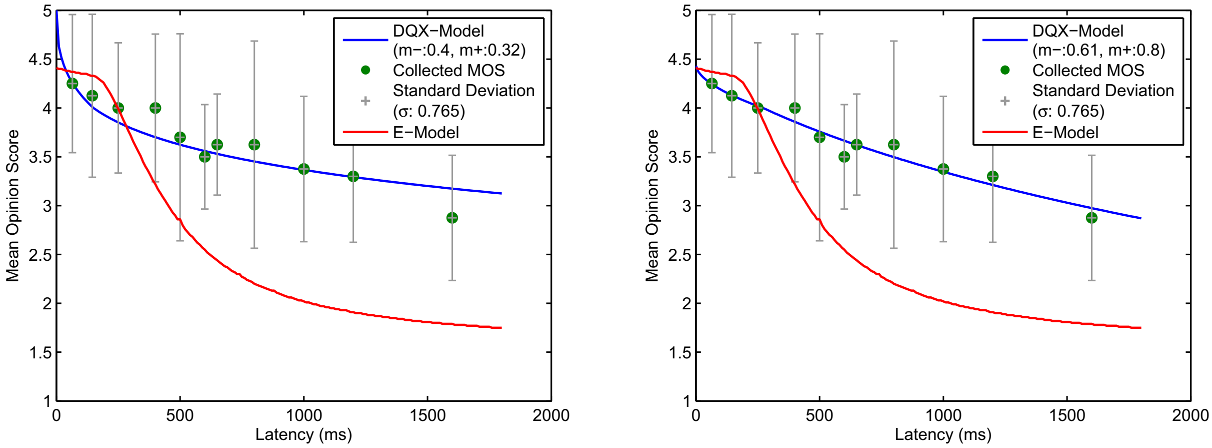

Figure 28: Comparison of Original and Adjusted DXQ-Model

But not only for latency the GOF could be improved. The following table summarizes the GOF as well as the *m* values for all the variables:

Table 6: Comparison of the m Values and R^2 Value for the Original and the Adjusted DQX-Model

	Latency	**Packet Loss**	**Jitter**	**Bandwidth**
m^+ original	0,40	0,09	1,06	0,47
m^+ adjusted	0,61	0,20	4,75	0,96
m^- original	0,32	0,73	0,59	4,53
m^- adjusted	0,80	1,06	0,84	5,69
R^2 of m^+ original	-0,65	0,85	0,96	0,75
R^2 of m^+ adjusted	0,98	0,81	0,99	0,84
R^2 of m^- orignial	0,75	0,95	0,96	0,94
R^2 of m^- adjusted	0,89	0,98	0,96	0,95

As can be seen in Table 6 all the R^2 have improved with this further calibration. But it is not only an improvement, they also reached a level where a correlation is highly plausible. The lowest value is 0.81 and there are five out of eight values above 0.9.

Besides the improvement in single variable scenarios thanks to these two further calibrations, a better prediction of the mixed variable scenarios was also possible as shown in the Table 7.

Through the further calibration the mean difference could be dropped from 0,53 to 0,21. This mean difference is low having regard to the fact that the weight factors are not adjusted so far. However, it should be considered that there are not a big number of data points collected and the standard deviation is rather high. Therefore this results should be regarded with caution.

Table 7: Comparison between the Original and Adjusted DQX-Model Concerning Mixed Variables

Mixed Scenario	MOS collected	MOS DQX-Model original	MOS difference original	MOS DQX-Model adjusted	MOS difference adjusted
No. 1	3,13	2,59	0,54	2,98	0,14
No. 2	3,56	2,82	0,74	3,25	0,30
No. 3	3,00	2,63	0,37	3,05	0,05
No. 4	3,25	2,05	1,20	2,63	0,62
No. 5	3,38	3,09	0,28	3,61	0,24
No. 6	3,25	2,87	0,38	3,33	0,08
No. 7	2,50	1,95	0,55	2,41	0,09
No. 8	2,80	2,77	0,03	3,25	0,45
No. 9	3,88	3,08	0,80	3,46	0,41
No. 10	3,88	3,26	0,62	3,69	0,19
No. 11	3,25	2,60	0,65	3,02	0,23
No. 12	3,25	2,89	0,36	3,30	0,05
No. 13	3,13	2,64	0,48	3,09	0,04
No. 14	2,63	2,21	0,42	2,54	0,09

The GOF of the new Equation 18 can also be seen by comparing the 3D-plots. In the following figure there are two 3D-Plots compared. The one on the left is the deterministic DQX model and the one on the right is the adjusted version of it:

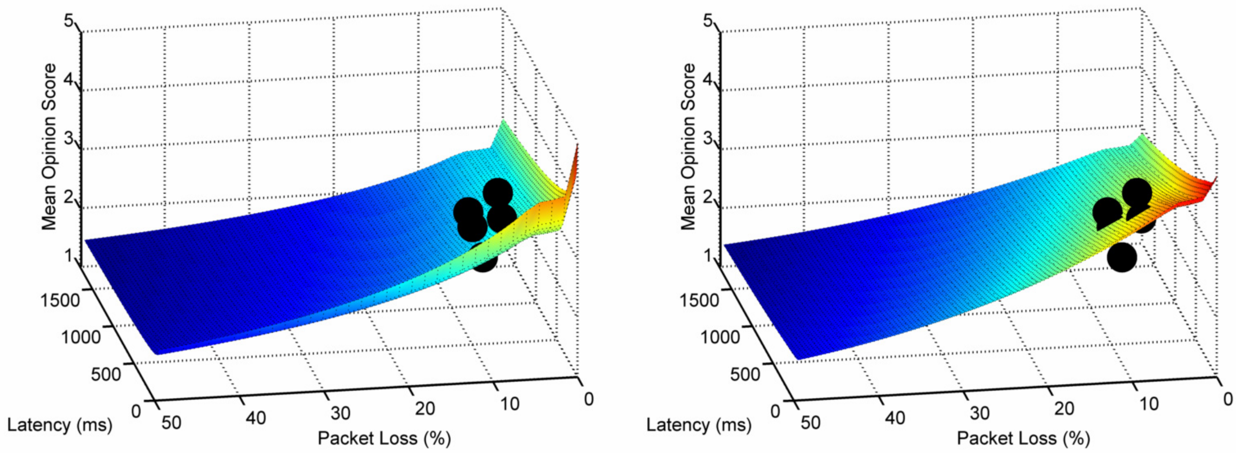

Figure 29: Comparison of the Original and Adjusted 3D-Plots

5 Summary

During this work, a QoE measurement setup was created that is able to save and replay a sequence of different network scenarios emulations. Moreover this setup provides the possibility to save user ratings and perform basic analysis. The adjustable variables that can be emulated by this measurements setup are jitter, latency, packet loss and bandwidth. The setup uses the Linux tool NetEm [26] to emulate the network and should therefore be accurate enough [21] to perform significant experiments.

This setup was used with a self-made VoIP messenger to collect over 500 data points in experiments with total 34 subjects. This messenger, called QoEssenger, is based on the WebRTC technology.

The collected data was in a further step used to calculate the MOS for each scenario and these MOS were used to define the parameters of the DQX model for VoIP services. In this evaluation there were basically three steps performed for every variable. First, the DQX model was fitted through the collected MOS and the resulting GOF and the influence factors m value were analyzed. Secondly, the resulting DQX model was compared to the ITU's E-Model and the IQX-Hypothesis. As a last step the variables were evaluated in a mixed scenario with other variables.

It could be shown that the DQX model reaches a high GOF through the data points. Moreover, an outcome of the analysis of the m values is that the they are not constant and further research is possible in this area. Besides, this thesis brought up a new proposal on how the MOS can be calculated in mixed scenarios. This new equation produced promising results, at least for the couple of measurements with mixed variables that were performed.

Furthermore, this thesis contains suggestions for further calibrations of the DQX-Model like an adjusted MOS scale or adapted x_0 values. On that way, even more accurate fits and MOS predictions could be generated, once there are available data from end-users feedback.

6 Conclusion and Critical Thoughts

In this evaluation there are some good fits what lead to the conclusion that the DXQ model is accurate and reflects/predicts well the reality. Moreover the expectation that the *m* values are not constant was proved and therefore further research is possible. Promising is also the adopted DQX equation for mixed value, especially if it is used with the further calibration techniques concerning the MOS co-domain and the appropriate x_0 selection.

However, there are also some critical thoughts towards the experiments and the analysis. First of all, a sample of 34 subjects is not enough to generate accurate data. However, there were limited time-resources in the scope of this thesis for having a larger sample. Therefore, all the findings should be regarded with the necessary caution. Moreover there is a large cluster in the subject's age and gender distribution what is also not favorable in an experiment. The largest part of the subjects were men between 20 and 25 years since the experiments took place in the department of informatics and the majority of students there are males and belong to this age group.

Another critical aspect is the headset used in the experiment. In the interviews with the subjects it came up that for some of them it was unfamiliar to wear such a headset which such a noise attenuation that subjects could not hear their own voice. On one hand such a headset was important to guarantee that people can focus on the audio and that they are not disturbed by environmental noise. On the other hand it is always a bias whenever people feel uncomfortable during an experiment. Thus, it is difficult to give a final recommendation on which headset is ideal for QoE measurements in VoIP services.

7 Future Work

There is some future work to be done concerning the measurement setup, the VoIP messenger as well as the DQX model.

The measurement setup could be easily extended with other adjustable variables like packet corruption, reordering and duplication. This would not be a big effort since the adjustment of the previously mentioned variables is already provided by the framework in the background. Moreover it could be used to perform more QoE experiments of the same or other IP-based services with different variables and variable values.

The VoIP messenger, used in this thesis, is on a level where it is almost fully functional. For that reason it will be finalized and expected to be deployed at the end of 2014 to the servers of the Communication Systems Group (CSG) in the University of Zurich. The code will be available for testing purposes.Thus, an open-source WebRTC client is available which is free to use for everyone and without any commercial interests.

Concerning the DQX model, the assumptions made in this thesis about the influence factors *m* could be further analyzed and possibly confirmed in a larger scale experiment. Moreover the proposal of the equation for mixed variables should be further evaluated also with a more elaborated use of the weight factors. Last but not least the DQX model is thought to be used with a range of other services besides VoIP. Therefore it would be certainly interesting to evaluate the model in other IP based services like video streaming, Internet browsing, multi-player gaming and finally in services where non technical variables can also affect QoE.

References

[1] Apache Friends, *XAMPP Installers and Downloads for Apache Friends*, URL: https://www.apachefriends.org/de/index.html, Visited in July 2014.

[2] Apposite Technology, *Apposite Technologies :: The Leader In WAN Emulation*, URL: http://www.apposite-tech.com/, Visited in July 2014.

[3] Awio Web Services LLC, *W3C Counter - January 2014 Market Share*, URL: http://www.w3counter.com/globalstats.php?year=2014&month=1, Visited in June 2014.

[4] Cisco, *Quality of Service for Voice over IP*, URL: http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.html, Visited in July 2014.

[5] H. Dragomir, R. Nyman, *WebRTC Articles & Mozilla Hacks – the Web developer blog*, URL: https://hacks.mozilla.org/category/webrtc/as/complete/page/2/, Visited in July 2014.

[6] ESTI, *Speech and multimedia Transmission Quality (STQ); Multimedia quality measurement; End-to-end quality measurement framework*, August 2011.

[7] M. Fiedler, T. Hossfeld, P. Tran-Gia, *A Generic Quantitative Relationship between Quality of Experience and Quality of Service*, March/April 2010.

[8] GigaNet Systems, *Network Emulators For Test Precision, Performance, And Repeatability,* URL: http://www.giganetsystems.com/, Visited in July 2014

[9] F. Hendrikx, WANsim allows you to emulate a WAN connection, URL: https://code.google.com/p/wansim/, Visited in July 2014.

[10] ITU-T, *E-Model Tutorial*, URL: http://www.itu.int/ITU-T/studygroups/com12/emodelv1/tut.htm, Visited in June 2014.

[11] ITU-T, *End-user multimedia QoS categories*, ITU-T Recommendation G.1010, November 2001.

[12] ITU-T, *Estimating End-to-End Performance in PI Networks for Data Application*, ITU-T Reccomendation G.1030, November 2005.

[13] ITU-T, *Mean Opinion Score (MOS) terminology*, ITU-T Recommendation P.800.1, November 2006.

[14] ITU-T, *Methods for Subjective Determination of Transmission Quality*, ITU-T Recommendation P.800, June 1998.

[15] ITU-T, *One-way transmission time,* ITU-T Recommendation G.114, Mai 2003.

[16] ITU-T, *R Value Calculation*, URL: https://www.itu.int/ITU-T/studygroups/com12/emodelv1/calcul.php, Visited in July 2014

[17] ITU-T, *Subjective Evaluation of Conversational Quality*, ITU-T Recommendation P.805, October 2007.

[18] ITU-T, *The E-Model, a computational model for use in transmission planning*, ITU-T Recommendation G.107, March 2003.

[19] ITU-T, *Transmission impairments due to speech*, ITU-T Recommendation G.113, July 2014.

[20] Joyent, *Node.js - About*, URL: http://nodejs.org/about/, Visited in July 2014.

[21] A. Jurgelionis, J. Laulajainen, M. Hirvonen, A.I. Wang, *An Empirical Study of NetEm Network Emulation Functionalities*.

[22] Keloo Network, *10000 general knowledge questions and answers,* URL: http://www.keloo.ro/doc/10000_intrebari.pdf, Visited in July 2014.

[23] S. Khirman, P. Henriksen, *Relationship between Quality-of-Service and Quality-of-Experience for Public Internet Service,* 3rd Passive Active Measurement Workshop, March 2002.

[24] K. Knopper, *Koppnix - Live Linux Filesystem on CD*, URL: http://www.knopper.net/knoppix/index-en.html, Visited in July 2014.

[25] T. *Krenn, Linux Netzwerk Analyse mit mtr*, URL: http://www.thomas-krenn.com/de/wiki/Linux_Netzwerk_Analyse_mit_mtr, Visited in July 2014.

[26] Linux Foundation, *netem | The Linux Foundation,* URL: http://www.linuxfoundation.org/collaborate/workgroups/networking/netem, Visited in July 2014.

[27] Mathworks, *Evaluating Goodness of Fit - MATLAB & Simulink,* URL: http://www.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html, Visited in July 2014.

[28] Mathworks, *Fit curve or surface to data - MATLAB fit*, URL: http://www.mathworks.com/help/curvefit/fit.html#bto2vuv-7, Visited in July 2014.

[29] Mathworks, *Least-Squares Fitting - MATLAB & Simulink,* URL: http://www.mathworks.com/help/curvefit/least-squares-fitting.html, Visited in July 2014.

[30] Mathworks, *Parametric Fitting - MATLAB & Simulink*, URL: http://www.mathworks.com/help/curvefit/parametric-fitting.html,Visited in July 2014.

[31] Mathworks, *Standard deviation - MATLAB std*, URL: http://www.mathworks.com/help/matlab/ref/std.html, Visited in July 2014.

[32] Mathworks, *Statistics Toolbox - MATLAB*, URL: http://www.mathworks.com/products/statistics/?s_cid=sol_des_sub2_relprod3_statistics_toolbox, Visited in July 2014.

[33] Mathworks, *Summary statistics organized by group - MATLAB grpstats*, URL: http://www.mathworks.com/help/stats/grpstats.html, Visited in July 2014.

[34] Novascola, *Quizfragen und Antworten,* URL: http://www.novascola.ch/quizfragen-und-antworten, Visited in July 2014

[35] Opus, *Voice Coding with Opus*, URL: http://www.opus-codec.org/presentations/opus_voice_aes135.pdf, Visited in July 2014.

[36] E. Rescorla, *Proposed WebRTC Security Architecture*, URL: http://www.ietf.org/proceedings/82/slides/rtcweb-13.pdf, Visited in June 2014.

[37] R. Riegel, *nload: monitor network traffic and bandwidth usage,* URL: http://www.roland-riegel.de/nload/, Visited in July 2014

[38] TATA, *WANem: The Wide Area Network Emulator*, URL: http://wanem.sourceforge.net/, Visited in June 2014.

[39] C. Tsiaras, B. Stiller, *A Deterministic QoE Formalization of User Satisfaction Demands (DQX)*, To appear on 39th IEEE Conference on Local Computer Networks (LCN), Sep. 8-11, 2014, Edmonton, Canada.

[40] J. Uberti, S. Dutton, WebRTC Plugin-free realtime communication, URL: http://io13webrtc.appspot.com/, Visited in July 2014

[41] Ubuntuusers.de, *iperf › Wiki › ubuntuusers.de*, URL: http://wiki.ubuntuusers.de/iperf. Visited in July 2014.

[42] Voiptroubleshooter.com, *Indepth Articles | Jitter* URL: http://www.voiptrouble-shooter.com/indepth/jittersources.html, Visited in July 2014.

[43] W3C, *WebRTC 1.0: Real-time Communication Between Browsers*, June 2014, URL: http://dev.w3.org/2011/webrtc/editor/webrtc.html, Visited in June 2014.

[44] WebRTC, *Issue 2025: Opus in low bandwidth conditions*, URL: https://code.goo-gle.com/p/webrtc/issues/detail?id=2025, Visited in July 2014.

[45] WebRTC.org, *General Overview - WebRTC* [Online], URL: http://www.webrtc.org/ref-erence/architecture, Visited in June 2014.

[46] E. Zachte, *Wikimedia Traffic Analysis Report - Browsers e.a.,* URL: http://stats.wikime-dia.org/archive/squid_reports/2014-01/SquidReportClients.htm, Visited in June 2014.

Abbreviations

APIs	Application Programming Interfaces
CD-ROM	Compact Disc Read-Only Memory
CQ	Conversational Quality
CSG	Communication Systems Group
CSS	Cascading Style Sheets
DQX	Deterministic QoE model
DTLS	Datagram Transport Layer Security
DV	Decreasing Variable
ESTI	Eidgenössisches Starkstrominspektorat
eV^2	Expected Variable Value
FOSS	Free and Open Source Software
GOF	Goodness of Fit
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICE	Interactive Connectivity Establishment
iLBC	Internet Low Bitrate Codec
IP	Internet Protocol
ISPs	Internet Service Providers
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
IV	Increasing Variable
LQ	Listening Quality
LAN	Local Area Network
MNOs	Mobile Network Operators
MOS	Mean Opinion Score
MOSs	Subjective MOS
MOSo	Objective MOS
MOSe	Estimated MOS
OLR	Overall Loudness Rating
OS	Operating System
P2P	Peer-to-Peer
PHP	PHP Hypertext Preprocessor
qdu	Quantization Noise
QoE	Quality-of-Experience
QoS	Quality-of-Service
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SSR	Sum of Squares of the Regression

SST	Total Sum of Squares
STMR	Sidetone Masking Rating
STQ	Speech and multimedia Transmission Quality
STUN	Session Traversal Utilities for NAT
ToS	Type of Service
TURN	Traversal Using Relays around NAT
UI	User Interface
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
W3C	World Wide Web Consortium
WebRTC	Web Real-Time Communication
WLAN	Wireless Local Area Network

Glossary

Bandwidth	Defines how many bits (1 byte = 8 bit) are sent every second
DQX model	An extension to the IQX Hypothesis that considers the SLA of the provider
E-Model	A computational transmission rating model which is the common ITU-T transmission rating model
End-to-End quality	Quality related to the performance of a communication system, including all terminal equipment, for voice services it is equivalent to mouth-to-ear quality
Goodness of Fit	Expression for how good the regression approximates a graph to the datapoints.
Jitter	Random delays in a certain range are called Jitter
Latency	Measurement for the constant delay of a data packet from the sender to the receiver
m value	The influence factor of the DQX-Model that determines the shape of the graph
Packet Loss	A measurement for the number of data packets that are lost from the sender to the receiver in percent
QoEssenger	The VoIP messenger based on WebRTC that was created in this thesis
R^2	It is a value from 0 to 1 representing the GOF as a percentage
R-Value	The output of the E-Model, can easily transformed into MOS
Skype	A popular VoIP application from Microsoft.
Standard Deviation μ	A measurement for the dispersion around the mean
x_0	The eV2 of the DQX-Model derived from the SLA. This value acts as an anchor point for the graph in the DQX-Model
IQX Hypothesis	An exponential approach to link QoE and QoS.

List of Figures

List of Tables

Appendix A: Installation Guidelines

The participant's computers

The installation of the measurement setup requires at least three computers. On two of them the application to be tested should be set up. In the case of this thesis, it is enough to install an Ubuntu 14.04. This version comes with the most recent Firefox browser and therefore it is not necessary to install anything else. There is only a little adaption of the browser that should be done. By typing "about:config" in the Uniform Resource Locator (URL) field the hidden settings of the Firefox browser will appear. In this menu the following setting should be set to true:

media.navigator.permission.disabled true

This step is necessary to start calls directly without asking the user permission.

Moreover the IP addresses of the two computers must be defined manually. This can be done by opening the network settings and typing the following IP addresses in.

Computer 1: 192.168.1.10

Computer 2: 192.168.1.20

The experimenter's computer

To set up the computer that runs WANem and the control panel, the following steps must be performed:

1. An Apache server and a MySQL database must be installed by downloading and installing **XAMPP.**
2. Download and installation of **Node.js**.
3. In the **command line** the following commands must be executed to install additional Node.js packages:

   npm install socket.io

   npm install node-static

   npm install http
4. **The QoEssenger folder** (from the attached CD-ROM) **must be copied** into the home directory of the apache server. By default it is the following path on Windows:

   C:\xampp\htdocs\
5. Download and installation of **VirtualBox.**
6. **A new virtual machine must be created. The WANem ISO file must be inserted** into the virtual disc tray.
7. In the network settings of that virtual machine **two adapters must be activated.** The first one as NAT and the second as bridged networking.
8. WANem can be booted by **starting the previously created virtual machine** and **pressing enter** when WANem asks to do so.

9. The following lines must be copied and pasted into the /etc/apt/**sources.list** file

   deb http://ftp.ch.debian.org/debian stable main

   deb http://ftp.debian.org/debian/ wheezy-updates main

   deb http://security.debian.org/ wheezy/updates main

10. In the **command line** the following commands must be executed to install the Iceweasel browser

    exit2shell

    apt-get update

    apt-get install iceweasel

    All the appearing messages should be confirmed by pressing “y”. After the installation procedure, the Iceweasel browser can be started.

11. In the **networks settings** of WANem (virtual machine) and Windows (physical machine) the following IP addresses must be defined manually:

    WANem: 192.168.1.77

    Windows: 192.168.1.20

12. The state of the virtual machine can be kept by **saving the machine state whenever the virtual machine is closed**.

After performing all the above mentioned steps, the computers are ready for performing experiments. This installation steps of Appendix A have to be performed only once. The actual setup of the experiment is described in the Appendix B and has to be performed after every reboot.

Appendix B: Experiment Setup and Execution Checklist

Experiment Setup

The following steps need to be performed before every new experimental session:

1. All the computers must be **connected** to the switch, the head-sets must be plugged in and all the computers must be **booted**.
2. The **XAMPP** control panel must be started to run the Apache server and the MySQL database
3. On the two computers which run the QoEssenger, the following URL must be opened in the **Firefox** browser:

   192.168.1.20/QoEssenger/?page=call&r=experiment&n=Peer

The Peer in the URL can be replaced with Peer1 and Peer2 to identify the computers.

4. On this two computers that run the QoEssenger, the following **commands** must be executed in the command line in to add the proper entries for the routing table:

   sudo route add 192.168.1.10 gw 192.168.1.77 (for device with IP 192.168.1.30)

   sudo route add 192.168.1.30 gw 192.168.1.77 (for device with IP 192.168.1.10)

   WARNING: Whenever a LAN cable is unplugged and replugged this step has to be performed again!
5. The virtual **WANem** machine in VirutalBox can now be started and the following URL should be opened in the Iceweasel browser:

   192.168.1.20/QoEssenger/?page=settings
6. **New network scenarios** can be saved them by right clicking the numbered buttons on the control panel. (More about that in Section 3.5 Control Panel)

Experiment Execution

The experiment takes place with two subjects at the same time and it takes around 30 minutes. The following steps are performed in these 30 minutes:

1. Explanation of the experimental setup to the subjects: (2 min)

   What is this experiment about?

   What is the task of the participant?
2. Explanation of the **rating system** (mean opinion score) and demonstration of the rating system of QoEssenger to the subjects. It should be mentioned in this step that the ratings are **anonymous** and that there are no video or audio recording during the experiment. (2 min)
3. One participant should be brought to the **remote room** and should be asked to wait for the first incoming call. (1 min)
4. **Explanation** to the other participant how to **start and end a call**. (1 min)
5. Now, the **play button** in the control panel can be pressed and the participant can be ask to start the first call.

6. Tasks **during the test call**: As soon as the calling time is up, the participant must be asked to **end the call.** During the voting time, it must be **controlled** that the rating of both subjects is saved in the database. The participant must be asked at the end of the rating time to **reconnect.** Unnecessary voting time can be skipped by pressing the next button. (20 min)
7. After all the experimental calls, the participant in the remote room has to be **picked up** and brought back to the main room. (2 min)
8. The subjects should now be asked **questions** concerning their QoE during the test calls. (5 min)
9. A little present should be offered to the subjects and in case they are interested, the control panel can be shown to them. (graphical representation and basic analysis of the collected data)

Appendix C: More about the Analysis

Precision of the Control Panel

As written before, with the Control Panel it is possible to make first basic analysis. It is possible to draw data points, fit the graph of the DQX model through the points and get the *m* values. This entire calculation is done with JavaScript. To test its functionality and precision test calculations has been performed and its results have been compared to leading statistic programs. The result of this comparison is summarized in the following table:

Table 8: Comparison of the Precision between Different Software

Software	Calculated Value
Control Panel:	0,550827649619541
Excel:	0,550827652344947
MATLAB:	0,550827653083384
R:	0,550827484209820

As can be seen in Table 8, the result from the control panel differs from other common statistic tools in a negligible manner. Therefore it can be considered as accurate enough for the purposes of this work.

Automated Analysis in MATLAB

To simplify analysis and the data visualization the process was implemented as a MATLAB script. It can be easily used by performing the following steps:

1. As a first step, the **getDataCSV script** must be opened in a browser. If the server runs on the localhost it is the following URL:

   http://localhost/QoEssenger/dbScripts/getDataCSV.php

The message: “Done! csv files can be found in the out folder.” should appear.

2. Next, the whole **Analysis folder** that can be found on the attached CD-ROM must be **copied into the MATLAB working directory** and it must be opened.
3. The CSV files from the **/QoEssenger/dbScripts/out** directory from the web server must be copied into the **Analysis/in** folder.
4. The analysis can now be started by executing the command **makeAll** in MATLAB. All the calculated values and the plotted diagrams can be found in the **Analysis/out folder**.

More Plots for single and mixed Variables

The following figures show plot with single and mixed variables. On the left side of each figure is always the original DQX-Model and on the right is the further adjusted one.

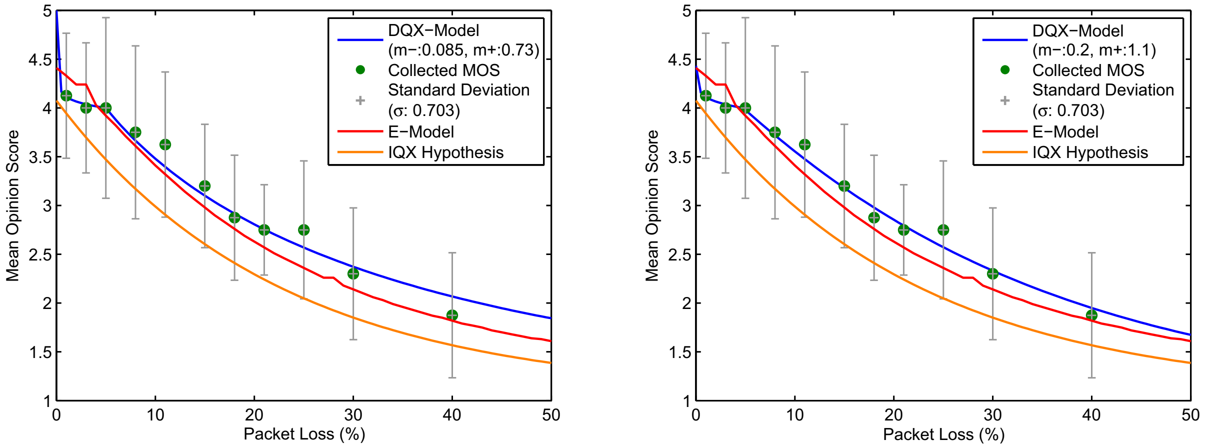

Figure 30: Comparison of the 2D Plots of Packet Loss

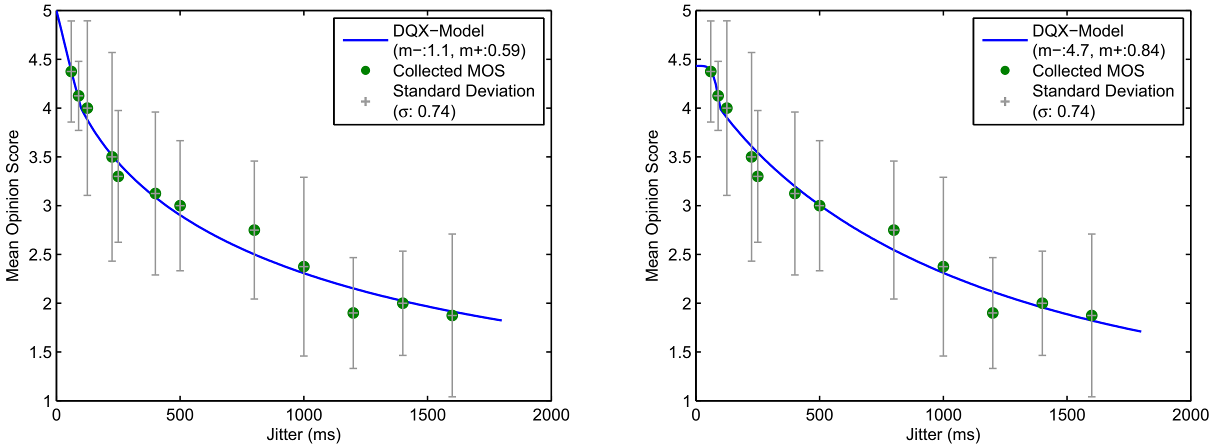

Figure 31: Comparison of the 2D Plots of Jitter

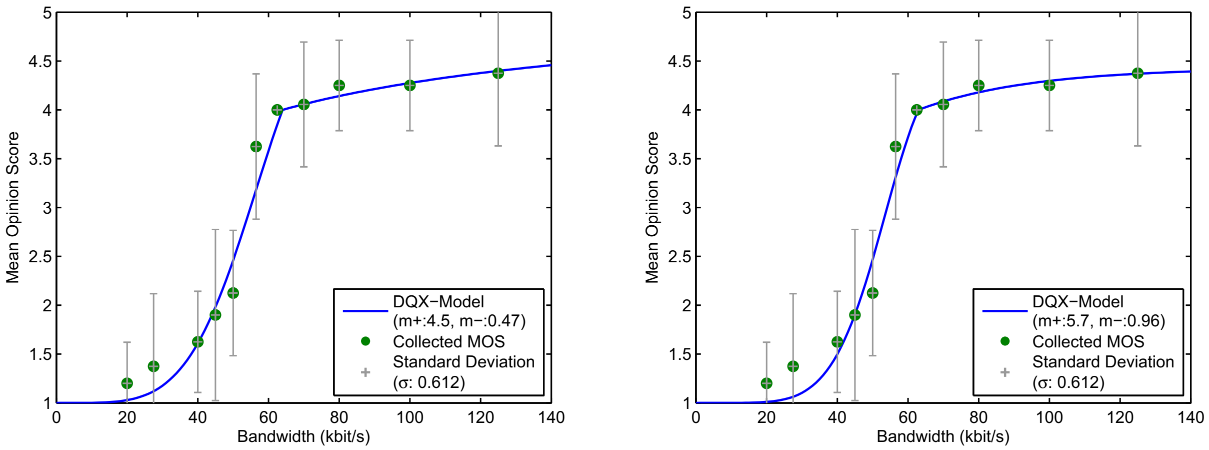

Figure 32: Comparison of the 2D Plots of Bandwidth

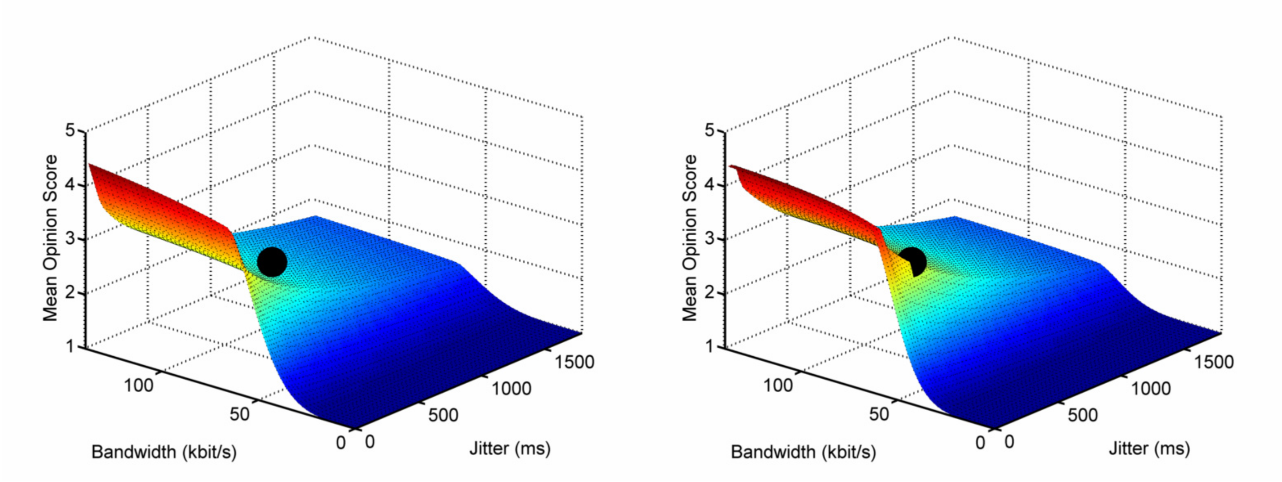

Figure 33: Comparison of 3D Plots with Jitter and Bandwidth

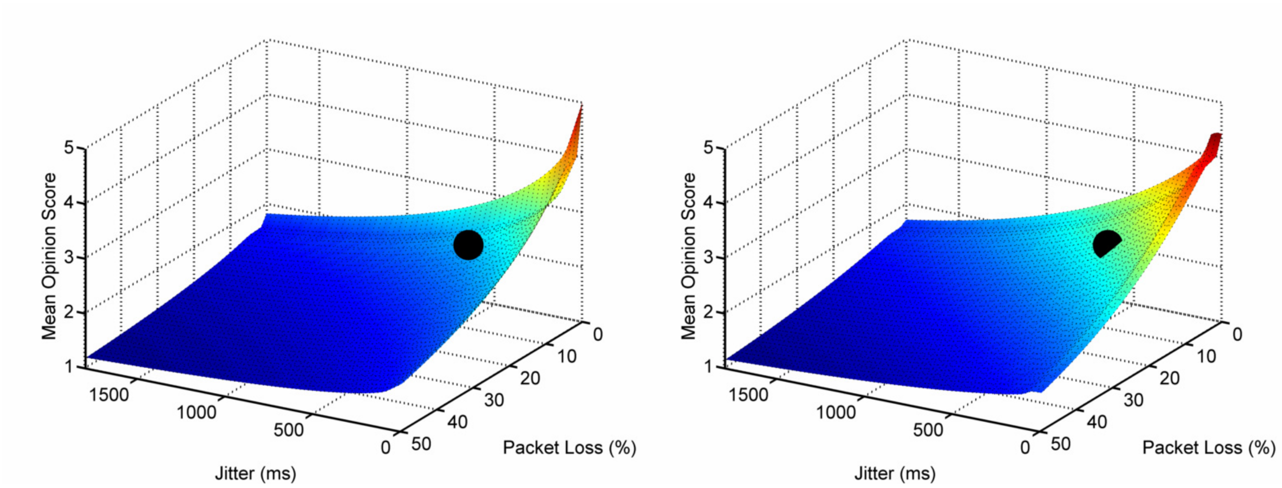

Figure 34: Comparison of 3D Plots with Jitter and Packet Loss

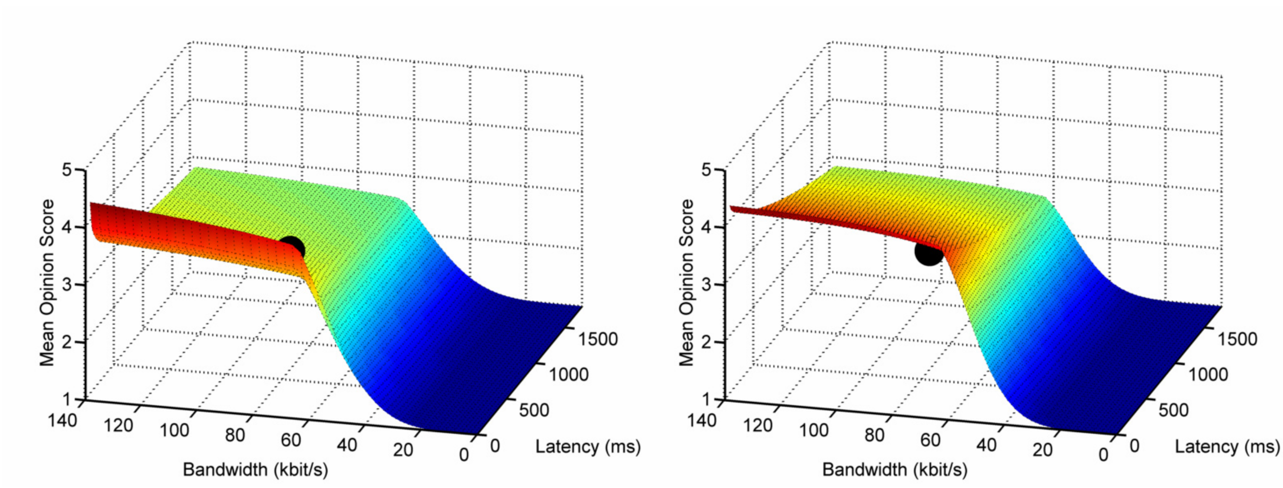

Figure 35: Comparison of 3D Plots with Latency and Bandwidth

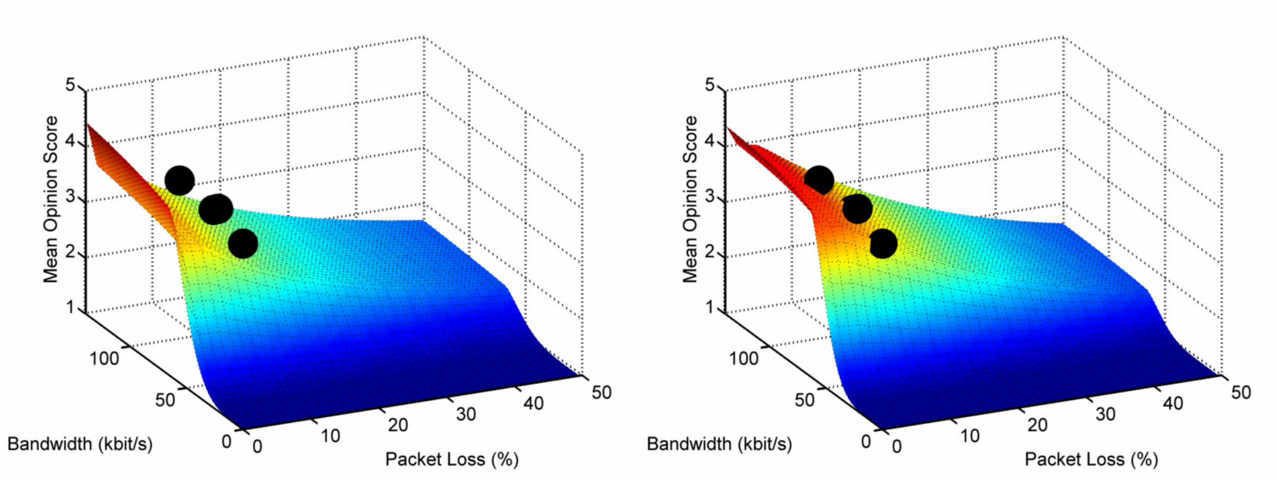

Figure 36: Comparison of 3D Plots with Loss and Bandwidth

Appendix D: Software and Libraries

Software and Libraries used in the final Architecture

Node.js: v0.10.29, http://nodejs.org/

Firefox: v30.0, http://www.mozilla.org/

WANem: v3.0 beta 2, http://wanem.sourceforge.net/

XAMPP: v3.2.1, http://www.apachefriends.org, (Apache, MySQL, phpMyAdmin)

VirtualBox: v4.3.12, https://www.virtualbox.org/

Ubuntu: v14.04, http://ubuntu.com/

Windows: v8.1, http://windows.microsoft.com/

jQuery: v1.11.1, http://jquery.com/

Rickshaw: v1.4.5, http://code.shutterstock.com/rickshaw/

MathJax: v2.4, http://www.mathjax.org/

Software used for the development process

FrameMaker: v10, http://www.adobe.com/

TexStudio: v2.8.0, http://texstudio.sourceforge.net/

Aptana Studio: v 3.4.2, http://www.aptana.com/

GIMP: v2.8, http://www.gimp.org/

Inkscape: v0.48.4, http://www.inkscape.org/

Audacity: v2.0.5, http://audacity.sourceforge.net/

Software used for the analysis process

Microsoft Excel: v2007 SP3, http://office.microsoft.com/

MATLAB: v2012b, http://www.mathworks.com/

R: v3.0.3, http://www.r-project.org/

Appendix E: Contents of the CD

QoEssenger: Folder that contains the source code of the WebRTC-based VoIP messenger and the control panel

BachelorThesis.pdf: This Bachelor Thesis in PDF format.

BachelorThesis.ps: This Bachelor Thesis in PS format.

BachelorThesis Source: Folder that contains the source files of the written thesis

Presentations: Folder with the intermediate and final presentation in PDF and PPT format

Related Work: Folder with related Papers that were used for this thesis

Analysis: Folder with the collected data in the CSV format, the automated MATLAB analysis scripts and other files which related to the analysis process

Abstract.txt: The abstract in English.

Zusfsg.txt: The abstract in German.