DEGREE PROJECT

Quality of Experience Provisioning in Mobile Cloud Computing

Jonathan Glenn Berrios Moron

Computer Science and Engineering, master's level (120 credits)
2018

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
PERCCOM Master Program

Master's Thesis in
Pervasive Computing & COMmunications
for sustainable development

Jonathan Glenn Berrios Moron

QUALITY OF EXPERIENCE PROVISIONING IN MOBILE CLOUD COMPUTING

2018

Supervisors: *Professor Jari Porras* (Lappeenranta University of Technology)
Assistant Professor Karan Mitra (Luleå University of Technology)
Assistant Professor Saguna Saguna (Luleå University of Technology)

Examiners: *Professor Eric Rondeau* (University of Lorraine)
Professor Jari Porras (Lappeenranta University of Technology)
Associate Professor Karl Andersson (Luleå University of Technology)

This thesis is prepared as part of an European Erasmus Mundus programme PERCCOM - Pervasive Computing & COMmunications for sustainable development.

Co-funded by the
Erasmus+ Programme
of the European Union

This thesis has been accepted by partner institutions of the consortium (cf. UDL-DAJ, n°1524, 2012 PERCCOM agreement).

Successful defense of this thesis is obligatory for graduation with the following national diplomas:

- Master in Complex Systems Engineering (University of Lorraine)
- Master of Science in Technology (Lappeenranta University of Technology)
- Degree of Master of Science (120 credits) –Major: Computer Science and Engineering, Specialisation: Pervasive Computing and Communications for Sustainable Development (Luleå University of Technology)

ABSTRACT

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
PERCCOM Master Program

Jonathan Glenn Berrios Moron

Quality of Experience Provisioning in Mobile Cloud Computing

Master's Thesis

2018

67 pages, 35 figures, 17 tables.

Examiners: *Professor Eric Rondeau* (University of Lorraine)
Professor Jari Porras (Lappeenranta University of Technology)
Associate Professor Karl Andersson (Luleå University of Technology)

Keywords: Quality of Experience; Mobile Cloud Computing; Cloud Computing; Bayesian Networks

It is common in these times to find people relying on the functionalities of their mobile devices while walking. In a higher perspective, moving around the city, or among cities. It is also common to find the same people utilizing remote services via their mobile devices. Among different options, it could be presumed that they are navigating on the internet, or making use of a streaming service. There is a wide spectrum of options for applications. While this can be thought of as a common scenario, there is a lot of reasoning and considerations in systems which enable this to happen. In this context, this thesis focuses on the Quality of Experience (QoE) measurement and provisioning in Mobile Cloud Computing (MCC) environment as many of the applications used in these days are hosted remotely in a *cloud*. As mentioned, the aim is to *assess* and *provision* the *best* possible Quality of Experience as the end user utilizes their application on the move. This thesis makes use of Bayesian Networks (BN) principle to estimate the QoE, while it is considered a system called M2C2 for mobility management and cloud selection.

ACKNOWLEDGEMENT

I dedicate this thesis to my family. My mother Dunie, my sister Katia, and although not with us, my father Oscar, you are the biggest motivation in my life. The love and support from you are always present for me despite the distance. From here I would like to thank you for the love and unconditional support. I would like to state that I always keep in mind the good and difficult moments for us. I remember those especially since I have learned through the last years, that it is from the difficult moments that we learn the most. Thanks to my mother for showing my the greatest values in life and for taking care of me when I was a little child, and supporting my decision until this time. This thesis is part of the decisions I took and I hope it counts for the good moments of the family. Every one of you is important in my life and I learn from you every day. The effort put in the present thesis goes out to you.

I want to express my gratitude to my supervisors. Without them, this work could not be possible. I learned the dedication and the meaning of doing research. Thank you for the guidance and disposition to share your knowledge with me and for marking the way to go in the development of this thesis. Professor Jari Porras, who supported my decision to go for a topic I found more interesting and challenging. Professor Karan Mitra, who constantly followed the technical aspects of this work and helped me to improve its robustness. Professor Saguna Saguna, who showed me the importance of the way of presenting this work. I appreciate the introduction when we started to work together and also when helping me to be prepared for the tough winter in Skellefteå.

Thanks to the PERCCOM Consortium [60] for the opportunity and letting me be part of this family. Special thanks to program coordinators in each host university: Professors Eric Rondeau, Jari Porras, and Karl Andersson, who not only shared with us their knowledge within the academic field but also made a great effort to make us feel at home. Many thanks also to professors Jean-Philippe Georges, Francis Lapage, Ah-Lian Kor, Alexandra Klimova and Colin Pattinson who took an active role in the seminars and Summer Schools.

Finally, thanks to all the people along this journey. The professors in each course and all of the friends I made.

Skellefteå, June 28, 2018

Jonathan Glenn Berrios Moron

CONTENTS

1 Introduction **10**
1.1 Background . . . 10
1.2 Motivation . . . 12
1.3 Thesis Aim . . . 14
1.4 Research Question and Methods . . . 14
1.5 Thesis Contribution . . . 15
1.6 Delimitations . . . 15
1.7 Sustainability Aspects . . . 16
1.8 Thesis Outline . . . 16

2 Background and Literature Review **18**
2.1 Mobility Management in Heterogeneous Access Networks . . . 18
2.1.1 Mobile IPv6 . . . 20
2.1.2 Multi-homed Mobile IP . . . 22
2.2 Quality of Experience in Mobile Cloud Computing . . . 23
2.2.1 Network Probing . . . 25
2.2.2 Cloud probing . . . 27
2.3 Context-Aware Quality of Experience Provisioning in Mobile Cloud Computing environment . . . 28
2.3.1 Perceptual Evaluation of Video Quality - PEVQ . . . 33
2.3.2 Bayesian Network for QoE measurement . . . 35
2.4 Summary . . . 36

3 The QoE Provisioning in Mobile Cloud Computing **38**
3.1 QoE Provisioning in Mobile Cloud Computing . . . 38
3.2 Mobile Node . . . 41
3.3 Home Agent and Anchor Point . . . 42
3.4 Corresponding Node (CN) . . . 44
3.5 Summary . . . 44

4 Implementation and Results Evaluation **46**
4.0.1 Requirements . . . 47
4.0.2 Test Plan and Execution . . . 48
4.0.3 Results . . . 49
4.0.4 Dataset initialization . . . 51
4.1 QoE Inferring from CNs in the Cloud . . . 51
4.1.1 Requirements . . . 51

4.1.2 BN and EU building: Test Plan and Execution 52
4.1.3 Simulation of Expected Results 55
4.1.4 Fetched Data for BW from CNs 56
4.1.5 Fetched Data for Delay from CNs 56
4.1.6 Comparison of QoE between CNs 57
4.1.7 RESTful Service in the Anchor Point 58
4.2 Summary 59

5 Conclusion and Future work **60**

List of Figures

1 Number in billions of mobile phone users worldwide 13
2 Number in billions of devices connected to the cloud worldwide 13
3 Base architecture for thesis contribution 16
4 Sustainability assessment for the proposed system 17
5 Handoff scenarios [19] 19
6 Pyramid model to infer a situation 24
7 M2C2 base architecture [1] 26
8 Layers that can be considered for QoE [4] 29
9 Mesh of context attributes that can be used by a BN 31
10 PEVQ algorithm principle workflow 34
11 Bayesian Network using conditional probability tables 37
12 Sequence Diagram for QoE-aware Provisioning System 40
13 MCC considered scenario [1] 40
14 Testbed Architecture for Training Dataset Building 47
15 Architecture for Testbed querying the CNs in the Cloud 52
16 [4] 53
17 Expected Result simulated with Genie BN Modeling tool 56
18 Bandwidth Comparison between CNs in the Cloud 57
19 Delay Comparison between CNs in the Cloud 57
20 Output QoE Comparison between CNs in the Cloud 58
21 QoE output display from RESTful service 58

List of Tables

1 Utility table *U(a,Pr)* for QoE value determination 32
2 Range defined on the bipolar interval scale for mapping overall QoE (R^t_{QoE}) . . 33
3 Parameters for Network selection 42
4 Parameters for comparison task for the Home Agent 43
5 Hardware Metric for Cloud Instances in AWS 45
6 Parameters in Different Network Scenarios for Training Dataset Building . . . 48
7 Stream Parameters Considered for Training Dataset Building 48
8 Results of Experiments Under Different Network Conditions 49
9 PEVQ algorithm sample output 50
10 Realistic Data Fetching from Public Clouds 53
11 Measurements perform in the state of the art [4] representation 54
12 QoS Parameters Discretization from CNs 54
13 Expected Utility Function Values 54
14 QoE Normalization Range 55

ABBREVIATIONS AND SYMBOLS

EU	European Union
MCC	Mobile Cloud Computing
CC	Cloud Computing
QoE	Quality of Experience
QoS	Quality of Service
BN	Bayesian network
CPT	Conditional Probability Table
ICT	Information and communications technology
MN	Mobile Node
HA	Home Agent
AP	Anchor Point
REST	Representational State Transfer Protocol
IP	Internet Protocol
CN	Corresponding Node
PEVQ	Perceptual Evaluation of Video Quality
EU	Expected Utility function
BW	Downlink Bandwidth
AWS	Amazon Web Services
GeNIe	Graphical network interface

1 Introduction

This chapter presents the introductory aspects which will form the basis for this thesis. It describes the background and lands concepts related to Quality of Experience (QoE), Mobile Cloud Computing (MCC), QoE in MCC and their impact towards the sustainable development. Here, it is also presented with the motivation, research questions, and contributions. Finally, an outline of the entire content of this thesis is presented in the closure of this chapter.

1.1 Background

In the recent years, it could be seen an increasing demand of mobile devices which at the same time entail an increasing demand of remote services, namely *services running in the cloud*, which we consider the base of the Mobile Cloud Computing (MCC) model. There is an implicit relationship between the growing number of mobile devices, cloud services and the model of MCC considered in the proposed scenario as shown in Fig. 3. While the demand for mobile devices rises, the services in the cloud need to grow as well. Many popular services nowadays have million of requests every day, entailing high availability from the service side. For example, services like those, offered by Netflix[1] or Spotify[2] are a common topic among regular users. While there is a satisfied demand, meaning users getting a requested service from the cloud, there exists many systems, subsystems, and components that are directly responsible to enable these services to happen. By the latter we mean systems like the Internet itself, to enable the communication to happen between devices. Subsystems, such as a home or cellular networks, that could be considered as part of the Internet. Components, such as devices in both ends of communication, servers and mobile devices, required to meet certain performance. For these components seen in both sides of the communication, the mobile device is expected to meet a determined performance from the end user, but also does a server, where ultimately the service will be requested from. We consider that, if all the previously mentioned systems and subsystems meet certain performance, it becomes accepted by the end user. Thus, it can be inferred a certain quality of experience level is achieved.

For now, in this introductory chapter, we understand for Mobile Cloud Computing (MCC) as the technological environment where the mobile characteristics of the devices make use of com-

[1] http://www.netflix.com/
[2] http://www.spotify.com/

putational resources in the cloud [3], also presented in Fig. 3. In the field of MCC, there have been efforts made from different perspectives. For example, previous work developed such the authors in [1], intend to mitigate some of the constraints for Cloud Computing, such as the end-to-end latency and significant battery life drainage for end devices. Other studies, such as the one in [2] approach another MCC limitation and analyzes the possibility to relieve of the computational load for end devices by relying on cloud technologies. In this particular work, the approach is addressed to cognitive assistance applications. By mentioning examples such the previous work done in the MCC field, we imply that the options of applications provided by the cloud technologies can be wide, meaning that besides the well known massive applications, there are also applications intended for a smaller group of users.

As previously mentioned, behind the demands of the end users exists the functional part of the Mobile Cloud Computing (MCC) model described. In such scenarios, as shown in Fig. 3, we can find one or more mobile node(s) (MN) in the user end, roaming around different networks, and possibly establishing connections to different corresponding nodes (CN) in the other end of the communication. Both MN and CN are explained in detail in Chapter 2 for the presented scenario. The process of changing from one network to another, for example from WLAN to WAN, is determined as *handoff*. This process is done by the MN and it can be classified according to the kind of new network for connection. Further detail is given in the next chapter. To the extent of the network communication establishment, it is of key importance to maintain the Quality of Service (QoS) [3] by the MN so that a certain level of QoE can be satisfied. The authors in [1] presented a system capable to determine what is the best network for connection for the MN while also the selecting best cloud. The mentioned work is merely based on the QoS approach. Nevertheless, in this thesis, we introduce a QoE-aware system. We intend to assess and provision QoE in MCC to ensure the user satisfaction when requesting a service from the cloud, disregarding the kind of service requested.

As mentioned before, the aim of the presented work is to propose a system which addresses the problem presented from the QoE perspective. In this way, it is proposed a system capable to measure and provision QoE to end user requesting a service in through their MN within a MCC scenario.

According to different resources [4], it can be briefly described the following concepts. In the next chapter, these will be explained to a deeper extent.

[3] https://www.ibm.com/cloud/learn/what-is-mobile-cloud-computing
[4] https://www.ibm.com/cloud/learn/what-is-mobile-cloud-computing

- Cloud Computing: The delivery of on-demand Computing resources.
- Mobile Cloud Computing: Technology environment where the user has the ability to access or offload from a mobile device to outsourced computing resources while moving.
- QoE: user perception metric which indicates the level of the end user‘s satisfaction when utilizing an application or service.

The previously mentioned concepts will be worked along the entire thesis.

1.2 Motivation

From a broad point of view, there are two aspects that motivate this work for its development. By one side there is an increasing demand for devices requesting services over an expected quality, meaning that the services requested should fulfill the quality requirements. By the other hand, based on findings on previous work in this field, such as in [4], there is a lack in current systems to infer QoE in MCC.

It is known in these days that there is an increasing demand for mobile devices. The Figure 1 [5] and Figure 2 [6] intend to show the growing demand for mobile devices by year. This trend comes as natural if we consider that cloud services have evolved because it is easier to provide services in an efficient way. This implies a growth in service requests in the cloud.

From the Figures 1 and 2 we assert that there is an increasing demand to be satisfied regarding mobile devices using resources on the cloud.

At the same time, we found different systems in the literature review such as [5–7] where approach the QoE assessment differently from each other but similarly in regards to direct ways of

[5] https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[6] https://www.economist.com/business/2018/01/18/the-era-of-the-clouds-total-dominance-is-drawing-to-a-close/

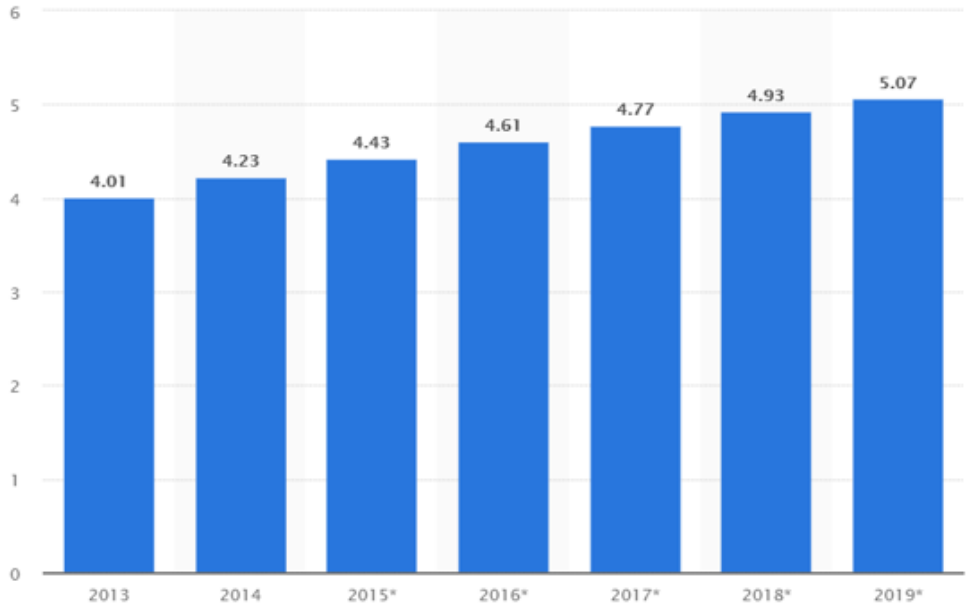


Figure 1. Number in billions of mobile phone users worldwide

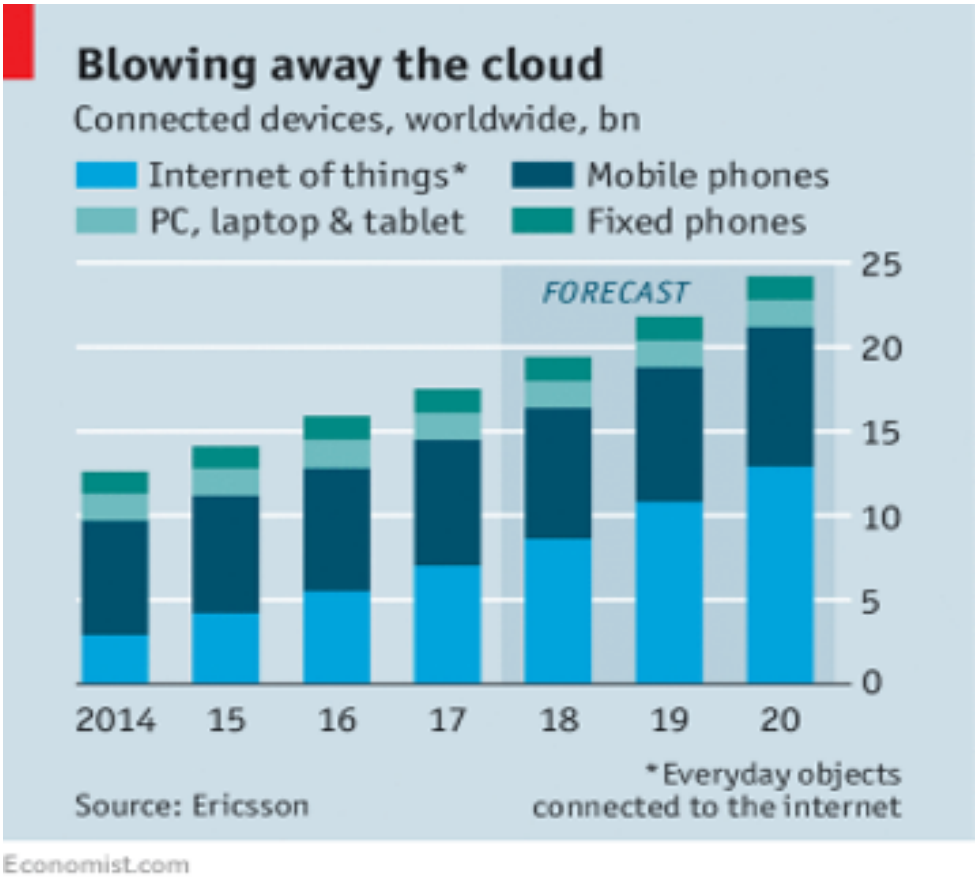


Figure 2. Number in billions of devices connected to the cloud worldwide

measurement. Nevertheless, in [8] we found an approach with potential to be applied to different applications. In its basis, this system infers with Bayesian Networks [9] to infer QoE metrics that were previously measured from realistic experiments, also known as Expert Knowledge [8].

1.3 Thesis Aim

The presented thesis aims to propose a new system for QoE assessment and provisioning applied into a MCC scenario as presented in Fig. 3. In line with this, we implement a testbed for proof-of-concept and gather results accordingly to sustain the presented proposal.

1.4 Research Question and Methods

After making an assessment of the motivation and delimitation, we proceed to determine the following research questions.

- Why is QoE needed in Mobile Cloud Computing?
- How is QoE modeled in Mobile Cloud Computing?
- How can QoE be provisioned in Mobile Cloud Computing?

Consequently, the research method conducted is Experimental Research Methodology, entailing the following aspects.

- Literature review
- Developed proof-of-concept system
- Development of a testbed for the proof-of-concept system
- Conducted experiments
- Results analysis
- Make modifications, when applicable

With the previous points, we intend to propose an answer to our research questions. We would research in the previous work, assisted with the existing database for studies done in the MCC field, about the reasons why is QoE needed in the field of MCC. Subsequently, by gathering different models, compare and analyze how is QoE in MCC modeled currently or most recently. In the same line, we aim to propose an answer to our third research question.

1.5 Thesis Contribution

The research contributions of this thesis are the following.

- Proposed and developed a QoE measurement service for a Mobile Cloud Computing system (M2C2 [1])
- This service used Bayesian Networks and runs in the Anchor Point (AP) so that any Mobile Node can access the QoE output estimated
- Validate the proposed system using video-on-demand service

The architecture where the proposed contributions are to take place is shown in Figure 3.

In the next chapter explain in detail one of the main contributions of this thesis. We elaborate how the proposed approach for QoE assessment and provisioning in MCC will be directly applied in the Anchor Point (AP) according to the architecture shown.

1.6 Delimitations

As it will be seen later in detail, the task of QoE measurement itself can be complex and related to multiple human factors [4]. For this reason, the application selected is video streaming over HTTP within an MCC environment. This application was selected due to the level of sensitivity when it is analyzed in a mobile environment.

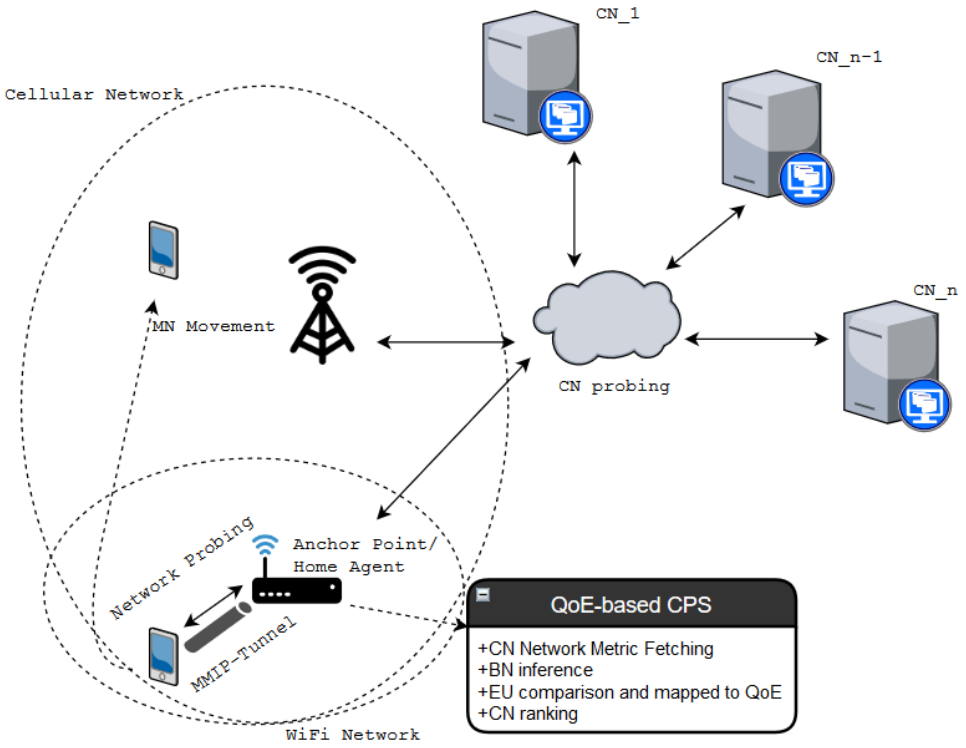


Figure 3. Base architecture for thesis contribution

1.7 Sustainability Aspects

Since this thesis is focused on the sustainable development, a sustainability assessment is performed for the system proposed in this thesis. The analysis is displayed in the Fig. 4.

In this assessment, it is intended to show the potential impacts towards the sustainable development. Direct and not direct approaches are considered for this evaluation.

1.8 Thesis Outline

The next sections of this thesis will be presented as follows.

Chapter 2 presents the technical background from which the development of this thesis is sustained. It also presents the literature review which explains the technical context for this thesis. Concepts in detail such as QoE and MCC are given in detail.

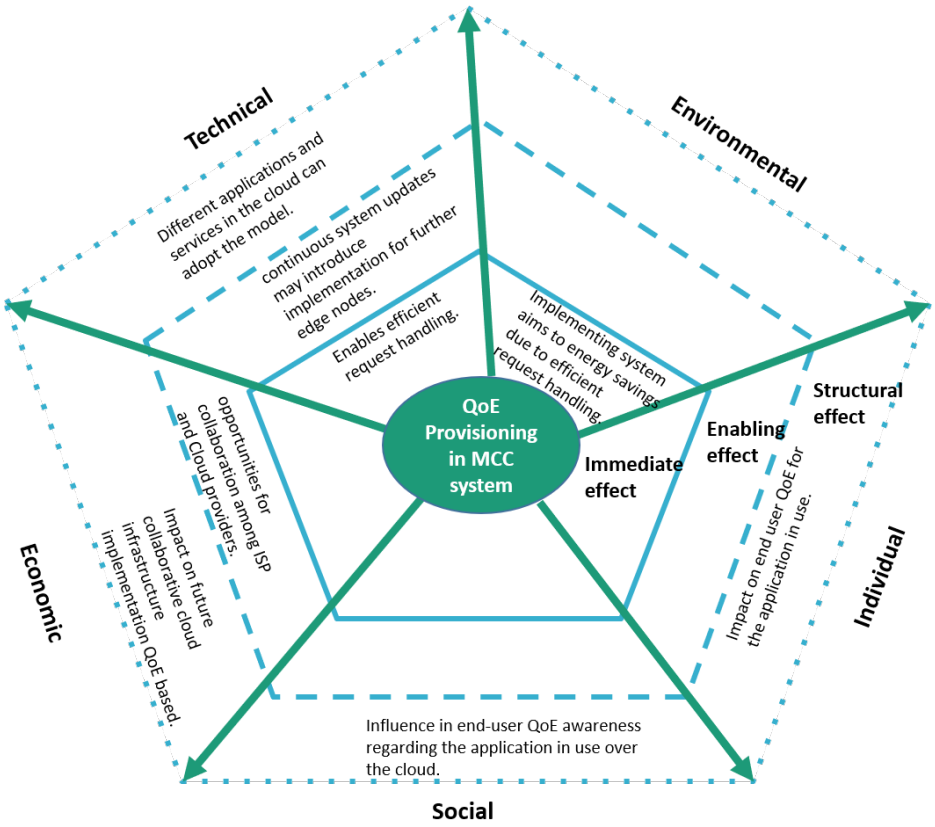


Figure 4. Sustainability assessment for the proposed system

Chapter 3 presents and describes the proposed system, its architecture, and functionalities.

Chapter 4 presents the implementation of the testbed and results gathered from the experiments. Discussion about comparisons between results is also presented in this section.

Finally, in Chapter 5 this thesis is concluded by presenting the conclusion from the overall work and possible directions for future work.

2 Background and Literature Review

This chapter discusses and shows a review of the previous work done in this field. This chapter presents the current work into segments to be thereafter integrated into the summary section. Granularly, Section 2.1 describes the Mobility Management. Section 2.2 elaborates about context awareness for the applications for mobile environments. Section 2.3 presents a discussion regarding Quality of Experience in Mobile Cloud Computing. Context-Aware Quality of Experience Modelling, Measurement and Prediction is explained in Section 2.4. Finally, Section 2.5 presents how can Quality of Experience be assessed and provisioned in Mobile Cloud Computing scenarios.

2.1 Mobility Management in Heterogeneous Access Networks

The Mobility Management has been addressed by techniques using Network protocols, such as Mobile Stream Control Transport Protocol (mSCTP) [10] and Mobile IPv6 [11]. There is dedicated work focused on the comparison of both protocols in [12]. Nevertheless, the discussion is still open and there is still no definite solution as of what is the best protocol for managing mobility for Mobile Nodes (MN), since there is still a lot of improvements and proposed plug-ins to the existing protocols, which take into consideration the non-mobility nature of the TCP/IP protocol [13].

Handoffs occur when a MN moves among different access points (AP), also known as base stations (BS). When using Mobility Management protocols such as MIP [11] the application session can be affected by being disconnected from the network. A handoff in mobile environments is the process of migration from one AP/BS to another one [14, 15] when these events occur, the impact can be reflected in users QoE, according to different previous research, such as [5, 16–18]. There are many applications that are especially sensitive to the handoff effect, such as video streaming, for example. For the case of the latter, it uses TCP in many cases, nevertheless, it is still prone to packet loss when roaming around APs and hence is not guaranteed reliable packet delivery. In the case of MIPv6, this protocol does not support seamless handoffs, hence the need aim seamless handoff process. "seamless" is a term used to denote handoffs with minimal packet loss, delay, and jitter.

The process of handoff can be classified into two types: horizontal or vertical [19] as shown in Fig. 5 Horizontal handoffs, shown on the left side of Fig. 5, are performed when a MN switches from one cell to another within the same network domain or technology i.e., HSDPA to HSDPA

or WLAN to WLAN [19]. Vertical handoffs, shown on the right side of Fig. 5, are performed when a MN migrates from one wireless network technology to another [19], e.g. WLAN to HSDPA.

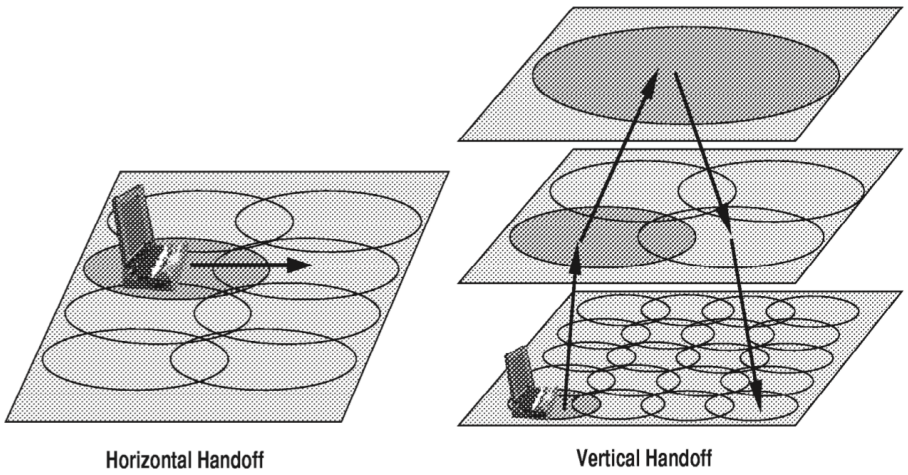


Figure 5. Handoff scenarios [19]

Besides the previously mentioned classification, There is another one regarding the connection or disconnection of the device, in this sense, Handoffs can be also classified in hard handoffs and soft handoffs. It is considered a hard handoff when a MN disconnects from its current network to connects to a new completely alien network, which means there has not been any previous registration and configuration phase with this new network. For example, we can consider MN moving among different WLAN sub-networks. In this scenario, there will be a loss of packets during the handoff process. As for soft handoffs, in contrast with the previous scenario, the session is maintained while the MN connects hooks to the new network meanwhile it is still connected to the network to be detached. If the latter approach is kept, handoffs can be carried out entailing minimal packet loss [1, 20].

For the previously mentioned vertical handoff classification, it can be upward vertical handoff and downward vertical handoff [19]. The differences between them are listed as follows:

- In an upward vertical handoff, the MN changes from a network with higher bandwidth but lower signal coverage area, such in a home area, to a network with lower bandwidth but higher signal coverage area, such as the streets.
- In a downward vertical handoff, it is the opposite scenario, which means the MN disconnects from a network offering lower network bandwidth but higher signal coverage to a

network with higher bandwidth but and lower signal coverage area.

For example, in WLAN to HSDPA handoff, the theoretical bandwidth drops from 54 Mb/s (in case of IEEE 802.11g WLAN) to 14 Mb/s and coverage area increase from 300 m to 20 km [21].

There has been work and development in regard to the different protocols available in the layers 3 and 4 to support mobility. For the third layer there has been work done with Multihoming protocols, including Multi-homed Mobile IP (M-MIP) [22] and Port-based Mobile IPv6 (PM-MIPv6) [23] those directly intended to support seamless vertical handoffs in HANs. By utilizing the multihoming approach, a MN is capable to be connected to several networks in real-time. This scenario is especially beneficial if considered the QoE approach, which is the scope of this work as the MN is able to select the best network interface. Multihoming can also be supported at various layers of the protocol stack according to the literature in [22].

Regarding the previous work done in layer 4 protocols, such as SCTP [24, 25] along with its variant, mSCTP [10] are work to support multihoming at the transport layer. Nevertheless, for the purpose of this thesis, it is considered network layer mobility management protocols such as MIPv6 and M-MIP for the approach it has been decided for the proposed contribution on regards of QoE which will be explained in Chapter 3.

2.1.1 Mobile IPv6

Mobile IPv4/v6 [11, 26], as the name suggests are protocols intended to be used in mobile environments, their main feature is the session continuity during the handoff process. Nevertheless, it can be pointed at the strongest drawback that for MIPv4, in its early implementation it suffered from the triangle routing problem [26]. This problem entailed delays, packet loss and it also could be seen connection dropout when a correspondent node (CN) intended to send packets to a MN via a home agent (HA) on a foreign network (FN) even when the CN was on the same FN. MIPv4 supports IPv4 addressing scheme, and while it is still used, the intention in the community is to smoothly replace it by IPv6 [27] addressing scheme, since it provides a larger pool of addresses, as well as flexibility using other header formats. It overcomes problems such as triangle routing and scalability limitations. The relevant concepts for this protocol are the following:

1. Mobile Node (MN): Any mobile device, it could be a laptop or a smartphone, it has multiple interfaces for simultaneous connectivity to different network technologies.

2. Home Network (HN): This is the network where the MN is permanently and has a permanent IP address.

3. Foreign Network (FN): An outer network where a MN comes from its HN or from another FN.

4. Home Agent (HA): A HA is a router placed on the HN which keeps the connectivity information of the MN. It is in charge of the transfer of packets to the MN when it is on a FN. The MN is always registered with its HA.

5. Correspondent Node (CN): A correspondent node is the other part on the network which is communicating with the MN, such a server or another peer.

An important characteristic in Mobile IPv6 is that the MN has three IP addresses. These are the following:

- Global address: This is the address associated with the HA.
- Care-of address (CoA): This is the address the MN gets when it is in a FN.
- and a Link-local address: This address is used when the MN node intends to communicate with other nodes within the same network

As a basic scheme for MIPv6 functionality, the MN gets a new CoA generated by the stateless address configuration [28] using the network prefix of a FN. Then after getting a new CoA, the MN then sends a binding update (BU) packet to the HA, to get in response a binding acknowledgment (BA). When the CN sends packets to the MN, the packets will go through the HA who will forward them to the MN. Then MN sends the BU to the CN, in this way CN can interact directly with the MN and no need of HA is required. The latter is how the triangle routing problem is solved. Nevertheless, problems are still encountered with MIPv6 such as fair delays. This is due to its operations: network discovery or detection period, network registration and network configuration interval [29].

- Network Discovery or Detection Period: It is the necessary time a MN takes when it moves between many FNs to discover the available networks. This entails that MN checks continuously for the availability of a certain network by computing the signal to noise ratio (SNR) on a network interface or by using link-layer beacons. When the connection is established between the MN and the new AP, it receives the router advertisements (RAs)

from the access routers (AR) present on a FN. The RAs indicate to the MN that it can connect to a certain network by establishing a connection with the local AR. According to [11], RAs are usually sent between 30 ms and 70 ms to perform faster detection period. There is previous work alerting about this constraint, such as [29], relates about bandwidth consumptions for such period of discovery.

- Network Configuration Interval: After the period deemed for Detection, the MN selects a network for connection, which is regularly the strongest signal reaching the MN. Once the network is selected, a virtual network tunnel in the link-layer is set between MN and HA. The MN gets a new CoA based on the network prefix in the RA. When a handoff occurs to WLAN, a new CoA is generated and a tunnel is created with its HA. Within this tunnel, the flow of packets happens between the nodes.

- Network Registration or Configuration Interval: This is the time needed after the CoA is generated so it is ready to send the BU to both HA and CN. After this required times for setting up, the MN sends a BU to the HA and CN with its location information.

For example, when a MN moves to a new WLAN, the AR on this WLAN, it will attach the MN to the network and will set a new CoA to the MN based on its network prefix, the FN's prefix. After this occurs, the MN will send a BU message to the HA and CN informing them about its new location. For every time the MN changes its location and therefore the FN, it will set a new tunnel and the previous one will be deleted. While this solution solves the triangulation problem, the delay is still a concern. It can take from a few milliseconds to many seconds subject to network conditions. Thus, a need for additional features arises.

2.1.2 Multi-homed Mobile IP

Many previous work support multihoming features for seamless handoffs such as [20,30]. Multihoming features a simultaneous connection to many networks for the MN. The MN makes use of its available interfaces so it gets a CoA for each of them, independently from one to another. The main focus of this method it transfers the packets between interfaces with no need for re-establishing a new tunnel when moving to a new FN and thus a new handoff. In the previous section, it was discussed that mobility characteristics can be supported at different layers in the protocol stack. For the purpose and scope for the current work, the multihoming network layer is considered for mobility management, M-MIP [22,31].

2.2 Quality of Experience in Mobile Cloud Computing

Quality of Experience (QoE) as such, is an aggregate of Quality of Service (QoS). From the previous statement, it could be inferred that by having guaranteed what is considered a "good" QoS, in consequence, there would be a "good" QoE as per end-user perception. This is not necessarily correct, the difference between them basically comes because QoS is closely bound to Service Level Agreement (SLA) whilst QoE is closely bound to the user who at the same time happen to be closer to the application itself. To put an example, if there is an office with a limited bandwidth of 4 Mbps that is relatively enough for browsing in the web and it is meeting with the agreed QoS, but at the same time, with this same bandwidth, I will not be possible for the same user in the same office to watch a video on demand, reason for which the QoE will be considered as "low".

QoE is a way to measure a service or application from the user perspective. The challenge entailed in this is to take objective values e.g. network connectivity parameters and transform them into values in a manner that the QoE can be measured [4].

At this point in the current work, we will calculate the QoE accordance with the equation 1:

$$QoE(t) = f\{latency, jitter, bitrate, screensize, videodimensions...\} \tag{1}$$

Note that for the equation 1, the values inside the function are intended to give a clear understanding for the reader and these may change according to the application.

The approach of context awareness for QoE in Mobile Cloud Computing can be illustrated as the Fig. 6 [8]. It can be implied that a situation scenario as given in Fig. 3 is given by the state or states of different context attributes. The last statement is particularly important for this work as it reveals what is the information data that can be processed in order to achieve QoE for an eventual end user. In a real-life scenario, the variables can be many. But if one application is selected, e.g. video streaming, network parameters can be selected such as nominal bandwidth, the current network of attachment, signal strength or latency. As mentioned previously, it will depend on the kind of application to determine whether the QoE is *acceptable*, *good* or *poor*, in the considered scale for assessment, for that specific application. There is also another type of parameters or variables that are not closely bound to anything related to the Cloud environment. These kinds of variables can be the location, the mood of the person, level of light, among others. Nevertheless, since these kinds of variables are deemed out of the scope for this research, it is limited this work into manage variables that can be processed from the perspective that is a

variable that can be processed to infer a situation but not possible to manage it. For example, it is possible to infer that the network is fading if a low-level Signal to Noise Ratio (SNR) is given, which could imply movement of the user, but at the same time, it is not possible to manage at what is the desired pace of the user, unless there is a clear intention to measure this kind of variables, which is not within the scope of the present thesis.

The main motivation to use Context-awareness as a technique is to build or adequate applications and systems to be capable enough to bring end-user performance based on the context information [32]. In this kind of approach, it is suitable to manage information regarding an overall situation as these are meant to indicating states which will be primary input for context-aware applications.

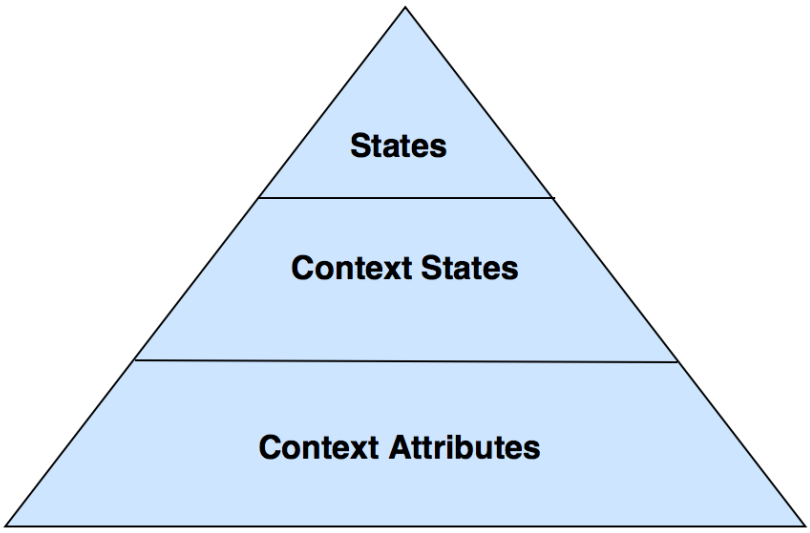


Figure 6. Pyramid model to infer a situation

Figure 6 represents a context-situation pyramid model for context and situation representation. The bottom layer of the pyramid represents raw data collected in form of context values from sensors mainly, but could be another physical interface; the middle layer represents context states which are inferred by processing raw context data. With the latter states, it is possible to determine facts about a user or system in a particular time. Finally, at the top level of the pyramid, it is represented a specific situation which is at the same time inferred from the context states in the previous level [21]. As mentioned before, the aim is to use a context information to determine a situation as accurately as possible. This task could be challenging for many reasons as the overall context can entail a fair level of uncertainty. For example, in a networking environment, context values change often as the network conditions can be stochastic, in very

adverse condition, it could be very difficult to gather information accordingly. For this reason, several techniques were recently proposed by researchers [21].

For the scope of this thesis, we use a context-aware approach which will be detailed in the next section. To make a summary, it can be said that for context-aware QoE modeling, measurement and prediction it is deemed the following main requirements [21]:

Context collection:

- **Context collection:** Data should be gathered as much as possible, the more collection, the better accuracy. This information should be collected with adequate mechanisms available for the fetch of data and it should also be cost efficient in regards to the resources of the device in use.
- **Context representation:** From the collected information it should be done a proper segmentation in order to have a future accurate processing.
- **Context processing:** Extensive algorithms should be used to process the data to finally give the ultimate reasoning for the purpose that is needed. In this case, QoE. It is important to consider efficiency in this approach on regards of resource availability. It is not the same to process in a PC than in a smartphone.

2.2.1 Network Probing

Within the proposal of this work, it is intended to perform an assessment of the entities involved in the scenario as shown in Fig. 3, namely MN, HA/AP, and CN. In order to select what it will be the best option in regards to QoE, there are different approaches that could be considered.

For network probing, it is considered the Multihomed Mobile IP (M-MIP). In previous work, such as in [1], it is shown how it is possible to take advantage of the features of this protocol to aim for seamlessly handoffs in Heterogeneous Access Networks (HAN) environments. This is particularly important for the scope of this work as it considers mobile scenarios, and the aim is to get a certain performance of applications while attaching to different networks. It presents MN performing passive path probing which can be found in Figure 7. This feature consists in sending Binding Update packets (BU) to the HA or to the CN, to get in response Binding Acknowledgment packets (BA). As per the M-MIP protocol itself features, the MN sends these packets constantly to all the network/interfaces it is linked, by doing this, The BUs and BAs from each connection are computed to obtain the Round Trip Time (RTT) which will

lead to getting the Relative Network Load (RNL), which is the parameter which will ultimately be considered for the network selection. These parameters are computed from the following equations [1]:

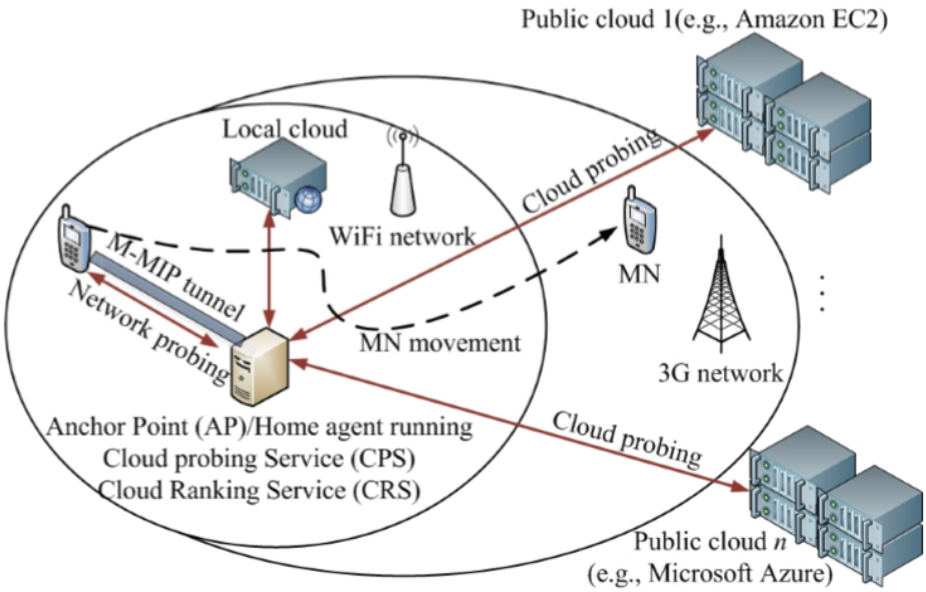


Figure 7. M2C2 base architecture [1]

$$RNL = Z_n + cJ_n \tag{2}$$

$$Z_n = \frac{1}{h} RTT_n + \frac{h-1}{h} Z_{n-1} \tag{3}$$

$$RTT_n = R_n - S_n \tag{4}$$

$$D_n = RTT_n - RTT_{n-1} \tag{5}$$

$$J_n = \frac{1}{h} |Dn| + \frac{h-1}{h} J_{n-1} \tag{6}$$

Where S_n is the time stamp in which is sent a BU packet $n \in N$ from the MN to the next node to reach, it could be either the HA or the CN. R_n is the time stamp of the arrival of the BA packet sent in response from the last BU sent to node reached. h is the value for the history window for calculating the weighted average, where $h = 5$ is considered as an optimal value according to [20]. c is the constant for the weight of the RTT value in comparison with the RTT jitter value. For example, is $c = 2$, it means that RTT jitter value contributes to the overall RNL twice as much as the RTT does in Eq. 2. Variables Z, D and J are iterating parameters and are initialized as follows: $Z_0 = RTT_0$, $D_0 = 0$, and $J_0 = D_1$. At last, the network having the lower

RNL value computed will be the selected one to establish the connection with.

2.2.2 Cloud probing

There are different approaches to assess to which cloud an MN would rather connect to expect acceptable QoE.

In [1] it is considered a way of ranking the available CNs so that there is an assortment of a list for selection of the best CN. The criteria for this ranking is based on QoS parameters gathered from the CN such as CPU Utilization, network throughput, and end-to-end latency. In the scope of this work Amazon Web Services (AWS) [7] is used for fetching this data from the CNs.

The cloud probing in this work uses what is called a Cloud Probing Service (CPS) and a Cloud Ranking Service (CRS). The functionality consists of a built-in RESTful application supported by an Application Programming Interface (API) which will be hosted in the AP. The CPS will request QoS values of the cloud (or CN), such availability or latency, this information is requested from this service every certain time and is then processed by the CRS. The CRS will give a weight for each cloud (CN) in order to give it a rank in its list. For this purpose, it uses the Simple Additive Weighting (SAW), which is a multi-criteria decision-making method (MCDM). The main computing for this raking is given by the following equation:

$$\Re_k = w_{ma}(QoS_{nk}) + (1 - w_{ma})(Cost_{nk}) \tag{7}$$

In order to get an accurate measurement of the parameters values for QoS_{nk} and $Cost_{nk}$ they need to pass through a normalization, the equations for this purpose can be found on [1]. After the ranking computation, the CN with the highest ranking given by $\Re_k$, is deemed the best CN for performance, and hence, selected for connection.

In other approaches such as in [33,34], it is given a broader method to assess the services offered by the CN in the cloud. They segment the assessment in different services as follows:

- Cloud Selection Service: Selects the best cloud based on the Benchmark service feed and the User Feedback Service feed.

[7] `https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html`

- Benchmark Service: Performs the measurement of the clouds envolved for the requested service. It uses a trusted third party for this purpose, which designs a variety of scenarios.
- User Feedback Management Service: After using the services users are prompted to give feedback according to the service that was used.
- Assessment Aggregation Service: It makes the assessment giving weight to what is most important for the requesting user. For example, if the user's requirements are high on security, this aspect will get more weight in the assessment.

For the interest of this work, it could be possible to take advantage of the benchmarking tools available in the field such as HTTPERF [8], and for example, fetch relevant data for the application used such response time from the server. Its increasing popularity tells about its accuracy and usability for benchmarking web server performance. Eventually, tools like this could provide a solution for near real-time scenarios. Nevertheless, for the system proposed in this work, the tradeoff could be further costs in memory. This topic could be an interesting proposal for future investigation.

While the previous approaches for cloud assessment show different methods to tackle the problem according to their own requirements, they do match in some aspects. On this regard, it could be said that objective parameters do matter when it comes to the assessment. Nevertheless, when it comes to the application, it is relevant to measure that the values gathered from the cloud (or CN for this work), meet the requirements for the application in use. Thus, achieving QoE.

2.3 Context-Aware Quality of Experience Provisioning in Mobile Cloud Computing environment

Taking as a baseline that the QoE in Mobile Cloud Computer environments can be more sensitive to real-time applications, the challenge remains into fetching the required data found in parameters from a determined context to process them every certain time and put them out for the intended use. For the video streaming case, it would be to collect the necessary data from the context, these could be parameters taken from either the network, location, device, to mention a few.

There are different approaches to tackle this goal. In [35] it is presented a method to provision

[8] `https://linux.die.net/man/1/httperf`

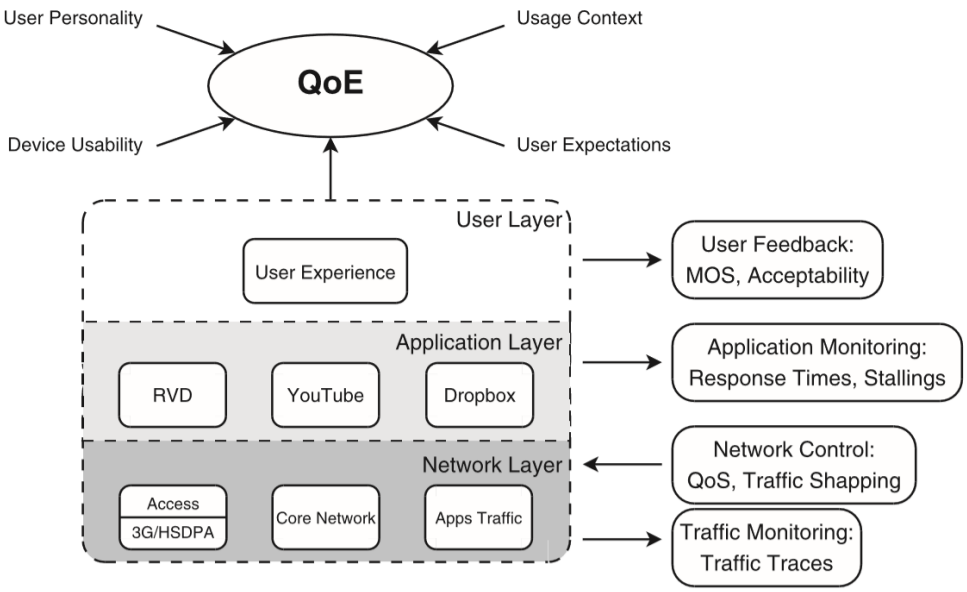


Figure 8. Layers that can be considered for QoE [4]

QoE from QoS parameters for HTTP video streaming applications. In this work, 3 layers are presented in order to reach QoE 9. QoS parameters usually make reference to the network values such as network bandwidth, latency, or jitter. But further from this, this work proposes to consider the application QoS as shown in Fig. 8, to afterward finally infer the QoE values. Application performance metrics (APM) are presented to be applied to quantify the QoS of the application, specifically, they take into consideration the following metrics:

- Initial buffering time: the time the application takes from starting to load the video and the time the video is played.
- Mean re-buffering duration: the mean of how long a re-buffering event takes
- Re-buffering frequency: the times a re-buffering event occurs

The correlation between application QoS and QoE is done with a radar chart mapping, in which metrics such as bandwidth, packet loss rate, and delay are considered. One of the most important conclusions in this work, after making the correlation, is that the re-buffering frequency, which is highly dominated by the packet loss rate metric, is the major factor affecting the QoE.

Another aspect in which is considered the provisioning of QoE in cloud computing environments is given in [6]. The main contribution approach of this work can be divided into the following three segments:

- application placement policy comprising fuzzy logic to prioritize different application requests.
- linear optimization mapping for application placement requests to the cloud instances available.
- evaluation simulation using *iFogSim* [36] , a simulator for cloud environments.

For this work, the CAQoEM [8] proposed methodology is considered: A Context-Aware Quality of Experience Measurement and Prediction in Mobile Computing Scenarios system. This work, presents a high-level approach for QoE measurement and segmentation for interested users, such as stakeholder for a mobile carrier, for example. This model can be better appreciated in Fig. 9. As the name suggests, this system intends to measure QoE based on context information, as discussed previously, these parameters need to be established according to the application in use. This work handles certain key definitions which are the basement of the system as from them the BN will gather the information to be able to estimate the final QoE value.

- Context attribute: It is the data element at a time t which can be deemed as the prime stone to infer a particular situation. Specifically, this information will be fetched directly from specific hardware, e.g. sensors, to be next processed and make reasoning from it.
- Context state: It represents the state of a context attribute. This information can be multiple and needs to be assorted to infer the resulting state. One single context attribute can have multiple context states. At the same time, more context attribute in combination with others can result in different context states.
- Situation Space: It is the result of the inferred context states, it represents a real-life situation. It is basically a statement referring to an actual scenario that is a result of the values gathered from the combination of context attributes and their according states.

The proposed method utilizes the Bayesian Networks and Utility theory to process the context attributes and infer the estimated QoE. It employs the use of context attributes which will be next used as feed to the BN system to learn from them as a starting point and perform future iterations in order to reinforce the learning method and get more accurate results in time.

The Bayesian Network approach is a model which let collect information to infer a hypothesis probabilistically. The following 9 represent the many context attributes which lead to the BN mesh, that could be inferred in the Utility function shown in the Equation 8 from a real scenario.

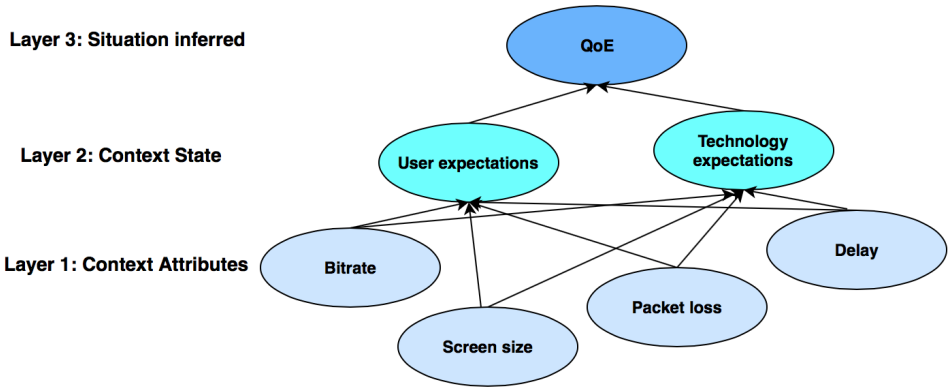


Figure 9. Mesh of context attributes that can be used by a BN

To result in the following Utility Function:

$$\Re_k = w_{ma}(QoS_{nk}) + (1 - w_{ma})(Cost_{nk}) \tag{8}$$

It the Figure 9 it can be appreciated how the context attributes in the lower level are considered, to infer context states in the upper context states and finally in the top level achieve the QoE estimation.

The system needs to go through a normalization or scaling process in order to succeed in the correlation between the QoS parameters with the QoE evaluation. For this purpose, this works presents techniques uses Bayesian Networks (BN) and utility theory as a baseline, and bipolar scale for the mapping according to mapping to the actual QoE value which is often in MOS scale from 1 as "poor" to 5 as "Excellent"

As mentioned before, the functionality of the system overall relies on Bayesian Networks, Utility function, and bipolar scale. The proposal entails the use of Maximum Expected Utility (MEU) [21] principle, which states that a rational agent, for this case: users, should do an action, for this case: give a MOS rating, that maximizes this agent expected utility, for this case it would mean for each context state such as User Expectation (Figure9). From this point, it is defined global utility function ($GU(QoE)$), which can be deemed as an aggregate for all the QoE states and will determine the best QoE-related situation (R^t_{QoE}) overall for that particular scenario with the context attributes and their given states. To determine the R^t_{QoE}, it is needed the Expected Utility for an individual context state, represented S^t_i, which is given by:

$$EU(S_i^t) = \sum_{h \in H} U(h)P(h) \tag{9}$$

Where the $U(h)$ is the utility value assigned according to qualification in the MOS scale and then transformed with linear scale transformation. It is given by The following Table 1:

Table 1. Utility table *U(a,Pr)* for QoE value determination

hypothesis(h)	Excellent	Very Good	Good	Fair	Poor
Utility(h)	1.00	0.75	0.50	0.25	0.00

The normalized values shown in Table 1 are given by 10:

$$U(a) = \frac{V(a) - V(a_{min})}{V(a_{max}) - V(a_{min})} \tag{10}$$

The variable a is a representation for the avaialable option, such in this case in MOS, they would be from 1 (a_{min} = "poor") to 5 (a_{max} = "excellent").

For the other value $P(h)$, it represents the belief of an agent in the hypothesis, for this case, it would represent the probabilistic value of a context state. For example: $P(S_{UE}^t = "excellent") = 0.67$. UE means User Expectation state for the case in this thesis.

By last, the Global Utility function (R_{QoE}^t) is computed following the Eq. 11:

$$GU(QoE) = \sum_{i=1}^{k} w_i * EU(S_i^t) \tag{11}$$

Where w_i represents the weight associated with an individual QoE state, $\sum_{i=1}^{k} w_i = 1$. After this computation, the result of GU is mapped back into the interval scale to select a from Table 2, and this would be the final QoE (R_{QoE}^t)

For sake of clarity. This thesis validates its contribution by doing laboratory measurements with VoIP codecs in a different context so that model can demonstrate its performance in different scenarios. At the same time, it is presented that, disregarding the application in use, the system performs well as long as the context attributes as accurately mapped. The main goal of this

Table 2. Range defined on the bipolar interval scale for mapping overall QoE (R^t_{QoE})

QoE	Range
"excellent"	0.8750 to 1
"very good"	0.6251 to 0.8750
"good"	0.3751 to 0.6250
"fair"	0.1251 to 0.3750
"poor"	0 to 0.1250

work is to determine the overall situation inferred from context attributes which are gathered from raw data obtained from physical interfaces e.g. sensors. This information is fused to make a final statement about QoE in that specific situation.

This method has been selected for the presented work due to the broad approach and the many application that it could be applied to, such as video streaming, augmented reality (AR), or face recognition. In the next subsection, it is explained the basis of Bayesian Networks.

2.3.1 Perceptual Evaluation of Video Quality - PEVQ

As mentioned in the previous section, the CaQoEM system needs a first data input to have a reference and have a scale available to finally give a QoE output measurement. [37]. In this context, we use [**?**] for the objective measurement of quality for video streaming. This approach is used in this work as it follows the recommendation ITU-T Rec. J247 [9].

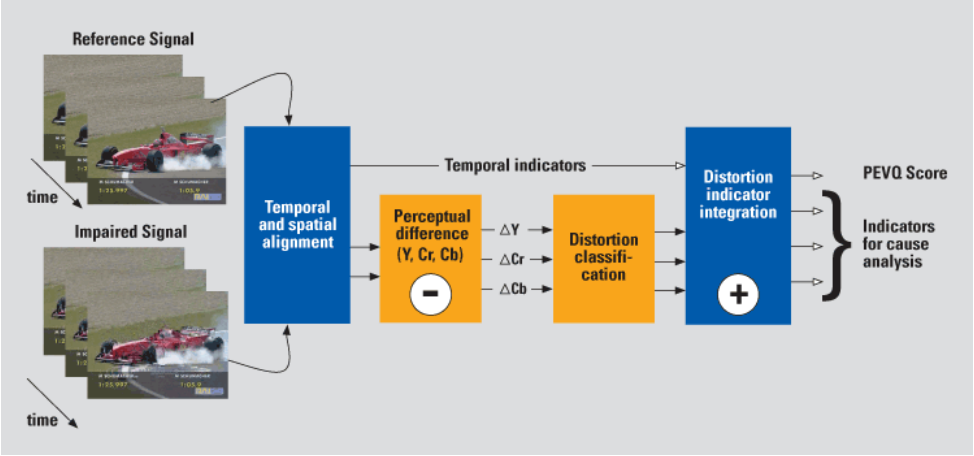
This standard is developed by OPTICOM and it is the Perceptual Evaluation of Video Quality, it provides a MOS score for video applications such as video streaming. It is a full-reference (FR) perceptual measurement algorithm. This kind of algorithm needs both the source signal video as well as the received video signal. These inputs are processed to this algorithm to get a PEVQ score.

PEVQ Measures the following outputs [10]:

- PEVQ MOS: Perceptual rating value in the range from 1 (bad) to 5 (excellent), it is calculated according to ITU-T Rec. J.247.
- Distortion indicators: Analysis of luminance, chrominance, and temporal variables.

[9] `https://www.itu.int/rec/T-REC-J.247/en`
[10] `http://www.pevq.com/pevq.html`

- Delay: frame delay compared with the reference signal
- Brightness and contrast: between both signals
- PSNR: This is a very used metric for image quality by Rec. ITU-T J.340 [11]
- Jerkiness: it refers to the smoothness of a video playback which is affected by down-sampling, coding processes, and perturbed transmissions.
- Blur: Distortion of contour edges and spatial detail.
- Blockiness: more obvious when reduced bit rate. Block matching algorithm for the motion estimation and a coarse quantization for the image blocks.
- Frame Skips and Freezes: Temporal artifacts in video transmissions caused by overloaded networks.
- Effective Frame Rate: Down-sampling of a video signal which results in degradation for the receiving signal.
- Temporal and Spacial Activity: Indicators of activity /movement in the video. These indicators are derived from ITU-T recommendation P.910 [12].



13

Figure 10. PEVQ algorithm principle workflow

The Figure 10 gives a graphical overview about how the algorithm works. Basically, it can be segmented into four blocks.

[11] https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=10551&lang=en
[12] https://www.itu.int/rec/T-REC-P.910/en

1. Block one is for the pre-processing stage. It entails the spatial and temporal alignment for both references and received signal. This way it makes sure the same frames are being compared for both signals.

2. The second block does a calculation of the perceptual difference between the aligned signals. This means that only is considered the difference that a human can perceive, as this is the conceptual approach. At this stage, it is important to consider that motion activity of the signal gives one more indicator for temporal information. Temporal information is important as the perception of details is more sensitive in frames with slow or low activity that in frames with a quick motion.

3. Third block gets the previously calculated indicators and classifies them. It could also detect distortion depending on the kind.

4. The last fourth block does an aggregation of all the previous indicators distortions detected. At this point, according to this aggregation, the Mean Opinion Score (MOS) is given.

2.3.2 Bayesian Network for QoE measurement

References in this field, such [9], consider that as per the unpredictable nature of large, multiply connected networks, relying on Bayesian Network is useful as it considers approximate inference methods to determine probabilities under uncertain conditions. The resemblance between this approach and the context space (CS) model [**?**] representing contextual information as geometrical objects (Figure 9) that lets this approach to develop and make inferences and statements with the results obtained with high accuracy. In this work, it is taken advantage of the CS approach as its basics concepts entail: context attribute, application space, context space, and situation space.

The Bayesian Network as such is considered a direct graph. In this graph each *node* is associated with a probability value which gives information. Taking as a reference the presented Figure 9, The specification of a BN, is as follows [9]:

- A set of random variables (continuous or discrete) are the nodes of the network.
- A set of direct links or arrows connecting each node to its pair. If the arrow comes from node X to node Y, X is said to be a parent of Y.
- For each node X_i they have a conditional probability distribution $P(X_i|Parents(X_i))$ that quantifies the effect of the parameters on the node.

- The graph has no directed cycles, hence it is a directed, acyclic graph (DAG)

The whole setting of the variables network, including the set of nodes and links, will imply the conditional independence relationship between nodes. It could be inferred with the help of the direction of the arrows, who is the parent node with each other. This means that the *intuitive* meaning of an arrow in a given network is usually that *X* has a direct *Influence* on *Y*. On this regard, it easier to get an external source, such an expert in the domain, to assign what is the relationship between nodes in a way that it is clear who is dependent respect with the other. Thus, the BN principle states that once the topology of this BN is set, it is needed to specify a conditional probability distribution for each variable, given its parents. By having this overall setting, meaning the combination of BN topology and conditional distribution, it is sufficient to implicitly specify the full joint distribution for all the variables.

Thus, to represent the full joint distribution, a BN provides a complete description of the presented domain. Every entry in the entire distribution can be calculated from the information in the network given by the following expression in Eq. 12:

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} (x_i | parents(X_i)) \tag{12}$$

Where $parents(X_i)$ gives the specific values of the variables in $Parents(X_i)$, which means that this condition is satisfied by labeling the nodes in an order that is consistent with the order implicit in the graph structure. For the expression presented in Eq. 12 it can be stated that each entry in the joint distribution is represented by the product of the appropriate elements of the conditional probability tables (CPT) in the BN.

The Fig. 11 illustrates the use of CPT in BN. In this example, it is used a simplistic way to represent it for sake of clarity. The principle is that each distribution is shown as a conditional probability table (CPT), where each row has a probability of each node value for a conditioning case, being a conditioning case a possible combination of values for the parent nodes. Thus, it is represented the relationship between nodes for the presented conditions.

2.4 Summary

In this chapter, the intention is to give the base of the concepts used for the proposed system in this thesis. We go into detail with the terminology used along with this work and explain in

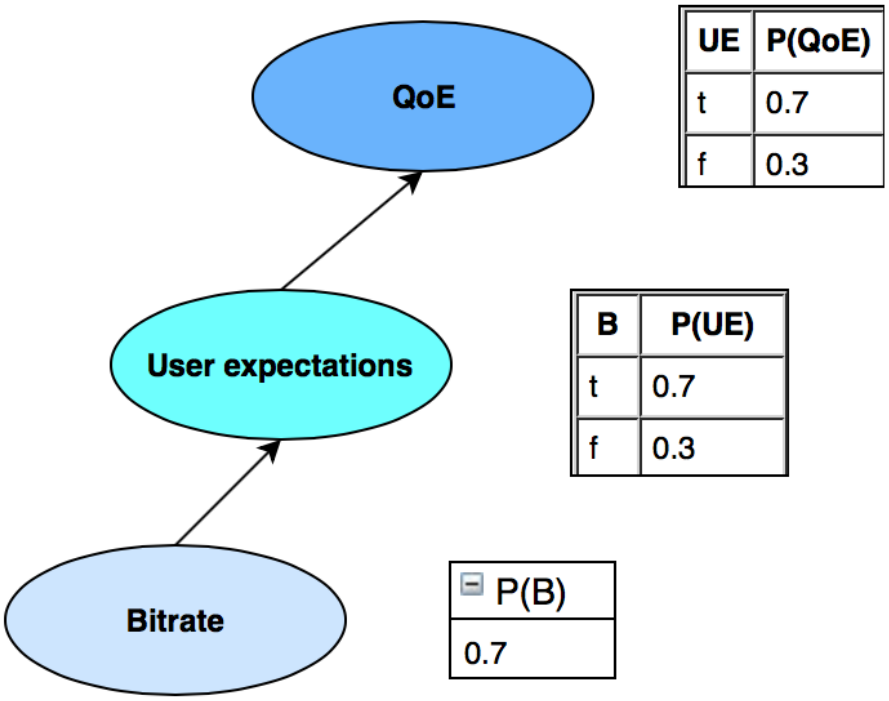


Figure 11. Bayesian Network using conditional probability tables

detail the relations between them. In the following chapters, we explain how these were placed together for proof-of-concept.

3 The QoE Provisioning in Mobile Cloud Computing

This chapter presents our core contribution, the QoE-aware Mobile Cloud Computing System, that ensures QoE when a user roams among HANs and can connect to multiple clouds.

In order to make an objective assessment of a service in the cloud, a specific application is considered, for example, Video streaming over Hypertext Transfer Protocol (HTTP). Nevertheless, the intention of this thesis is to go broader so that the quality of experience can be provisioned to any application running in the cloud requested by the end user while he/she is on the move.

In Chapter 2, we discussed M2C2 [1] as a MCC system, as mentioned previously M2C2 is not QoE-aware. In this chapter, *we extend the concept of QoE to M2C2*. The basic architecture of the proposed system is shown in Fig. 7 in Chapter 2 and is composed by a Corresponding Node (CN), a Mobile Node (MN), Heterogeneous Access Networks (HANs) and an Anchor Point (AP), which in this scenario will implement the Home Agent (HA) for Mobility Management, runs the QoE-aware services for Cloud selection, and also performs as default gateway. In the presented scheme, the AP is in charge of the assessment of both the network and Cloud performance at the t time requested.

3.1 QoE Provisioning in Mobile Cloud Computing

For QoE provisioning in MCC, several parameter values can be gathered from the cloud(s) according to the application in use. For example image resolution, round trip time (RTT), bandwidth (BW), packet loss percentage (PL) and jitter. In this way, we can estimate beforehand what are the parameter values to assess so that the CN and network conditions can meet a satisfactory performance. At the same time, bound to the specific application requested to the cloud, hence provisioning QoE to the end user via the MN.

The cloud environment follows a multi-tenant model [**?**], which means it is a shared resource. Due to this reason, for the interest of the end user, it is of high importance regarding QoE to be aware of the current state of the cloud in real time. Thus, we remark the importance of the presented system. In order to achieve QoE-awareness, it is proposed that the Anchor Point gathers the parameters previously mentioned for assessment in near real time.

QoE assessment steps for the proposed system are described as follows:

1. A request to the service, such as video streaming, is sent from the MN to the CN in the cloud via the AP.

2. The request will let the AP fetch the application and network objective parameters from the CN so that these can be processed in the Cloud Probing Service (CPS) and Cloud Ranking Service (CRS) implemented in the AP.

3. In parallel, the MN periodically probes the network interfaces that are registered in the HA. This keeps a ranking of the best network interface to use. For example, WiFi or the Mobile network. This ranking is updated periodically to the HA running in the AP.

4. The AP probes the available CNs in the cloud. Upon request from the MN, the AP will have to assess which of the listed CNs will provide the best QoE for the end user in real time. To achieve this goal, the RESTful service implemented in the AP.

5. Before running the CRS, the AP will use the input training dataset which is preloaded from previous Expert knowledge data. With the data preset, we infer a BN for assessment of the gathered values, which will be next passed to the utility function, to give as a result the QoE value, for the probed CN.

6. After the assessment of the probed CNs available, the service will output the rank for the best performing CN, this one will be selected based on the level of QoE given by the CRS.

7. Finally, the end user will be provisioned with the best combination of network-cloud for the application he/she desired to use. This process can also be appreciated in the Sequence Diagram in Fig. 12.

As described in the previous steps, and displayed in Fig. 13, the goal of this thesis is to provision QoE to the end user by using services implemented in the AP. At this point, it is assumed that a record of every AP the MN accesses is registered in the HA so that every hand-off is handled smoothly. In this way, accessing the best network while the AP will host the QoE-aware CPS and CRS in the RESTful application in order to rank the CNs from the cloud(s) and hence select the best ranked CN.

When the user requests the service, for example, video, through the MN. The goal of the proposed system is to provide QoE by computing the highest QoE metric available to the end user. In order to achieve this, depending on the application used, there are key parameters that are found more sensitive at the time to evaluate the QoE. As mentioned in previous chapters 1 and 2, these parameters can be many and usually not all of them are related to the system utilized, but also relate in many cases with the human factor [4, 38]. Nevertheless, we consider that if

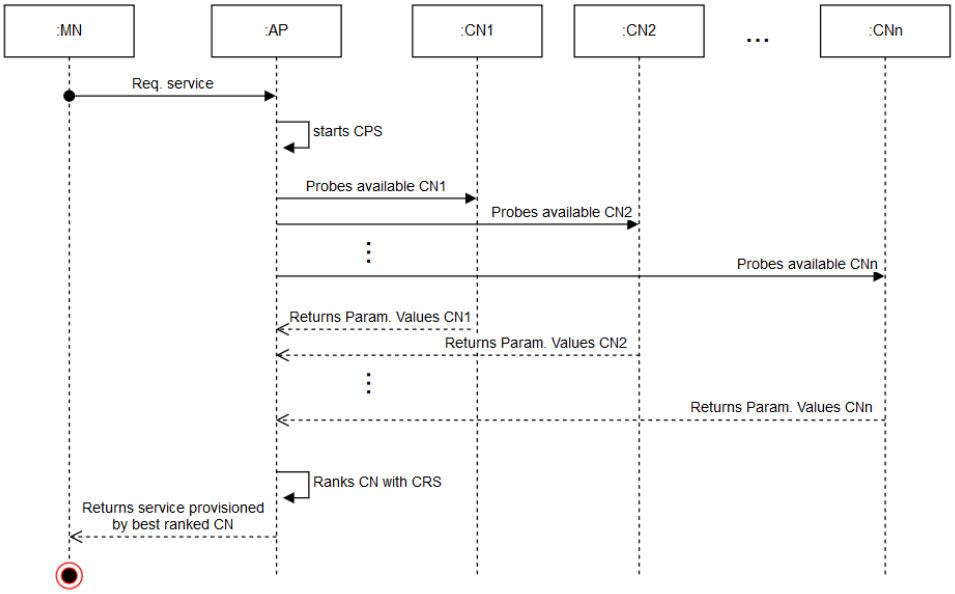


Figure 12. Sequence Diagram for QoE-aware Provisioning System

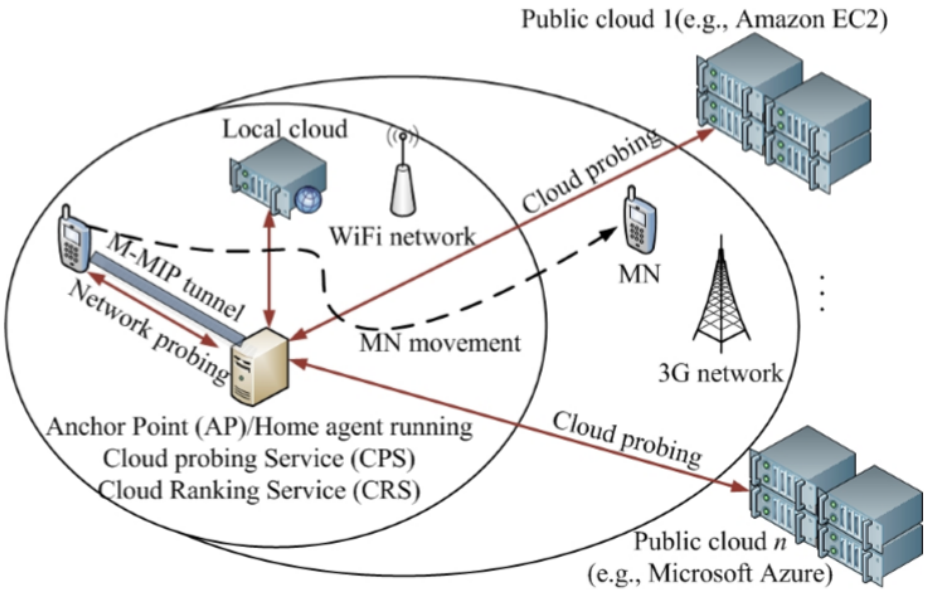


Figure 13. MCC considered scenario [1]

the problem is addressed considering the most sensitive objective parameters closely correlated with the QoE assessment for the application in use, QoE assessment and provisioning could be managed. This can be possible to estimate and compute a QoE metric value close to real time according to the application specific demands [4] and hence provision QoE. For example, for video streaming, it is found that the QoE value is very sensitive in relation with the re-buffering parameter [7, 39]. As it has been shown previously in chapter 2, the end user is keen to tolerate any other fault but this one, video stalls due to an insufficient re-buffering factor, which could

even lead to making the user stop watching a video online [35].

With the latter statement, we identify that it is possible to assess and provision QoE based on what can the CN in the cloud offers in near real time. In the case of the previously given example, to achieve a satisfactory QoE it would be needed to provide a sufficient re-buffering factor, which at the same time relies on the real-time given bandwidth (BW) available from that particular CN requested in the cloud [4]. This means that if the current BW available by requesting a CN in the cloud can be estimated, and followed by comparing this value with the ones provided by the other CNs available in the cloud, a ranking can be performed among CNs based on this parameter. Since the AP is able to reach the other CNs running the requested application in the cloud, by computing this parameter value and then raking the CNs based on it, we can determine what is the best CN in the cloud that gives the best QoE metric in a time t with the given conditions. At this point, it is considered that the network has been already chosen by the HA and thus the combination of best network-cloud QoE-aware is given to the end user in the MN.

To summarize, in a workflow manner and with reference to the Figures 12 and 13. The MN connects to the HA/AP, the HA gathers information to register the network and assess performance to compare with the other networks available for the MN to connect along its way. For example, when the MN is leaving a home-based Access Point (HA/AP), to use instead of the cellular network given by the contracted carrier [1,20]. After this step, the AP collects information from the CNs available in the cloud and this information is computed and processed by inferring with a BN which is preloaded in the AP as a training dataset. In here, the CPS and CRS can assess the considered parameters and map them to finally correlate it to a certain QoE metric output which will be finally the criterion for which the CN on the cloud will be chosen amongst the others. Finally, this one will be proposed as the best one to provision QoE for the end user in the MN [21].

3.2 Mobile Node

In the scenario presented in Fig. 13, the MN is constantly probing the network. This is done in cooperation with the HA in the AP. Here we consider the use of Mobile IP Protocol v6 (M-IPv6) [11] as this thesis is based on related previous work analyzed in the literature review in Chapter 2 [1, 20, 31]. A Binding Update (BU) message is sent to retrieve a corresponding Binding Acknowledge (BA) message. The time difference between those will be considered as the Round Trip Time (RTT). While these packages are meant to be sent periodically, the jitter is also a considered parameter to assess the connectivity quality with the current network among

the others networks available.

The MN will consider the parameters shown in Table 3 for network selection.

Table 3. Parameters for Network selection

Parameter	Description
RTT	With the Round Time Trip parameter it will be possible to measure the time spent to reach the Anchor Point (AP).
Jitter	To measure how stable is the network the MN is attached to.

As the MN uses the Multi-Homed Mobile IP Protocol v6 [23], it will connect the available networks on its range, this will allow the MN to perform passive network path probing to compute the Relative Network Load (RNL) metric [20] for each network available. This is the parameter for selection of the best performing network as explained in 2 in Chapter 2, which will be used by the MN at the requested time t on its location.

After the best-ranked network is selected after probing, the MN will make a request to launch the desired application to the CN in the cloud, such as video streaming over HTTP. This action will start the RESTful service in the AP to run the CPS and CRS as explained in the QoE assessment steps. After this process, it will be possible to get the best network-cloud pair among the options available to provide the requested application.

3.3 Home Agent and Anchor Point

In a mobility scenario as presented in Fig. 13, the HA is hosted in the AP located in the home network, which is at the same time the immediate next node from the MN perspective to reach the internet. The HA main task is to support the MN in the selection of the best network based on the QoS parameters such as RTT and Jitter. This is achieved as the BU and BA messages are sent and received to the HA in to make the network probing [1, 20].

The selection of the best performing combination of network-cloud is provided based on QoE criteria derived from QoS parameters. This will depend directly on the ranking given to the CNs available in the cloud by the RESTful service running the CPS and CRS built in the AP. To achieve this goal the parameters compared vary according to the application requested. For example, for the presented case in video streaming, the AP would gather network parameters such as RTT and BW. Also, the re-buffering parameter can be taken into account to know which

is the fixed minimum QoS requirements for the application in the CN to perform within the expectations for the end user and provide QoE. Meanwhile, computing this parameter also enables to calculate the current BW available to reach the requested CN on the cloud since the re-buffering value is directly dependent on BW, as presented in this case.

In this scope of the RESTful service built in the AP, the following parameters are considered:

Table 4. Parameters for comparison task for the Home Agent

Parameter	Description
RTT	With the round time trip metric, it will be possible to measure the time spent to reach the queried CN on the cloud.
Packet Loss	To measure how reliable is the network the MN is attached to.
re-buffering	Fixed metric closely related to the QoE which will help set the minimum requirements for the other QoS metrics to provision QoE for the end user.
real-time BW	To measure the current bit-rate available from that particular CN on the cloud.

Thus, the AP is in charge to run the CPS and CRS built in the RESTful service. At this stage, the AP will gather the intended metrics data from the CNs available in the cloud every certain time to process them and make the cloud ranking based on the derived QoE parameters. The AP implements the RESTful service in it and will run the mentioned probing and raking services based on an algorithm which processes the data through an inferred BN where it is found a preloaded training dataset, to next pass it to a Utility function. The result is an outputs QoE metric value that will be displayed in the AP and stands for the best ranked CN.

The algorithm **??** shows the logic behind the Cloud Probing (CPS) and Ranking (CRS) services built in the AP.

Algorithm 1 The CPS QoE-based algorithm

1: **for all** $k \in K$ Cloud Sources **do**
2: Fetch QoS values for k: BW, RTT, PL
3: Query the BN with values discretized
4: Get Expected Utility (EU) value for k
5: $QoE_{array}[k] \leftarrow EU[k]$
6: **end for**
7: Return k CN with MAX(QoE_{array})

As mentioned previously, the AP is not only intended to be in the Home Network but also in the cellular network. Reason for which the service is intended to be deployed in every AP.

3.4 Corresponding Node (CN)

To complete the description of the node elements in the proposed architecture as shown in 13, we have the CN in the cloud. The Corresponding Node (CN) in the cloud is a computational resource from where the application in the cloud will be requested. It is at the same time, the other end in the communication established with the MN when requesting an application service in the cloud.

The CN in the cloud can be classified as one of the following categories.

- Public Cloud: Freely accessed by any MN throughout the internet.
- Private Cloud: Could be accessed by any MN over the internet but for a monetary cost.
- Hybrid Cloud: Utilizes both private and public services.

A CN in the cloud is subject to its hardware (HW) characteristics and limitations, depending on which is the cloud service provider, these characteristics can differ from one CN to another. The following is a table in which the HW is described for a CN. In this case, the cloud provider is Amazon Web Services (AWS) [14] where the CNs are called Cloud Instances [15].

While HW characteristics are relevant for the scope of this thesis, it is of key importance to take into account the geographical location of the CN in the cloud requested. This condition could be a determinant for the case studied in this thesis as it will be shown in the results obtained in the next chapter.

3.5 Summary

This chapter presented the concepts and elements we use in this thesis for QoE provisioning in MCC. In particular, we emphasize the terminology used, the technologies, techniques, and how

[14] http://aws.amazon.com/
[15] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/

Table 5. Hardware Metric for Cloud Instances in AWS

Metric Name	Description
CPUUtilization	The percentage of allocated EC2 compute units that are currently in use on the instance. This metric identifies the processing power required to run an application upon a selected instance.
DiskReadOps	Completed read operations from all instance store volumes available to the instance in a specified period of time.
DiskWriteOps	Completed write operations to all instance store volumes available to the instance in a specified period of time.
DiskReadBytes	Bytes read from all instance store volumes available to the instance.
DiskWriteBytes	Bytes written to all instance store volumes available to the instance.
NetworkIn	The number of bytes received on all network interfaces by the instance. This metric identifies the volume of incoming network traffic to a single instance.
NetworkOut	The number of bytes sent out on all network interfaces by the instance. This metric identifies the volume of outgoing network traffic from a single instance.
NetworkPacketsIn	The number of packets received on all network interfaces by the instance. This metric identifies the volume of incoming traffic in terms of the number of packets on a single instance. This metric is available for basic monitoring only.

the elements in the MCC scenario considered the link with each other. At this point, we clear concepts related to mobility and QoE provisioning within the scope of this thesis.

4 Implementation and Results Evaluation

In this chapter, we present and explain the implementation and tests performed for the proposed QoE provisioning system in MCC. We include details of the software tools utilized, and also the testbeds deployed. We also mention the challenges faced during the implementation and tests during at this stage.

The results are presented in chronological order according to the occurrence of testbeds implementations, these are two in total. The first is about expert knowledge data generation from an objective assessment algorithm for video quality. In this testbed deployment, we show how we were lead and helped with the development of this thesis to come to a preliminary conclusion which will be mentioned later in chapter 5 for conclusions and future work. We consider this preliminary conclusion of key importance for the deployment of the second testbed. The second testbed is about directly querying CNs on the cloud from the simulated AP, which in turn processes the data queried with the proposed model for QoE provisioning in MCC as proposed in Fig. 15. We get the output through a RESTful service implemented in the AP and displayed for reference. A notorious difference between both deployments relies on the source from where the training dataset for BN is taken. Both deployments are explained in detail in the following subsections.

For the deployment of this testbed, the focus was to build a Bayesian Network (BN) based to a training dataset to manage a proprietary set of data so it can be queried from a BN. The final QoE value would be given by the RESTful service built in the AP. As mentioned in Chapter 2, for the deployment of this testbed, it was used the Perception Evaluation Video Quality (PEVQ) Algorithm [40]. The main intention of using this particular tool is to run experiments recreating different network scenarios taking as a baseline what was analyzed in the literature review in Chapter 2 [35].

For sake of clarity, in the Fig. 14 we present the basic architecture utilized for the testbed deployed. Next, we proceed to explain its components accordingly in the Requirements subsection.

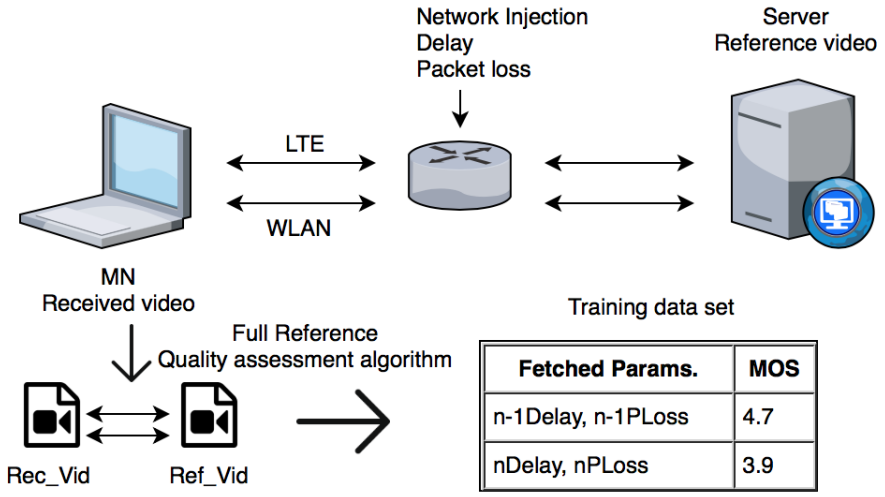


Figure 14. Testbed Architecture for Training Dataset Building

4.0.1 Requirements

The testbed deployed in this stage is compound by the following elements:

- Physical Machines: 3 Ubuntu Desktop v. 18.04 [16] as physical machines where each one represents the client, router, and server respectively as shown in Fig. 14.
- Network Simulator: NetEm [17] built-in package to simulate network impairments according to the required scenario. This software tool was used entirely in the machine intended for it, isolated.
- Video Stream Client: VLC Media Player [18] was used in the client machine to fetch the received video signal from the server side.
- Video Stream Server: PLEX [19] media server was used in the server machine. It was selected due to its flexibility to offer a web interface from the client.
- Install Platform: A Docker [20] container was used in the server machine to install the PLEX server over it.

[16] http://releases.ubuntu.com/releases/18.04/
[17] https://wiki.linuxfoundation.org/networking/netem
[18] https://www.videolan.org/vlc/index.fr.html
[19] https://www.plex.tv/
[20] https://www.docker.com/

4.0.2 Test Plan and Execution

At this point in the deployment, we execute a test plan taking as a baseline the literature reviewed [35] in chapter 2. The Table 6 presents the parameters considered to simulate the scenario for the Training Dataset building.

Table 6. Parameters in Different Network Scenarios for Training Dataset Building

Parameters	**Values**
Network bandwidth (Mbits/s)	1, 5, 10, 15, 100
RTT (ms)	0, 25, 50, 75, 100
Round-trip packet loss rate (%)	0, 2, 4, 6, 8
Video dimensions	640x480

It was relevant for testing purposes to make the assessment with long enough video durations to meet the requirements of the PEVQ algorithm. Also, other requirements were considered for the video streaming test, these are shown in the Table 7.

Table 7. Stream Parameters Considered for Training Dataset Building

Item	**Value**
Video Length	20 seconds
Video format	H. 264
Audio Format	ACC
Resolution	864x480
File size	4.3 MBytes
Frame rate	23.97 fps
File format	AVI
Stream Protocol	HTTP

The video used for the stream is found in the literature review [41] and it was selected as it follows the ITU recommendations for QoE evaluation.

The experiment workflow was conducted as follows. The PLEX Media Server was launched in the 1 of the 3 Ubuntu physical machines, while the Client accessed the web server interface of the Server through the Ubuntu Router machine. Next, the video would be streamed in the client

while it is loaded in the buffer and finally downloaded when finishing the video play.

Results and analysis of this stage are given in detail in the next subsection.

4.0.3 Results

The Table 8 shows the parameter values ran for the experiments, taking as reference the method used by the authors in [35]. The Table 9 shows the information in the output given by the PEVQ algorithm, where amongst the video information parameter values, the QoE MOS value is given.

Table 8. Results of Experiments Under Different Network Conditions

Network Bandwidth (bits/s)	**RTT (ms)**	**Round-trip Packet Loss rate (%)**	**PEVQ (MOS)**	**File Name**
1.05 M	2.266	0	4.7	ref-0
512 K	2.168	0	4.7	ref-1
256 K	2.169	0	4.7	ref-2
1.05 M	25	0	4.7	ref-3
1.05 M	50	0	4.7	ref-4
1.05 M	75	0	4.7	ref-5
1.05 M	100	0	4.7	ref-6
1.05 M	2.194	2	4.7	ref-7
1.05 M	2.469	4	4.7	ref-8
1.05 M	2.294	6	4.7	ref-9
1.05 M	2.361	8	4.7	ref-10

With the results obtained, it was sufficient to realize that with the current approach it was not possible to build a training dataset due to its own intended test method itself. We observed that the received signal was in every case the same than the source, it did not contain corruption in the received signal due to network impairments. Whit this outcome, by applying the PEVQ algorithm, the resulting QoE MOS value was always the same, this is because the HTTP protocol is intended to rebuild the network packets so that the end file does not get corrupted. Nevertheless, this will be done at a certain cost, which is the download time. After coming to this

Table 9. PEVQ algorithm sample output

PEVQ Version	2.000 ;;	PEVQ Frame Skips [%]	20.717 ;;
;;		;;	
Reference file	tmp.avi ;;	PEVQ Jerkiness	0.000 ;;
Ref Fps [1/s]	24.000 ;;	PEVQ Blur	0.008 ;;
;;		PEVQ Blockiness	0.000 ;;
Degraded file	tmp-0.avi ;;	PEVQ Mean Temporal Dist	0.177 ;;
Deg Fps [1/s]	24.000 ;;	PEVQ Worst Temporal Dist	1.516 ;;
;;		;;	
NrFrames	251 ;;	PEVQ Temporal Activity Reference	51.323 ;;
Work Fps [1/s]	25.000 ;;	PEVQ Temporal Activity Test	51.336 ;;
Frame duration [ms]	40.000 ;;	PEVQ Spacial Activity Reference	125.797 ;;
Pixels per line	854 ;;	PEVQ Spacial Activity Test	125.800 ;;
Lines per frame	480 ;;	PEVQ Brightness Reference	5.872 ;;
;;		PEVQ Brightness Test	5.872 ;;
;;		PEVQ Contrast Reference	7.684 ;;
PEVQ DSCORE	0.344 ;;	PEVQ Contrast Test	7.684 ;;
;;		;;	
PEVQ SCORE	4.656 ;;	PEVQ PSNR Y	74.453 ;;
;;		PEVQ PSNR Cb	59.758 ;;
PEVQ Est FPS Reference [1/s]	23.8 ;;	PEVQ PSNR Cr	61.177 ;;
PEVQ Est FPS Test [1/s]	23.8 ;;	PEVQ Luminance Indicator	0.000 ;;
PEVQ Delay Min [ms]	0.000 ;;	PEVQ Chrominance Indicator	0.244 ;;
PEVQ Delay Max [ms]	0.000 ;;	PEVQ Correlation Indicator	0.000 ;;
PEVQ Delay Mean [ms]	0.000 ;;	;;	
;;		Pass through data	;;
PEVQ Frame Freeze [%]	0.000 ;;	Calculation time	172.54;;

preliminary conclusion, we noted that due to the HTTP protocol property mentioned, another approach could be considered. For example, an algorithm managing the download time. Because as mentioned before, the download time was the parameter most affected when running the tests and manipulating network impairments in the testbed. Hence, a relationship between Download Time - QoE could approach the problem from that perspective.

Due to time constraints for the development of this thesis, A different approach was utilized to

overcome this problem, which will be explained in the next subsection.

4.0.4 Dataset initialization

For this testbed deployment, in order to have an entity capable to assess and rank the best Cloud instance, as explained in Chapter 3, it requires to have pre-set data in the Anchor Point (AP).

4.1 QoE Inferring from CNs in the Cloud

At this stage in the simulation, we assume that MN makes use of Multi-Homed Mobile IP to be able to connect simultaneously to the available networks and perform the network assessment and selection according to the proposed model in Fig. 13. The deployment of this testbed entails the implementation of a service built in the AP.

For the deployment of this testbed, we come with the knowledge and experience acquired from the previous deployment. The focus was to build a Bayesian Network (BN) as a training dataset to have a set of data for inference with a BN. After inferring the processed data, it will be passed to the utility function and finally, a QoE value would be given by the RESTful service built in the AP. This time, we relied upon measurements already performed in the literature reviewed in chapter 2 [4] to build the BN. As mentioned in the Introduction in Chapter 1, one of the contributions is to demonstrate that it is possible to make an inference with a BN with Expert Knowledge. In this case, these are measurements that were used in previous validated experiments to build the Training Dataset.

For sake of clarity, in Fig. 15 we present the architecture of the testbed deployed. Thereafter, we proceed to give detail of the elements involved accordingly in the Requirements subsection.

4.1.1 Requirements

For this testbed deployment the following elements were considered:

- Physical Machine (Laboratory): Ubuntu machine v. 18.04 as Anchor Point (AP).

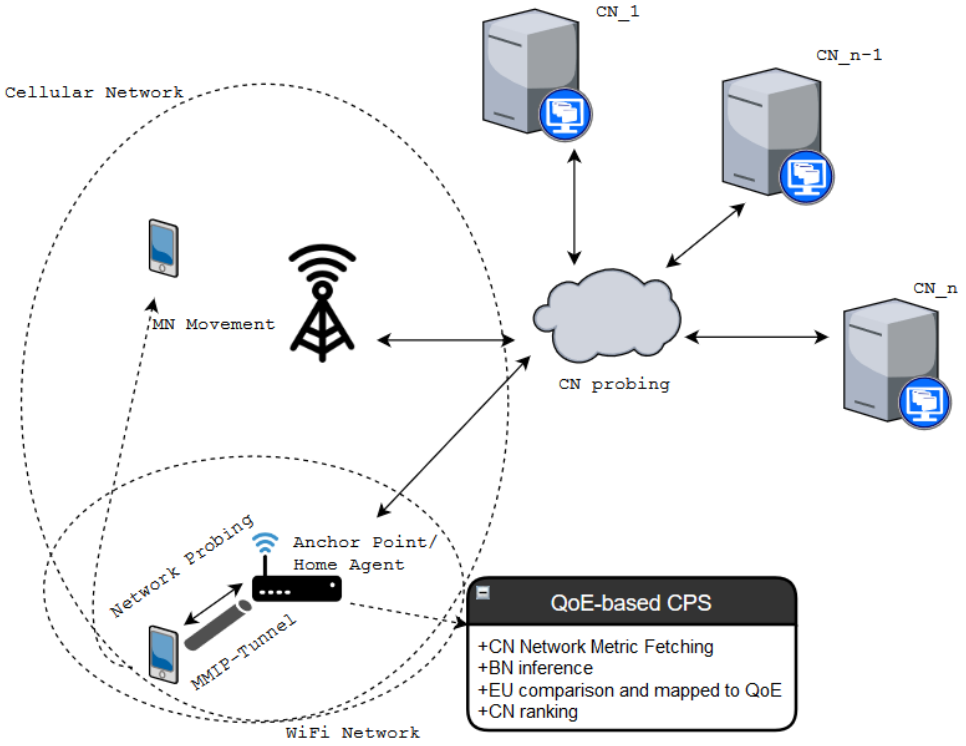


Figure 15. Architecture for Testbed querying the CNs in the Cloud

- Cloud: AWS EC2 [21] instances as CNs in the Cloud to query realistic data from them.
- Software: Falcon Web Server Gateway Interface (WSGI) library [22] for building APIs, Python [23] as the programming language supported by Falcon, and Nginx [24] as a web server. The three mentioned SW, to be used in the implementation of the QoE provisioning service in the AP.

4.1.2 BN and EU building: Test Plan and Execution

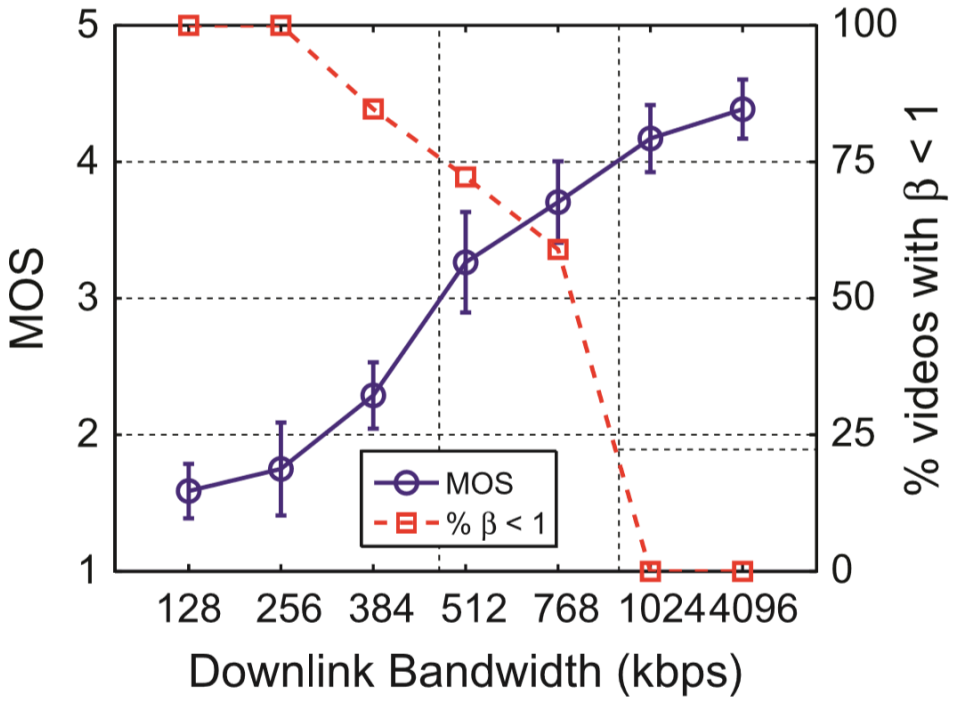
The elements described in the Requirements subsection for this testbed were used under the certain conditions to cope with the scope of this thesis. These are described in Table 10.

The experiment workflow was conducted as follows. The following measurements performed in the state-of-the-art found in the literature review by the authors in [4] where used to build the Training Dataset for inference with the BN.

[21] https://aws.amazon.com/ec2/
[22] http://falcon.readthedocs.io/en/stable/
[23] https://www.python.org/
[24] https://www.nginx.com/

Table 10. Realistic Data Fetching from Public Clouds

Item	**Value**
Cloud Provider	Amazon EC2
Datacenter Location	US, Tokyo, Sydney
QoS Parameters	Bandwidth, RTT, Packet Loss
Frequency	3 times/day. Morning: 09h, Afternoon: 15h, Night 23h
Network tools	iPerf, Ping
Protocols	HTTP, ICMP
Bayesian Network software tool	GeNie Academic for MS Windows



(a) MOS and β vs DBW

Figure 16. [4]

This information is represented in the Table 11.

In this case, we take measurements based on the Downlink Bandwidth. In the same Fig. 16 on the right axis it is also shown a β parameter = DBW/VBR and it is used as a video quality indicator. In this thesis, we use only a fixed quality value (360p, progressive scan). For reason, this parameter is not considered for the BN building. Delay and Packet Loss (PL) were considered because these are deemed relevant the for the application selected. Nevertheless, the BW would be assigned more weight in the Utility Function for inference in the BN.

Table 11. Measurements perform in the state of the art [4] representation

BW in Kbps	MOS range	Normalized range
128	1.4 - 1.8	0.28 - 0.36
256	1.4 - 2.1	0.28 - 0.42
384	2 - 2.6	0.4 - 0.52
512	2.9 - 3.6	0.58 - 0.72
768	3.5 - 4	0.7 - 0.8
1024	3.9 - 4.5	0.78 - 0.9
4096	4.2 - 4.6	0.84 - 0.92

Thus, the parameters were fetched from the CNs in the Cloud according to the requirements shown in the Requirements subsection. The Table 12 shows the criteria selected based on the techniques applied in the literature review by the authors in [21] for discretization of the selected parameters.

Table 12. QoS Parameters Discretization from CNs

Parameters	Values (range)
Bandwidth	Good (> 1024 Kbps), Fair (512 Kbps - 1024 Kbps), Poor (< 512 Kbps)
Delay	Good (0 ms - 150 ms), Fair (150 ms - 300 ms), Poor (> 300 ms)
Packet Loss	Good (0% - 5%), Fair (5% - 7%), Poor (7% - 10%)

After the creation of the BN for inference, the result passes along to a Utility function as explained previously in Chapter 2. The Utility function is represented as follows in the Table 13.

Table 13. Expected Utility Function Values

Hypothesis(h)	Excellent	Very Good	Good	Fair	Poor
Utility(h)	1	0.75	0.5	0.25	0

The fetched data was inferred to the BN which will then output its result to the Expected Utility (EU) function. Thereafter this result would be mapped into the QoE normalization range table as shown in Tab. 14 in Chapter 2. We also show it in this section for sake of clarity with the workflow in the Table 14.

Table 14. QoE Normalization Range

QoE	Range
"excellent"	0.8750 to 1
"very good"	0.6251 to 0.8750
"good"	0.3751 to 0.6250
"fair"	0.1251 to 0.3750
"poor"	0 to 0.1250

After treating the data it was proceeded to corroborate results and display the actual outcomes. These are explained in the next subsections.

4.1.3 Simulation of Expected Results

In this subsection, we present the results obtained for this deployment. First, we present the results from the simulations made to get the final QoE output value, and next, the realistic data fetched from the CNs in the cloud.

The Fig. 17 shows the result from the simulation done with Genie BN Simulator [25] for MS Windows [26].

As it can be seen in the Fig. 17, the final expected QoE value is gotten from the inference of the BN built and then passed after the EU function. This value corresponds to an only one CN amongst the total of three CNs available in the cloud. For this testbed, 3 CNs in the cloud were queried in different locations as presented in the Requirements subsection.

After this value is gathered from all the CNs available in the cloud, these values are passed through the Cloud Probing Service (CPS) and Cloud Ranking Service (CRP), implemented in the AP as explained in Chapter 3.

In the next subsections, we show the results of the data collected from the CNs in the cloud.

[25] https://www.bayesfusion.com/academic-users
[26] https://www.microsoft.com/en-us/windows

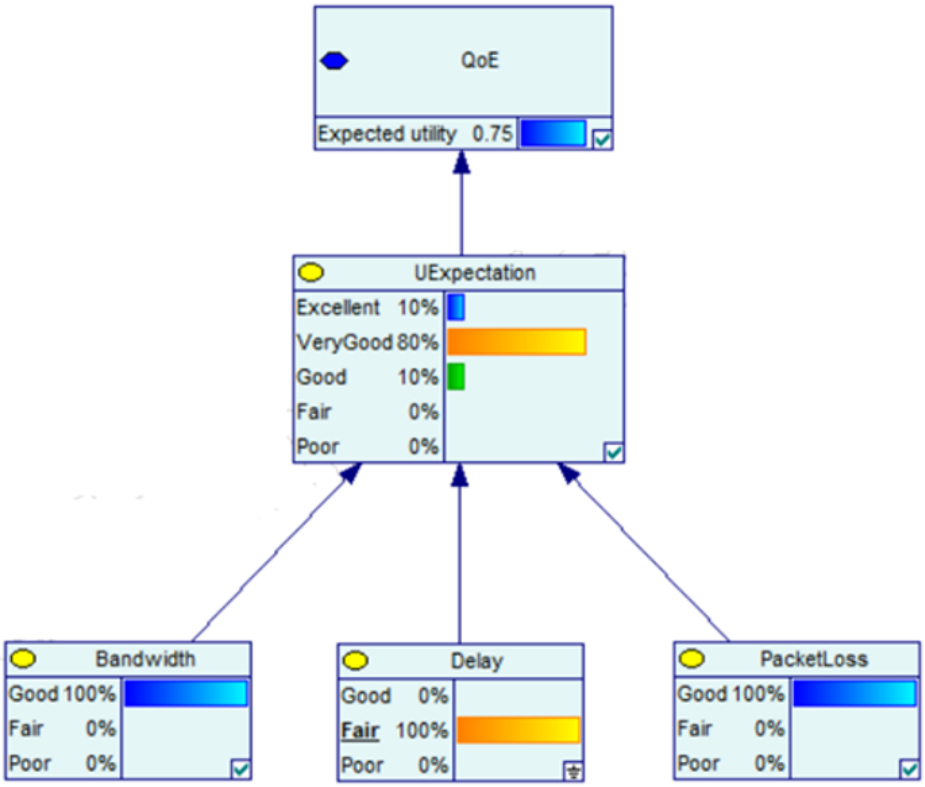


Figure 17. Expected Result simulated with Genie BN Modeling tool

4.1.4 Fetched Data for BW from CNs

The Fig. 18 presents a comparison of BW parameter between the queried CNs in the cloud.

As it can be appreciated, the US CN ranks above the others with around 3.5 Mbps downlink BW. This would be considered as "Good", according to the presented Discretization Table 12.

4.1.5 Fetched Data for Delay from CNs

The Fig. 18 presents a comparison of Delay parameter between the queried CNs in the cloud.

As it can be appreciated, the US CN ranks above the others as it presents around 200 ms od delay. This value would be considered as "Fair", according to the presented Discretization Table 12.

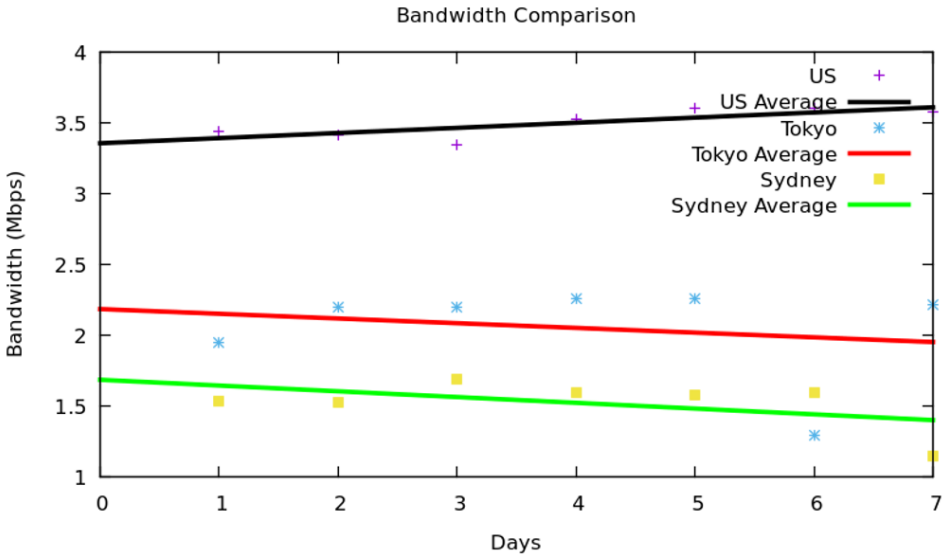


Figure 18. Bandwidth Comparison between CNs in the Cloud

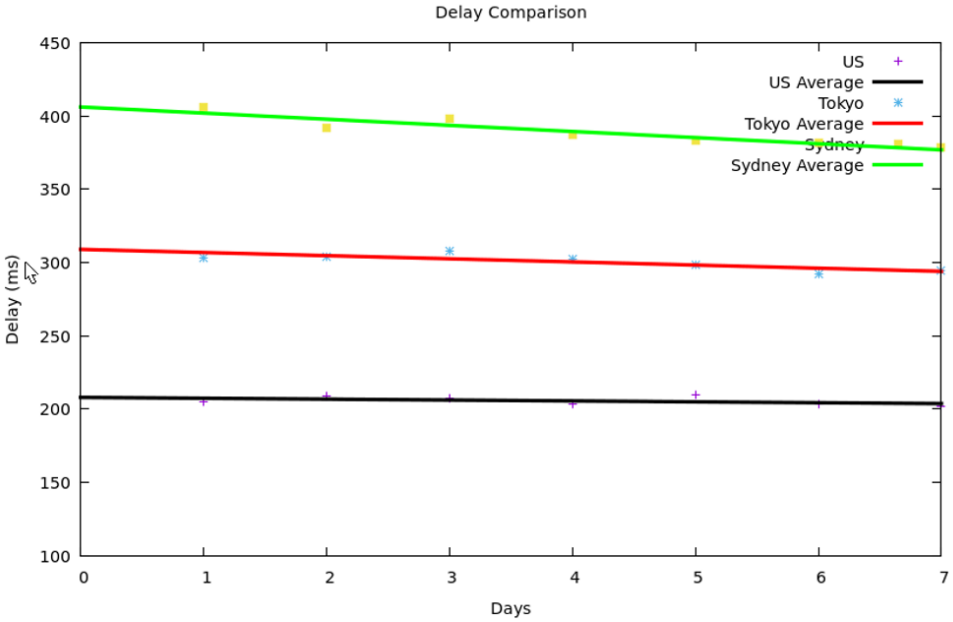


Figure 19. Delay Comparison between CNs in the Cloud

4.1.6 Comparison of QoE between CNs

At this stage, the PL parameter is also considered "Good" for this case, as it was given the value of 0% for all the cases. With all the data treatment as explained before, the QoE value comparison is performed between the queried CNs as shown in the Fig. 20.

As it can be appreciated in the Fig. 20, The US CN outranks the others as it gets an inferred value of 0.75 which is mapped to the QoE Normalization Table 14 as "Very Good".

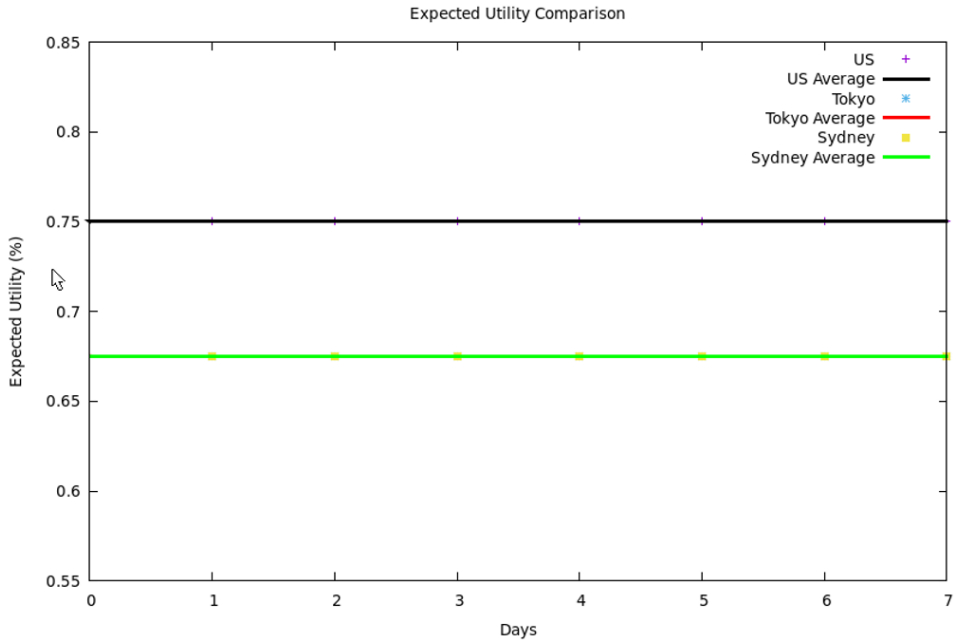


Figure 20. Output QoE Comparison between CNs in the Cloud

4.1.7 RESTful Service in the Anchor Point

In this subsection, as part of the results, we present an example of the QoE value presented by the RESTful service implemented in the AP. The Fig. 21 shows how would it look when a MN access the information in the AP.

Figure 21. QoE output display from RESTful service

4.2 Summary

This chapter presented the results for QoE provisioning in MCC. In particular, we did some specific experiments and showed that BN can be used to infer a QoE provisioning service. We mention the tools available and the access to those. Also, about time constraints and how to face changes for situations encountered along the way. It is not more to mention, that despite any difficulty presented in the development of testbed, this was cleared accordingly and the final goal could be achieved.

5 Conclusion and Future work

The presented thesis addresses the problem of Quality of Experience provisioning in Mobile Cloud Computing environments by developing a system which entails the deployment of a RESTful service in the Anchor Point according to the architecture model proposed [1]. This system considered two main approaches to approach the problem overall: One of them is ensuring the smooth hand-off process by the use of Multi-Homed Mobile IPv6 [20], and the other is the task of provisioning QoE through inference of Bayesian Networks for the application selected, which for this thesis is video streaming over HTTP due to the reason and under the conditions specified in Chapter 2 of this manuscript. On this regard, it was of key importance to state how does the QoE relate to the QoS parameters as treated in this thesis. It was concluded that the re-buffering factor is a sensitive concern towards QoE, as it affects the video streaming by producing stalling. At the same time, this factor can be controlled if the enough QoS parameters are provided when requesting the application from the cloud.

Based on the testbed deployments, it was concluded that different ways to build a Conditional Probability Table (CPT) [8] from a training dataset for Bayesian Network inference could also be considered to get results for the final QoE output as per the model proposed. This point was considered when it was needed to approach a different addressing of the problem when gathering results after the deployment of the first testbed implementation. As explained in that section, it was concluded that the approach to building a proprietary dataset should have been different from the one performed due to the properties of the HTTP protocol itself. The measurements should have been done by developing an algorithm to control the time spent on downloading a received video file from the server instead of evaluating the video degradation. Nevertheless, the results were fruitful in the terms that let this thesis overcome the problem by investigating more in the state-of-the-art and find the method finally proposed, which entails to get an input dataset from measurements already done, proved, and accepted by the Academia and that follows the ITU recommendations.

In line with the statistical approach proposed in this thesis. It could be successfully proven that Quality of Experience metrics could be inferred using the probabilistic approach such as Bayesian Networks built from statistical data as shown with the Conditional Probability Tables (CPT) and the Training Dataset. It is asserted that, by the results gathered from the simulations performed and the actual data processed from the cloud, the proof-of-concept is validated.

As per the technology environment for the proposed solution, monetary costs were incurred for the deployment of the testbed, a reason for which it is also concluded that cost related matters are also relevant. This criterion was as well one limitation on regards selecting multiple cloud providers for the proof-of-concept deployment with the testbeds presented. Due to this reason, this case was selected for one cloud provider only. Another criterion considered for this was the familiarity with the cloud provider, in this case, Amazon Web Services, as the authors of this thesis had already familiarity with the provider environment and the API provided, Nevertheless, the intention to work with different cloud providers is left for future work.

In the development of the system, while assessing the many human factors that can influence the Quality of Experience perceived from the end user, the knowledge from the end user regarding Quality of Experience for the application requested over the cloud could be addressed differently. On this regard, the end user could select the application they are going to use into an interface before they make the actual request, this interface could be a web form, for example, and from there gather the expected parameters involved so they can successfully get Quality of Experience. Nevertheless, this is one approach that requires more investigation and hence it is left for future work.

It was proved that with the tools utilized for the testbed deployments, the proposed system can be built in the Anchor Point as a RESTful service so that the Mobile Node (MN) can access to the Quality of Experience information and be provisioned when required within a Mobile Cloud Computing scenario. With this work, the authors expect to contribute in the field of Mobile Cloud Computing and the application offered in the cloud environment.

Finally, as this thesis aims for the sustainable development of society, and for that reason also an impact analysis was made in Chapter 1, two main aspects were considered: One is the technical impact, as it is expected to improve application performance in Mobile Cloud Computing environments when the requests from end users are made for the applications over the cloud. It is expected that, for more successful queries, the fewer resources would be waste, hence aiming for efficiency. The other one aspect for sustainability is considered the economy since it is expected to provide useful information for Internet Service Providers (ISP) and Cloud providers who in cooperation, can build together or otherwise work towards the Quality of Experience for the end user.

REFERENCES

[1] Christer Åhlund Daniel Granlund Karan Mitra, Saguna Saguna. M2c2 : A mobility management system for mobile cloud computing. In *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*, 2015.

[2] Wenlu Hu Kiryong Ha Brandon Amos Padmanabhan Pillai Alex Hauptmann Mahadev Satyanarayanan Zhuo Chen, Lu Jiang. Early implementation experience with wearable cognitive assistance applications. *WearSys '15 Proceedings of the 2015 workshop on Wearable Systems and Applications. Pages 33-38*, 2015.

[3] Selim Ickin, Katarzyna Wac, Markus Fiedler, Lucjan Janowski, Jin-Hyuk Hong, and Anind K Dey. Factors influencing quality of experience of commonly used mobile applications. *IEEE Communications Magazine*, 50(4), 2012.

[4] Pedro Casas and Raimund Schatz. Quality of experience in cloud services: Survey and measurements. *Computer Networks*, 68:149–165, 2014.

[5] Kandaraj Piamrat, Adlen Ksentini, Cesar Viho, and Jean-Marie Bonnin. Qoe-based network selection for multimedia users in ieee 802.11 wireless networks. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 388–394. IEEE, 2008.

[6] R Mahmud; SN Srirama; K Ramamohanarao; R Buyya. Quality of experience (qoe)-aware placement of applications in fog computing environments. 2018.

[7] Pantelis A. Frangoudis Adlen Ksentini Sunny Dutta, Tarik Taleb. On-the-fly qoe-aware transcoding in the mobile edge. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, 2016.

[8] Christer Åhlund Karan Mitra, Arkady Zaslavsky. Context-aware qoe modelling, measurement, and prediction in mobile computing systems. *IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 14, NO. 5, MAY 2015*, 2015.

[9] Peter Norvig Stuart Russell. *Artificial Intelligence:A Modern Approach*. Pearson Education, 2nd. edition, 2009.

[10] Seok Joo Koh, Moon Jeong Chang, and Meejeong Lee. msctp for soft handover in transport layer. *IEEE communications letters*, 8(3):189–191, 2004.

[11] Charles Perkins, David Johnson, and Jari Arkko. Mobility support in ipv6. Technical report, 2011.

[12] Dong Phil Kim and Seok Joo Koh. Analysis of handover latency for mobile ipv6 and msctp. In *Communications Workshops, 2008. ICC Workshops' 08. IEEE International Conference on*, pages 420–424. IEEE, 2008.

[13] Atheer Dhiaa Al-Rubaye et al. *Vertical Handover Management with Quality of Service Support*. TU Ilmenau Universitätsbibliothek, 2017.

[14] Debabrata Sarddar, Tapas Jana, Souvik Kumar Saha, Joydeep Banerjee, Utpal Biswas, and Mrinal K Naskar. Minimization of handoff failure probability for next-generation wireless systems. *arXiv preprint arXiv:1006.3376*, 2010.

[15] Shantidev Mohanty. *Architecture and Cross-Layer Mobility Management Protocols for Next-Generation Wireless Systems*. PhD thesis, Georgia Institute of Technology, 2005.

[16] Karan Mitra, Arkady Zaslavsky, and Christer Åhlund. Pronet: Proactive context-aware support for mobility in heterogeneous access networks. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 675–676. IEEE, 2009.

[17] Kandaraj Piamrat, Adlen Ksentini, César Viho, and Jean-Marie Bonnin. Qoe-aware vertical handover in wireless heterogeneous networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 95–100. IEEE, 2011.

[18] Karan Mitra, Christer Åhlund, and Arkady Zaslavsky. Qoe estimation and prediction using hidden markov models in heterogeneous access networks. In *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*, pages 1–5. IEEE, 2012.

[19] Mark Stemm and Randy H Katz. Vertical handoffs in wireless overlay networks. *Mobile Networks and applications*, 3(4):335–350, 1998.

[20] Christer Åhlund Karl Andersson, Daniel Granlund. M4: multimedia mobility manager: a seamless mobility management architecture supporting multimedia applications. *MUM '07 Proceedings of the 6th international conference on Mobile and ubiquitous multimedia Pages 6-13*, 2007.

[21] Karan Mitra. *Quality of experience measurement, prediction and provisioning in heterogeneous access networks*. PhD thesis, Luleå tekniska universitet, 2013.

[22] Arkady Zaslavsky Christer Åhlund, Robert Brännström. M-mip: Extended mobile ip to maintain multiple connections to overlapping wireless access networks. *International Conference on Networking ICN 2005: Networking - ICN 2005 pp 204-213*, 2005.

[23] Christer Ahlund, Robert Brannstrom, Karl Andersson, and Orjan Tjernstrom. Port-based multihomed mobile ipv6 for heterogeneous networks. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 567–568. IEEE, 2006.

[24] Mohammed Atiquzzaman and Abu S Reaz. Survey and classification of transport layer mobility management schemes. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 4, pages 2109–2115. IEEE, 2005.

[25] Aggeliki Sgora and Dimitrios D Vergados. Handoff prioritization and decision schemes in wireless cellular networks: a survey. *IEEE Communications Surveys & Tutorials*, 11(4), 2009.

[26] Charles E Perkins. Ip mobility support for ipv4, revised. 2010.

[27] Stephen E Deering. Internet protocol, version 6 (ipv6) specification. 1998.

[28] Thomas Narten, Susan Thomson, and Tatuya Jinmei. Ipv6 stateless address autoconfiguration. 2007.

[29] Rajiv Chakravorty, Pablo Vidales, Kavitha Subramanian, Ian Pratt, and Jon Crowcroft. Performance issues with vertical handovers-experiences from gprs cellular and wlan hotspots integration. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 155–164. IEEE, 2004.

[30] John Fitzpatrick, Séan Murphy, Mohammed Atiquzzaman, and John Murphy. Using cross-layer metrics to improve the performance of end-to-end handover mechanisms. *Computer communications*, 32(15):1600–1612, 2009.

[31] Arkady Zaslavsky Christer Åhlund. Multihoming with mobile ip. *IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC 2003: High-Speed Networks and Multimedia Communications pp 235-243*, 2003.

[32] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[33] Lie Qu, Yan Wang, and Mehmet A Orgun. Cloud service selection based on the aggregation of user feedback and quantitative performance assessment. In *Services computing (scc), 2013 ieee international conference on*, pages 152–159. IEEE, 2013.

[34] Marc Torner Tomàs, Arto Hämäläinen, Jouni Ikonen, and Jari Porras. Service discovery and selection approaches in peerhood environment. In *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, pages 257–262. ACM, 2010.

[35] Ricky K. P. Mok; Edmond W. W. Chan; Rocky K. C. Chang. Measuring the qoe of http video streaming. *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, 2011.

[36] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47(9):1275–1296, 2017.

[37] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J Karam. Objective video quality assessment methods: A classification, review, and performance comparison. *IEEE transactions on broadcasting*, 57(2):165–182, 2011.

[38] Parikshit Juluri; Venkatesh Tamarapalli; Deep Medhi. Measurement of quality of experience of video-on-demand services: A survey. *IEEE Communications Surveys & Tutorials (Volume: 18, Issue: 1, Firstquarter 2016)*, 2016.

[39] Florian Wamser ; Michael Seufert ; Pedro Casas ; Ralf Irmer ; Phuoc Tran-Gia ; Raimund Schatz. Yomoapp: A tool for analyzing qoe of youtube http adaptive streaming in mobile networks. *Networks and Communications (EuCNC), 2015 European Conference on*, 2015.

[40] M Keyhl, Mathias Obermann, SM Satti, and Graham Rousell. Perceptual quality evaluation of ott streaming video tv services. In *Proceedings of the 68th NAB Broadcast Engineering Conference*, volume 2014, pages 252–260, 2014.

[41] Video Library. Download Guide. Big buck bunny, sunflower version. Available online: http://bbb3d.renderfarming.net/download.html. (Accessed on 18 April 2018).

[42] Kishor Trivedi Haiyang Qian, Deep Medhi. A hierarchical model to evaluate quality of experience of online services hosted cloud computing. *12th IFIP/IEEE International Symposium on Integrated Network Management 2011*, 2011.

[43] Michael Jarschel; Daniel Schlosser; Sven Scheuring; Tobias Hoßfeld. An evaluation of qoe in cloud gaming based on subjective tests. *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2011.

[44] João Falcão e Cunha Teresa Galvão Pedro Maurício Costa, Jeremy Pitt. Cloud2bubble: Enhancing quality of experience in mobile cloud computing settings. *MCS '12 Proceedings of the third ACM workshop on Mobile cloud computing and services Pages 45-52*, 2012.

[45] Abdelhamid M. Sajid Mushtaq, Brice Augustin. Empirical study based on machine learning approach to assess the qos/qoe correlation. *Networks and Optical Communications (NOC), 2012 17th European Conference on*, 2012.

[46] Martin Varela Christian Timmerer Tobias Hoßfeld, Raimund Schatz. Qoe management in emerging multimedia services. *IEEE Communications Magazine • April 2012*, 2012.

[47] Michail Alexandros Kourtis Vaios Koumaras Emmanouil Kafetzakis, Harilaos Koumaras. Qoe4cloud: A qoe-driven multidimensional framework for cloud environments. *2012 International Conference on Telecommunications and Multimedia (TEMU)*, 2012.

[48] Eui Nam Huh Wei Tang, Tien-Dung Nguyen. A survey study on qoe perspective of mobile cloud computing. In *Information Science and Applications (ICISA), 2014 International Conference on*, 2014.

[49] Jari Porras Anaël Racine. Elements affecting remote execution decision on peerhood environment. In *2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2009.

[50] Haidar Chikh. Apollo: A system for proactive application migration in mobile cloud computing. Master's thesis, Luleå University of Technology Department of Computer Science, Electrical and Space Engineering, 2016.

[51] Alexander Tolstoy Sergey Zapechnikov Natalia Miloslavskaya, Mikhail Senatorov. Big data information security maintenance. *SIN '14 Proceedings of the 7th International Conference on Security of Information and Networks Pages 89*, 2014.

[52] Amy R. Reibman ; David Poole. Characterizing packet-loss impairments in compressed video. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2007.

[53] Anind Dey Mahadev Satyanarayanan Dan Siewiorek Asim Smailagic Brandon Taylor, Yoshihisa Abe. Virtual machines for remote computing measuring the user experience. Technical report, School of Computer Science Carnegie Mellon University Pittsburgh, PA15213, 2015.

[54] Christoph Becker ; Stefanie Betz ; Ruzanna Chitchyan ; Leticia Duboc ; Steve M. Easterbrook ; Birgit Penzenstadler ; Norbet Seyff ; Colin C. Venters. Requirements: The key to sustainability. *IEEE Software (Volume: 33, Issue: 1, Jan.-Feb. 2016)*, 2016.

[55] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[56] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *International Conference on Pervasive Computing*, pages 167–180. Springer, 2002.

[57] Xiaohuan Yan, Y Ahmet Şekercioğlu, and Sathya Narayanan. A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks. *Computer networks*, 54(11):1848–1863, 2010.

[58] Denis Rosário, Matias Schimuneck, João Camargo, Jéferson Nobre, Cristiano Both, Juergen Rochol, and Mario Gerla. Service migration from cloud to multi-tier fog nodes for multimedia dissemination with qoe support. *Sensors*, 18(2):329, 2018.

[59] Pari Delir Haghighi, Shonali Krishnaswamy, Arkady Zaslavsky, and Mohamed Medhat Gaber. Reasoning about context in uncertain pervasive computing environments. In *European Conference on Smart Sensing and Context*, pages 112–125. Springer, 2008.

[60] Alexandra Klimova, Eric Rondeau, Karl Andersson, Jari Porras, Andrei Rybin, and Arkady Zaslavsky. An international master's program in green ict as a contribution to sustainable development. *Journal of Cleaner Production*, 135:223–239, 2016.

[61] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1):1–11, 2011.