ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

BACHELOR THESIS

“Monitoring Quality of Experience for LTE Advanced Pro networks”

Student

Saoulidis Theocharis

Student Number 134060

Thesis Supervisor

Dr. Periklis Chatzimisios

Associate Professor, CSSN Research Lab

Thessaloniki

June 2018

PREFACE

The advancement of wireless communication network the last years has led to an increasing demand on mobile Internet services. One of the main issues for future wireless networks is users' Quality of Experience (QoE) especially when designing personal and customized services in order to maintain and attract more users. Thus, the researchers are focusing on the field and they moved forward from trying to make the system better, faster and enhancing objectively the system's performance to a more user-driven area so they can improve the subjective experience of the users. This switch made the resource allocation policy to be preferred as QoE-oriented and brought out many new challenges, including how to quantify and measure QoE, how to design a set of unified wireless resource management strategies and how to make use of a huge amount of available data to derive an optimal QoE model, etc. Therefore, personalized QoE management, efficient monitoring, accurate estimation, and optimal resource allocation need to be studied and implemented in future wireless networks.

Part of the current work have been accepted for publication as follows:

Elisavet Grigoriou, Theocharis Saoulidis, Luigi Atzori, Virginia Pilloni and Periklis Chatzimisios, "An agent-based QoE monitoring strategy for LTE networks", *IEEE International Conference on Communications (ICC 2018)*, 20-24 May 2018, Kansas city, MO, USA.

Elisavet Grigoriou, Theocharis Saoulidis, Luigi Atzori, Virginia Pilloni and Periklis Chatzimisios, "A QoE monitoring solution for LTE-Advanced Pro networks", *IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD 2018)*, 17-19 September 2018, Barcelona, Spain.

ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της εργασίας είναι να παρουσιάσει δύο προσεγγίσεις για την παρακολούθηση της Ποιότητας Εμπειρίας (Quality of Experience - QoE) σε δίκτυα Long Term Evolution (LTE) Release 13 του 3rd Group partnership project (3GPP). Πιο συγκεκριμένα, το πρώτο κεφάλαιο παρουσιάζει την εξέλιξη της κινητής τεχνολογίας και το κίνητρο για την παρακολούθηση του QoE. Το δεύτερο κεφάλαιο διερευνά την έννοια του QoE (ορισμός, παράγοντες επηρεασμού και μοντέλα QoE), την αρχιτεκτονική και τα κύρια συστατικά του δικτύου LTE (ειδικά του Release 13). Το κεφάλαιο αυτό, αναλύει επίσης το μοντέλο ARCU και την επιχειρησιακή προσέγγισή του που ονομάζεται QoE layered model, που χρησιμοποιήθηκαν και στις δύο προτεινόμενες προσεγγίσεις παρακολούθησης. Επιπρόσθετα, στο τέλος του κεφαλαίου περιγράφεται μια σύντομη επισκόπηση του Big Data. Το Κεφάλαιο 3 περιέχει την περιγραφή των στρατηγικών QoE-agent, δηλαδή δύο προτεινόμενων προσεγγίσεων παρακολούθησης. Στο κεφάλαιο 4 παρουσιάζεται η αξιολόγηση των επιδόσεων των δύο προσεγγίσεων. Τέλος, το κεφάλαιο 5 παρουσιάζει τα συμπεράσματα της πτυχιακής εργασίας καθώς και σχετικές ανοικτές ερευνητικές προκλήσεις.

ABSTRACT

The purpose of this thesis is to present two approaches for monitoring the Quality of Experience (QoE) in 3rd Generation Partnership Project (3GPP) Release 13 Long Term Evolution (LTE) network. More specifically, the first chapter presents the evolution of the mobile technology and the motivation for monitoring the QoE. The second chapter explores the concept of QoE (definition, influencing factors and QoE models), the architecture and the main components of the LTE network (specifically Release 13). This chapter also analyses the ARCU model and its operational approach that is called the QoE layered model that they were used in both proposed monitoring approaches. Moreover, in the end of the chapter, a short overview of Big Data is described. Chapter 3 contains the description of the two proposed monitoring approaches which are the QoE-agents strategies. Chapter 4 presents the performance evaluation of the two approaches. Finally, Chapter 5 presents the conclusions of the thesis together with related open research challenges. .

ACKNOWLEDGEMENTS

This thesis was the work of my practical placement in University of Cagliari (Italy). There I collaborated with researchers and professors in order to research on Quality of Experience. I take this opportunity to express gratitude to my thesis supervisor, Dr. Periklis Chatzimisios, (Associate Professor at the Department of Informatics, CSSN Research Lab), for providing me with all the necessary facilities for this research and for sending me to the receiving institution. I am also grateful to Elisavet Grigoriou, Early Stage Researcher in QoE-NET. I am extremely thankful and indebted to her for sharing expertise, and sincere and valuable guidance and encouragement extended to me as I started to learn about the concept of QoE and the 3GPP LTE technology. Finally, I wish to express my sincere thanks to Luigi Atzori and Virginia Pilloni, Associate Professor and Assistant Professor at the University of Cagliari respectively, for their help and guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AF	Application Function
AI	Artificial Intelligence
DPI	Deep Packet Inspection
eNB	Evolved Node B
EPC	Evolved Packet Core
EPS	Evolved Packet System
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
IF	Influence Factor
IoT	Internet of Things
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MA	Master Agent
Mbps	Megabits per second
ML	Machine Learning
PCC	Policy and Charging Control
PCRF	Policy Control Rules Function
PDN	Packet Data Network
PGW	Packet Gateway
QoE	Quality of Experience
QoP	Group of Picture
QoS	Quality of Service
SA	Slave Agent
SNMP	Simple Network Management Protocol
TDF	Traffic Detection Function
VoIP	Voice over IP

CHAPTER 1 - INTRODUCTION

In this chapter we discuss the evolution of mobile technology in the last years and the technological advancement of the mobile communications. We also discuss the reasons and motivation that led to Quality of Experience (QoE) monitoring and management.

1.1 Mobile Technology Evolution

Similar to the evolution of computers, mobile phone technology has changed over the past decades, and mobile networks/services are becoming indispensable for many people by having an increasingly important role for them. In addition to traditional voice communication service, many new data services are emerging and becoming popular especially with the rise of mobile phones. They made user’s life much easier with their advancement from feature phones (a mobile phone that incorporates features such as the ability to access the Internet and store and play music but lacks the advanced functionality of a smartphone) to Smartphones running on different operating systems such as Android and iOS. According to Cisco Visual Network Index (VNI) by 2021, there will be 11.6 billion mobile-connected devices, including Machine-to-Machine (M2M) modules, and 79% of the total traffic will come from 4G networks [1].

In the last fifty years there has been a tremendous development of mobile network technologies including the evolution of mobile services. Mobile 1G, which is the first generation of mobile communications, established seamless mobile connectivity introducing mobile voice services. The Second Generation technologies (2G), e.g., Global System for Mobile communications (GSM) and Code Division Multiple Access (CDMA), increased the voice capacity delivering mobile services to the masses and has extended the voice-only service to data access service such as Short Messaging Service (SMS) and Wireless Application Protocol (WAP) services. Afterwards, the Third Generation (3G) technologies, e.g., Wideband CDMA (WCDMA) and Time Division Synchronous CDMA (TD-SCDMA), have improved the data access and led to the optimization of mobile for data enabling mobile broadband services. The Orthogonal Frequency Division Multiple Access (OFDMA) technology enables the Long Term Evolution (LTE) and LTE Advanced (LTE-A), i.e., the Fourth Generation (4G), to provide an even better Quality of Service (QoS) to users with improved data rate and to support more capacity for faster and better mobile broadband experiences, and is also expanding in to new frontiers (Extending LTE Advanced to unlicensed spectrum, dynamic LTE broadcast beyond mobile for terrestrial TV etc.). Mobile 4G LTE is the first global standard for mobile broadband and the standardization process was finished in 2011. Many projects in different countries and by different organizations around the world are researching about the Fifth Generation (5G) mobile technologies, and relevant testing is made by Huawei [2], Ericsson [3] and other companies that are contributing to the development of 5G [4].

1.2 Motivation for QoE monitoring and management

The way of our communication has changed with the advancement of cellular technology. Voice telephony was impacted at first but now the 4G networks enable real-time access to application/services with richer content. [5]. The development of 5G is in progress, and it will provide new stunning technologies such as virtual reality, 3D videos etc., where user's acceptability and satisfaction of the users is a major issue. In previous technologies, like the era of 2G, where the main service of communication systems was voice service, the network providers didn't consider QoE evaluation of of the communication systems but only QoS. QoS parameters like the delay, the coding rate etc. were well suited to perform the network evaluation but the QoE parameters are needed to describe user experience during a service/application. In 3G and 4G networks, the increased popularity of smart phones led to the expansion of wireless data services types that are supported by mobile network operators. According to the report from Cisco Visual Networking Index the global mobile data traffic was 7 % of total IP traffic in 2016, and will be 17 % of total IP traffic by 2021 [1]. The mobile users tend to pay more attention to their experiences, which leads operators and vendors to provide better services that may lead to better user experiences. In order to avoid customers' churn and attract new users, mobile service providers need to monitor and manage the experience of their users. In the era of big data, personal user data collection and reservation are feasible under the premise of privacy protection. Note that, in the wireless communication infrastructure, the data from a service on both mobile terminal and network side can be preserved within a certain period, such as users' cookies stored in the mobile terminals [6] and web logs stored in a web server [7] .

In this chapter we discussed about the evolution of mobile and cellular networks, the reasons and the motivation that led to QoE monitoring on those networks. The next chapter analyses the concept of QoE along with the factors that influence it, and describes the assessment methods and the types of QoE modelling.

CHAPTER 2 - BACKGROUND AND RELATED WORK

In this chapter, we present an overview of the QoE, the LTE networks and Big Data aspects. In particular, the definition of QoE considering different aspects, the influence factors of QoE, the QoE assessment methods, the Application-Resource-Context-User (ARCU) model with its operational extension and the LTE network entities are presented.

2.1 QoE

The network parameters such as throughput, loss rate, delay, etc. are used to assess the quality of a service provided by the network provider. However, QoS assessment considers only the network parameters and it doesn't consider other factors related to the usage of a service, such as user's satisfaction, application parameters, etc. This led to the need of QoE assessment because it takes into account the contextual factors and the factors related to the user such as its expectations and preferences. The QoE concept is used in telecommunications and determines if a service is satisfying the end user. In contrast to the QoS which is a concept of network operating conditions, such as noise, lost or dropped packets, QoE takes into consideration the end-to-end connection and the applications that are currently running over that network. Moreover, it considers how multimedia elements such as Internet video or Internet Protocol Television (IPTV), are satisfying or meeting the end user's requirements. In general, QoS depends on the quality of interactions between applications and network, while QoE is based on the interactions between users and applications.

2.1.1 QoE Definition

QoE is a highly multi-disciplinary concept and despite the growing research activities around the end-user experience, the concept of QoE is still an ambiguous concept that lacks a commonly accepted definition and a coherent theoretical basis [8].

Many international organizations proposed QoE definitions such as International Telecommunication Union (ITU) that defines QoE as "*the overall acceptability of an application or service, as perceived subjectively by the end user*" [9], while the European Telecommunications Standards Institute (ETSI) defines QoE as a "*measure of user performance based on both objective and subjective psychological* *measures of using an ICT service or product*” [10]. The QUALINET community proposed the definition of QoE “*as the degree of delight or annoyance of the user of an application or service”* [11] based on the results of the research project COST Action IC1003.

Not only international organizations but also different researchers proposed different QoE definitions according to their own research. Khalil in [12] observed that it is hard to involve objective human factors in ITU’s definition. Therefore, he proposed his own definition which is: *a blueprint of all human subjective and objective quality needs and experiences arising from the interaction of a person with technology and business entities in a particular context*. Pyykko in [13] took into consideration a mobile video scenario and presented a definition in which QoE is “*the binary measure to locate the threshold of minimum acceptable quality that fulfills user quality expectations and needs for a certain application or system”*.

Concluding, that QoE is related to user satisfaction between users and services.

2.1.2 Influencing Factors

The user’s satisfaction is influenced by a number of characteristics such as level studies, gender, knowledge, device type, network type, etc. These characteristics derived as parameters from a service provider, a network operator and/or the user itself. Ickin et al. [14] called all these parameters as QoE Influence Factors (QoE IFs), and defined them as “*any characteristic of a user, system, service, application, or context whose actual state or setting may have influence on the Quality of Experience for the user*”.

A categorization of IFs for the end-users are shown in Table 1 below. They are divided in two categories that are service dependent or independent, and presented in each aspect [15].

The factors listed in Table 1 are associated with the QoE and if these values are changed that means that QoE may be affected but in order to evaluate user’s satisfaction, we need to use specific QoE evaluation and estimation schemes. On the next sub-chapter, we present the QoE assessment methods that may be used to evaluate it.

Table 1 Major QoE Influence Factors

Aspect	Quality Influence Factors
Service-independent	
Transport/Network layer	Round trip / one-way delay, jitter, packet loss ratio, delay burstiness distribution, loss burstiness distribution, congestion period, packet size.
Physical Layer	SNR/SIR/SINR, throughput, bottleneck bandwidth, bit rate, BLER, outage probability, packet / symbol / bit error probability, outage capacity, ergodic capacity / rate / throughput, diversity order / coding gain, area spectral efficiency.
Equipment factors	Codec, dejittering buffer characteristics (overflow, delay), voice activity detection (VAD) / temporal clipping, echo cancellation, noise suppression artefacts, packet loss concealment (PLC) algorithm, talker echo loudness rating (TELR).
Mobile networks additional factors	Transient loss of connectivity (due to handovers), battery consumption, session establishment delay, accessibility, availability, reliability, grade of service (GoS), quality of resilience (QoR).
Common factors	Charging policy and cost, service support, privacy, security, fidelity, conversational task, usability, accuracy, efficiency, context of use (environment, etc.), ambient noise level and variation, equipment brand, service provider reputation, comfort.
Service-dependent	
Video specific	Frame rate, video bit rate, video content (almost static / high motion, etc.), packet loss visibility, re-buffering, group of pictures (GoP) size and structure, video & audio synchronization, terminal type, monitor specifications, display size, type and resolution, ambient luminance, codec type and implementation, video resolution and video format, key frame interval, freshness, blocking.
Video on demand	Video streaming: Number and duration of stalling events, total video duration, initial delay (start-up delay) – For HTTP adaptive streaming (HAS): time on highest layer, frequency and altitude of switches, chunk size, buffer size, etc.
Download-type services	Web-browsing: web-page download time – For FTP: data rate, file download time, delivery synchronization.
Voice	Service-independent factors apply (e.g. packet loss ratio, delay, codec, coding rate), call setup success ratio / blocking probability, call setup time, call cut-off ratio, start-up time, response time.

2.1.3 Assessment Methods

There are different approaches that we can follow to evaluate the QoE level of a service but in general QoE evaluation assessment methods can be divided into 3 categories which are the subjective method, the objective method and a combination of these two, that is called hybrid method. Their detailed description is presented below.

2.1.3.1 Subjective Assessment

Subjective assessment is conducted by tests that are based on real life psychological/visual experiments with human participants who evaluate their experience. It is the most complicated and expensive method but the most reliable one. They are two ways of involvement by the participants. There is the “passive” way that the users only view/listen or the “interactive” way in which the users can be part of a conversation and later they evaluate the quality by the test manager by answering/filling a form. The output of these tests is measured using Mean Opinion Score (MOS) metric [16] that corresponds to the average opinion of the users’. These tests need to be predefined and thoroughly designed, especially subjective experiments in controlled laboratory that strictly follows the guidelines of standardizations, before the users’ participation. These guidelines describe all the aspects that needs to be followed, such as room conditions, audio headset or generally the dedicated equipment used for hearing/viewing/talking, test methodologies, guidelines for the selection of the panel, etc. Moreover, the user sample can’t be random but it must be properly selected by following the guidelines considering also user’s age, gender, etc. There are various techniques such as Single Stimulus Continuous Quality Evaluation (SSCQE), Simultaneous Double Stimulus for Continuous Evaluation (SDSCE), etc. that are used for QoE evaluation, in which the users can evaluate each service separately or compare sequential and then select the better one. [17]. However, lately there is also a trend to evaluate QoE in a new and more relaxed way, called “Crowd-sourcing” technique [18]. In this case, the user is at a familiar environment and uses his own equipment. In this kind of settings, the audio/video/etc. is evaluated using “streaming” or “download” approaches. Moreover, this method is considered more realistic and it is open to a much broader public.

Note that, the results of a subjective evaluation method are most accurate because they include a direct feedback from the users. Nonetheless, the cost of this method is too high, and this method cannot be used in real-time scenarios.

2.1.3.2 Objective Assessment

Subjective tests maybe most accurate but they have a high cost, they are time consuming and they can't be used in real-time scenarios like real time monitoring. In general, objective evaluation methods provide a mapping model from an objective metric to a subjective metric such as MOS. The objective QoE assessment methods can be classified into three categories. We will describe these categories using a video streaming service example but these methods can be used for the QoE assessment of any application. The first category is called “full reference” and compares a reference video and a test video frame by frame. There is “no reference” method which analyzes the test video without the need of the reference video. “Partial reference” is the last category and is somewhere in the middle of the full and no reference methods, that considers some characteristics from the reference video and then analyzes the test video taking into consideration the characteristics from the test video.

Objective models estimate the quality perceived by the end-users, without their intervention. But objective assessment is not perfect because it has the disadvantage of inaccuracy. Thus, with objective assessment, we consider that the QoE is estimated and not measured.

2.1.3.3 Hybrid Assessment

Hybrid method combines both previous methods and it works as an automatic objective quality estimator in combination with subjective results. The hybrid methods are based on Machine Learning tools and use a training QoE model based on the subjective tests results. One example of these methods is the Pseudo Subjective Evaluation Method (PSQA) [19] that uses Random Neural Network (RNN). The PSQA usually goes through four stages: (1) Generate influencing factors, (2) Measure subjective quality, (3)Train neural network model and (4) Use the neural network model for evaluation.

At the first step many samples taken from a service, will be generated in the database, after parameter optimization. In the second step, the testers will grade the samples from the first step. Step 3 comes with the training data set, which is the data collected in the two previous steps, that will train the random neural network. The neural network can use other machine learning tools such as the Support Vector Machine (SVM) or the Bayesian network. In the last phase, the neural network model trained and is ready to be used for evaluation. The only thing that is needed is to input data from the service that we want to assess into the trained RNN.

2.1.4 Objective quality assessment models

In the literature, we can observe that the proposed QoE Models are user-oriented, network-oriented or application-oriented. We observed that quantitative parts of QoE are affected by the network QoS parameters. QoE can be measured and quantified, which leads subsequently to a mapping correlating the QoS parameters with the measured QoE metrics. Thus, the development of an effective QoE-aware QoS model is feasible. There are objective models that have been designed for QoE estimation. The ITU G.1011 [20] has classified these models/objective quality assessment methodologies. The objective quality assessment models have several uses [20]:

- **Planning** takes place before the services of the networks/systems are implemented. It refers to the estimation of the perceived quality of services and since it is not used in real time environment, it doesn’t need any real-time inputs to the objective model.
- **Lab-testing** takes place in the laboratory and it refers to the estimation of the perceived quality of services while the equipment is being developed.
- **Monitoring** is about networks/systems that are operational, and it refers also to the perceived quality of them. Monitoring requires as input the necessary information that is collected from the network and later analyzed to reflect the degradation of the quality experienced by users.

The types of objective quality assessment methodologies are the following [20] :

1. Media layer model
2. Parametric packet-layer model
3. Parametric Plannning model
4. Bitstream layer model
5. Hybrid model

2.1.4.1 Media layer models

Media layer models predict the QoE by analyzing the media signal (audio/video) and use it as input. They consider information related to codec compression and channel characteristics. The models can estimate QoE by comparing (full reference/reduced reference) the output (degraded) signal to the input (clean) signal or by just analyzing the output (degraded) signal (no reference). The full reference models are used mainly for QoE assessments in the laboratory (codec comparison/optimization) because they use both signals. On the other hand, reduced reference and no reference models can be applied for QoE monitoring at some mid-point or end-point e.g. in an IPTV network.

The major representative of this category is described in ITU-T P.862 [21]. It compares the original reference signal with the degraded output signal while it passes through the communication system. Later a Perceptual Evaluation of Speech Quality (PESQ) score is mapped to subjective MOS listening quality.

If the media signals are not available, then this model cannot be used. In order to use this model, we must implement them in specific environments where the input signal is available for processing.

2.1.4.2 Parametric Packet-Layer models

Packet-Layer models predict QoE from packet-header information without handling the media signal itself. Because it doesn’t look at the payload information, it has difficulty to include aspects of QoE that are related to media content but the advantage is that they have a very light measurement of computational efficiency. These models are mostly used as network probes at mid-point or end-points of the network.

2.1.4.5 Parametric Planning models

Parametric Planning models take quality planning parameters for networks and terminals as input. Usually, this type of model requires a priori information about the system under testing.

Standard example of these models is the E-Model [22]. In VoIP applications, similarly to conventional telephony, the QoE is expressed in terms of how clearly the user can listen and understand the transmitted speech, and how easy or not the communication is, due to potential arrival delays of speech internet packets. A classic linear model is the E-model that is recommended by ITU-T G.107 and it is used to predict the overall quality. It is a parametric objective method and is the most reliable and representative approach [22]. The main characteristic of the E-model is that it is restricted in voice service over telecommunication networks. To conclude, the speech related quality parameters in these models could be the coding scheme, bitrate, packet loss rate, delay, etc and the video-related quality parameters could be delay, image resolution, display size, etc.[G.1070]

Some QoE assessments are based on the “IQX hypothesis” [23] that is a perception-centric QoS-QoE mapping. This approach presents that the relationship between QoE and one QoS degrading parameter is negative exponential and the perceived change of QoE actually depends on the current level of QoE. This means that the sensibility of the QoE becomes very noticeable when we have higher experienced quality. Thus, if the QoE is very high, a small disturbance will strongly decrease the QoE and if the QoE is already low, a further disturbance is not perceived significantly. One example is the restaurant QoE example: if we go to a five-star restaurant, a single spot on the clean tablecloth would strongly disturb the atmosphere but the same incident would go unnoticed in a simple tavern. It is found that QoE is related to the QoS impairment factors such as packet loss or network delay in the model of IQX hypothesis, while QoE is related to the perceivable QoS resource like bandwidth or bit rate in the logarithmic model. Some parametric models that conform to the IQX hypothesis can be found for video streaming, such as YouTube [24] , as well as HTTP Adaptive Streaming (HAS) [25].

The ITU-T P.564 [26] is a representative parametric model and it predicts the impact of observed IP network impairments on a one-way listening quality experienced by the end-users. It is a no-reference approach and it exploits the packet header / payload information that later uses to assess the QoE score. It uses the time-stamps and the sequence numbers of the packets that travel in the network. The model is applicable for (passive) quality assessment and live QoE monitoring and assessment.

2.1.4.6 Bitstream-Layer models

Bitstream-Layer model can be positioned between the Parametric planning models and the Hybrid models because it is not only considers information the encoded bitstreamas as input. These models extract information from the bit stream that is delivered through the network. However, they cannot be applied if payload is encrypted.

2.1.4.7 Hybrid models

Hybrid models is, as its name implies, a combination of subjective and objective models. It is an effective model in terms of extracting as much information as possible to assess the objective QoE.

2.1.5 ARCU Model

There are different types of services and they have different QoE requirements. The authors in [27] had proposed a solution that includes different parameters to meet those requirements. These parameters are mapped in different spaces composed by different dimensions related to service characteristics. The ARCU model is a generic QoE model and it is independent of any service type. The name of this model comes from the combination of the four multi-dimensional spaces initials "Application, Resource, Context, and User".

ARCU model provides a methodology to identify the Influence Factors (IFs) of QoE in a systematic way and it models those factors in four multi-dimensional spaces:

1. **Application space (A):** This space represents the application/service parameters and factors. Some examples of such factors are media encoding resolution, sample rate, frame rate, buffer sizes etc. In this space the content-related factors also are included like the type of content (2D or 3D), the color depth etc.
2. **Resource space (R):** The resource space represents the technical part of the delivery of the service. This space is related to the characteristics, the performance of the system and the network resources. Some examples of such factors are delay, jitter, loss, error rate and the throughput (parameters of the network QoS) and also the system resources like the server processing capabilities, along with the end-user device capabilities such as computational power, memory, screen resolution, user interface, battery lifetime, etc.
3. **Context space (C):** This space includes the circumstances under which a service or application is being used. That means that parameters like the time of the day, the user's location and ambient conditions like lighting belongs to this space. Moreover, the dimensions that represent the

economic context such as the cost that a user is paying for a service, are also included in this space.

4. **User space (U):** This space refers to a specific user of a given service or application. The users' preferences, requirements, expectations and prior knowledge are some factors considered in this space. Also the mood, the motivation and attitude can influence the quality perception of a user [28] and they belong to this space.

The advantage of this model is that it distinguishes the factors that are related to the actual applications and the media configuration parameters from the network/system resources. That happens because these sets of parameters may be considered and varied independently and by different actors (QoE models, monitoring algorithms etc.).

The proposed model was introduced by L. Skorin-Kapov and M. Varela [27] and is illustrated in Figure 1. The dimensions of each space may have different types of scales (e.g. ordinal, interval and ratio scale). A simple form of the model is the direct sum of the spaces (Figure1):

$$\text{ARCU} = \text{A} \oplus \text{R} \oplus \text{C} \oplus \text{U} \qquad (2.1)$$

The factors that are influencing the quality in the A, R, C, U spaces can be seen as independent from each other but in practice there is often a correlation between different subsets of parameters, both within intra- and inter-space. The points from the A, R, C, U spaces are later mapped to points in the QoE space. The QoE space is composed of dimensions that represent the quality features that can be perceived by the end user in different quantitative and qualitative metrics (e.g. MOS, efficiency, comfort etc.). For different services, a choice of the quality dimensions needs to be made in a way that it includes all the available aspects of the particular service's QoE. Then from the results of the QoE space a new mapping function (linear or one more complex) can be used to estimate a scalar QoE value if needed (such as MOS scale).

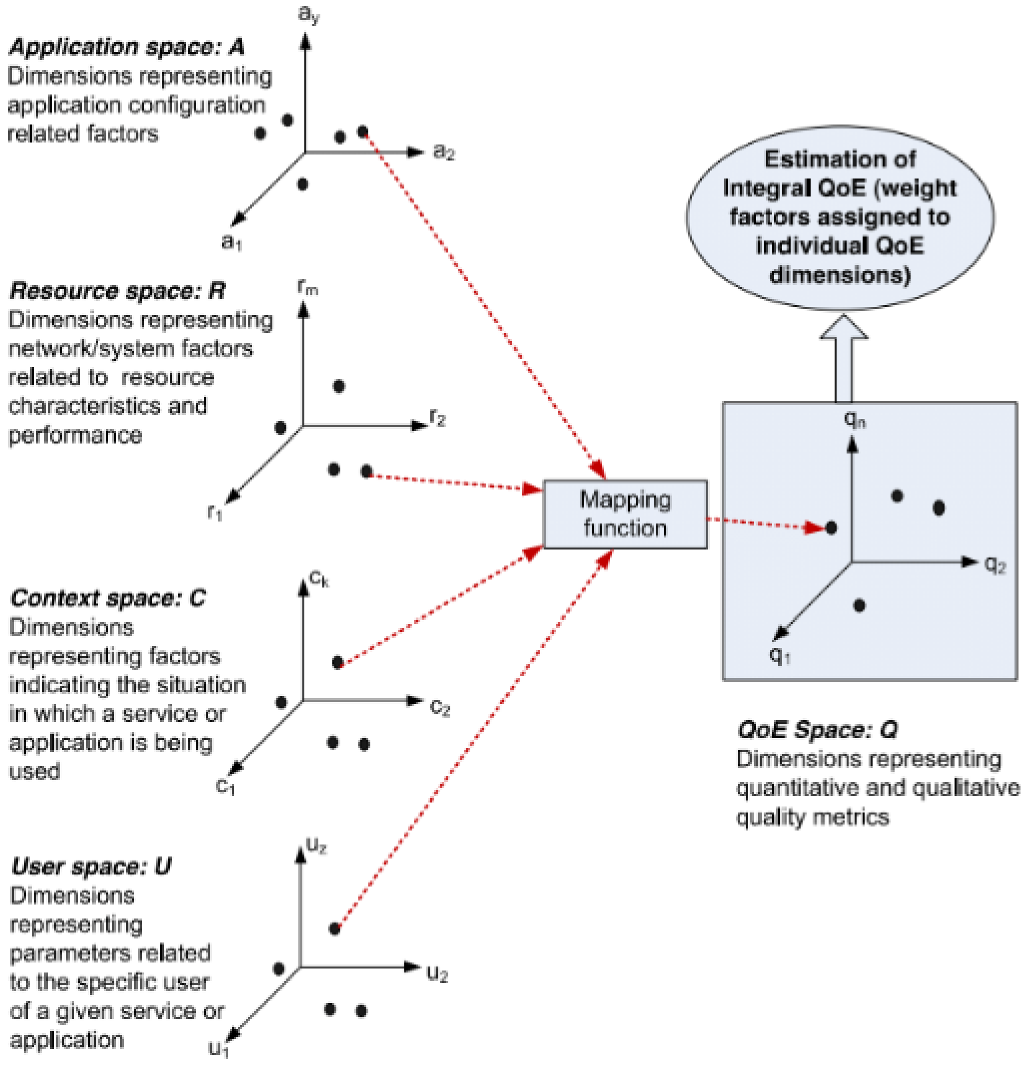


Figure 1 The ARCU model [27]

In some cases, some factors are limited by others and this leads to constraints on the regions of each space (e.g. bandwidth and video resolution) and creates an impact on the feasible regions of the QoE space. A simple form of the mapping function can have the form:

$$Q : ARCU \rightarrow Q, where\ Q = R^n \qquad (2.2)$$

Where Q is the QoE space, and n is the number of dimensions in the QoE space of the service that the model is used. The overall evaluation of subjective user perceived quality, according to the authors of [27], should be based on a weighted, nonlinear, combination of the metrics that affect the quality (dimensions). An issue is to determine the weights of each space and to do so, we need to know the way and to what extent the different dimensions affect the overall QoE. Due to the fact that for different types of services there are different dimensions to the QoE. For instance VoIP's QoE is affected by noisiness, intelligibility but if we consider the gaming service we have different dimensions to consider.

2.1.6 Operational Approach – The QoE layered Model

The ARCU model is a theoretical model and it doesn't distinguish the spaces (there is no hierarchy between the A, R, C, and U spaces) and the correlations between factors that belong to different spaces (inter-space correlations). These factors are dealt only through the mapping function Q (see formula 2.2). Thus, a need for an operational approach of the ARCU model was created. To have a complete knowledge of the mapping Q and to identify all the factors in each constituting subspace of ARCU is not possible and at best you can expect some approximation of them.

Generally, a user doesn't assess the quality of a service/application by the state of the network (delay, loss, throughput, etc.) but through the influence and the impact that the network has during the usage of the application. In a similar way, the behavior of the application could be perceived through the interface that can be for example a device or a screen, and this interface can be affected by ambient conditions (e.g. user doesn't have good readability due to bad lighting conditions). These ideas were presented by F. Guyard et al. [29] and a layered approach was proposed based on those ideas. On this remark the authors of [27] proposed the operational layered approach of the ARCU model.

The proposed layered model has an extended 6 layers that are mapped to the 4 spaces (Figure 2) of the ARCU model:

- **Layer 1 – Resource:** This layer is related to the Resource space of the ARCU model.
- **Layer 2 – Application:** The application layer is related to the Application space in the ARCU model.
- **Layer 3 – Interface:** This layer is related to a part of the Context space of the ARCU model. In particular it is related to the "technical" part of the space which means that it represents the physical equipment and interface that a user is interacting with an application or a service.

- **Layer 4 – Context:** The non "technical" part of the Context space is related to this layer, and it corresponds to the factors that are not included in the Interface layer. Factors related to physical context, usage context and economic context can be found here.
- **Layer 5 – Human:** The human layer is related to the User space of the ARCU model that is about the psycho-physical part. Factors that are related to the perceptual characteristics of users are found here.
- **Layer 6 – User:** This layer is also related to the User space of the ARCU space that doesn't belong to the Human layer. These factors are about the aspects of humans as users of an application or a service.

The layers of the QoE layered model include parameters, processes, inputs and outputs. Therefore, this model is distinct from the ARCU model and it is presented as an operational model in order to design objective models for QoE. The authors note that the two bottom layers of the QoE layered model can be seen as the 7 layers of the OSI model and that means that the QoE layered model can be considered as an 11-layer extension of the OSI model.

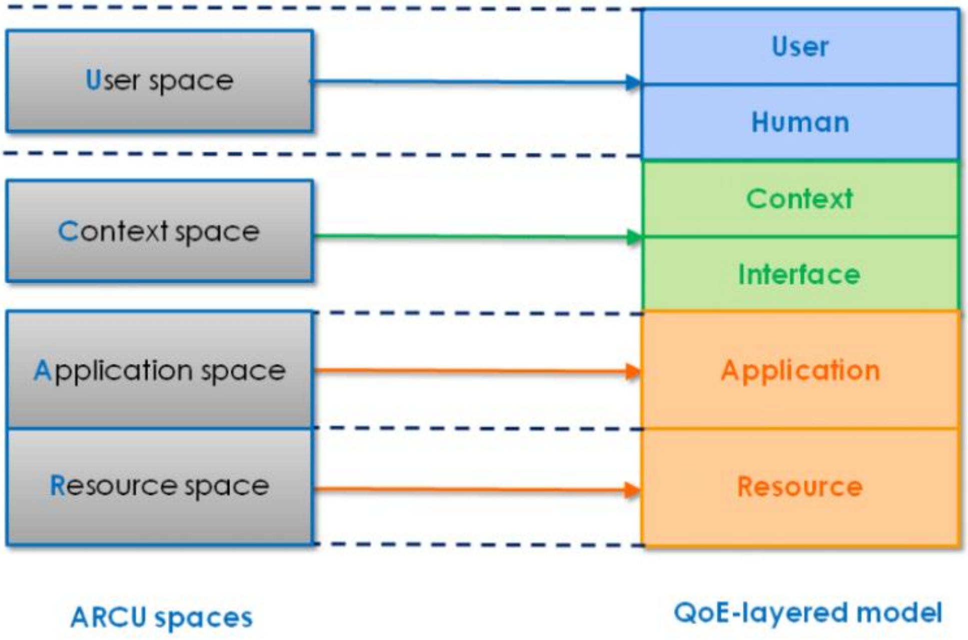


Figure 2 Mapping of the ARCU spaces into layers of the QoE layered model [8]

Each layer follows the specific structure of the layers that the QoE layered model defines. In this model, each layer "L" can be considered as a black box that has an input and an output. The layer "L" contains a vector "I_L" that is the internal parameters and it also contains a process “P_L”. The process “P_L” takes as input a vector “E_L”, which is related to external parameters and the input of the layer, and with the internal parameters of “I_L” which transforms them into the output of the layer “Q_L”. This can be written in as formula $Q_L = P_L (I_L, E_L)$. “P_L” is considered as the objective quality function and “Q_L” is the objective quality, of layer “L”. This process of the layer is presented at Figure 3. Due to the fact that there are many layers in this model, this process is considered as recursive, meaning that with the output of the previous layer is the input of the next layer.

$$Q_{L+1} = P_{L+1} (I_{L+1}, Q_L) \qquad (2.3)$$

The output Q_L of a given layer L represents the “quality” of the system’s behavior up to that layer L.

The implementation of the QoE layered model, as proposed by ETSI specification [8], is applied in an agent-based architecture, called “QoE-Agent”. The QoE-Agent provides a flexible way to obtain the needed information about the QoE of any service in a distributed environment by deploying QoE estimators. The flexibility of the QoE-Agent and of the layered model is that it provides the opportunity to import any quality model in any layer related to a service/application as long it conforms to the APIs of the QoE-Agent. We can observe a high level UML-like description of the QoE-Agent in Figure 4. Based on QoE-Agent structure, it’s possible to implement different quality models in different layers.

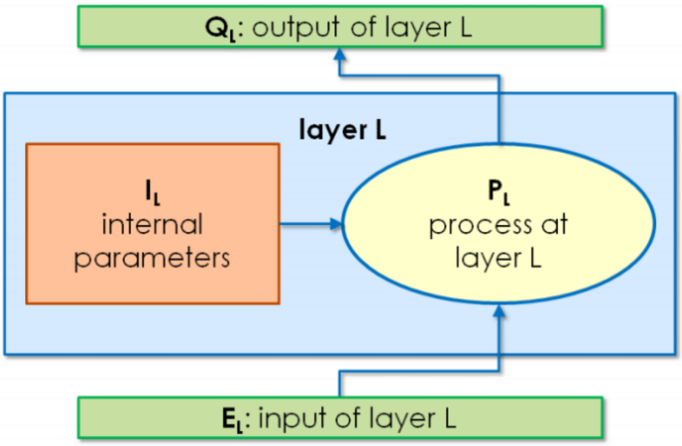


Figure 3 Structure of a layer in the QoE – layered model [8]

However, quality models for the top layers are hard to find and depending on the different applications that the QoE-Agents can be applied to, some layers might be left vacant. The structure of the QoE-Agent helps with by-passing the “empty” layers and the estimations about the quality can be done up to the layer that contains a model. We can observe the recursive process of the QoE estimation done by the agents in Figure 5. In Figure 5 , we also observe that in the last layer i.e. the User layer, all the previous outputs from the layers bellow are included, that leads to the objective QoE estimation of the service.

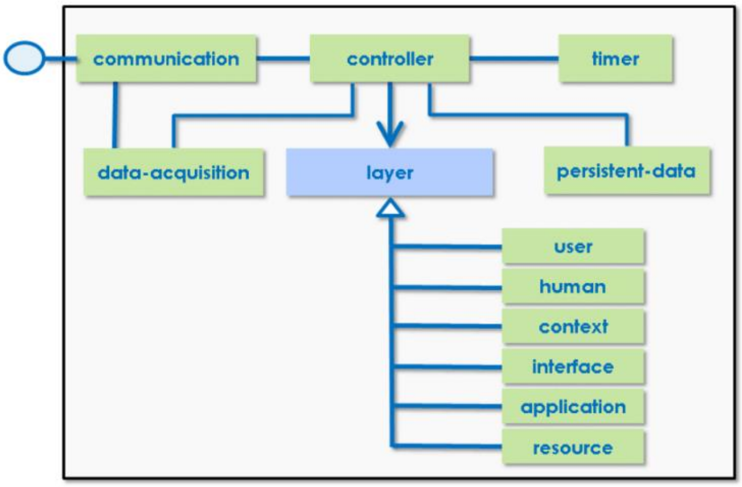


Figure 4 High-level description of the QoE-Agent [8]

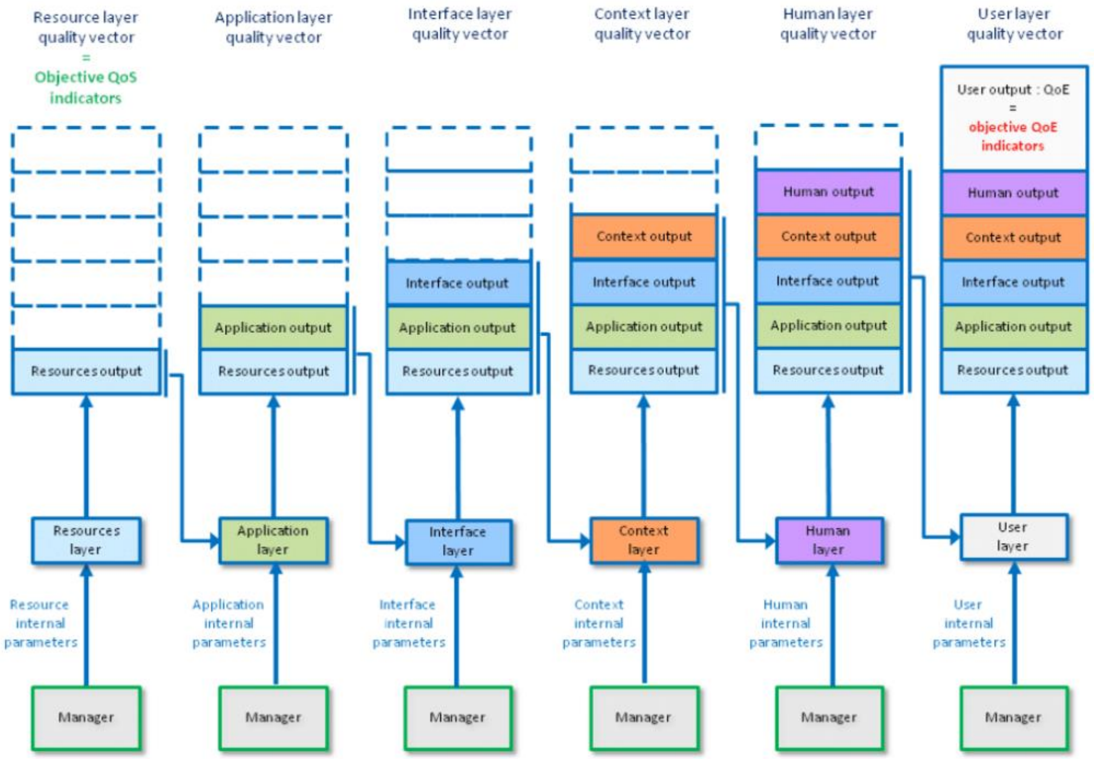


Figure 5 Process of QoE estimation in the QoE-Agent [8]

The probes are used to provide the internal parameters at each layer and they can be either existing commercial tools like tools to measure QoS, or tools specifically designed for the QoE-Agents like a probe that obtains performance data via an API. The protocol that is used for the communication between the Agents and the probes is SNMP which provides an easy way to integrate the Agents into existing tools. Moreover, the communication between the Agents is also performed with the SNMP protocol.

A QoE-Agent except from being embedded in a single device/monitoring equipment (ME), it can collect the data from remote devices/equipment (RE) via the communication API using the SNMP. There are two types of QoE-Agents that can be implemented, the stand-alone QoE-Agent and the distributed QoE-Agent. In Table 2, we observe the components that a QoE-Agent should implement.

The distributed QoE-Agent is used more often than the stand-alone type. The reason for this is because it is more flexible and it deals with situations where the code of the model is not public (because some services create their own private models for their own service) or it has high computational requirements or it runs only in a specific platform. The configurations of both Stand-Alone and Distributed Agents are presented in Figure 6 and Figure 7 respectively.

Table 2 QoE-Agents components

Stand-alone type	**Distributed type**	
Stand-alone QoE-Agent	*Master QoE-Agent*	*Slave QoE-Agent*
Communication	Communication	Communication
Data-Acquisition	Data-Acquisition	Data-Acquisition
Controller	Controller	Controller
Timer	Timer	Timer
Persistent-Data	Persistent-Data	-
Recourse Layer	Recourse Layer	Recourse Layer
Application Layer	Application Layer	Application Layer
Interface Layer	Interface Layer	Interface Layer
Context Layer	Context Layer	
Human Layer	Human Layer	
User Layer	User Layer	

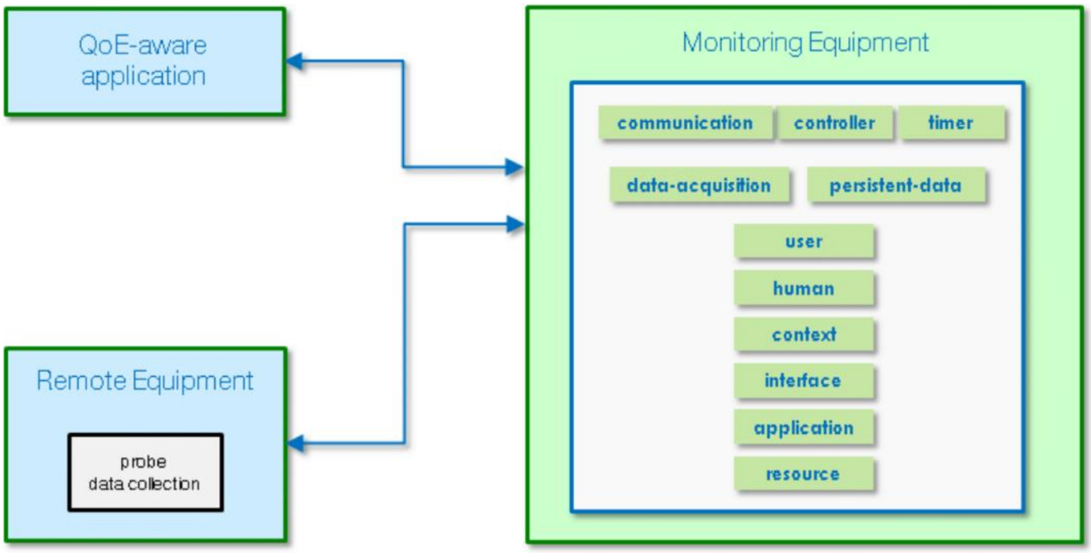


Figure 6 Typical configuration with the Stand-alone QoE Agent [8]

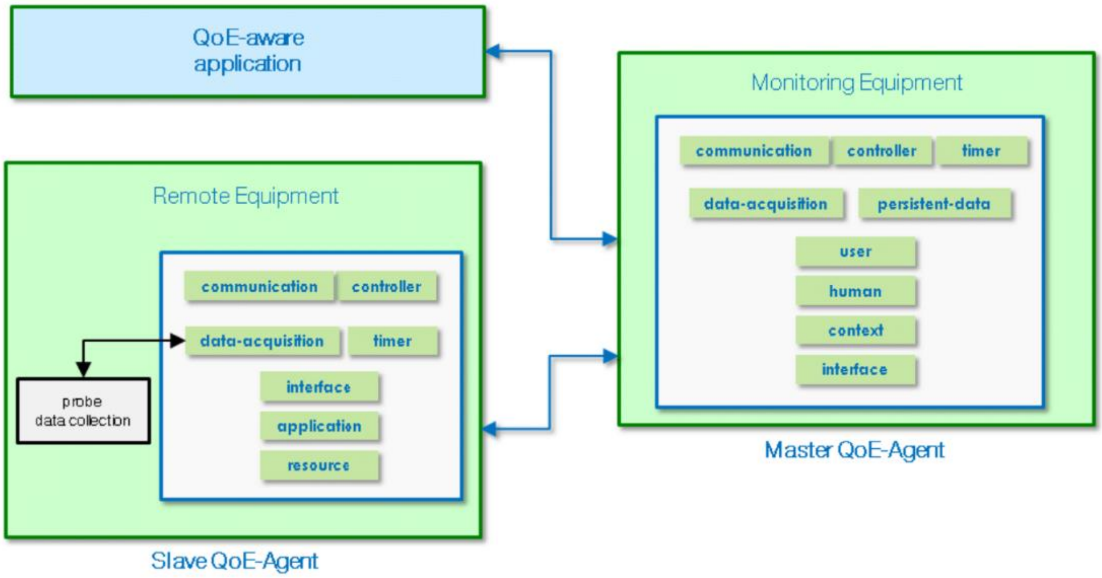


Figure 7 A configuration with the distributed master and a slave QoE-Agents [8]

2.2 LTE-Advanced Pro

LTE, which is also known as Evolved Universal Terrestrial Access Network (E-UTRAN), was first introduced in 3GPP Release 8 [30] and it is the access part of the Evolved Packet System (EPS). The main requirements of LTE were high spectral efficiency, flexibility in frequency and bandwidth as well as high peak data rates. EPS is completely based on packet switching and it includes the core network (Evolved Packet Core (EPC)), the wireless networks (E-UTRAN), the equipment of the end user (UE) and the services. EPC constitutes the core network of LTE. It enables the operation and coordination of various wireless networks, ensuring mobility, handover and roaming subscribers. Handover is when the mobile terminal (mobile station) moves from a radio cell to another during a call or a data connection without interrupting the connection, while the term roaming refers to the ability of a mobile network-participant, automatically receive or send data on a different network than its home network or to have access to other mobile network services away from the home network.

LTE-Advanced (LTE-A) was introduced by 3GPP as part of 3GPP Release 10. The need for this standard came because the previous versions of LTE didn't satisfy all the ITU-R 4G requirements [31] such as that the network should be based on an all-IP packet switched and that the peak data rates must be up to approximately 100 Mbit/s for high mobility like mobile access, and up to approximately 1 Gbit/s for low mobility such as nomadic/local wireless access. LTE-A is a major enhancement of the LTE standard and can provide peak data rates of 1 Gbit/s. Later introduced the LTE-A Pro that defines features and technologies that meet the 3GPP specifications related to Release 13 such as the mission-critical Push-To-Talk which is the essential functionality for LTE to be used by 'blue light' services (an ambulance production company) for private mobile radio voice communication. LTE-A Pro was announced in October of 2015 in order to deal with the increasing data usage and specialized resource management that are required by Internet of Things (IoT) applications and other LTE network cases.

LTE-A Pro can be considered as 4.5G because it is an intermediate technology between 4G that was defined by the first releases of LTE and the upcoming new 5G air interface. It has many use cases from retail applications, live HD video broadcasting to public safety networks that can be used for example by ambulances and other critical networks that safety is very important [32]. Figure 8 shows the overall architecture of the EPS along with the network elements of the LTE-A Pro and the standardized interfaces which they are used for communication. The LTE Advanced pro network can be divided into two separate networks as the LTE architecture, the E-UTRAN which is also called access network and the EPC which is also called core network. The E-UTRAN consists only of one node, the eNodeB (eNB) which provides connectivity of the network with the users' equipment (UE), while the EPC is made up of many logical nodes.

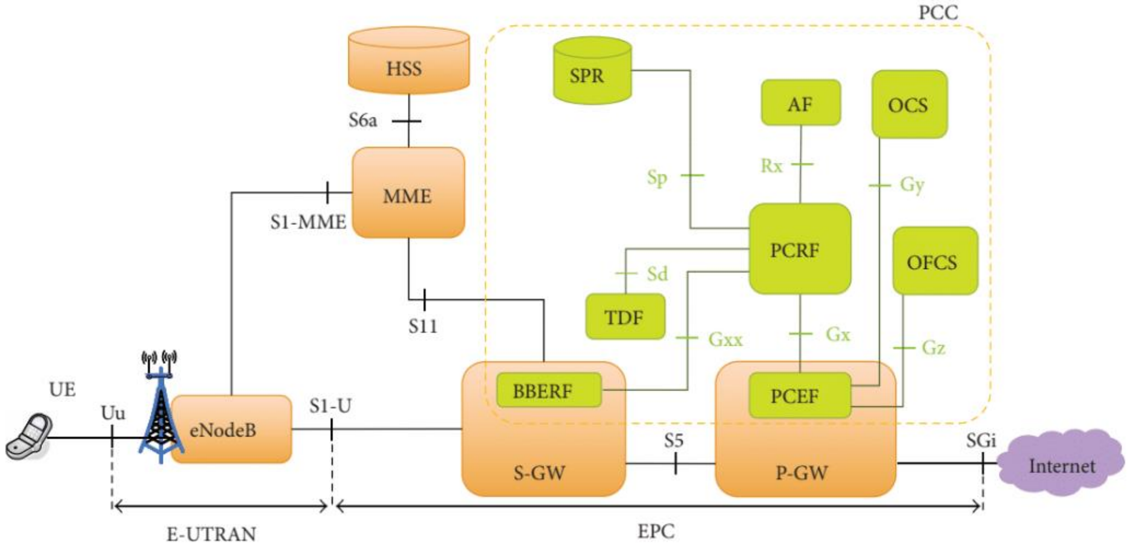


Figure 8 High level architecture of LTE-A Pro EPS [33]

The users access the Internet, run services such as VoIP via the EPS that provides them IP connectivity to a Packet Data Network (PDN), which is a network that provides data services like the Internet. There is a concept in LTE that is called “EPS bearer” that is a logical connection of the terminal with the EPC. The EPS bearer is used for QoS level measurements and multiple bearers can be established for one user. This happens due to the fact that a user can be in a voice call through a VoIP bearer and at the same time download a file through a best-effort bearer. These multiple bearers are used in order to provide different QoS streams or connectivity to different PDNs. The EPS also has a Policy Charging Control (PCC) subsystem that provides service data flow detection and other advanced tools for QoS purposes and charging control. It is responsible for the management of the connections that are made and determines how the bearer resources will be allocated in a given service and it also responsible for the charging control that is applied. PCC is composed of rules that they used to identify the service, to provide appropriate applicable charging and to apply the proper policy control.

The main logical nodes of the EPC are described below.

- **Policy control and charging rules function (PCRF):** This node functions in real-time to determine policy rules. It operates in the network core and it has access to subscriber databases and other nodes like the OCS or OFCS, which are described below, to apply certain policies. It is also responsible for

QoS-policy management and QoS authorization like determining the QoS class identifier and the bit rates based on the user’s subscription profile.

- **Policy and charging enforcement function (PCEF):** This node is part of the PCRF and is inside the PDN GW. The PCEF communicates with the PCRF through the Gx interface about operations that are about the PCC rules such as the outcome of a PCC rule operation. The major difference between PCRF and PCEF is that PCEF does support offline and online charging interactions while PCRF does not support these.
- **Home subscriber server (HSS):** It contains users’ subscription data and information such as the International Mobile Subscriber Identity (IMSI) which is a unique identification associated with all cellular networks and the Mobile Station International Subscriber Directory Number (MSISDN) which is a number used to identify a mobile phone number internationally. It also contains key parameters for authentication when the users’ try to attach on the network. Basically it is a database that contains user-related and subscriber-related information.
- **PDN gateway (P-GW):** It allocates IP addresses and IP prefixes to the UEs and it is responsible for policy control and charging. P-GW is the point to interconnect the EPC with the external IP networks.
- **Serving gateway (S-GW):** It serves as a local mobility anchor for data bearers when the UEs move between eNBs. S-GW is the point of intra-LTE mobility between LTE and other 3GPP accesses. Both P-GW and S-GW deal with the user plane.
- **Mobility and management entity (MME):** The control node that processes the signaling between UE and the EPC is called MME. It handles the signaling which is related to mobility and security for E-UTRAN access. MME deals with the control plane.
- **Application function (AF):** It extracts session information from the application signaling and it sends them to the PCRF so it can apply certain policies.
- **Subscription profile repository (SPR):** It keeps track of the subscriber profile/policies related to QoS. So the PCRF uses SPR as a basis for the decisions that it makes and the policies that it applies. An example of an SPR

information is the user category, which divides the users in business users and regular consumer.

- **Traffic detection function (TDF):** The TDF enforces traffic policies based on pre-set rules or dynamically determined rules by the PCRF on data flows in real time. TDF was first introduced in the Release 11 of 3GPP specifications. It helps with traffic optimization and it has mechanisms for service detection.
- **Online charging system (OCS):** It collaborates with the PCEF in order to retrieve policy and charging authorization for quotas and credit control based on time, traffic volume etc.
- **Offline charging system (OFCS):** It is a mechanism where charging information does affect and have an impact, in real time, on the service that is rendered. It also has interaction with PCEF to generate charging data that they will be later use in the billing system.

2.3 Big Data

Big data concept is related to the collection and the analysis of large amount of data, which has the ability of changing rapidly within a specific period. Network traffic has a large and diverse data set that is generated. The telecommunication industries are interested in data-driven decisions to provide optimal solutions based on the information and knowledge derived from the data [34]. Big data has of five characteristics which are the volume, the velocity, the variety, the value and the veracity. These characteristics are described below:

- **Volume** describes the mass or the quantity of the data.
- **Velocity** deals with the speed that the data are being created, which is how quick the data are generating and how they are being processed to meet the current network demand and prepare for future challenges.
- **Variety** comprises of different types of data, like data that are being generated in the same network traffic.
- **Veracity** describes the accuracy of the data sources. Also it describes the quality of the data sources along with the noise and abnormality that may exist in the data.
- **Value** describes the type of information that can be extracted from the data.

ITU asserts that the data in mobile network traffic characterized by these five characteristics [35]. Hence big data gathered from mobile networks can be used from QoE modelling, estimations and monitoring in a heterogeneous environment like the smart city use case [36].

There are big data analytic tools/platforms such as the ElasticSearch, Logstash and Kibana (ELK) Stack [37]. Elasticsearch is a search and analytics engine, and a database that search capabilities which uses JSON. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. That process is called Extract, Transform and Load (ETL) pipeline. Kibana is a web-based data analysis and lets users visualize data with charts and graphs in Elasticsearch in seconds.

In this chapter, we discussed about the concept of QoE, about the LTE networks architecture and especially Release 13. Later we described the ARCU model with its operational approach, which is the QoE layered-model and the implementation of them using the Agents. All these concepts are considered in order to create the QoE monitoring strategies that are presented in the next chapter.

CHAPTER 3 - DESCRIPTION OF QoE-AGENT STRATEGY

In the previous chapter, we discussed about the concepts and technologies (QoE, LTE-Advanced Pro architecture, ARCU model etc.) that were studied in this thesis in order to construct the two QoE-monitoring approaches over LTE-Advanced Pro networks. In this chapter we will describe these approaches.

3.1 An agent-based QoE monitoring strategy for LTE networks

The first approach [38] was accepted and presented on the IEEE International Conference on Communications of May 2018 in Kansas City. This monitoring approach was based on the operational QoE-layered model as seen on chapter 2. We applied the QoE-Agents in specific entities of the LTE-Advanced Pro network so they can extract information in order to use them on models to estimate the objective user's QoE level. In order to evaluate this strategy we took into consideration the network load to observe if the usage of agents increase it, which is something that we are trying to avoid, and we also considered the accuracy of the QoE estimations. The proposed strategy is only for monitoring and not for management but that doesn't mean that network operators can't further conduct intelligent QoE management. Thus a network operator can use this monitoring strategy and then extend it by applying his own management scheme. The target of this strategy is to make accurate measurements in order to have accurate QoE estimations, without having a negative impact on the network's performance. To evaluate that, we are adjusting the time frequencies of the measurements and we observe the network performance and the accuracy of the estimations.

3.1.1 The QoE-Agents

In order to implement the QoE-layered model, there are two types of QoE-Agents, the Distributed and the Stand Alone, (Figure 7 and Figure 6 respectively). In this strategy, we used the distributed version of the QoE-Agents which means that we have implemented one Master QoE-Agent (MA) and many Slave QoE-Agents (SAs). However, in this approach we propose one MA and one SA. The QoE-layered model consists of 6 layers so the first thought is to deploy one agent to each layer. However, that approach had many problems because it deprives us the flexibility, it gives higher network load and we are not able to achieve our objective.

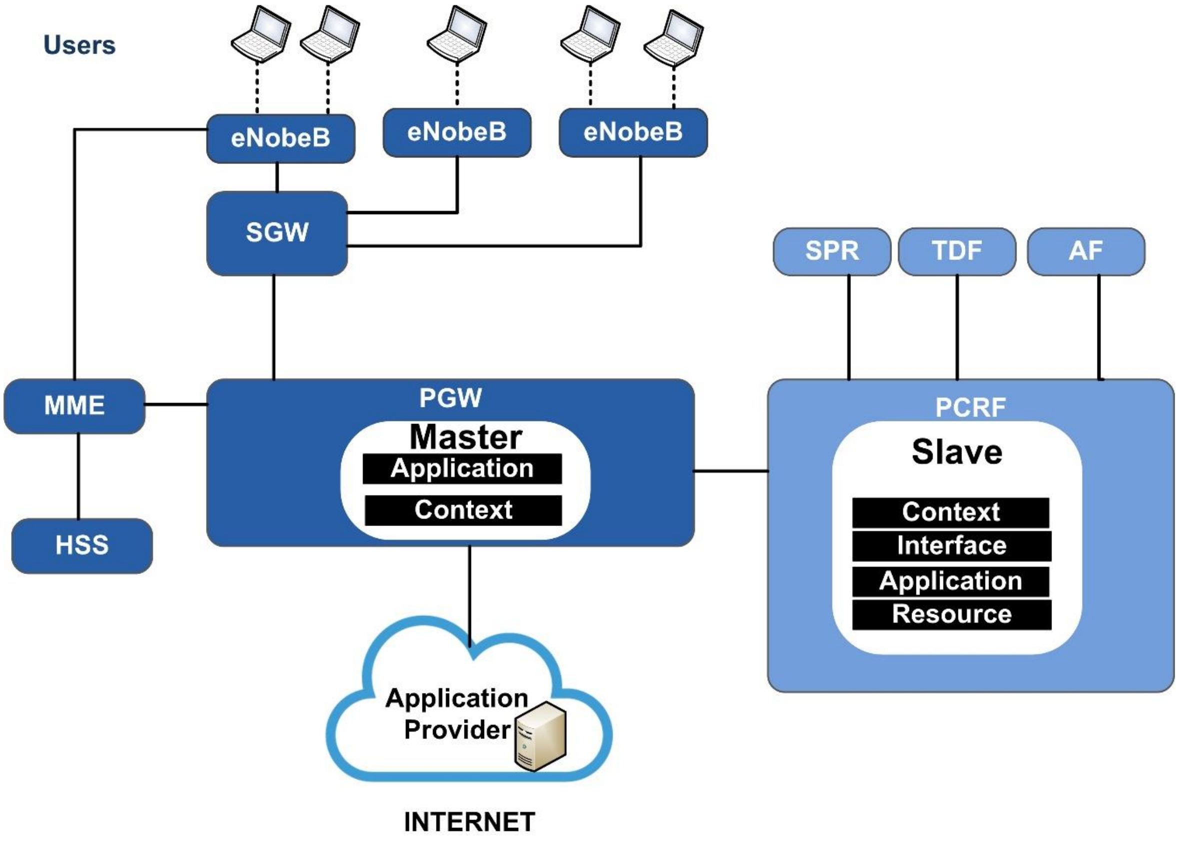


Figure 9 An Agent-based QoE monitoring in LTE-Advanced Pro network architecture [38]

The policy that the PCRF follows in the LTE EPC Core Network Technology, which supports flow-based charging, service data flow detection and policy enforcement, offers a comprehensive solution. It also has the ability to communicate with the network entities that are necessary, such as PDN, AF etc., using different interfaces. Based on that policy and the distributed agents we can achieve our objective and we propose the monitoring strategy that is shown below.

In our strategy, the MA applied in the PGW entity of the LTE-Advanced Pro network and the SA is applied in the PCRF entity as we can observe on Figure 9 and their functionalities were discussed in sub-chapter 2.2. All the estimations are done by the MA, thus the SA only collects the data (in our approach) The MA after the collection of all the necessary data from SAs can estimate the QoE using the appropriate QoE models.

For each layer, the QoE monitoring strategy retrieves different parameters, executes different processes in the form of QoE estimation models and defines different input/outputs. The SA in the PCRF entity implements the first 4 layers (Resource, Application, Interface, Context) of the QoE-layered model and the MA in the PGW entity implements the Context and the Application layers which are the fourth and fifth layer respectively. On Table 3, we can observe for each layer, the process of data retrieval by the Agents, the data sources and Agents' responsibilities.

Table 3 Data retrieval for each layer by the Agents

Layer	Dimension	Sources	QoE-Agent
1-Resource	Delay, jitter, loss, error rate and throughout	TDF- network entity	Slave Agent
2-Application	Application ID, application description, media type etc.	TDF, AF - network entities	Slave Agent
3-Interface	Type of device, screen size, mouse etc.	Collaboration with Application Provider	Master Agent
4-Context	Cost of the service	SPR- network entity	Slave Agent
	Location of the user	BBERF- network entity	Master Agent

By observing Table 3, we notice that layers User and Human of the QoE-layered model are missing, and the reason for that is because those layers are directly related to the user and it's not so easy to retrieve information about the user. In the literature there are several approaches, but we will mention some of them. The first approach is to install an agent in the user's device in order to retrieve feedback, with the user's permission by respecting his privacy, or we can follow the second approach which is the collaboration between the Internet Service Provider and Application provider [39].

Agents' Responsibilities for each layer:

- For the first Layer, which is the Resource layer, the SA retrieves data from the TDF entity related to network parameters in terms of delay, jitter, loss, error rate and throughout. TDF is an interesting entity in the LTE-Advanced Pro network that brings application detection to a higher level than the current DPI capabilities by collecting dynamic, precise real-time information based on device, access type, location, traffic patterns, application information and subscriber plan details that can be used to develop highly personalized offerings.
- The PCRF receives information from the network elements such as AF and TD, that are related to Application layer. The AF is a logical element of the 3GPP PCC framework which provides session information related to the

PCRF with the support of the PCC rule generation. The whole policy control and charging rules function scheme helps with the monitoring strategy and with the retrieval of the needed information. The SA applied in PCRF and retrieves information that are application driven like the Application's ID, description, media type, media format, bandwidth, application service provider etc. and forwards it to the MA.

- The third layer which is the Interface layer, is related to the user. It's related with the interface that the user is interacting with, and we could use the collaboration approach between the ISP and the Application provider in order to retrieve physical equipment and interface related parameters like the type of the device, screen size, mouse etc. that the network provider doesn't have any knowledge. Also in case that the agent is deployed in the user's device, we can retrieve the needed information but we didn't follow this solution because we prefer to follow an automatic approach so the user should have the least participation, in order to avoid his dissatisfaction.
- For the fourth layer, we use the Subscription Profile Repository (SPR) that is a logical database inside the LTE communications standard, containing subscriber related information related to police and charging control. The SA can retrieve information like the cost that a user is paying for a service.

3.2 A QoE monitoring solution for LTE-Advanced Pro networks

When someone looks over the first approach notices that is not flexible and it doesn't give many options to the network provider e.g. choosing real-time or not real-time monitoring, monitor specific area of the network, visualize the data etc. These options along with an intelligent algorithm that we developed are going to be presented and analyzed in this approach, which is as an extension of the previous one. As we dug deeper on the technologies and the models we used on the previous approach, we realized that we have to extend the previous work and propose a more flexible monitoring solution over the LTE-Advanced Pro networks. We meet two requirements in this proposed strategy which are the frequency of the measurements and the network latency. We want to keep the latency as low as possible in order to avoid a bad impact on the network performance. The performance estimation is done by the platform we developed that called "AGENT-

MONitoring” (AGENT-MON) which measures the accuracy, the network load, the computational complexity (in terms of CPU and memory usage) and implements the monitoring algorithm that adjusts the monitoring frequency based on these measurements.

Thus, in order to design the involved strategy, we need to identify in which network components the QoE-Agents should be installed and which are the available applications and their related parameters. It is a challenge to consider a variety of IFs due to the fact that the collected data belongs to multiple dimensions and we need to deal with a diversity of collected data and decide which parameters we need to estimate the QoE of a service in a mobile context. For that reason it’s more efficient to use the distributed agents to collect them. The “Distributed QoE-Agent” approach includes one MA and many SAs, as shown in Figure 10.

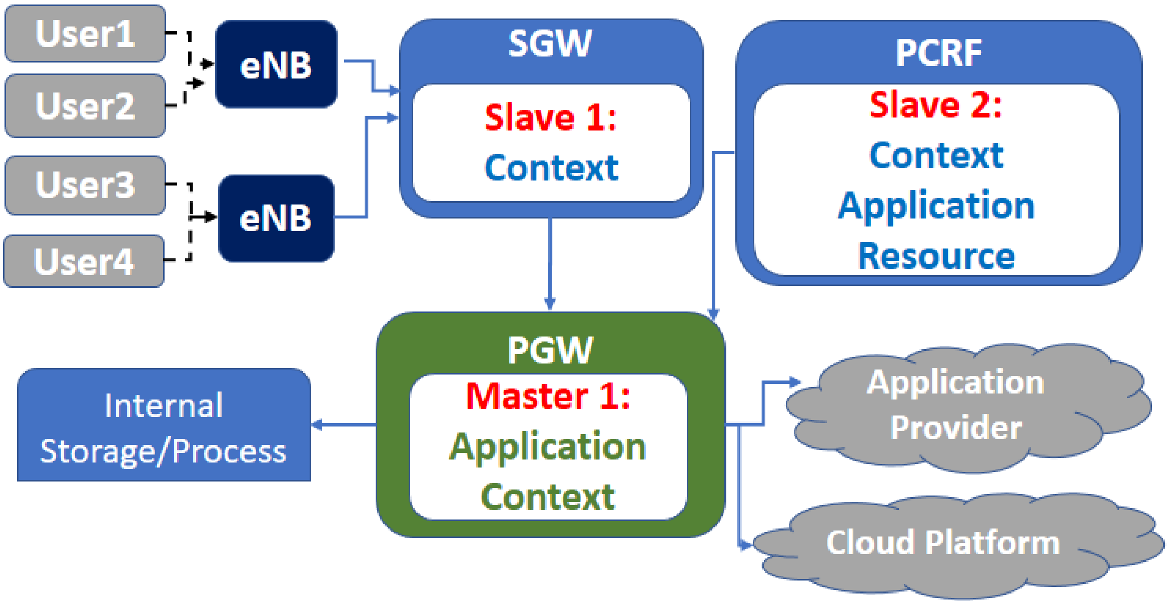


Figure 10 QoE monitoring solution architecture

In the Figure 10, we notice a “Cloud Platform” entity that is collaborating with the MA. In particular the MA collects the data from the SAs and forwards them to the “Cloud Platform” for further analysis using big data analytic tools such as ElasticSearch.

Table 3 presents the parameters that each agent can retrieve and also the agents’ location in the network.

Table 4 QoE-Agents parameters and location

QoE-Agent	Parameters	Location
Slave Agent 1	User's location, user's mobility and device type	Service Gateway
Slave Agent 2	Throughput, application ID and description, economic context	PCRF
Master Agent	Application related information from the Application Provider	Packet Gateway

From the Table 4 we observe that the MA is deployed in the Packet Gateway whereas the two SAs are deployed near to the Packet Gateway. With these locations we have the advantage that we can retrieve data from the E-UTRAN where the SA 1 is located, and we can retrieve information about usage context of the service (Context IFs) such as user's location and mobility through the eNBs. We can also retrieve data from the EPC, where the SA 2 is located, such as network related parameters, application-related parameters and economic-related parameters. Later these data end up in the MA whose location allows him to communicate with the rest of the Internet and acquires data from the SA and forwards them to the Cloud platform and to the internal system to store and analyze them. There is an option given to the ISP to choose one or both approaches to store/analyze the data that were collected in order to estimate the QoE level.

3.2.1 The monitoring algorithm

To make this monitoring strategy more flexible we developed a monitoring algorithm that takes into consideration the computational complexity and the network load, in order to adjust the frequency of measurements and its objective is to improve QoE estimation's accuracy. The algorithm can identify all the available inputs and IFs, sends/retrieves data, communicate with the agents and adjusts the frequency of the measurements. The flowchart of the proposed monitoring algorithm is shown in Figure 11 and its analytical procedure will be analyzed below.

In the proposed algorithm, there are 12 steps which are the following:

- Step 1: The first step is conducted by the network manager, because he needs to define the monitoring type through the "Monitoring Application Interface" for the network provider which can be:
    - Location-based e.g. select a specific geographical area and do the monitoring of the QoE for that area,

  - User-based e.g. monitor the QoE for a specific group of user based on some criteria that separates them like gender,
  - Service-based in which the manager chooses to monitor the QoE for a service that he wants.

The manager defines also the time scale in this step, e.g. if the QoE monitoring is on-per session basis.

- Step 2: The frequency of the measurements is applied by the agents is set to its default value that is every 3 minutes. The default value found from the experiments done in [38].
- Step 3: The identification of all the available inputs and IFs is done in order to realize which parameters should be retrieved by the agents.
- Step 4: Agents know which data to collect and they collect them in step 4a and 4b with the difference that step 4a concerns SA1 who collects context-related information and step 4b concerns SA2 who collects resource, context and application related parameters using standard APIs.
- Step 5: The SA agents retrieve the data and forwards them to the MA. Then, MA it collects these data and sends them to the Cloud platform using the REST protocol and to the local storage for further storage and analysis.
- Step 6: The Cloud platform receives the data and identifies the application type in order to extract the related KPIs in the next step.
- Step 7: The Cloud platform extracts the related KPIs and forwards them to the MA.
- Step 8: MA retrieves the data, and consequently the MA is ready to conduct the QoE estimation.
- Step 9: The MA includes the QoE models and along with the data he received is able to estimate the QoE, taking in consideration the application type and the identified KPIs.
- Step 10: Those results of the QoE estimation are stored to the Cloud storage and in a local database of the network provider.
- Step 11: The algorithm checks the network load in order to adjust the frequency properly.

- Step 12: The algorithm checks the computational resources in order to adjust the frequency properly.

In the algorithm there is a function called “Monitoring Report” which measures the network load, the network latency and the computational resources in terms of CPU and memory utilization, during the data collection and the analysis procedure. This function uses the “qperf” tool [40] to measures bandwidth and latency between two nodes. Based on the results from the “Monitoring Report” function, the algorithm adjusts the frequency of the measurements if needed, which means that if the results are good meaning that the network load is low and the computational resources are low, then it doesn’t need to adjust the frequency. In step 11, after the QoE-estimation, the platform checks the network load and if it’s high, it adjusts (increases) the frequency of the measurements (step 2) and then continues the monitoring. However if the network load is not high, the platform checks the computational resources in step 12 and it does the same procedure, meaning that if they are high it adjusts (increases) the frequency of the measurements(step 2) then and continues the monitoring. In the case that both of them are low then the algorithm continues until the monitoring procedure is stopped by the network manager.

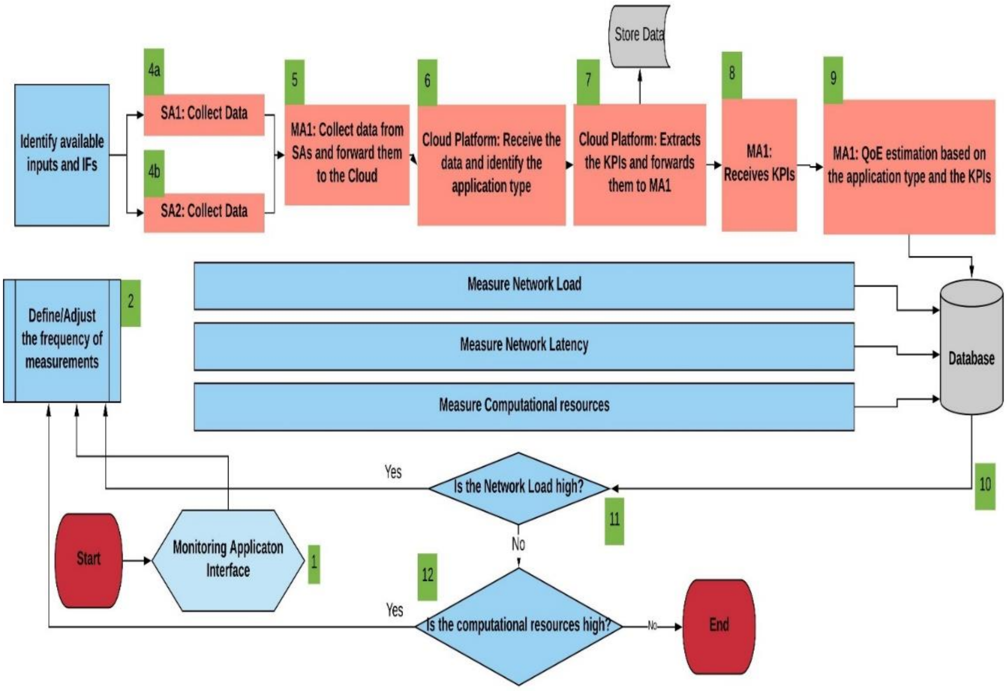


Figure 11 Flowchart of proposed monitoring algorithm

CHAPTER 4 - PERFOMANCE EVALUATION

In this chapter we present the results from the performance evaluation of the two approaches presented in Chapter 3 along with the QoE models used for estimation. Mininet network emulation environment [41] was used in both approaches to emulate the network traffic, to create a media server in order to stream video, and the network tools that are used to identify the network load and the network latency

4.1 Performance evaluation of "An agent-based QoE monitoring strategy for LTE networks"

The proposed strategies are applied in a LTE-Advanced Pro network during a video streaming service. We consider a video streaming application by emulating its traffic, and then we estimated the network performance of the proposed approach.

4.1.1 QoE model

The network performance evaluated by considering the packet loss, packet delay, packet jitter and bandwidth. We computed an overall QoS level as a weighted average of the QoS level for the major performance indicators as presented in *Eq. 4.1.* The normalized QoS value (*QoS*), which refers to the network performance, is calculated with the total sum of the values that are the QoS parameters in the network layer multiplied with their allocated weights. These weights are selected according to the type of the access network for the service.

$$QoS = PL \times W_{PL} + PJ \times W_{PJ} + PD \times W_{PD} + BW \times W_{BW} \qquad (4.1)$$

The above equation calculates the QoS for each path between the Application Provider and the users. We denote that the considered QoS parameters are Packet Loss (PL), Packet Jitter (PJ), Packet Delay (PD) and Bandwidth (BW) and their weights are Weight Packet Loss (W_{PL}), Weight Packet Jitter (W_{PJ}), Weight Packet Delay (W_{PD}) and Weight Bandwidth (W_{BW}) respectively. The value of the weights is assigned according to the quality standard bounds and their relative importance degree which is given from [42], and their values along with which unit they are being measured is shown on Table 4. Later a correlation model also from [42] is used to map the QoS parameters (in our case the QoS equation 4.1), to a QoE metric considering the application's parameters such as the video resolution, the GoP and the estimated video quality.

Table 5 QoS parameters weights

Weight	Importance degree	Measured unit
Packet Loss	58,9 %	Percentage
Packet Jitter	15,1 %	Milliseconds
Packet Delay	14,9 %	Milliseconds
Bandwidth	11,1 %	Mbps

In the equation 4.2 that is shown below we observe that:

- The normalized QoS value is related to equation 4.1
- The constant "A" is related to the video resolution such as 240p, 360p, 480p and 720p.
- Constant "R" reflects to the structure of the video frames according to the GoP length and its value is 24 based on [42].
- Constant "Q_r" is the quality factor that determines the overall QoE of video streaming service and is set to 0.90 based on [42].

$$Est.QoE = Qr \times (1 - QoS)^{\frac{QoS \times A}{R}} \qquad (4.2)$$

In order to estimate the accuracy of the measurements, we used Mean Squared Error (MSE) metric which is the deviation of our estimated value from the true one and it is equal to the variance plus the squared bias. Moreover, if the sampling method and the estimating procedure of the MSE lead to an unbiased estimator, then the MSE is simply the variance of the estimator. Moreover, in order to estimate the computational complexity in the Agent-based monitoring system, we measured the CPU utilization and the memory utilization of the devices that the agents are located each time during the monitoring procedure.

4.1.2 Performance evaluation

The experiments were carried out in an Asus computer with Intel Core i7 @3.6 GHz, 4GB RAM installed with Linux Ubuntu 14.04 64bit with Mininet, and inside of Mininet there was the VLC player version 3.0. In the experimental setup there are 5 hosts, 1 media server (as the Application Provider) and 11 network devices which are considered as the logical entities of an LTE-Advanced Pro network. The logical entities are: 1 SGW, 1 PGW, 3 eNBs, 1 HSS, 1 MME, 1 TDF, 1 SPR and 1 AF. The links are considered as wireless for users and eNBs and all the rest considered as wired. The experimental setup is shown at Figure 9. The hosts (users) request a video from the media server (Application Provider). The media server delivers the video through the LTE network and the users receives it from the media server via the PGW entity. Note that, the MA is installed in the PGW entity and the SA is installed in the PCRF entity.

We have created two experimental scenarios with different network parameters such as delay, bandwidth, jitter, packet loss, and application parameters such as video resolution, to evaluate the performance of our strategy. Using these two scenarios our target is to observe if the estimated QoE level during a video streaming session is affected by the network conditions and by the frequency of the measurements as well as if the video resolution has an impact on that. A specific QoS level is provided and we used Equation (4.1) to estimate that level. Moreover, we evaluate the estimated QoE of each video sequence.

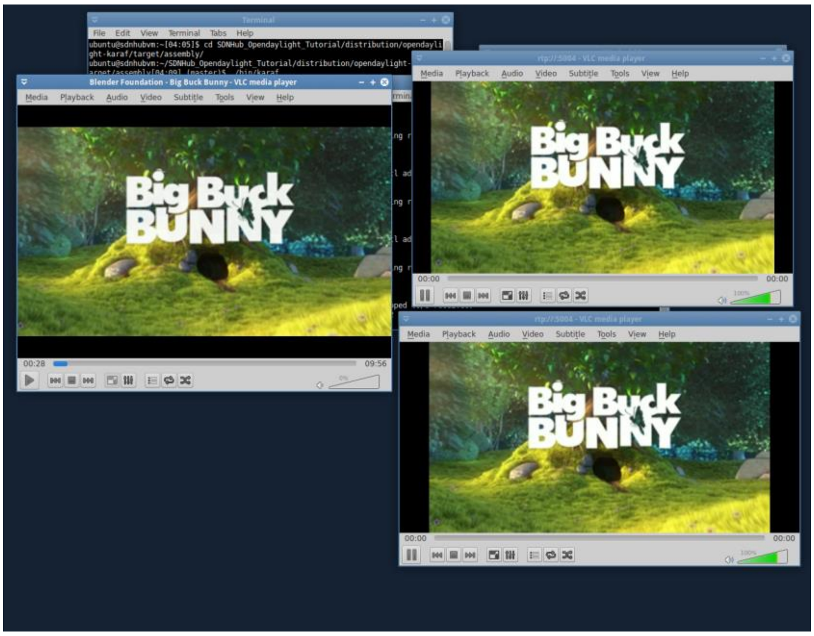
The average end-to-end transmission delay for each path in the network in all scenarios is 4ms. The available bandwidth fluctuates between 5 Mbps and 10 Mbps, respectively in the two scenarios. In each scenario, the media server serves at least two clients and we have added additional background traffic, in order to have real network conditions. A background traffic is another type of traffic which is used simultaneously along with the primary traffic of interest (in our case the traffic came from video streaming sessions). Under ideal network conditions, we have all the necessary network resources and bandwidth. Thus, we want to prove that our strategy during a video streaming service has better performance even in the worst network conditions. For that reason we need to simulate the worst case using bottlenecks and traffic overhead.

Furthermore, in order to find the ideal frequency of the measurements, we have conducted measurements every 1 minute, 3 minutes, 5 minutes, 7 minutes and each second for a duration of 30 seconds. Our target was to observe the results and find the ideal frequency of measurements, in order to avoid overloading the network and consuming many resources in terms of CPU and memory while keeping the QoE at an acceptable level.

The agents execute the following processes in order to estimate the QoE level:

- “QoS level” process is related to Equation (4.1) and calculates the QoS level for the best path [43] from the user to the Application Provider considering the bandwidth, the delay, the jitter and the packet loss. The SA executes this process and it's related to “Resource Layer” of the QoE-layered model.
- “Est.QoE” process is related to Equation (4.2) and estimates the QoE level for the best path [43] from the user to the Application Provider considering the “QoS level”, the resolution of the video, the GoP and the video quality. The MA executes this process and it's related to “Application Layer” of the QoE-layered model.
- “Utilization” process measures the CPU and Memory utilization when the agents are used to monitor the network. It is executed in the background in our system, in order to observe if the utilization is decreased or increased.

In order to measure the values of bandwidth, delay, jitter and loss, we used network analyzing tools such as iperf [44] and qperf [40]. We also used “Big Buck Bunny” (BBB) [45] video sequences, considering four different resolutions such as 240p, 360p, 480p and 720p and the H.264 codec. Moreover, the frame rate was 24fps and the container was mp4.In Figure 12 we observe an example with two hosts (on the right) that are receiving the video BBB and one media server (on the left) that steams it to them.



```
ubuntu@sdnhubvm:~[04:05]$ cd SDNHub_Opendaylight_Tutorial/distribution/opendayli
ght-karaf/target/assembly/
ubuntu@sdnhubvm:~/SDNHub_Opendaylight_Tutorial/distribution/opendaylight-
arget/assembly[04:09] (master)$ ./bin/karaf
```

Figure 12 Example of the video streaming emulation with two hosts and one media server

4.1.3 Results

The QoS values are normalized with range from 0 to 5. The frequencies at each scenario is 1 minute, 3 minutes, 5 minutes and 7 minutes in a 30 seconds continuing observation. We have estimated the QoS, the QoE, the accuracy (using MSE), the CPU utilization and the memory utilization under different video resolutions (240p, 360p, 480p and 720p). First we measured the QoS and the QoE level every second for the whole duration of the video session in order to compare them with the tested frequencies. Figure 13 and Figure 14 show the QoS level and the QoE level respectively for both scenarios. In these figures we observe that “scenario 1” has higher variation during a session because there is a higher jitter and there are more delays. On the other hand “scenario 2” seems to be more stable. Thus, we conclude that “scenario 2” has a higher QoE level and it is more stable due to better network conditions.

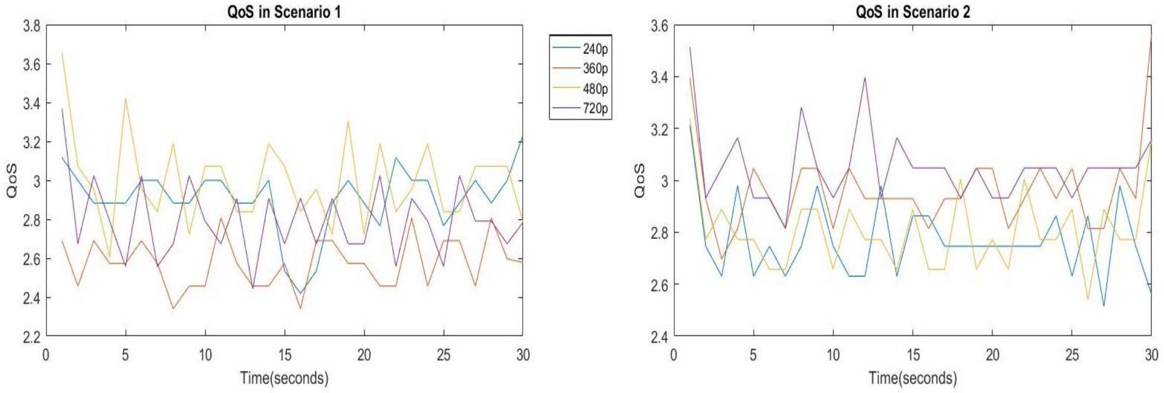


Figure 13 QoS value for each scenario during a video session [38]

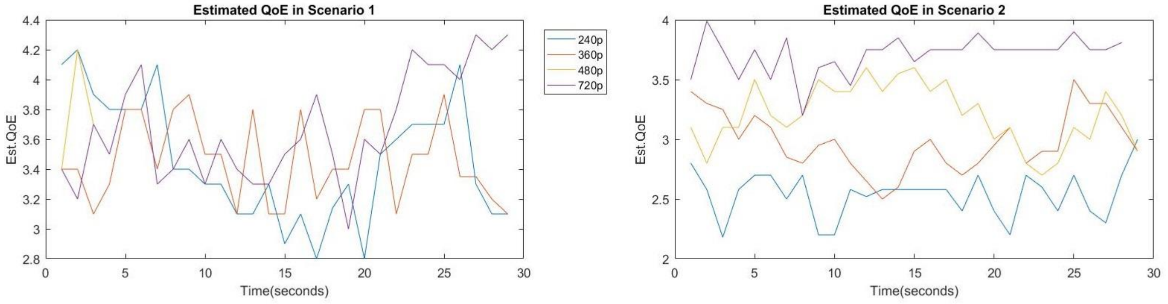


Figure 14 Estimated QoE for each scenario during a video session [38]

In Figure 14, we can observe the average values for each resolution during a session. Figure 15 and Figure 16 illustrate the average values of QoE and QoS respectively, throughout the session with different resolutions. The network load is lower when the resolution is 240p in “scenario 1” in Figure 16 than in the other cases because the network conditions are better.

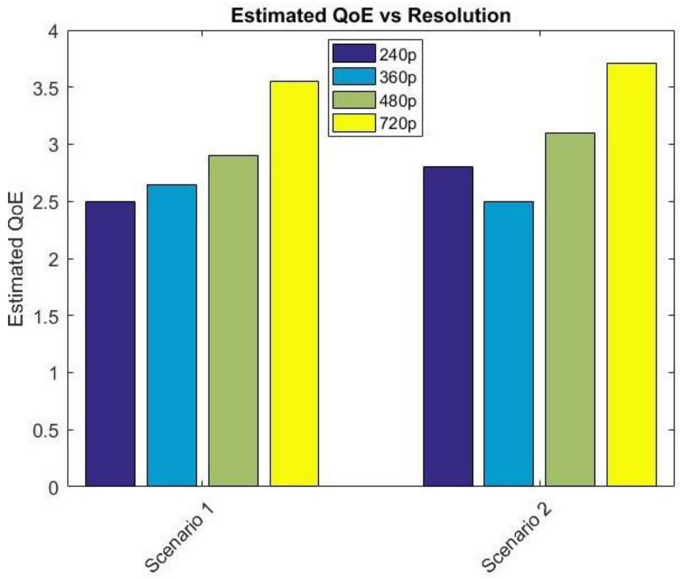


Figure 15 Estimated QoE value for each scenario [38]

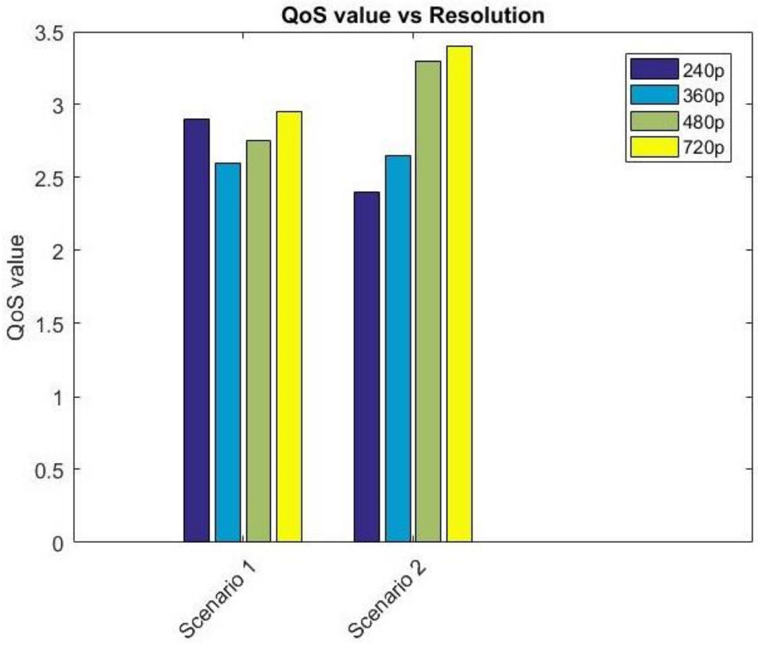


Figure 16 Estimated QoS value for each scenario [38]

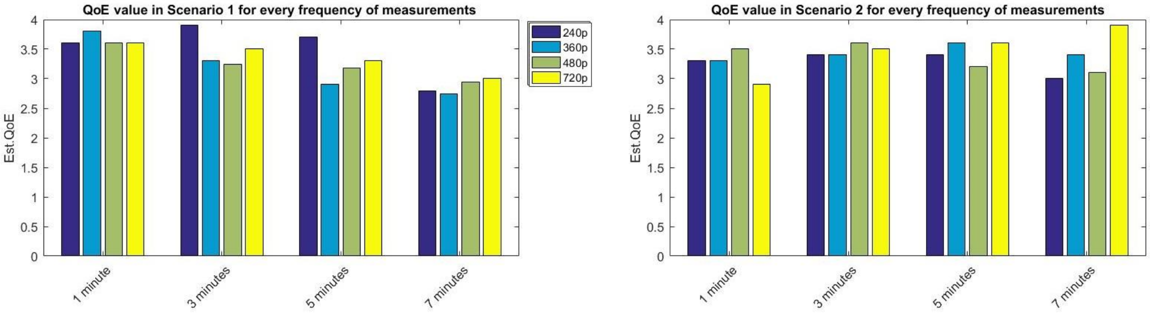


Figure 17 Estimated QoE value for every frequency of measurements [38]

Table 6 Accuracy estimation with MSE, CPU and Memory Utilization during a video session

Scenario - Resolution	MSE	CPU Utilization	Memory Utilization
1 – 240p	0.65	15,11%	95,49%
1 – 360p	3.61	12,48%	95,93%
1 – 480p	1.8	24,9%	96,59%
1 – 720p	1.31	26,76%	96,89%
2 – 240p	6.48	16,9%	94,26%
2 – 360p	6.74	11,96%	96,98%
2 – 480p	5.5	18,94%	96,86%
2 – 720p	6.3	23,76%	96,80%

The objective of these results was to show that the frequency of the measurements can affect the QoE and has a significant impact on its level. That objective is shown in Figure 17 for both scenarios along with the different time frequency measurements. We observed that in “Scenario 2”, with better network and different sampling rate, that there is a significant impact on QoE level.

The accuracy of the measurements is important and in that end we calculated the MSE between the estimated values and the true values. These calculations are shown in Table 5. The results from the table show that we achieved higher accuracy in “Scenario 2”. The memory utilization is on high levels because of the capabilities of the virtual machine that the experiments were conducted.

4.2 Performance evaluation of “A QoE monitoring solution for LTE-Advanced Pro networks”

The proposed solution is based on a QoE-monitoring algorithm that adjust the frequency of measurements while considering the computational complexity, the network load and the estimation accuracy. Later, the results are imported into a big data analytic platform called Elastic Stack [37]. This platform can help with the analysis and the visualization of the data.

4.2.1 QoE models

In order to evaluate a video service, we used the eMOS model [42] because it considers both network (e.g. delay, jitter, loss, and bandwidth) and application parameters (e.g. resolution, frame rate and video quality). The video quality is chosen because it is a main influence parameter of the video's QoE. Based on the analysis of video sessions, we are able to identify the quality perceived by the user and mapped it to QoE using eMOS, which is a QoE psychometric model [42]. The range of the QoE values is from 0 to 1.

In order to evaluate a Web browsing service, we used a web-based QoE parametric model that considers the page load time and its formula is.

$$U(d) = 5 - \frac{578}{1+\left(11.77+\frac{22.61}{d}\right)^2} \tag{4.2}$$

Where "d" represents the service response time measured in seconds. The range of the QoE values is from 1 to 5.

Due to the flexibility of the QoE-Agents, we can be replace the QoE estimation models depending on the application and the use case.

4.2.2 Performance Evaluation

The experiments were carried in an Asus computer that runs with Intel Core i7 @3.6 GHz, 4GB RAM and is installed with Linux Ubuntu 14.04 64bit with Mininet. Also there is an Apache HTTP server and a MySQL RDBMS installed to store the data locally. The emulated network consists of 4 hosts and 10 devices that are considered as the logical entities of an LTE-Advanced Pro network. The logical entities are: 3 eNBs, 1 SGW, 1 PGW, 1 HSS, 1 MME, 1 TDF, 1 SPR and 1 AF. We consider wireless links from the users to the eNBs and the rest of the links are wired. The MA is deployed in the PGW and the two SAs to the SGW and PCRF, respectively. The experimental setup is shown on Figure 18 and it was repeated for 5 times.

For this performance evaluation, we have created two scenarios using the AGENT-MON platform. In the first scenario the proposed QoE monitoring algorithm is not used, whereas in the second scenario it is used, in order to compare the two scenarios. The use cases are a video streaming service, where a media server is streaming a video and the users receive it, and a Web browsing service, where the QoE model considers the page load time. The video that was used for the video streaming service was “Elephants Dream” [45] with 480p resolution, H.264 codec, 24fps frame rate and mp4 as container. Our target was to analyze the variation in the accuracy of the estimated QoE for these two services by adjusting only the frequency of monitoring. In the emulated network the average end-to-end transmission delays were set to 3ms and the available bandwidth at 5 Mbps. In each scenario the network administrator starts the monitoring process and users request for a service (Web browsing or video streaming in our case). AGENT-MON’s target is to consume less resources in terms of CPU and memory, keep the QoE at an acceptable level by changing the monitoring frequency and avoid network overload.

The MA agent is responsible for the QoE estimation after the SA sends all the necessary data to MA and then it stores it to the MySQL RDBMS. The database exports the data and then we transform them into JSON format in order to import them to ElasticSearch. In ElasticSearch, we can create all the kinds of graphs from the information gathered from the network and organized them in an easy and flexible way.

The frequencies that were used for the tests were 1 minute, 2 minutes, 3 minutes, 4 minutes, 5 minutes, 8 minutes and 10 minutes. In the first scenario, which is without the monitoring algorithm, the frequency was stable e.g. every 4 minutes (until the experiment ends without the ability to change it. However, in the second scenario there was a dynamic adjustment of the frequency. We must note that the only thing that changes between the two scenarios is the ability to change the frequency (dynamic or static). The input parameters of QoE estimations are the same. The network measurements conducted using network analyzing tools such as iperf [44].

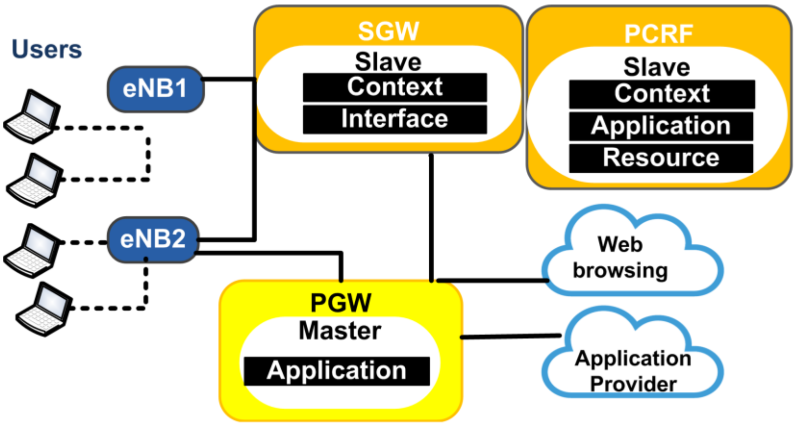


Figure 18 Experimental setup of A QoE monitoring solution for LTE-Advanced Pro networks

4.2.3 Results

After the experiments we came up with some results from the QoE estimations of the video service and of the Web browsing services. These results are shown in Figure 19 and Figure 20 respectively for a variety of measurement frequencies. From those figures we observed that the usage of the monitoring algorithm increases the QoE estimation for both services. The monitoring frequency was increased and decreased according the network load and the computational complexity in order to avoid the network overload. The thresholds that were used to adjust the frequency were set to 3 Mbps for network load, 5% in terms of CPU usage and 96% in terms of memory usage. Above these thresholds the QoE estimations, based on our experiments, were in an acceptable level and that's the reason why we chose them. The CPU usage is running low because it is related to the QoE-Agents which they run as processes to the network devices.

The measurements of the throughput, the computational resources (memory utilization, CPU) and the loss with and without the algorithm for both services are shown in Figure 21 and Figure 22 respectively for all the monitoring frequencies. The main target was to increase the throughput and decrease the computational resources and the loss. Both figures show the throughput measured in Mbps, the computational resources in terms of CPU as well as memory utilization, and the loss is measured in percentage. In Figure 21 and Figure 22 we observe that we were successful.

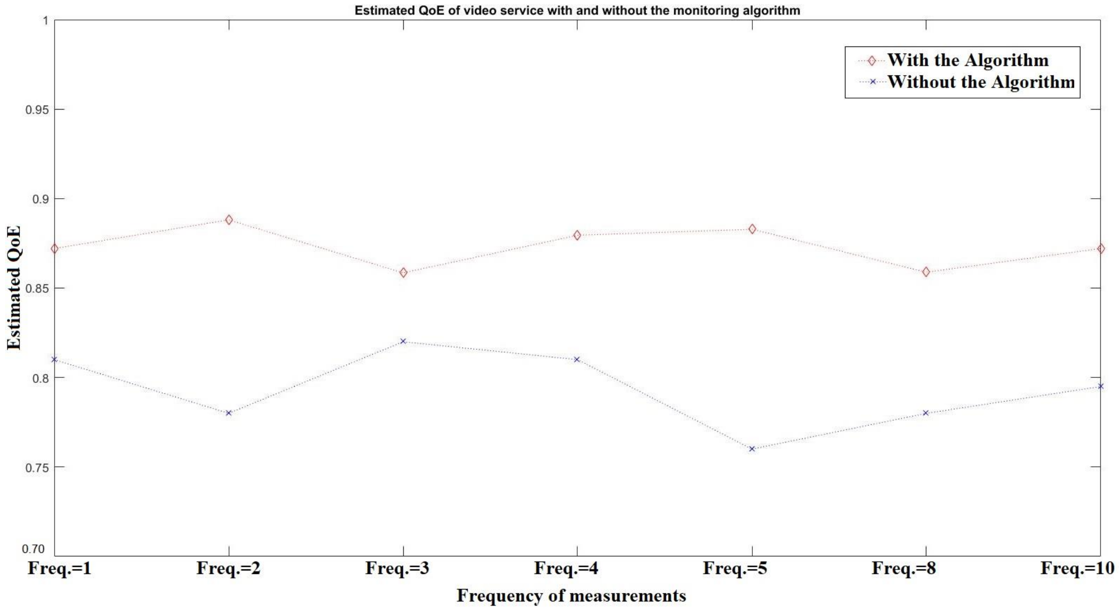


Figure 19 Estimated QoE for video service with and without the monitoring algorithm

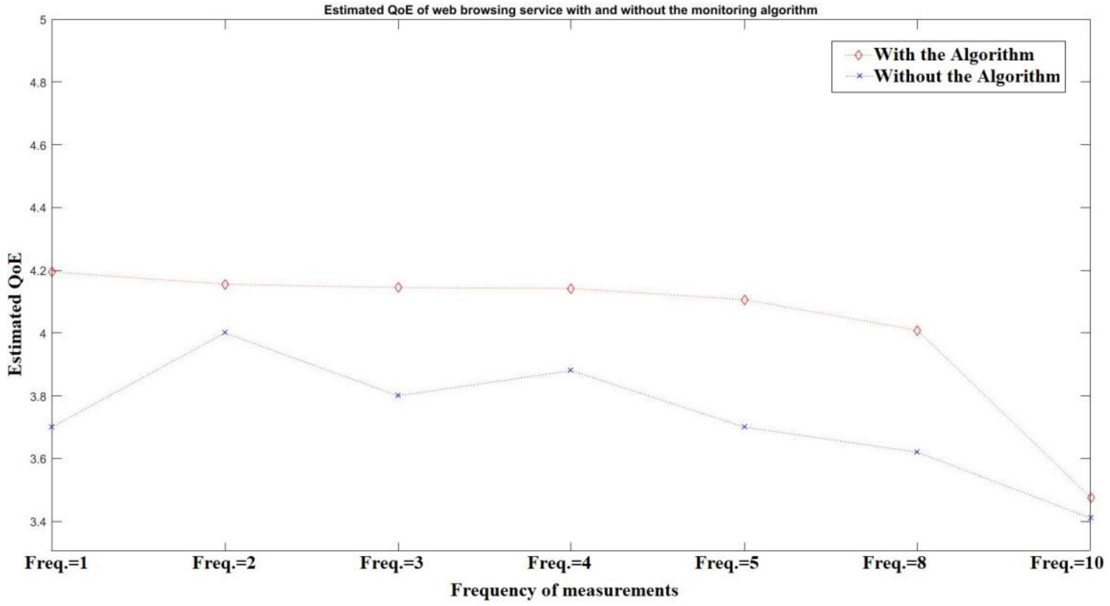


Figure 20 Estimated QoE for web browsing service with and without the monitoring algorithm

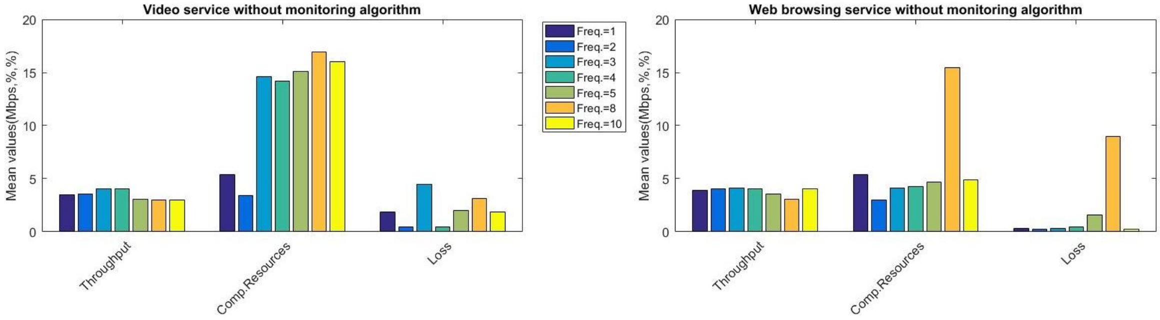


Figure 21 Video and Web browsing services results without the monitoring algorithm

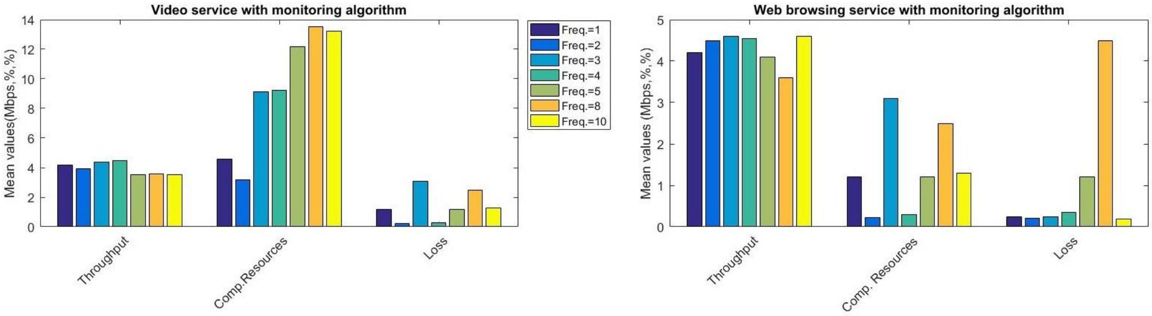


Figure 22 Video and Web browsing services results with the monitoring algorithm

Table 7 Accuracy of the measurements using MSE

Frequency	Video (Without)	Web Browsing (without)	Video(with)	Web browsing(with)
1	7.683	6.700	6.2	6.100
2	0.143	33.13	0.123	29.26
3	7.503	4.937	6.012	4.212
4	8.014	4.407	6.345	2.369
5	0.130	3.672	0.110	3.015
8	0.103	8.276	0.096	6.875
10	8.599	2.594	6.514	2.111

MSE was used to observe how reliable our platform is and how accurate the AGENT-MON measurements are. MSE was used for both services, which are video service and web browsing service, and is the deviation of our estimated value from the true one. The MSE is equal to the variance plus the squared bias. It calculates the error between the QoE estimations with and without the usage of the proposed algorithm. The results of the accuracy are shown in Table 7.

In all cases, we observed that the error with the usage of the monitoring algorithm is lower than these without the algorithm.

As big data platform, we used the ELK stack [37] that allows network managers to extract results and create graphs such as this in Figure 23 after the data analysis. Moreover, the network manager can observe the usage of services in a specific monitoring network area as shown in Figure 24.

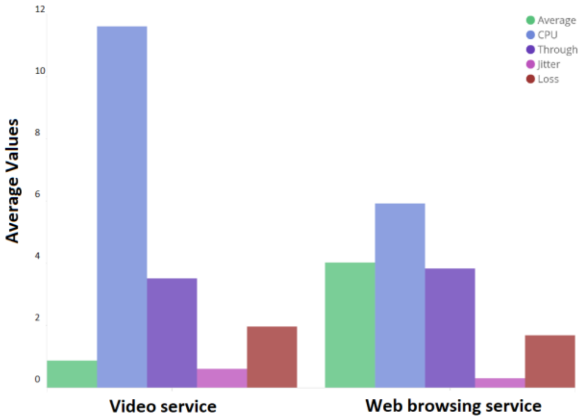


Figure 23 Video and Web browsing services results with the algorithm using the ELK stack

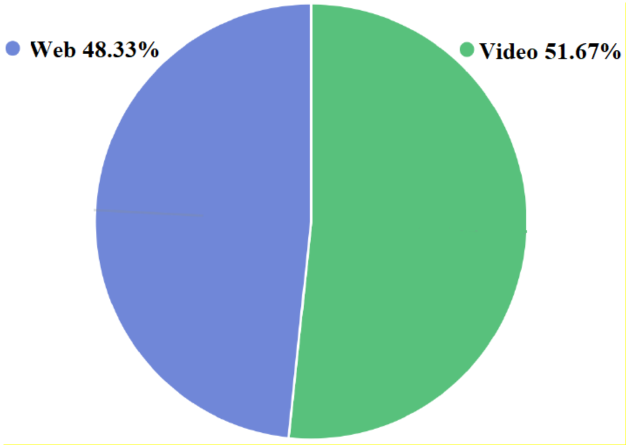


Figure 24 Identified services during a monitoring session

CHAPTER 5 – CONCLUSIONS

5.1 Conclusions

QoE monitoring of video services over wireless networks is essential for performance optimization. This thesis presented two QoE monitoring strategies based on the QoE layered model and the QoE-Agents (Master-Agent and Slave-Agent). Both approaches based on the results showed that with the use of the Agents we have low CPU utilization (that means that the agents don't overload the network) and acceptable memory utilization. Especially the monitoring algorithm of the second approach that did dynamic adjustment of the monitoring frequencies has a significant impact of the QoE estimations. Also in that approach, the analysis of all the measurements and the creation of graphs were made using big data analytic tools that can be found in the big data analytic platform called ELK stack [37].With the usage of this platform the network provider can categorize the data (by service, country, etc.), visualize the data by creating graphs and charts and thus can see an abstract view of the network's performance.

5.2 Future work

An extension of the monitoring algorithm using machine learning techniques is been discussed for future plans. The objective is to use these techniques to train a neural network to build a QoE evaluation model and improve the monitoring algorithm. The monitoring algorithm, with the usage of the machine learning techniques, will make better adjustments of the monitoring frequencies. Usually there is a gap between the application/service that the operator is offering with the user's perception of that service. We are going to use neural networks so we can reduce that gap.

Another future work plan is to use Semantic Web technologies, such as Ontologies and Reasoning, to extend the QoE-Agents so they can consider different parameters and metrics. Semantic Web Services can solve issues about heterogeneity and extensibility, which will be a challenge for the new mobile technologies that are coming in cellular communication.

REFERENCES

[1] T. Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," Cisco Public Information, 2017.

[2] "Huawei Leads with Completion of China's Second-Phase 5G R&D Test," Huawei, 09 June 2017. [Online]. Available: http://www.huawei.com/en/press-events/news/2017/6/Second-Phase-5G-R-D-Test.

[3] "5G: Networks of the future," Ericsson, 2018. [Online]. Available: https://www.ericsson.com/en/networks/trending/hot-topics/5g-networks.

[4] "Top companies leading 5G development," 09 November 2017. [Online]. Available: https://www.netscribes.com/top-companies-leading-5g-development/.

[5] G. Fettweis and S. Alamouti, "5G: Personal mobile internet beyond what cellular did to telephony," *IEEE Communications Magazine,* vol. 52, no. 2, pp. 140 - 145, February 2014.

[6] L. I. Millett, B. Friedman and E. Felten, "Cookies and Web browser design: toward realizing informed consent online," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, 2001.

[7] M. Burton and J. Walther, "The value of web log data in use-based design and testing," in *Journal of Computer-Mediated Communication*, 2001.

[8] "ETSI. (2014) Etsi ts 103 294 speech and multimedia transmission quality (stq), quality of experience; a monitoring architecture - version 1.1.1.," [Online]. Available: http://www.etsi.org/deliver/etsi_ts/103200_103299/103294/01.01.01_60/ts_103294v010101p.pdf.

[9] ITU, ITU Telecommunication Standardization Sector and OF, "Definition of quality of experience (QoE), liaison statement," ITU-T Recommendation P.10/G.100, Amd 1, 2007.

[10] European Telecommunication Standards Institute TC HF (Human Factors), "Quality of Experience (QoE) requirements for real-time communication services," European Telecommunication Standards Institute, 2010.

[11] P. L. Callet, S. Moller and A. Perkis, "Qualinet white paper on definitions of quality of experience (QoE)," 2013.

[12] K. U. R. Laghari and K. Connelly, "Toward Total Quality of Experience: A QoE Model in a Communication Ecosystem," *Communications Magazine, IEEE,* vol. 4, no. 50, pp. 58-65, 2012.

[13] S. Jumisko-Pyykkö, V. K. Malamal Vadakital and M. M. Hannuksela, "Acceptance threshold: A bidimensional research method for user-oriented quality evaluation studies," in *International Journal of Digital Multimedia Broadcasting*, 2008.

[14] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J. Hong and A. K. Dey, "A factors influencing quality of experience of commonly-used mobile applications," *IEEE Communications Magazine,* pp. 48-56, 2012.

[15] E. Liotou, D. Tsolkas and N. Passas, "A roadmap on QoE metrics and models," in *23rd International Conference on Telecommunications (ICT)*, Thessaloniki, 2016.

[16] I. Rec., "P. 800.1, mean opinion score (mos) terminology," in *International Telecommunication Union, Geneva*, Geneva, 2006.

[17] I. Rec., "Bt. 500-11, methodology for the subjective assessment of the quality of television pictures," in *International Telecommunication Union*, 2002.

[18] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold and P. Tran-Gia, "Best Practices for QoE Crowdtesting: QoE Assessment With Crowdsourcing," *IEEE Transactions on Multimedia,* pp. 541 - 558, February 2014.

[19] G. Rubino, "The PSQA project," INRIA Rennes, 2010. [Online]. Available: http://www.irisa.fr/armor/lesmembres/Rubino/myPages/psqa.html,.

[20] *Recommendation ITU-T G.1011(2015), Reference guide to quality of experience assessment methodologies.*

[21] *Recommendation ITU-T P.862 (2001), Perceptual evaluation of speech quality (PESQ).*

[22] *Recommendation ITU-T G.107 (2014), The E-model: a computational model for use in transmission planning.*

[23] M. Fiedler, T. Hossfeld and P. Tran-Gia, "A generic quantitative relationship between Quality of Experience and Quality of Service," *IEEE Network,* vol. 24, no. 2, pp. 36-41, Mar. 2010.

[24] T. Hoßfeld, R. Schatz, E. Biersack and L. Plissonneau, "Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience," in *Data Traffic Monitoring and Analysis*, E. Biersack, C. Callegari, and M. Matijasevic, Eds, Springer Berlin Heidelberg, 2013, pp. 264-301.

[25] T. Hoßfeld, M. Seufert, C. Sieber and T. Zinner, "Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming," in *IEEE QoEMEX*, Singapore, 2014.

[26] *Recommendation ITU-T P.564 (2007), Conformance testing for voice over IP transmission quality assessment models.*

[27] L. Skorin-Kapov and M. Varela, "A Multi-Dimensional View of QoE: the ARCU Model," in *proc. of the 35th International Convention MIPRO*, 2012.

[28] I. Wechsung, M. Schulz, K.-P. Engelbrecht, J. Niemann and S. Möller, "All Users Are (Not) Equal - The Influence of User Characteristics on Perceived Quality, Modality Choice and Performance," in *Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop*, Springer, 2011.

[29] F. Guyard, Ed., QoE Concept, Celtic QuEEN deliverable, D2.100, Celtic Project Call 8, April, 2012.

[30] 3GPP, "Release 8," 2008. [Online]. Available: http://www.3gpp.org/specifications/releases/72-release-8.

[31] ITU-R, "Requirements related to technical performance for IMT-Advanced radio interface(s)," 2008. [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2134-2008-PDF-E.pdf.

[32] "Home," 3rd Generation Partnership Project, [Online]. Available: http://www.3gpp.org/release-13.

[33] G. Gómez, J. Lorca, R. García and Q. Pérez, "Towards a QoE-Driven Resource Control in LTE and LTE-A Networks," *Journal of Computer Networks and Communications,* vol. 2013, p. 15, 2013.

[34] J. Spiess, Y. T'Joens, R. Dragnea, P. Spencer and L. Philippart, "Using Big Data to Improve Customer Experience and Business Performance," *Bell Labs Technical Journal,* vol. 4, no. 18, pp. 3-17, 2014.

[35] ITU, "Measuring the information society report," International telecommunication union, Geneva, 2014.

[36] X. He, K. Wang, H. Huang and B. Liu, "QoE-Driven Big Data Architecture for Smart City," *IEEE Communications Magazine,* vol. 56, no. 2, pp. 88 - 93, 2018.

[37] Elasticsearch-kibana-logstash. [Online]. Available: https://www.elastic.co/elk-stack.

[38] E. Grigoriou, T. Saoulidis, L. Atzori, V. Pilloni and P. Chatzimisios, "An agent-based QoE monitoring strategy for LTE networks," in *Communication Software, Services, and Multimedia Applications Symposium*, Kansas city, 2018.

[39] A. Ahmad, A. Floris and L. Atzori, "OTT-ISP joint service management: A Customer Lifetime Value based approach," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, 2017.

[40] "qperf - measure rdma and ip performance," [Online]. Available: https://linux.die.net/man/1/qperf.

[41] "Mininet," [Online]. Available: http://mininet.org.

[42] H. J. Kim, D. G. Yun and K. Hwa-Suk, "QoE assessment model for video streaming service using QoS parameters in wired-wireless network," in *2012 14th International Conference on Advanced Communication Technology (ICACT)*, PyeongChang, 2012.

[43] E. Grigoriou, A. A. Barakabitze, L. Atzori, L. Sun and V. Pilloni, "An sdn-approach for qoe management of multimedia services using resource allocation," in *IEEE International Conference on Communications 2017*, 2017.

[44] "Iperf: Tdp/Udp bandwidth measurement tool," [Online]. Available: http://iperf.fr.

[45] "Video test media," [Online]. Available: https://media.xiph.org/video/derf/.

[46] "All-in-one SDN App Development Starter VM | SDN Hub," [Online]. Available: http://sdnhub.org/tutorials/sdn-tutorial-vm/.

[47] "Download VMware Workstation Player," VMware, [Online]. Available: https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/14_0.

[48] "Elastic Cloud Signup | Elastic," [Online]. Available: https://www.elastic.co/cloud/elasticsearch-service/signup.

APPENDICES

1) SETUP OF MININET

Mininet network emulation environment [41] was used to run the experiments and it was inside of a virtual machine called “All-in-one SDN App Development Starter VM” [46]. VMware Workstation 14.1.2 Player was used as a player for the virtual machine. VMware Workstation 14.1.2 Player is free and is available for download at [47]. The steps that we followed to setup the VM and run the experiments are:

1. Opened the VMware Workstation from the Windows 10 desktop
2. Load “All-in-one SDN App Development Starter VM” file by going to File ⇒ Open and after the loading is complete we can power on the VM
3. Power on the VM
4. Log in to the VM by using “ubuntu” as username and “ubuntu” as password
5. Open a Terminal emulator and run the following commands with this order to setup and run the controller:
    a. “cd SDNHub_Opendaylight_Tutorial” and “git pull –rebase”
    b. “mvn install –nsu”
    c. “cd distribution/opendaylight-karaf/target/assembly” and “./bin/karaf” to run the controller
6. Open a new Terminal emulator to run Mininet with our custom topology. That happens with the commands:
    a. “cd SDNHub_Opendaylight_Tutorial” because in that folder we have saved the file with our custom topology
    b. “sudo mn --custom ./scenariomaseLTE.py --topo scenariomaseLTE --mac --switch ovsk,protocols=OpenFlow13 --controller remote --link tc”
7. Go to the Terminal emulator where the controller runs (step 5) and run the following command so the controller can add intelligence in the switches. That intelligence helps them to learn the MAC addresses of each host and forward traffic to the correct switch ports. That command is:
    a. “feature:install sdnhub-tutorial-learning-switch”
8. Finally go back to the Terminal emulator where Mininet is running (step 6) and add default rules in the switches so they can send packet-in messages to the controller. That rules are added with the following commands:
    a. “s1 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”
    b. “s2 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”
    c. “s3 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”
    d. “s4 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”

   e. “s5 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”
   f. “s7 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller” “s1 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”
   g. “s8 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”
   h. “s9 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13 priority=1,action=output:controller”

- The scenariomaseLTE.py code is:

```
#!/usr/bin/python
from functools import partial
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.net import Mininet
from mininet.node import OVSKernelSwitch
from mininet.node import RemoteController
from mininet.topo import Topo
from mininet.util import dumpNodeConnections

class scenariomaseLTE( Topo ):
def __init__(self):
Topo.__init__(self)
u1 = self.addHost('u1') #UE1
u2 = self.addHost('u2') #UE2
u3 = self.addHost('u3') #UE3
u4 = self.addHost('u4') #UE4
u5 = self.addHost('u5') #UE5
u6 = self.addHost('u6') #UE6

s1 = self.addSwitch('s1') #enb1
s2 = self.addSwitch('s2') #enb2
s3 = self.addSwitch('s3') #enb3

s4 = self.addSwitch('s4') #SGW
s5 = self.addSwitch('s5') #PGW
s6 = self.addSwitch('s6') #PCRF
s7 = self.addSwitch('s7') #AF
s8 = self.addSwitch('s8') #TDF
s9 = self.addSwitch('s9') #SPR

self.addLink( u1, s1, bw=5, delay='3ms' ) # UE1 <-> enb1
self.addLink( u2, s1, bw=5, delay='3ms' ) # UE2 <-> enb1

self.addLink( s1, s2, bw=5, delay='3ms' ) # enb1 <-> enb2

self.addLink( u3, s2, bw=5, delay='3ms' ) # UE3 <-> enb2
self.addLink( s2, s3, bw=5, delay='3ms' ) # enb2 <-> enb3
```

```
self.addLink( u4, s3, bw=5, delay='3ms' ) # UE4 <-> enb3
self.addLink( u5, s3, bw=5, delay='3ms' ) # UE5 <-> enb3
self.addLink( u6, s5, bw=5, delay='3ms' ) # UE6 <-> PGW

self.addLink( s1, s4, bw=5, delay='3ms' ) # enb1 <-> SGW
self.addLink( s2, s4, bw=5, delay='3ms' ) # enb2 <-> SGW
self.addLink( s3, s4, bw=5, delay='3ms' ) # enb3 <-> SGW


self.addLink( s4, s5, bw=5, delay='0ms' ) # SGW <-> PGW
self.addLink( s5, s6, bw=5, delay='0ms' ) # PGW <-> PCRF
self.addLink( s6, s7, bw=5, delay='0ms' ) # AF <-> PCRF
self.addLink( s6, s8, bw=5, delay='0ms' ) # TDF <-> PCRF
self.addLink( s6, s9, bw=5, delay='0ms' ) # SPR <-> PCRF

topos = { 'scenariomaseLTE': ( lambda: scenariomaseLTE() ) }
```

This file was used for the performance evaluation of “A QoE monitoring solution for LTE-Advanced Pro networks”. The difference with the file that was used for the Performance evaluation of “An agent-based QoE monitoring strategy for LTE networks” is that the delay was set to 4ms for both scenarios and that the available bandwidth was fluctuating between 5 Mbps and 10 Mbps, respectively in the two scenarios.

- The calculation of the QoS in both approaches was done by the same file called “calc_qos.py” and its code is:

```
#!/usr/bin/env python
import math
import cmath
from datetime import datetime
import sys
from cStringIO import StringIO
net_parameters=([0.01, 0.04, 0.02, 0.2],[0.01, 0.04, 0.02, 0.10],[0.01, 0.04, 0.02, 0.15]) #net parameteres for 3 scenarios- adjust only the bandwidth
weights=[0.589, 0.149, 0.151, 0.111]
net_size=len(net_parameters)
x=0
y=0
qos_calc=[]
qos_table=[]
text2=[]
for x in range(net_size):
qos_calc=net_parameters[x][y]*weights[y]+net_parameters[x][y+1]*weights[y+1]+net_parameters[x][y+2]*weights[y+2]+net_parameters[x][y+3]*weights[y+3]
qos_table.append(qos_calc)
x = x + 1
text= qos_calc
text2.append(text)
```

```
timestamp=datetime.now().strftime('%Y-%m-%d %H:%M:%S')
filename= ("qos_" + timestamp)
f = open(filename,'w')
f.write(str(text2))
f.close()
```

- The calculation of the QoE in both approaches was done by the same file called “calc_qoe.py” and its code is:

```
#!/usr/bin/env python
import math
import cmath
from datetime import datetime
import sys
from cStringIO import StringIO

emos=[]
qos = [0.0370, 0.03152, 0.02597]
qr=0.9
resolution = [240, 360, 480, 720]
gop=24
qos_length=len(qos)
resolution_length=len(resolution)
i=0
j=0

net_parameters=([0.01, 0.04, 0.02, 0.2],[0.01, 0.04, 0.02, 0.15],[0.01, 0.04, 0.02, 0.1])
weights=[0.589, 0.149, 0.151, 0.111]

net_size=len(net_parameters)
text2=[]

for i in range(len(qos)):
for j in range(len(resolution)):
qosresolution= qos[i] * resolution[j]
test1 = 1-qos[i]
test2 = (qos[i] * resolution[j])/gop
power = format((math.pow(test1,test2)), '.2f')
qr = float(qr)
power = float(power)
line='\n'
mos = qr * power
text= "Resolution=%s  eMOS=%s  " % (str(resolution[j]),str(mos))
text2.append(text)
j = j + 1
i = i + 1

timestamp=datetime.now().strftime('%Y-%m-%d %H:%M:%S')
filename= ("qoe_" + timestamp)
f = open(filename,'w')
f.write(str(text2))
f.close()
```

The “Utilization” that process measures the CPU and Memory utilization when the agents are used to monitor the network was done by the file called “calc_util.sh” and its code is:

```
#!/usr/bin/env bash

CPU_USAGE=$(top -b -n2 -p 1 | fgrep "Cpu(s)" | tail -1 | awk -F'id,' -v prefix="$prefix" '{
split($1, vs, ","); v=vs[length(vs)]; sub("%", "", v); printf "%s%.1f%%\n", prefix, 100 - v }')

DATE=$(date "+%Y-%m-%d %H:%M:")

MEMORY_USAGE=$(free -m | awk 'NR==2{printf "%.2f%%", $3*100/$2 }')
USAGE="$DATE CPU: $CPU_USAGE MEMORY: $MEMORY_USAGE"

now=$(date +"%Y.%m.%d-%H.%M.%S")

echo $USAGE >> /home/ubuntu/Documents/"util_$now"
```

2) ElasticSearch, Logstash and Kibana (ELK) Stack

As a big data analytic platform we used the Elastic Stack [37]. In particular, the 14-day free trial of the Elasticsearch at [48]. First using the Kibana interface by going at Dev Tools>Console we set up a mapping for our data set and if the logs contain only an id and the service (Figure 24) the mapping would be:

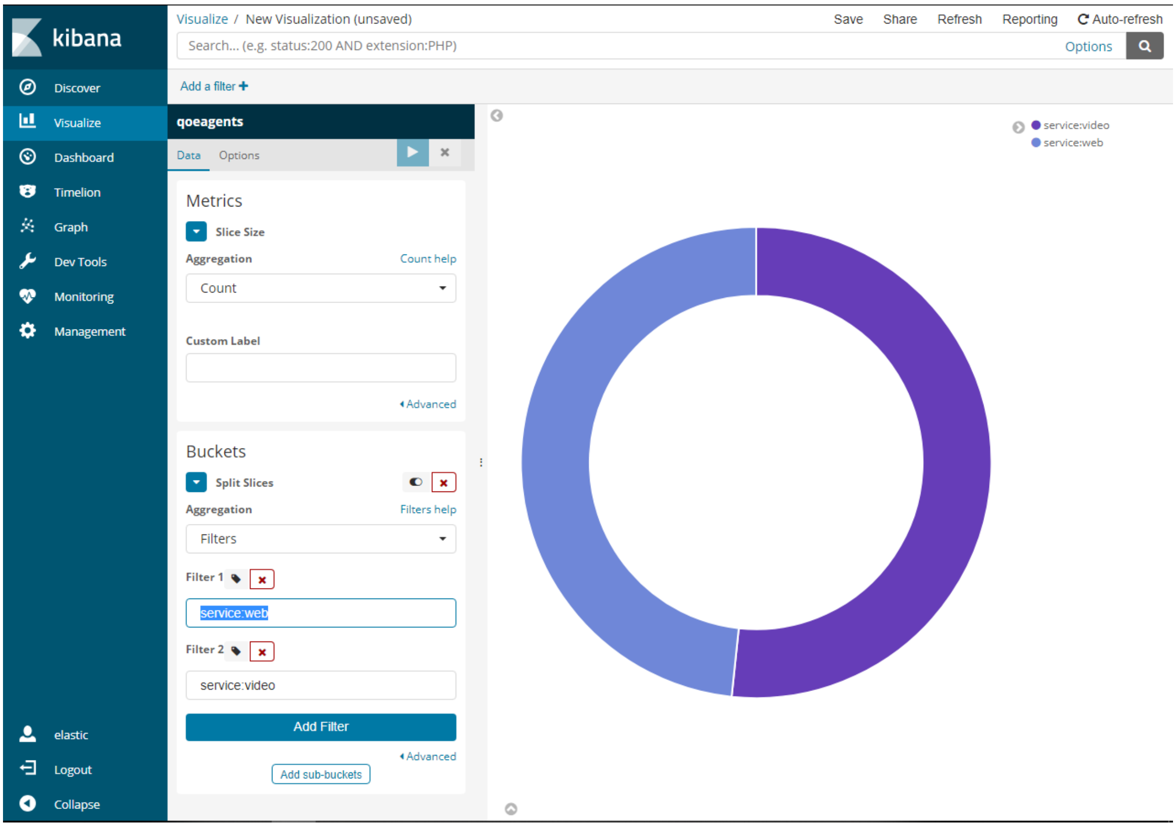
```
PUT /qoeagents
{
"mappings": {
"log": {
"properties": {
"id": {"type": "integer"},
"service": {"type": "keyword"}
}
}
}
}
```

Later we upload the json files (one for video service and one for web service) using the Elasticsearch bulk API with the following command from a Terminal Emulator:

-
  ```
  curl -H 'Content-Type: application/x-ndjson' -s -X POST -u elastic:UlJJZYqmGxH2CwrGGhm5XqVP --data-binary @/home/ubuntu/json/web.json 'https://31e5f57c8a0c436c9933b223e78afb0d.europe-west1.gcp.cloud.es.io:9243/_bulk'
  ```

-
  ```
  curl -H 'Content-Type: application/x-ndjson' -s -X POST -u elastic:UlJJZYqmGxH2CwrGGhm5XqVP --data-binary @/home/ubuntu/json/video.json 'https://31e5f57c8a0c436c9933b223e78afb0d.europe-west1.gcp.cloud.es.io:9243/_bulk'
  ```

After the upload of these files we go at Discover and because it’s our first data Kibana warns us that there is no default index pattern and prompts us to create an index pattern. So we create an index pattern called “qoeagents”. Later we go at Visualize and we create a pie chart from the “Basic Charts” category and choose the index “qoeagent”. Our metric will be the Count and we split the pie slices with filters because we want the count of the web services and the video services. Those filters are “service:web” and “service:video”. Figure 25 illustrates the final stage of the creation of the pie.



Aggregation: Count

Aggregation: Filters

Filter 1: service:web

Filter 2: service:video

Figure 25 Creating a pie chart counting the services in use.