



# Analysis and influence of business parameters on quality of experience for Over-The-Top services

Diego Rivera Villagra

## ► To cite this version:

Diego Rivera Villagra. Analysis and influence of business parameters on quality of experience for Over-The-Top services. Networking and Internet Architecture [cs.NI]. Université Paris Saclay (COMUE), 2017. English. NNT : 2017SACLL004 . tel-01498214

**HAL Id: tel-01498214**

**<https://theses.hal.science/tel-01498214>**

Submitted on 29 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2017SACLL004

THÈSE DE DOCTORAT  
DE  
L'UNIVERSITÉ PARIS-SACLAY  
PRÉPARÉE À  
TÉLÉCOM SUDPARIS

ECOLE DOCTORALE N° 580  
Sciences et Technologies de l'Information et de la Communication

Spécialité de doctorat Informatique

Par

**Mr. Diego Rivera Villagra**

Analyse et influence des paramètres d'affaires sur la qualité  
d'expérience des services Over-The-Top

**Thèse présentée et soutenue à Evry, le 28/02/2017 :**

**Composition du Jury :**

Mme. CAVALLI, Ana Rosa	Professeur, Télécom SudParis	Directrice de thèse
M. CUPPENS, Frédéric	Professeur, Télécom Bretagne	Président
M. TIXEUIL, Sébastien	Professeur, UPMC-LIP6	Rapporteur
M. SÉNAC, Patrick	Professeur, ENAC (École Nationale d'Aviation Civile)	Rapporteur
M. MELLOUK, Abdelhamid	Professeur, Université Paris-Est	Examineur
Mme. KUSHIK, Natalia	Maître de Conférences, Télécom SudParis	Examinatrice
Mme. ZAIDI, Fatiha	Maître de Conférences HDR, Université Paris Sud	Examinatrice
M. BUSTOS-JIMENEZ, Javier	Directeur de Recherche, NICLabs	Invité



*A mis hermanas, quienes han sido la  
motivación para que este documento exista.*



# Acknowledgments

First of all, I would like to express my sincere thanks to my thesis director Ana Cavalli for trusting me and giving me the opportunity to develop the work presented in this document. She also put her efforts and confidence to allow me to work to actively in the European Project *New Generation Over-The-Top Services (NOTTS)*. I will always appreciate the opportunity and the training she gave me to collaborate with partners across Europe while, at the same time, develop the work for the thesis.

Besides the support of my supervisor, I would also like to thank Nina Yevtushenko and Natalia Kushik for their constant support and interest on the work I was developing. They were always open to discuss the ideas and propose new ways to expand the work presented in this document. The words I am writing would not exist if it were not by the constant support.

Although this work was not conceived as a co-guidance, I have always received the support of the whole NICLABS team back in Chile, where I developed my whole previous work. They have supported my actual work by offering their support and disposition to take long surveys and evaluate the quality of videos. Their responses made possible the development of the last part of this work and constitute a principal part of the results presented here. I would like to thank in particular to Prof. Javier Bustos, who always supported me and offered all his help.

In addition, the whole team of the former LOR department at Télécom SudParis (nowadays RS2M department) was a huge support for this work, to whom I am really thankful. In particular, I would like to thank Jorge for the coffee breaks and discussions about interesting research topics not usually related with this work, Olga for being such a nice friend and provide support when I needed it, José for sharing histories, coffee and beer whenever I could, and Pamela for being my closer Chilean connection on whom I could rely when I required it.

There were other people with whom I could share great moments during this journey. First I would like to give special thanks to Rebeca. She has been one of the few people I can call “friend,” sharing uncountable moments with her – lunches, dinners and others – and offering me her support whenever I needed it. In a similar way, I would like to thank Javier. When we met in Santiago we only shared a professional relationship, however the time proved that I could rely on him not only with professional support but also as a real friend. Finally, I would like to thank all my friends from Chile who came to Europe (and Paris) and shared some time with me. Maybe you did not notice, but sharing even a few hours with you made me remember the good times we had at the University, and gave me strength to continue with the job I love.

Last but not least, I would like to thank my family, for being always there when I missed them. Despite the distance, I still love to hear their voices and see their faces when everything seems to be dark.

---



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problematic . . . . .	3
1.3	Scope and Plan of the Thesis . . . . .	4
1.4	Publications . . . . .	6
1.5	Publications non-related with the main topic . . . . .	7
<b>2</b>	<b>State of the Art</b>	<b>9</b>
2.1	Over-The-Top (OTT) Services . . . . .	9
2.1.1	The Beginning of Time: Telecommunication and Audiovisual Media . . . . .	9
2.1.2	The Internet Era, and the Raising of Internet-based Services . . . . .	10
2.1.3	The “Invasion” of The Content Market Land . . . . .	11
2.1.4	The “Unfair Competition” War . . . . .	15
2.1.5	The “Network Neutrality” Treaty . . . . .	16
2.2	The Quality of Service (QoS) and Quality of Experience (QoE) concepts . . . . .	17
2.2.1	Quality of Service . . . . .	17
2.2.2	Quality of Experience . . . . .	20
2.3	The Quality of Business (QoBiz) concept . . . . .	23
<b>3</b>	<b>EFSM-based Quality of Experience Evaluation Framework</b>	<b>25</b>
3.1	Introduction . . . . .	25
3.2	Preliminaries . . . . .	26
3.2.1	Finite State Machine (FSM) . . . . .	26
3.2.2	Extended Finite State Machine (EFSM) . . . . .	27
3.2.3	$l$ -equivalent of an Extended Finite State Machine . . . . .	29
3.3	Quality of Experience Evaluation Framework . . . . .	31
3.3.1	Modeling an Over-The-Top Service . . . . .	32
3.3.2	Augmenting the Model with Quality Indicators . . . . .	34
3.3.3	Computation of the $l$ -equivalent model and the Quality of Experience . . . . .	37
3.4	Discussion . . . . .	41
3.5	Conclusions . . . . .	42



---

<b>4</b>	<b>Implementation of the Quality of Experience Evaluation framework</b>	<b>43</b>
4.1	The Montimage Monitoring Tool (MMT) . . . . .	43
4.1.1	Architecture . . . . .	44
4.1.2	Montimage Monitoring Tool (MMT) Correlation Engine . . . . .	45
4.2	MMT Extension with the Quality of Experience Evaluation Framework . . . . .	47
4.2.1	$L$ -equivalent Derivation Algorithm . . . . .	48
4.2.2	Quality of Experience Computation Algorithm . . . . .	49
4.3	Applications of the Algorithms . . . . .	49
4.4	Conclusions . . . . .	50
<b>5</b>	<b>Quality of Experience Framework Validation</b>	<b>53</b>
5.1	Implementation of an Over-The-Top Emulation Platform . . . . .	54
5.1.1	Over-The-Top Emulation Platform . . . . .	54
5.1.2	Perturbed Video Generation Tool . . . . .	56
5.2	Preliminary Refinements . . . . .	60
5.2.1	Over-The-Top (OTT) Model Enhancement . . . . .	60
5.2.2	Enhancement of the Quality of Experience model . . . . .	61
5.2.3	Evaluation of the Videos . . . . .	64
5.3	Results and Model Validation . . . . .	66
5.3.1	Context Variables Validation . . . . .	66
5.3.2	Quality of Experience (QoE) Model Validation . . . . .	71
5.3.3	QoE Model Testing . . . . .	76
5.4	Conclusions . . . . .	76
<b>6</b>	<b>Static Analysis of an Over-The-Top Service</b>	<b>79</b>
6.1	Introduction . . . . .	79
6.2	Static Analysis Algorithm . . . . .	80
6.3	Analysis of a Real Over-The-Top Service . . . . .	82
6.3.1	Experimental Configuration . . . . .	82
6.4	Static Over-The-Top Model Analysis . . . . .	84
6.4.1	Linear Quality of Experience Model . . . . .	84
6.4.2	Expanded Quality of Experience Model . . . . .	85
6.5	Conclusion . . . . .	88
<b>7</b>	<b>Conclusion</b>	<b>91</b>
7.1	Perspectives . . . . .	93
	<b>Bibliography</b>	<b>95</b>
<b>A</b>	<b>beIN Sports Connect Service Description</b>	<b>103</b>
A.1	Presentation . . . . .	103
A.2	Service Options . . . . .	103
A.3	Technical Details . . . . .	104
A.4	Service Operation . . . . .	104

---

---

<b>B Javascript notation for the Extended Finite State Machine <math>M_3</math></b>	<b>107</b>
B.1 Representation of the Extended Finite State Machine $M_3$ . . . . .	107
B.2 Computed paths for the Extended Finite State Machine $M_3$ . . . . .	109
B.3 Augmented paths for the Extended Finite State Machine $M_3$ . . . . .	109
<b>C Video Generation Tool</b>	<b>111</b>
C.1 segment.sh . . . . .	111
C.2 concat.sh . . . . .	115
<b>D Validation Survey</b>	<b>117</b>
<b>E Version française abrégée</b>	<b>119</b>
E.1 Chapitre 1 : Introduction . . . . .	119
E.1.1 Motivation . . . . .	119
E.1.2 Problématique . . . . .	120
E.1.3 Objectifs et planification de la thèse . . . . .	120
E.2 Chapitre 2 : État de l'art . . . . .	121
E.2.1 Les services Over-The-Top (OTT) . . . . .	121
E.2.2 Les concepts de qualité . . . . .	124
E.3 Chapitre 3 : Système d'évaluation de la QoE basée sur les machines à états finis étendues . . . . .	126
E.3.1 Préambule – Machines aux états finis étendues . . . . .	126
E.3.2 Système d'évaluation de la QoE . . . . .	127
E.3.3 Conclusions . . . . .	129
E.4 Chapitre 4 : Implémentation du système d'évaluation . . . . .	129
E.4.1 L'outil de monitoring de Montimage (MMT) . . . . .	130
E.4.2 Extension de l'outil OTT avec le système d'évaluation de la QoE . . . . .	130
E.4.3 Application des algorithmes . . . . .	131
E.5 Chapitre 5 : Validation du système d'évaluation de la QoE . . . . .	131
E.5.1 Mise en œuvre d'une plateforme d'émulation OTT . . . . .	131
E.5.2 Raffinements préliminaires . . . . .	132
E.5.3 Résultats et validation du modèle . . . . .	134
E.5.4 Conclusions . . . . .	135
E.6 Chapitre 6 : Analyse statique d'un service OTT . . . . .	135
E.6.1 Algorithme d'analyse statique . . . . .	135
E.6.2 Analyse d'un service OTT réel . . . . .	136
E.6.3 Analyse statique du modèle d'un OTT . . . . .	137
E.6.4 Conclusions . . . . .	137
E.7 Chapitre 7 : Conclusions et perspectives . . . . .	137
E.7.1 Perspectives . . . . .	139

---



---

# Chapter 1

## Introduction

### Contents

<b>1.1</b>	<b>Motivation . . . . .</b>	<b>1</b>
<b>1.2</b>	<b>Problematic . . . . .</b>	<b>3</b>
<b>1.3</b>	<b>Scope and Plan of the Thesis . . . . .</b>	<b>4</b>
<b>1.4</b>	<b>Publications . . . . .</b>	<b>6</b>
<b>1.5</b>	<b>Publications non-related with the main topic . . . . .</b>	<b>7</b>

---

Nowadays the Internet has been turned into a reliable and effective platform to deliver value to the customers. In the last few years, Internet accesses have enjoyed a raise in their adoption as the main communication channel for the users. This has lead to a democratization process of the Internet access, allowing the users to have connectivity not only at their work posts, but also at home and even in their pockets using *smartphones*.

As this growth occurred, new companies saw in this platform a rich land to establish new multimedia business opportunities without investing large amounts in deploying expensive delivery networks required to distribute the content to the final user. This fact was supported by the development of two factors of the public Internet. On one hand, by using a constantly expanding network as the delivery system, the amount of potential customers of the proposed service grows every day, allowing the business to grow its impact at the same time. On the other hand, the development of high-speed links allows to offer scalable services that can adapt to the different conditions of a best effort-based network. However, an apparent contradiction can raise with this last fact.

### 1.1 Motivation

Despite the huge efforts to offer fast connections to a high number of people, since its beginnings the Internet was developed as a best-effort network. In simple words, the delivery of the data (originally messages) is not ensured by the network itself, offering instead the best effort to deliver the data with no errors in the less possible time. Among the possible problems, the interconnection of network introduces nodes that represent possible failure points on the delivery chain: each node can drop packages – or parts of them – due to congestions, packets can be reordered or they can even be corrupted, which invalidates the information contained in it.

---

These drawbacks are the main disadvantage when delivering services to users that expects high quality levels.

Services like telephony and TV were designed to have an amount of resources reserved to ensure the delivery of the content being transported. In the former case, these resources are reserved at the time the call is performed, and they are maintained up to the end of it. In the latter, a part of the magnetic spectrum is reserved for TV broadcasting, which allows a huge amount of viewers of the same TV channel at the same time. As stated before, both technologies were designed having this behavior in mind, which makes easier the quality control on them. When moving to similar – or even the same – services transported over an unreliable network, the ensured channels premises no longer hold.

As the new companies started to offer paid services on the Internet, quality assurance started to be an issue. Services (specially multimedia) now have to thrive for resources that are always busy, sometimes interfering with other services and resulting in “bad quality” services which do not satisfy the users. For example, traditional telephony services are distinguished for having almost no delay (even on international calls), no cuts on the conversation and low quality voice. The latter is achievable with almost no effort on real Internet networks, however the rest of the requirements can be hard to emulate on a best-effort infrastructure: packet loss can produce cuts in the conversation and congestion can augment the delays of up to points where a fluid conversation cannot be held. When considering video-based services (like TV or even, Video On Demand (VOD) services), the scenario is not that promising.

TV services usually transport audio at higher quality than telephony services and, in addition, moving images that need to be synced with the audio. These features dramatically raise the amount of data that need to be transported, which translates to higher bandwidth requirements. On the contrary, even when TV services require some degree of immediacy to deliver the content, it is not as critical as telephony, since the communication is done only in one way: from broadcasters to the audience. This requirement is not even present for VOD services, where pre-recorded videos are being delivered to the spectator when they require it. In both cases delivering audiovisual services over Internet poses important challenges on ensuring quality to the customer. Firstly, high visual and audio quality imposes high restrictions on the effective bandwidth of the connection between the user and the service provider, which cannot be guaranteed when multiple users share the same best-effort network. Secondly, the amount of data being transferred raises the probability to lose parts of them, which might result in cuts on the service and augmented delays. Finally, other impairments of the networks like the delay jitter might affect the synchronization between the audio and the video of the stream, which drastically decreases the quality perceived by the user [15], [23], [46], [60].

Even when the network conditions are the main factors impacting the quality noticed by the user, it is not the only dimension that impacts the user when using a service. Nowadays more and more services are being offered through Internet impacting a large amount of people with competitive prices compared to their traditional counterparts offered by telecommunication companies. Although these competitive prices are, in general, lower than the traditional competence, granting high levels of quality and revenues at the same time, can be a huge challenge for the service providers.

The main disadvantage of the Internet-based service providers lies on their main fortitude: the Internet. It is true that this platform lowers the deployment costs in order to deliver value, at the cost of losing the power of control the delivery process. In other words, Internet-based services transfer the responsibility of delivering their data over networks that are usually managed by telecommunication companies. Moreover, these last actor do not take any revenue

---

when transporting high amounts of data over their network, which usually raises the operational costs of the Internet access. In this scenario, the telecommunication companies have the power to limit the usage of these alternative services that trim their revenues. This represents the main motivation to find a solution to conceal both markets, aiming to minimize the impact on shared Internet infrastructure and maximizing the revenues of the companies involved in the whole distribution chain.

Considering all these effects, it is important to provide high levels of quality even on scenarios of high impairments, aiming to raise the satisfaction level of the user. In this way, it will be possible to keep customers more committed with the service and willing to recommend the service to other potential customers. By applying proper quality assurance techniques to an Internet-based service, it will be possible to deliver a reliable offer and meet the user's expectations, rising at the same time the revenues of the business. It is important then to supply the technical staff with the proper technology – in form of quality models and their respective monitoring tools – intended to monitor their streams, optimize their operational efforts and deliver a high-quality service to the customer.

## 1.2 Problematic

The quality concept has evolved as the Internet started to grow. At the beginning, the quality was associated with objective metrics under the name of Quality of Service (QoS). As an example of this last fact, typical Quality of Service (QoS) metrics involve the measurement of network-related parameters such as packet loss and delay of the streams. If we consider the services usually offered in the first era of the Internet – or the *Web 1.0* – these measurements are good enough to ensure the correct behavior of the services offered. For example, web pages that do not offer interactive content are usually downloaded once in the client's browser and cached for an accelerated later use. When the *Web 2.0* era started, the assumptions that validated this behavior did not hold any longer.

The *Web 2.0* introduced rich content to the Internet, taking interactiveness, audio and video to the websites. This last type of content made the hosting web sites more dynamic and demanding not only for the end-user hardware but also for the network transporting the content. With the raise of blog sites – and later video blogs – caching content was not possible any longer. The client has to contact the web site each time the user access it, asking for the updated content since their last visit. In the case of rich, multimedia content, not only the server and the client had to embrace huge changes in order to reproduce the content, but also the network was forced to transport a significantly higher amount of data than before.

As stated before, multimedia content breaks the paradigms that usually governed the network; huge amounts of data now traverse the network, and QoS parameters can now impact the service in unpredictable ways. For example, delay can lead to unsynchronized delivery with respect to live broadcasting – this effect can still be observed in live streamings. In the case of packet loss, it can produce freezings on the video and/or the audio of the service being streamed, generating the same desynchronization effect and requiring to skip parts of the stream in order to keep the liveness of the service. In any case, QoS are not good metrics to predict the effects on multimedia streams, since they strongly depend on other higher-level objective metrics – such as the encoding bitrate of the audio or video, the compression level, among others – and even other content-related variables, such as the type of the content transported or the amount of movement of the video.

Facing these challenges, it was required to provide a metrics that provides a wide view

---

over these effects and, at the same time, express the quality concept from the point of view of the user. The *Perceived QoS* is a concept that was developed by the needs to incorporate subjective dimensions into the quality analysis. Later, this concept evolved in the general term *Quality of Experience (QoE)*, which formally defines quality from the point of view of the user. Even when this definition recognized the multidisciplinary approach of the QoE, its analysis was focused on how do the QoS-related metrics affected the QoE value, i.e., the quality perceived – *experienced* – by the user. To this goal, the literature offers a plethora of studies that analyze these relationships, which then are used to elaborate quality models to predict the value of the QoE. The main goal of this type of studies is to provide the services providers with the appropriate tools to monitor their network and predict the experienced quality, aiding them to prevent bad quality scenarios on the services.

The QoE models proposed on the literature are useful to satisfy the basic needs of quality monitoring, however they lack of a multidisciplinary analysis of the QoE. In particular, user's expectations are influenced by economical, business-related decisions – such as the price of the service – and, therefore, the quality they expect from the service. Although the QoE was defined as a multidisciplinary concept – being business decisions one of the multiple factor influencing it – the literature does not provide deep analyses about how these variables impact the QoE.

The work presented here aims to expand the QoE studies by integrating the business dimension into the analysis of the QoE from a multidisciplinary point of view. In other words, this work expands the state of the art by providing a QoE analysis from both the objective and the business dimensions of the QoE, in order to provide broader quality models to the service providers.

### 1.3 Scope and Plan of the Thesis

In the last paragraphs the multidisciplinary nature of the QoE concept has been exposed and explained. Its analysis requires a broad, integrated understanding of the factors influencing it. This work aims to analyze how do objective and business variables impact as a whole instead of providing a separate analysis of each variable. This novel approach focuses on four main contributions to provide a complete analysis of the QoE.

At first, Chapter 2 presents a review of the literature related with the main topic treated in along this work. First, the concept of OTT services is presented, stating their principal distinction points, the innovation they brought to the whole Internet ecosystem, and the principal dispute they have with the telecommunication companies, which has not been resolved yet. Second, two principal concepts are introduced: the *QoS* and *QoE*. This section also points out the evolution of both concepts and how the QoE concept was coined in order to introduce the perceptual dimension on the quality concept. In addition, this section also analyzes the principal influence factors of the QoE, and which are the principal challenges that need are still open. The holistic approach of the QoE concept is one of the aforementioned challenges, being the inclusion of business dimensions a specific goal of the research. In this sense, the last section introduces the Quality of Business (QoBiz) concept, mentioning the most iconic efforts done to integrate business analyses into the QoE concept. This work tackles some of the open challenges by expanding the QoE domains with the inclusion of business and subjective parameter into the analysis.

The inclusion of business parameters on the QoE requires a proper understanding and formulation about how these two dimensions contribute on the final QoE value. To this end, the first contribution presented in Chapter 3 is focused on the proposal of a novel QoE evaluation

---

framework, aiming to provide a comprehensive methodology to evaluate the quality of Internet-based, multimedia services. This framework is based on modeling the interaction between the user and the service from the point of view of the user by using the mathematical formalism of Extended Finite State Machine (EFSM). These models have the advantage of retaining which are the possible traces or use cases, preserving at the same time which are the decisions the user took while using the service. With this information it is possible to use a proper QoE model that considers the user's decisions – including economical decisions – in order to obtain a QoE value. As an initial preliminary analysis, the usage of EFSM on the framework allows backtracking which are the decisions – in forms of transitions of the mathematical model – that lower the quality the most.

The next step on proposition of innovative technologies is the implementation of the approach in order to provide a concrete set of technologies ready to be used on real services. Chapter 4 presents an implementation of the previously described QoE evaluation framework. Although it is possible to implement the approach from the scratch, this work takes advantage of the EFSM emulation and monitoring capabilities of the MMT software by implementing the framework as an extension of this tool. In particular, two main algorithms form the principal implementation of the framework, computing the traces contained on the EFSM model and its respective QoE value respectively. This particular implementation can benefit from the MMT capabilities to express EFSM and use its Deep Packet Inspection (DPI)/Deep Flow Inspection (DFI) features to obtain the network and business parameters to feed both algorithms. In addition, it is shown how to use both algorithms in combination in order to provide a complete solution to predict the QoE value in an online manner for live users of the service.

The implementation prototype described above requires proper validation before it could be used with real software, presented in Chapter 5. This validation process also comprises the finding of a proper QoE model which maps the objective and business parameters into a single QoE value. This is done by implementing a web-based multimedia platform that mimics the behavior of a real service, which is used having two objective in mind. On one hand, it is used to determine the impact degree of the business variables by means of a survey applied to real users. On the other hand, the platform emulated the behavior of a real multimedia service, showing perturbed videos and asking for the quality value to the user. With this information it is possible to generate a proper QoE model to compute the quality value considering the impact factors under analysis. Once this process is completed, the implemented prototype is ready to be applied on real multimedia services to predict the quality values in combination with the monitoring tools of MMT.

Chapter 6 exposes how the model of the multimedia service can also be used to perform a static analysis of the service in early stages of the development. Since the formal model is built from the point of view of the user, it has the ability to retain which are the scenarios a real user might face. In order to aid the development of the process and determine how the business decisions impact on the final QoE value, the framework can be used before launching the service, obtaining a profile of the possible QoE values of the service. To this end, preliminary versions of the service model can be used with the framework, which computes the possible scenarios for a user and – using a proper quality model – their respective QoE value.

Finally, Chapter 7 presents the general conclusions and the perspectives for the technologies proposed and implemented in this work.

The set of contributions described above aim to provide a complete and integrated QoE computing solution to the service providers. Even when in this document a concrete implementation is proposed as an extension of the MMT software, the formulation of all the contributions are



general enough to be implemented on any other suitable software, as long as it can be used in combination with monitoring tools that can extract information from the interaction between the user and the service.

## 1.4 Publications

The works previously mentioned have been published in different journals, conference proceedings and workshops. A complete list of the publications related with this work is the following:

- [1] D. Rivera and A. Cavalli, “Qoe-driven service optimization aware of the business model”, in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Mar. 2016, pp. 725–730. DOI: 10.1109/WAINA.2016.105.
- [2] D. Rivera, N. Kushik, C. Fuenzalida, A. Cavalli, and N. Yevtushenko, “Qoe evaluation based on qos and qobiz parameters applied to an ott service”, in *Web Services (ICWS), 2015 IEEE International Conference on*, Jun. 2015, pp. 607–614. DOI: 10.1109/ICWS.2015.86.
- [3] D. Rivera, A. R. Cavalli, N. Kushik, and W. Mallouli, “An implementation of a qoe evaluation technique including business model parameters”, in *Proceedings of the 11th International Joint Conference on Software Technologies*, 2016, pp. 138–145, ISBN: 978-989-758-194-6. DOI: 10.5220/0006005001380145.
- [4] D. R. Villagra and A. R. Cavalli, “Analysis and influence of economical decisions on the quality of experience of ott services”, *IEEE Latin America Transactions*, 14, no., pp. 2773–2776, Jun. 2016, In Spanish, ISSN: 1548-0992. DOI: 10.1109/TLA.2016.7555253.

The publication [2] presents the new quality evaluation framework, introduces the required concept of EFSM and derives the corresponding methodology to be applied when analyzing online, multimedia services. In addition, this publication shows how this methodology is intended to be applied by running it on a real service, which is used to analyze its QoE in different scenarios.

The proposed methodology is formalized in publication [1], where the outline of three main algorithms are proposed. This work identifies which are the parts of the framework that can be performed in an automatic way, which allows implementing this approach either as completely new or as an extension of an existing tool. In addition, this paper also sets the base of an online tool that uses these algorithms in order to monitor a real OTT service, aiming to predict the quality of future scenarios and avoid the ones that might lead to unsatisfied users.

The framework and its corresponding methodology is then implemented by three algorithms introduced in the publication [3]. The principal goal is to provide a formal way to implement the procedures required to analyze any OTT service by using the proposed framework. These three algorithms were designed to automatize the three main procedures of the methodology, namely *l*-equivalent computation, QoE augmentation and computation of the total number of scenarios of a model. These algorithms are implemented as an extension of the MMT software and applied on a real OTT service in order to determine how do the QoE values are spread across all the scenarios. With this information, the service providers can extract conclusions about the service even during early stages of the development, aiming to aid them to focus their efforts by ameliorating the low quality scenarios.

Finally, and as a way to summarize the potential of the proposed framework, the whole approach is exposed in the article [4]. In this publication, the main advantages and how the

approach can be applied are the main points presented, which aims to enhance the visibility of the approach in new, emergent markets.

## 1.5 Publications non-related with the main topic

In addition with the publications previously mentioned, other works have been published during the development of the this work that are not related with the main topic of the research presented in this document.

- [5] S. Blasco, J. Bustos, and D. Rivera, “Detection and containment amortization udp sockets for multithreading on multicore machines”, *IEEE Latin America Transactions*, 14, no., pp. 2853–2856, 2016.
- [6] D. Rivera, E. Achá, J. Bustos-Jiménez, and J. Piquer, “Analysis of linux udp sockets concurrent performance”, in *Proceedings of the Chilean Workshop on Distributed Systems and Parallelism (WSDP)*, 2014.
- [7] D. Rivera, S. Blasco, J. Bustos-Jiménez, and J. Simmonds, “Spin lock killed the performance star”, in *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*, IEEE, 2015, pp. 1–6.

In general terms, these publications are related with the performance analysis of the User Datagram Protocol (UDP) sockets in Linux Operating Systems. It was found that concurrent accesses to UDP sockets – in particular, concurrent calls to `read` on the socket – do not grant a performance boost on the applications that use threads with sockets. In other words, any application that uses multiple threads to concurrently read on UDP sockets does not increase the overall performance of the application.

In [6] the initial analysis of the problematic is shown, studying the origins of the serialization of concurrent accesses. This work states the problem of the non-scalability on the Linux kernel, whose implementation of the sockets do not allow concurrent accesses to this data structure. In addition, this work also proposes a naïve solution, tested as a prototype implemented in a user space application.

A detailed analysis of the UDP performance is exposed in [7]. In this work, all the possible point of failures are identified and tested. First, the performance of the Linux sockets is compared with other kernel-provided structures: FIFO queues (“named pipes”), `/dev/null` and `/dev/urandom` virtual devices, and UNIX sockets. This test shows that, despite using the same reading primitive, the performance of the Linux Internet sockets degrades as long as more threads are introduced, which is not observed in the virtual devices. Finally, this work also presents a profiling study of the Linux Internet Sockets synchronization scheme, identifying a spinlock as the reason of the serialized accesses to the UDP socket.

Finally, the issue is further analyzed in [5], measuring experimentally the collapse of the locking system due to the amount of concurrent accesses. To this end, a complete profile of the memory accesses is performed, establishing that the underlying reason of the scalability collapse is due to a resource contention on the hardware communication channels. This work also proposes an optimization of the naïve solution proposed in [6], which aims to minimize the contentions at hardware level. This is based as a Linux module that redistribute the packets in different UDP sockets following different strategies, which aims to minimize the communication between tasks running on different CPU cores. The experiments showed that this approach is a

---

competitive solution compared with the recently-introduced `SO_REUSEPORT` feature of the Linux Kernel.

---

---

# Chapter 2

## State of the Art

### Contents

<b>2.1 Over-The-Top (OTT) Services . . . . .</b>	<b>9</b>
2.1.1 The Beginning of Time: Telecommunication and Audiovisual Media . . . . .	9
2.1.2 The Internet Era, and the Raising of Internet-based Services . . . . .	10
2.1.3 The “Invasion” of The Content Market Land . . . . .	11
2.1.3.1 Video-based OTT services . . . . .	13
2.1.4 The “Unfair Competition” War . . . . .	15
2.1.5 The “Network Neutrality” Treaty . . . . .	16
<b>2.2 The Quality of Service (QoS) and Quality of Experience (QoE) concepts</b>	<b>17</b>
2.2.1 Quality of Service . . . . .	17
2.2.2 Quality of Experience . . . . .	20
2.2.2.1 Influence Factors of the Quality of Experience . . . . .	21
<b>2.3 The Quality of Business (QoBiz) concept . . . . .</b>	<b>23</b>

---

## 2.1 Over-The-Top (OTT) Services

For most of the daily users of Internet, names like YouTube, Netflix, Spotify, Whatsapp and others services alike are not member of the unknown brands of the Web. However, most of the users do not know how do these services gained the popularity they have nowadays. In order to understand and contextualize the actual popularity of these service providers, it is required to know how they and their markets evolved across the years.

### 2.1.1 The Beginning of Time: Telecommunication and Audiovisual Media

Back in the 80’s and early 90’s, the communications needs for both people and companies were completely satisfied by telecommunications services: immediate transmission of data was done by voice using telephone, while document transfer was done via either old-fashioned postal service or fax [22]. During these years, the needs of communications were not exigent, and the market was satisfied with decades-old technologies whose ghosts still remain nowadays [22].

---

In both cases of the aforementioned services – telephony and fax – the technologies that supported their working was expensive and scarce and therefore controlled by a few big companies on each country [37]. In addition, these technologies were built upon strong assumptions that limited the extensibility of the network and defined the range of value-added products that could be implemented over them [37].

In the meantime, the content media distribution was performed either via physical ways (like cassettes, video tapes or CD-ROMs) or via broadcast networks that rely on technologies that use similar assumptions compared to the ones used with telephony networks (as, for example, television and radio). In both cases, the technologies used did not appear to share any common characteristics despite being offered by similar companies – TV cable companies were usually subsidiary companies of the telecommunication companies that offered telephony services. This last fact is due the amount of infrastructure required to distribute the content to the final user, which was only possible by using expensive cable-based networks similar to the ones used with telephones. These networks were usually deployed in a joint effort with the big telephone companies in order to avoid the high-costs of this infrastructure.

Back in those days, there were two different, clearly defined markets for investors. On one hand, the telecommunications market, based on the exploitation of the telephony-based networks, offering services like landline telephone, beepers and, lately, mobile phones. On the other, the broadcast market based its revenues in three main models: *ads-based* funding (usually exploited by public TV services), *subscriptions* (as, for example, cable TV operators) and *Pay-Per-View (PPV)* (or similar services that require a single-time payment to access the distributed content).

The aforementioned scenario showed a stable yet unsustainable business model against the rise of new technologies: a technology that does not require expensive handling of the content transported and could define new, flexible ways to transport different services over the same infrastructure... at the same time [37].

### 2.1.2 The Internet Era, and the Raising of Internet-based Services

While most of the communications industry was running over hundred-years old technology, the research oriented its efforts to provide a new generation methodology to distribute the media content. Isenberg describes this technology as the *stupid network* [37], due it completely changed the way data should be transported.

Under the telephony paradigm the network is the core of the system, which is in charge of allocating the required resources when the user needs to use it [37]; this is called *connection-oriented* network [48]. Following this model, all the logic of the service relies on the network itself, leaving a small part of the work to the terminals: the decodification of the data transported [48]. For this reason, the deployment of such networks is expensive, since intelligence-loaded telephony commutators are hard to build and maintain. In addition, the telephony service was designed to transport voice using hard assumptions; voice was codified using certain codecs and bitrates for this, therefore, services that used to run over this network required more logic in order to adapt its requirement to the ones imposed by the network [48].

On the contrary, the new network paradigm focused its efforts on inverting the model, which puts the data in the center of the network [37]. Centering the attention on the data, allowed multiple simplifications of the implementation of the network. First, all the logic of the network reside on the terminals, which allows the network to focus only on distributing the data [48]. Second, the data knows where it should be delivered, relaxing the dependency between the type of

data being carried and the network [48]. Last but not least, the use of digital signals to transport the data permitted the expansion of the capacities of the network, allowing transporting multiple services at the same time over the same infrastructure [37].

The development of the first major effort of such network was performed by the Advanced Research Projects Agency (ARPA), depending of the U.S. Department of Defense. The *ARPANET Project* was conceived in the late 60's in order to provide a *packet switching* network which implemented the data-centered communication methodology mentioned before [48]. At the beginning, ARPANET was only used with military purposes (in order to provide a "survival system that contained no critical central components," [22]) but later it was extended to be used with research purposes.

While this network allowed communicating hosts using the same underlying technologies, the development of this network was directed to the intercommunication of hosts using multiple technologies seamlessly. This idea set the base for the definition of the *internetworking* concept, also called *internet* [22], [48], [61]. Having this concept in mind, the Transmission Control Protocol (TCP) communication protocol was proposed as an abstraction layer to allow interconnecting different physical networks using to the same logic, packet-switch based network. The era of Internet is just starting.

It did not take a long time to have a more open, exploitable version of this network, which promised low cost and easy distribution of data no matter where the data wanted to be transferred. This last fact attracted the attention of the business industry who started to see a great potential on this network, migrating from telephony-based technologies to a more flexible solution to transfer its data [37]. This new implemented network was so flexible that even allows the implementation of Telephony service over it at lower costs than the offer of traditional telecommunication companies. The latter reacted to this menace by asking the government to rise the access prices to the Internet and even banning Internet Telephony [37]. This dispute represents a precedent in today's war on content distribution.

Even though the high costs of implementing this network made its uses only affordable to big companies who used it with business purposes, a mass use of this technology did not take so long. The prices of implementing and operating the network decreased during the early 90's, and each day more and more home users were connected to the Internet, and new companies started to see new business opportunities to deliver value using this infrastructure [37].

### 2.1.3 The "Invasion" of The Content Market Land

The natural reaction of the telecommunication companies was to take control of this new service, taking part on the new market of the Internet access. The principal telecommunication enlarged their offers by adding the Internet connection service. At the beginning, this service was offered in form of *Dial-up* connections and then migrating to Digital Subscriber Line (DSL) services, being both transported physically over the telephony network [22], [48]. Other companies started to develop coaxial networks, enlarging even more the bandwidth offered and allowing to transport different services over the same cable [22], [48].

At the same time, telecommunication operators started to extend their services by adding cable television and Internet access to their telephony service. This new offers were – and still are – called *Triple Play*, offering a cheap, integrated communications solution for end-users [62]. However, the Internet expansion was fast during the 1990s and the 2000s [16], raising the adoption of Internet technologies at home from 42% and 32,7% in 2005 for Europe and The Americas respectively, to 67,8% and 44,4% in 2010 [45].

---

As the Internet access became more widespread, the available bandwidth for data transfer also grew. This allows the transmission of even more complex data over the public Internet. With this in mind, a new industry that took advantage of this flexible implementation was born; the Internet is now seen as a way to deliver rich content to the final user.

*The Web 2.0* was the name for this new era of the Internet in which this technology started to be used as a common platform to share content [55]. Popular sites like Google, Amazon and Wikipedia are simple examples of the new business paradigm that the Internet founded, but there were other corners that started to be exploited.

In May 2005, the well-known site YouTube was born, offering to the Internet users a public platform to publish videos [21], [92]. This site marked the start of a change on multimedia distribution as it was known up to this point: YouTube proved that the stupid network could be used to deliver audiovisual content to the end user at low cost. This effect was predicted in 1997 by Isenberg in the famous article *The Rise of the Stupid Network* [37].

As the time passed, telecommunication and media distribution companies coexisted in these “different” market lands. On one side, telecommunication businesses did not make any changes on their offer, even when similar Internet-based services started to be offered for free. As an example of this last fact was noticed by Telefónica, determined to keep charging the customer per Short Message Service (SMS) rather than moving to more “innovative” pricing schemata. The effects of such business decision were reported in 2014, observing the starting of a stagnation or even a decreasing phase on the SMS usage [72]. As stated in the same manifesto, this service started to be superseded by their free, Internet-based options such as Whatsapp and Facebook Messenger.

At the same time, other companies also started to exploit the Internet as the base. Netflix was born as a film rental company in 1997, offering the possibility to rent movies over Internet delivered by postal service [36]. The main change in this company begun in 2007, when they decided to reinvent their delivery method by using Internet instead of the postal service to deliver audiovisual content to their customers. The easiness of use and their low price subscription schema attracted more customers, which started to see in this type of services a real substitution to the traditional TV service.

Finally, companies like Skype offered the *Voice over IP (VoIP)* service and even *Telephony over IP (ToIP)* at much lower prices than the telecommunications companies, the former being offered for free between users of the same application. Starting from now, the Internet has grown enough to show its full potential to deliver a plethora of services without implementing complex, service-specific networks. These are the main features that describe what an *Over-The-Top (OTT) Service* is. A formal definition has been given in many publications in the literature, however a suitable, enclosing definition has been given in 2006 by Green et al.: “Over-the-top (OTT) services is the buzz-expression for services carried over the networks, delivering value to customers, but without any carrier service provider being involved in planning, selling, provisioning, or servicing them – and of course without any traditional telco booking revenue directly from them.” [29]

Despite the fact that this definition is broad, it includes a bunch of services that share Internet as the main distribution method. In [35], OTT services are classified according to the type of service they offer, where a bunch of examples can be found for each category:

- *Community*: “Social networks” services can be found here, like Facebook, Twitter and Instagram,
- *Technology*: Cloud computing and online storage services are classified here, as Dropbox,

Amazon Web Services and Google Cloud Platform,

- *Productivity*: Online office solutions are members of these services like Google Apps, Microsoft Office Online and Evernote,
- *Communication*: All the text-, audio- and video-communication services can be included in this category, as for example Skype, Whatsapp and Facebook Messenger,
- *Music*: Services that allow the user to access music libraries in an online-basis are members of this category like Spotify, Apple Music and Amazon Cloud Player,
- *TV and Video*: Finally, multimedia services based on TV and video are also part of the pffer of OTT services as, for example, YouTube, Netflix, Hulu and Amazon Prime Video.

Although most of the services mentioned before are not direct substitutes of the ones offered by telcos, the last three categories can be considered as real replacements. The OTT communication services directly strike the revenues of the telcos by offering replacement services to the SMS service (with Whatsapp, Telegram and Facebook Messenger), the voice call services (with Whatsapp, Facebook Messenger and Skype) and even expanding the traditional offer with video calls (with Facebook Messenger, Apple's FaceTime and Skype). The expansion is even wider if we consider that some OTT companies offer access to music (with Spotify, Apple Music), VOD (with YouTube, Netflix) and even live TV (with FoxPlay, HBO Go). These last types of multimedia services are the ones that impact the network the most, since the amount of data and the stress on the network increases as long as different media streams and immediacy are added to the service. Even though the OTT analysis can be extended to all the categories aforementioned, the technical and business challenges are related with the substitute services that confront both telcos and OTT service providers.

Since this work focuses on the video-based OTT services, the next section provides a deeper description of this type of multimedia services.

### 2.1.3.1 Video-based OTT services

The first well-known OTT video service *YouTube* was born in 2005 [27], [92]. This service was conceived as an open, web-based platform to upload and share videos on the Internet. At the time of its initial development, this service was implemented using the Adobe Flash Video (FLV) to deliver the video, which allowed embedding the content into the web-based frontend [27]. The aforementioned technology was the de facto standard on the Internet to deliver rich content to the user. This is justified by the fact that Adobe's FLV technology relied on *progressive download* to reproduce the content.

Up to the launching of FLV, the videos were required to be completely downloaded before it reproduction started. FLV relaxed this requirement by allowing its reproduction while the download was still ongoing [9], [27]. This was achieved by a set of scripts that supplied the FLV data to the player at the same time it was being downloaded [27]. The popularity of the FLV technology can be explained considering the fact that it was designed to run with the Shockwave Flash (SWF) Player, which was already wide spread in the user's computers. The usage of Flash technology to embed videos on web pages was justified for the impossibility to deliver any multimedia content by using pure HyperText Markup Language (HTML) code, which up to the specification 4.1 did not support any mechanism to transport video and audio over HyperText Transfer Protocol (HTTP).



Without going any further, the initial release of progressive FLV technology supported multimedia streams using the *Sorenson Spark* codec for video and MPEG Layer 3 (MP3) codec for audio [10], both of them completely proprietary. In 2009, Adobe introduced support for the codec H.264 for video and AAC for audio into the FLV specification, as it offers better compression ratios and was emerging as the new format for many services [10]. In a similar way, in 2007 Netflix started to rise as a new Internet multimedia delivery company, offering VOD films to their customers. During its initial years, Netflix used a combination of Microsoft proprietary codecs for the video, until the development of a Microsoft Silverlight-based platform that is still used.

Following the example of these two successful companies and the media requirements of the Internet users, more and more web sites started to offer multimedia content. However the underlaying transport technology – HTTP – did not officially support these types of content. To this end, it was required to develop a new standard to support such demands.

In 2007, Opera Software proposed the `<video>` tag extension for the HTML language, showing a running example on a modified Opera Web Browser [54]. The development of the new HTML standard included support for both audio and video streams (a complete specification and usage manual can be found, for example in [58]). Nowadays, the HTML 5.0 specification – already implemented in all the modern web browsers – has support for transporting audio and video over Internet for web pages, but a more generic solution is required in order to deliver the content.

With this goal in mind, the International Organization for Standardization (ISO) has reviewed and published the Dynamic Adaptive Streaming over HTTP (DASH) format as the new multimedia de facto standard to deliver multimedia content over Internet [38]. This new technology was designed with the following main characteristics<sup>1</sup>:

- content is delivered using normal HTTP requests, which makes it suitable to effectively distribute content to the user even if firewalls filter other multimedia-oriented transport protocols as Real-time Transport Protocol (RTP);
- the development is done by using ordinary HTTP servers like Apache;
- support for Digital Rights Management (DRM) systems, which allow delivering copyright-protected content;
- support for HTML 5, meaning that this technology and DASH can be used together without interfering;
- agnostic to video and audio codecs, implying that the service provider is free to choose a suitable codec for the content being delivered;
- definition of *quality metrics*, which allows the user to report stalls, degraded quality, etc.

The DASH standard has been ratified by all the principal streaming technology providers united in the *DASH Industry Forum*<sup>2</sup>, whose members include Adobe, Microsoft, Samsung, and Netflix among others. However the actual offer of OTT multimedia services still does not make use a wide use of this technology even though has already been defined as the future of the Internet-based, multimedia stream delivery.

---

<sup>1</sup>This list is not exhaustive; it only points the main capabilities of the DASH standard.

<sup>2</sup><http://dashif.org/>

The definition in [29] clearly states that the base success of the new actors in the content distribution market is the penetration level of the Internet; they rely on this technology to deliver their value to the final customers, but they do not share any revenues with the companies deploying the underlying infrastructures. The latter have been ousted to serve as Internet access providers who offer a single product that replaces the rest of their offer.

In this scenario, the telecommunication companies started to see their revenues diminished by the newly arrived OTT companies [25], [80], which invaded the ‘Land of Content Distribution’ using cost-effective weapons. It is time for them to react against these menaces – by reinventing their offers – or lose the battle against their new competitors.

#### 2.1.4 The “Unfair Competition” War

This is not the first time the old-fashioned telecommunication companies have to face a change in their markets. As mentioned before, the expansion of the Internet already threatened this market with the introduction of ToIP. At that point, the reaction of the telcos was trying to forbid the entrance of ToIP companies or even impose charges on Internet access [37]. Now the menace is even bigger, threatening the telcos to transform into companies providing only access to Internet; “telcos might just become dump pipes for multimedia services,” [80] since the multimedia companies started to offer substitute services at lower prices[37].

The new market of multimedia services forces the telecommunication companies to react strategically while considering the actual situation. Multimedia services rise the operational costs of the network operated by the telcos while, at the same time, Internet access lowers the entrance barriers for new Internet-based service providers [80]. As mentioned by many authors and companies, the rising of OTT companies had a high impact on the revenues of the telcos, offering them an opportunity to reinvent their offer as they did in the past (see, for example [29], [72], [80]). However their reaction to these new challenges did not open them new business opportunities; “they have been caught napping in the face of the newest challenge to their revenues.” [80]

In a similar way as they did when ToIP service started to appear, the first reaction of the telecommunication companies was to declare a war against these new actors [37], accusing the OTT providers of unfair competition. The telcos justify this accusation on the fact that OTT companies use their networks to deliver their services without paying for them [80]. This represents the core of the argument between these two principal actors; OTT players started to erode telcos’ revenues by using the infrastructure implemented by the telecommunication companies to deliver value. This last fact puts pressure on the network in form of high traffic raising the operational costs of telcos [49].

This dispute forced the telcos to take aggressive measurements to contain the growth of the threat. The first known case about an offensive against OTT provides was in the United States by Comcast and many others American Internet Service Provider (ISP). According to a study conducted by Netflix, some big Internet companies started to degrade the bandwidth of their subscribers for data-intensive services, as it is the case of Netflix [33]. Using the data of periodical measurements the OTT company performs, Netflix detected how the ISP were applying traffic shaping measurements, demanding Netflix to pay *interconnection* fees. From the point of view of the ISP, these charges are justified because video streams put pressure on a network that lacks of sufficient interconnectivity [33]. This case ended with a payment agreement between Netflix and Comcast, even after a failed attempt of installing Netflix servers within Comcast’s network infrastructure [65].

This opened the discussion about what is the role of each company in the Internet ecosystem, where three main actors can be recognized in the scene: OTT, ISP and the users. The former two are the big factions fighting for the revenues of multimedia services, while in the middle the user receives the aftermath of the clashes in form of poor access to the services. Should ISP provide a flat, impartial access to the Internet to their customers? Should they provide “differentiated” access to the services according to the type of the content the user access? These are some of the principal questions that still require a clear answer, however the clear definition of the domains of both principal actors is the base issue that needs to be resolved; the *Network Neutrality* can heal the open wounds of the war.

### 2.1.5 The “Network Neutrality” Treaty

The fight between OTT and ISP arose since the former started to pose a threat to the latter companies, who saw a menace in the revenues of the old-fashioned telephony and SMS services [72]. This menace was due to the fact that new companies started to offer similar services at reduced prices thanks to their reduced implementation costs using Internet. The eroding effect was amplified when these companies started to offer multimedia services over Internet; companies like YouTube or Netflix posed a high amount of load on the network of the telcos, which ended in a rise of the operational costs [65]. This last fact was the trigger to the telcos to demand a part of the revenues of the OTT providers and, putting pressure on them by degrading the access to these services. However, it is not clear if this type of pressure can be considered legal, since the ISP companies – usually providers of telephony and cable TV services – are supposed to provide to the user an access to the whole Internet regardless the content demanded. This rose the main question: should ISP know the content the user is accessing and provide different “qualities” depending on them? This is the base of the *Network Neutrality* debate.

Nowadays this debate has arisen in many countries, mainly triggered by abusive behaviors from the ISP. For example in 2011, KPN and later Vodafone blocked the access to VoIP and messaging applications on their 3G networks, asking for a supplementary payment to access these services at full quality. This precedent marked the beginning of the required legislation on Netherlands, converting this country in one of the first with network neutrality regulations [35], [82]. This type of regulations started to open the debate of the regulation of the access to Internet, which concerned not only the access to OTT services, but also to any content available on the network.

In general terms, a neutral network should be able to provide an non-restricted, non-differentiated access to any content available on the network, without any quality disruption introduced arbitrarily by the access provider [91]. This wide definition of a neutral network not only allows distributing content of new multimedia services over the Internet, but also opens the possibility to use the network to distribute illegal content. In order to cope with this, the neutrality has been implemented following different approaches across the countries.

In the United States, for example, net neutrality rules have been issued several times by the Federal Communications Commission (FCC), being all of them challenged by the telcos in order to maintain their dominance on the market [30]. Nowadays, the FCC has issued a set of rules that ensure “the end of paid prioritization and blocking and throttling of lawful content and services,” [35], [88] yet a complete net neutrality law is still under debate [30].

In the European Union, some countries have already started the discussion of these laws considering the example of The Netherlands and Slovenia who already have a net neutrality legislation [30], [90]. This discussion is being carried by the European Parliament, who has already

voted laws that forbid “the slowness of or blocking of Internet access except on the cases to enforce a court order, preserve network security or prevent temporary network congestion.” [90] However, these rules still allow the ISP to offer certain specialized network services with prioritized access – usually referred as *fast-lanes* connections – without degrading other services to do so [30], [90].

In general terms, *network neutrality* is still an ongoing discussion in many countries in the world. Although the efforts of the governments to start regulating the Internet access, in many territories the lack of regulation allows the ISP to arbitrarily degrade the access to complimentary services that might impact their revenues. From the point of view of the ISP, these types of measurements aim to protect their business and investments but at the price of lowering the satisfaction of the end-user.

A less aggressive strategy is the creation of *fast-lane* connections, where particular services are prioritized on the ISP networks, aiming to promote the usage of certain services over the rest of the offer. These types of agreements are based on the differentiation of the content being delivered, and they are usually charged directly to the third-party service provider [30]. This types of arrangements are still an open challenge in net neutrality discussion, whether it should be acceptable to charge for this prioritization or not.

The bottom line of this discussion has not been said yet, although network neutrality is starting to bring together all the participants of this market. Network neutrality has been a non-agreed treaty signed – implicitly – by OTT players and some ISP, although against the will of the latter, who search to stay on the telecommunications market at any cost.

## 2.2 The Quality of Service (QoS) and Quality of Experience (QoE) concepts

The implementation of Internet-based services proposed an additional challenge to the providers of these service: they had to face the fact that the delivery network does not provide any guarantee of delivering the data to the customer. At last, this problem can easily be translated into ensuring *quality* to the customers of these new providers, which is not trivial while using a best-effort network. This lead to the main question: *What do we understand by quality on Internet?*

### 2.2.1 Quality of Service

In order to provide an answer, it is required to consider the principal objective that was considering when designing the Internet. The development of this network was conceived in order to deliver small amounts of information – called *packets* – from the original to the destination host. The key feature is, as stated by Isenberg, is that the data is the core is this design; it know where it needs to be delivered regardless the nature of the physical link used to transmit it [37]. Having this in mind, any definition of quality for packet-switched networks should be centered on the capability of the network to deliver the information.

The idea behind this logic was to provide reliable, *measurable* metrics with the ability to determine the “level of service” supplied by the network. In the same way as the classical postal service, the first attempt to determine this was to measure, for example, the time required to deliver a single unit of information through the network. This fact stated the *Quality of Service (QoS)* concept for packet-switched networks. With this goal in mind, two principal approaches to define QoS can be found [28].

On one hand, the International Telecommunication Union (ITU) organization defined quality – focused on telephony systems – as: “the collective effect of service performance which determine the degree of satisfaction of a user of the service.” [39] In this same technical report, the ITU also clarifies that this term “is not used to express a degree of excellence in a comparative sense nor is it used in a quantitative sense for technical evaluations.” [39] In addition, the ITU also defines a complete QoS framework that shows the multiple factors that can impact the QoS, being the *network performance* indicators some of them [39]. Despite being broad, the principal issue with this definition is that it relies on the user perception to understand the quality, while introducing the concept of network performance to cover all the technical facts [28].

On the other hand, the Internet Engineering Task Force (IETF) provides a more specific definition of QoS “a set of service requirements to be met by the network while transporting a flow.” [19] As it was designed having the Internet – a packet-switched network – into consideration, this definition clearly states that *quality* should be focused on the capacity of the network to transmit the data. It is also clear to see that this definition is closely related with the concept of network performance proposed by the ITU [28].

Both approaches are compared and discussed in depth by Gozdecki et al. in [28], in which the authors propose a common quality model that relate the concepts defined in both conceptions. Figure 2.1 shows the QoS models presented and compared in [28]. In the ITU approach, QoS fulfills the user’s expectations, naming network performance any technical measurement that directly impact these perceived quality. On the contrary, the IETF classifies the QoS as purely technical, measurable metrics.

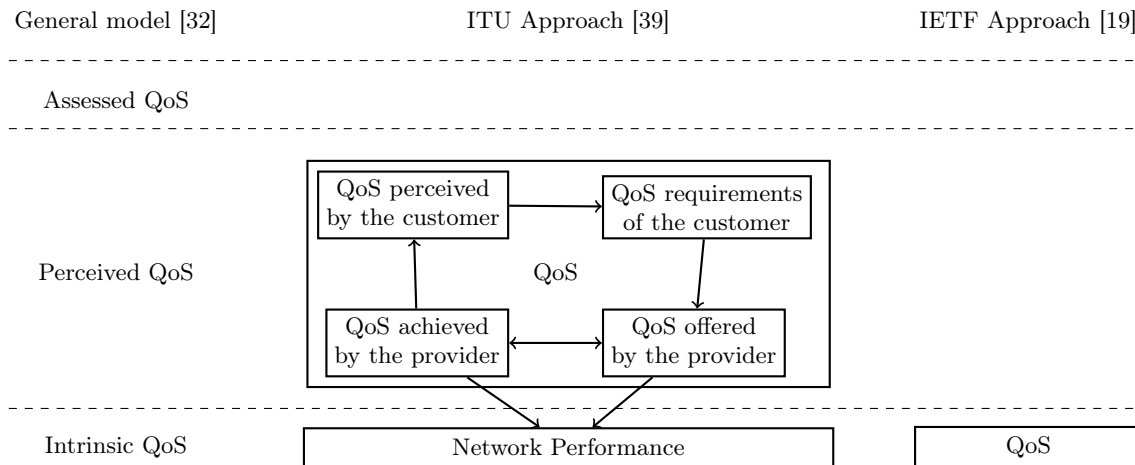


Figure 2.1: QoS model comparison presented in [28]

Later in 2007, the ITU superseded the old QoS definition, redefining the concept as the “totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.” [42] This definition matches the principal definition of *quality* given in the same document, where it is stated that the characteristics “should be observable and/or measurable. When the characteristics are defined, they become parameters and are expressed by metrics.” [42] Although this new definition still incorporates the user into the QoS concept, it clearly remarks that the quality should be a quantifiable measurement.

As seen in Figure 2.1, Gozdecki et al. put in comparison both approaches and introduce the

general QoS model proposed in [32]. The latter is formed by three main types of QoS, namely *Assessed QoS*, *Perceived QoS* and *Intrinsic QoS* [76], [84]. This distinction was made in order to provide a proper differentiation of the different domains where the quality concept has influence.

Following a bottom-up analysis, the Intrinsic QoS is the concept that matched the network performance concept from the ITU and QoS from the IETF. In this generic approach, the intrinsic QoS are the set of “service features stemming with technical aspects.” [32] From this point of view, any required quality level should be achievable by selecting the proper transport mechanisms and technologies that ensure the delivery of the data. In addition, this type of quality is compared and measured against the expected performance characteristics, which is not influenced by the user’s perception [28]. This is also pointed by Ghezzi and Dramitinos, identifying the technical parameters as one of the drivers of the future Internet [25].

As a way to extend the quality concept to include the user’s perception, Hardy introduces the concept of Perceived QoS, which aims to add the expectations of the user into the quality. As pointed by Gozdecki et al., the user compared the observed performance of the service and compares it with their respective quality expectations. The latter might be affected by user’s past experiences with similar services and other user’s opinions. This subjective dimension of the quality explains the fact that a service with similar technical intrinsic parameters can be evaluated with different quality values by two different users [28]. In Section 2.2.2 it is explained how the inclusion of subjective elements into the quality evaluation established the bases of new research topics.

Finally, Hardy also introduced the concept of Assessed QoS in order to represent the user’s evaluation that lead them to continue with the user or not. This quality level is mainly influenced by economical factors as the charging policy, the technical support among others [28].

In spite of the multi-layer view of the quality, the vision followed by the major portion of the telecommunication engineers was the IETF approach, understanding the *Quality of Service* concept as a synonym of network performance and *Intrinsic QoS*. In other words, *QoS* was seen as a set of *measurable, observable* indicators that characterize the ability of the network to deliver data [84]. The core concept behind these words was captured by Van Moorsel, defining *Quality of Service* as “metrics such as availability and performance.” [83] Following this logic, the efforts were centered in defining correct metrics that define the capabilities of the network.

The same study provided by Gozdecki et al. provides a minimal list of the QoS parameters that influence any Internet Protocol (IP)-based service [28]: *bitrate of transferring* – namely network throughput, *delay* of the packets, *jitter* – namely the statistical variation of the network delay, and *packet loss* rate (see, for example [76], [78], [84] to find commonly used QoS metrics). However, these are not the only technical metrics that can be impact the quality of an IP-based service.

IP networks are designed to provide an end-to-end connection that allows transmitting digital data through it. When these connections are used in addition with congestion control protocols, it is possible to consider the channel “more reliable” than one without these techniques. Considering this, other technical, “higher level” factors start to play an important role on the quality.

For example, in VoIP services, the use of different compression codecs, frame lengths and bit-rates can impact the quality of the transported audio (some works show these variables as, for example [18], [76], while a extensive study can be found in [63]).

Finally, as any audio-visual service requires a huge amount of data to be transported, therefore, the audio and images require to be compressed before being transmitted over Internet [76]. The compression process introduces losses on the stream that can impact the final quality of

the service. As a way to measure the impact of this process, a set of metrics have been defined that aim to measure the impact on quality based on the network packets, the bitstream (like the bitrate, the compression level, among others), picture-based (which extract visual information from the video, like distortion levels) or even hybrid metrics that combine the effect of the previous metrics [89].

Despite the fact that QoS can be defined at different levels, the concept has evolved to represent metrics that are measurable. In this sense where QoS is understood across this work, serving as objective, computable measurements that describe the technical characteristics of the service being delivered.

### 2.2.2 Quality of Experience

The initial formulation of the QoS concept by the ITU was proposed in order to include subjective aspects on the quality. Despite this effort, this concept evolved to represent the technical measurements that support the service, following the IETF approach. However, “technology-centric concepts like QoS<sup>3</sup> are not powerful enough to cover every relevant performance aspect of a given application or service.” [76] Following this logic, it was required to re-introduce the user into the quality concept.

The initial efforts to do it can be tracked to 2001 in [32]. As mentioned before, Hardy introduced the concept of *Perceived QoS*, which “reflects the customer’s experience of using a particular service. It is influenced by the customer’s expectations compared to observed service performance.” [28] However, the main contribution in this direction was made on the same year by Van Moorsel with the definition of the *Quality of Experience (QoE)* concept. In [83], Van Moorsel defines this new concept as “metrics related with the customer experience,” but also remarking that “QoE metrics may have a subjective element to it.” With this definition, the user became once more the center of the quality concept, focusing the analysis on the perceptual aspects of the quality. However, this time the definition of a new subjective quality-related concept marked the beginning of a new research area, highly influenced with the rise of the Web 2.0.

Despite the fact that the research in *experienced* quality continued, a proper, formal definition of QoE was missing. In 2011 the Qualinet community started the discussions to provide this definition in order to promote the collaboration and interchange of ideas between different work groups [51]. In [47], Le Callet et al. propose the first formal definition of QoE which is widely accepted nowadays: “QoE is the Degree of delight of the user of a service. In the context of communication services, it is influenced by content, network, device, application, user expectations and goals, and context of use.” [47]

The definition given by the Qualinet group concealed the problems with other definitions of QoE. For example, the ITU defined the concept in 2008 as “the overall acceptability of an application or service, as perceived subjectively by the end-user,” [41] which might lead to confusions between the QoE and *acceptability* concepts.

Considering that the Qualinet definition addresses the main issues of other definitions and exposes the holistic nature of the QoE, their definition will be used as the running description to continue the analysis of the concept. In [64] Raake and Egger expose a deep analysis about the experiencing process and how the perceived quality fulfills the expectations of the user, however a discussion about these topics goes beyond the scope of this work.

---

<sup>3</sup>In this quotation, QoS is used in the sense of representing technical, measurable network parameters following the IETF definition.

Despite the fact that the QoE measures the subjective appreciation of the user, any further analytical study require a proper way of measuring this experienced quality. In this sense, some metrics already existed in the field of telephony, defining the *Mean Opinion Score (MOS) scale*, in addition with the proper conditions to correctly extract these subjective values [40]. According with the specification, the scale is formed by five values, each one representing a characteristic meaning shown in Table 2.1.

MOS Value	Meaning
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 2.1: Mean Opinion Score (MOS) scale values.

This scale has been adopted for the evaluation of the QoE of multimedia services, turning into the de facto metrics used by the researchers to evaluate and estimate the experienced quality.

### 2.2.2.1 Influence Factors of the Quality of Experience

In the definition of QoE given by the Qualinet initiative – which is the running definition used in this work – it is clearly stated the multiple factors that influence the value of the QoE for a given service. These *Influence Factors (IF)* can be grouped into three main groups, aiming to simplify its analysis and understanding [47], [70].

- *Human IF*: These IF are defined as any variant or invariant characteristic intrinsically related to the human being. Typical examples of these IF are the demographic and/or economical background, mental constitution, emotional state, among others [70]. Since all these factors are usually related with the social and psychological aspects of the human perception, its analysis is a difficult task in the fields of multimedia services. Regardless of this implicit difficulties, a few studies on the literature have tried to show the influence on a theoretical and conceptual level (see for example, references 2 to 4 of [70]), and the impact of some specific human IF on the perceived quality and the QoE (see references 7 and 8 of [70]). The IF belonging to this category are the most complex variables, and its influence on the QoE is still not well understood [70].
- *Context IF*: Any factor that describe situational properties of the user environment are encompassed in this group [47]. These IF are usually sub-classified into *physical context*, *temporal context*, *social context* and *economical context* [47]. As stated in [70], context IF play a substantial role on selecting the appropriate quality level for a given experience, feature that might be crucial when optimizing the characteristics of the service. This is the main reason for providing *context-aware* analyses of the QoE, which aim to consider these types of variables into the QoE evaluation.
- *System IF*: In this category are classified any properties and characteristics of the service that determine the technical quality of the service [47]. These IF are typically related with the QoS concept discussed in Section 2.2.1, and they can be classified into *content*- (for example, the dynamic range of the audio, the amount of detail and motion, among others), *media*- (as the encoder type, the encoding bitrate, etcetera), *network*- (i.e. any metrics that are related with the IP transport mechanism as packet loss, or delay) and *device*-related (as for example, screen size, mobility of the device) [70]. The definition of these IF makes them the most easy to obtain, since their value can be retrieved by directly measuring the involved characteristic. This advantage and the fact that the multimedia services under study rely on the transporting capability of the network makes the system IF the most impacting factors of the QoE.



Although this classification is intended to provide an easy understanding of the QoE, usually the influence factors are interrelated among them, and its cumulative effect cannot be ignored. Even when the nature of the QoE exposes the holistic nature of the concept, most of the research has been done in the analysis of the impact of pure system IF on the QoE value. Despite the fact that an exhaustive review of the literature on this topic is out of the scope of this work, there are a few remarkable, generic analyses that need to be mentioned in order to understand the QoE.

Generally speaking, most of the works that try establish a relationship of any IF – most of the times a system IF – and the QoE value follow the same standard methodology. In [43], the ITU establishes the guidelines for both test and evaluation methods in order to obtain the assessed QoE value from the user. The main goal of this procedure is the deriving of a QoE model in the form of a mathematical formula that relates the analyzed impact factor with the final QoE value measured in the MOS scale.

Despite the multiple nature of the works that propose these types of QoE, all of them can be classified into two groups [71], [76]:

- *Exponential relationships:* In [20] Fiedler et al. propose a generic QoS-QoE relationship based on exponential functions, called the *IQX hypothesis*. This relationship is based on the two facts: on one hand, small changes on the IF strongly impact the experienced quality when this value is already high; on the other hand, once the quality is already low, big changes on the factor's do not have a significant impact on the QoE [20]. The principal assumption of these types of models is that the impact of the analyzed factor depends on the actual QoE value [76].
- *Logarithmic relationships:* In [66] Reichl et al. propose a generic QoE understanding based on the well-known *Weber-Fechner law*: based on the fact that “just noticeable changes” on the perception can be related with a fixed portion of the applied stimulus [66]. Despite having its origins on psychophysics, these types of relationships have also been found in the human perception of microeconomics and QoE [68].

The mathematical models extracted from these studies are useful to build QoE monitoring tools that are used to predict the quality in an online basis. These tools are designed to monitor the QoS IF required by the underlying QoE model and use the mathematical relationships in order to predict the future values of the QoE.

Even when most of the literature uses these mathematical models to build QoE monitoring tools, other approaches can also be followed. The following list is not exhaustive, yet it is intended to show how other techniques are also useful to build such quality monitoring tools. Some approaches are based on the fuzzy logic as the mechanism to predict the QoE [12], [31], [59]. Some other approaches rely on machine learning techniques that use user-assessed QoE values to train the tools [50], [52], [81].

Considering the context presented in the last paragraphs, the work presented in the next chapters expands the offer of these tools with a new one based on a novel approach that aims to integrate the three types of IF. This proposition aims to expand the analysis by proposing a more holistic approach rather than focusing on the impact of a single IF.

## 2.3 The Quality of Business (QoBiz) concept

The subjective nature of the QoE makes its evaluation a hard task, since it can be influenced by multiple factors. As mentioned in Section 2.2.2, technical concepts and measurements are not enough to provide a comprehensive understanding of the QoE. Since the Internet has evolved as a platform to deliver value to a high amount of potential customers, the assumptions about the user and business decisions no longer hold [83]. In an era where the customer is the center of services, monitoring quality and business decisions can make a huge difference with respect to the competition.

Van Moorsel remarks this fact, and proposes a new framework that introduces a new concept into the QoE ecosystem: the *Quality of Business*, defining it as “metrics expressed in terms of money.” [83] In addition with the definition, Van Moorsel proposes a new way of understanding the interrelations between technical – QoS, subjective – QoE and business-related parameters –QoBiz. In general terms, the framework proposes the high impact that business decisions can have on the user’s expectations and, therefore, the revenues of the service provider. Monitoring these three main dimensions will ensure proper working of the whole service ecosystem, granting at the same time a proper network management, optimal revenues and high levels of user’s satisfaction [56], [69]. This initial analysis introduced the a new dimension to include when evaluating the QoE of OTT services: the business parameters.

The inclusion of business-related parameters into the QoE analysis introduces two principal points of view. On one hand, the QoBiz definition given by Van Moorsel can be interpreted from the side of the service provider. In this sense the procedure of analyzing, monitoring and ensuring high levels of QoE directly impact to the revenues of the OTT service provider [83]. This last fact has been treated in some works that try to put in relation the amount of money spent in a service with the expected quality from the user (see for example [74]). Other works also propose economical strategies that the telcos can follow in order to indirectly grow their revenues by taking advantage of the high amounts of data required by multimedia OTT services (as for example [49], [80]).

On the other hand, business-related parameters are also important from the point of view of the user. Reich and Zwickl identify the price as one of the factors that impact on the expectations of the user and, therefore, the expected quality from the service [67], [69]. Following these concept, the research has focused the efforts to analyze the impact of pricing schematas into the QoE. For example, Sackl et al. analyzed the effect of the willingness-to-pay for enhanced video quality on the QoE value of Internet-based services [73]. In [74], the authors propose how the price of the service can impact on the expected quality, giving recommendations to the service provider to guide the pricing schemas in order to maximize the revenues of the business.

The above studies show the close relationship between business variables and the QoE concept, aiming to demonstrate that monitoring and ensuring QoE rather than simple QoS metrics ameliorates both the revenues of the business and the satisfaction of the user [53]. In this line, Wahlmueller et al. demonstrated this last by proposing and emulating an approach to price and regulate the served QoE rather than QoS parameters [85]. The approach is based on the *Paris Metro Pricing (PMP)* approach identified by the differentiation of the network traffic in two classes [53]. The idea behind is the prioritization of the services requiring high levels of QoE. Even when the approach can interfere with the Network Neutrality, the authors show that by applying this techniques, both the revenues and the user’s satisfaction are increased [53], [85].

The PMP approach was later extended to be used in the particular case of OTT services. In [11] Ahmad et al. proposed a PMP-based approach that enforces the collaboration between

ISP and OTT. The proposition is formed of a reference architecture that defines the required interfaces to allow collaborating between both service providers. This collaboration is focused on the maximization of the revenues based on multiple factors as the user chunk (which is assumed to be directly correlated to the QoE), the pricing and marketing actions. The authors show that the OTT provider is capable of maximize their revenues and the ones of the ISP by asking them to deliver the required QoS parameters in order to maintain the desired QoE level. The novelty of this approach clearly demonstrate that the collaboration between telcos and OTT providers – having in mind business parameters – can benefit the whole Internet-based services ecosystem.

Despite the efforts of analyzing the QoE concept, a complete, holistic analysis is still an open challenge, due the multiple factors that have impact on this value. Lately, the inclusion of the business factors in the study expanded the domains of the QoE, however a complete business-based quality evaluation tool is still an open challenge on the field. The next Chapters goes a step beyond the state of the art by proposing an implementable solution that integrates business parameters into the QoE analysis.

---

---

# Chapter 3

## EFSM-based Quality of Experience Evaluation Framework

### Contents

<b>3.1</b>	<b>Introduction</b>	<b>25</b>
<b>3.2</b>	<b>Preliminaries</b>	<b>26</b>
3.2.1	Finite State Machine (FSM)	26
3.2.2	Extended Finite State Machine (EFSM)	27
3.2.3	$l$ -equivalent of an Extended Finite State Machine	29
<b>3.3</b>	<b>Quality of Experience Evaluation Framework</b>	<b>31</b>
3.3.1	Modeling an Over-The-Top Service	32
3.3.2	Augmenting the Model with Quality Indicators	34
3.3.3	Computation of the $l$ -equivalent model and the Quality of Experience	37
3.3.3.1	Deriving the $l$ -equivalent of the EFSM	38
3.3.3.2	Computing the Quality of Experience at terminal states	38
<b>3.4</b>	<b>Discussion</b>	<b>41</b>
<b>3.5</b>	<b>Conclusions</b>	<b>42</b>

---

### 3.1 Introduction

In the previous chapters, the concept of QoE was addressed, analyzing the main components that impact on this concept. Among these factors, we identified the business aspects as one of the multiple areas that can influence the expectation of the user and, therefore, the QoE of the service. However, the literature does not present works that consider this facet into the QoE analysis. Being this said, we aim to extend the actual domain of the QoE analysis by including business elements as an impacting factor of the final QoE value.

As stated before, there are not so many works that tackle this thematic, thus a complete novel analysis and approach was required in order to correctly extend the QoE analysis. This requirement was satisfied by the proposition of a novel QoE evaluation framework, which aims to address the following points:

---

- *Extensibility*: The proposed solution must be extensible enough to allow the introduction of business-related aspects without interfering with other aspects influencing the QoE. In other words, the approach must support the coexistence of the business aspects with at least the technical ones; the most studied influencers of the QoE.
- *Traceability*: As stated before, the business elements have the potential to change the expectations of the users by means of changing their future expectations with the service. This temporal consciousness feature has not been exploited in the current QoE evaluation frameworks, which are limited to correlate actual values of technical variables to an actual value of the QoE. These values are then ignored when computing the QoE value in the future. By introducing business elements into the QoE concept, we need to consider the past values of the user's decisions in order to consider the effect of past decisions – usually related with business aspects – into the QoE analysis. In brief, the proposed approach must have a memory of the previous decisions of the user in order to consider them into the QoE computation.

Considering requirements presented above, we developed a QoE evaluation framework based on the mathematical formalism of the EFSM. The characteristics of this mathematical model allowed us to fulfill the constraints, making them a suitable solution for our novel approach. In the following sections we present the details of the mathematical framework on which the proposed approach is based.

## 3.2 Preliminaries

### 3.2.1 Finite State Machine (FSM)

The first, base model on which the EFSM underlie is called Finite State Machine (FSM). It is a mathematical model conceived as an abstract machine that can be in one – the *current state* – of a finite number of possible states. This state can change when triggering an event or conditions, which is called a *transition* [26]. These machines are particularly useful when designing protocols and modeling and testing software, since they allow to control the workflow of the system under study and know the exact state of the system at any given moment [57].

In a more formal way, an *Finite State Machine (FSM)* can be defined as a 5-tuple  $M = (I, \Sigma, O, \delta, s_0)$  [26] where:

- $I$  is the set of inputs of the machine,
- $\Sigma$  is the set of states of the machine,
- $O$  is the set of outputs of the machine,
- $\delta : \Sigma \times I \rightarrow \Sigma \times O$  is the transition function of the machine, and
- $s_0$  initial state of the machine.

This definition allows an FSM to start the execution at the initial state  $s_0$ . At any state, a single input  $i \in I$  is consumed and the machine computes the text state  $s' \in \Sigma$ , moves to it and computes the output  $o \in O$  according to the function  $\delta$ . This process of consuming an input, moving to the next state and producing an output conforms the *execution* of a *transition*.

Using this definition it is possible to characterize the *recognized language*  $\mathcal{L}_i \subseteq I^*$  and *generated language*  $\mathcal{L}_o \subseteq O^*$  of the machine. The first is the set of all possible inputs accepted by the machine while the latter is the set of all generated outputs under valid inputs of the machine.

An example of an FSM is shown in Figure 3.1, which is a graph representation of the FSM defined by:

- $I = \{i_1, i_2\}$ ,
- $\Sigma = \{s_0, s_1\}$ ,
- $O = \{o_1, o_2\}$ , and
- $\delta = \{((s_0, i_1), (s_1, o_1)), ((s_0, i_2), (s_0, o_2)), ((s_1, i_1), (s_0, o_1)), ((s_1, i_2), (s_1, o_2))\}$

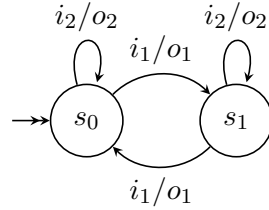


Figure 3.1: An FSM  $M_1$

Following the given specification, the machine  $M_1$  starts in the state  $s_0$ , where two possible actions can take place. If the machine receives  $i_1$  as input, it will generate  $o_1$  as output and it will move to the state  $s_1$ ; on the contrary if it receives  $i_2$  as input, it will generate  $o_2$  as output and it will stay on the same state  $s_0$ . Moreover, an analogous behavior can be observed on the state  $s_2$ . Being this said, it is possible to note some remarks about the language this machine accepts and the outputs produced. For instance, the tuple  $(i_1 i_2 i_2 i_1 i_1, o_1 o_2 o_2 o_1 o_1)$  is a valid pair for the input and output of the machine  $M_1$  but  $(i_1 i_1 i_2 i_2 i_1, o_1 o_1 o_2 o_2 o_2)$  is not, since the last input ( $i_1$ ) produced an invalid output ( $o_2$ ) given the specification of the machine.

Finally, by observing the specification of  $M_1$  it is possible to note that the function  $\delta$  is defined for all possible input values at each state. This is an example of a *completely specified* FSM. In the case there is a behavior not specified for any input at any state, the machine is said to be *incompletely specified* [26].

### 3.2.2 Extended Finite State Machine (EFSM)

The definition of an FSM given in Section 3.2.1 can be expanded in order to introduce more elements that can model, for example, the data flow of the system under analysis. An EFSM is the mathematical model that allow the expansion of the FSM model with input and output parameters, context variables, and operations and predicates defined over context variables [57].

For a formal definition of this extension it is required to consider the following. As before, let  $I$  be set of inputs of the machine,  $O$  be the set of outputs of the machine. Let  $R$  and  $V$  be two disjoint sets of *parameters* and *variables names*. For  $i \in I$ , we note  $R_i \subseteq R$  the set of input parameters and  $D_{R_i}$  the set of valuations of parameters in  $R_i$ . For  $o \in O$ , we note similarly  $R_o$  and  $D_{R_o}$ . Finally, let  $D_V$  be the set of context variables valuations  $\mathbf{v}$ . Considering these sets, an *Extended Finite State Machine (EFSM)* is a tuple  $M = (\Sigma, T)$  [57], where:

- $\Sigma$  is the set of states of the machine, and
- $T = \{(\sigma, i, P, op, o, up, \sigma')\}$  is the set of transitions of the machine where:
  - $\sigma, \sigma' \in \Sigma$  are the initial and final states of the transition,
  - $i \in I$  is the input of the transition,
  - $o \in O$  is the output of the transition,
  - $P : D_{R_i} \times D_V \rightarrow \{True, False\}$  is the predicate function of the transition,
  - $op : D_{R_i} \times D_V \rightarrow D_{R_o}$  is the output parameter function of the transition, and
  - $up : D_{R_i} \times D_V \rightarrow D_V$  is the context update function of the transition.

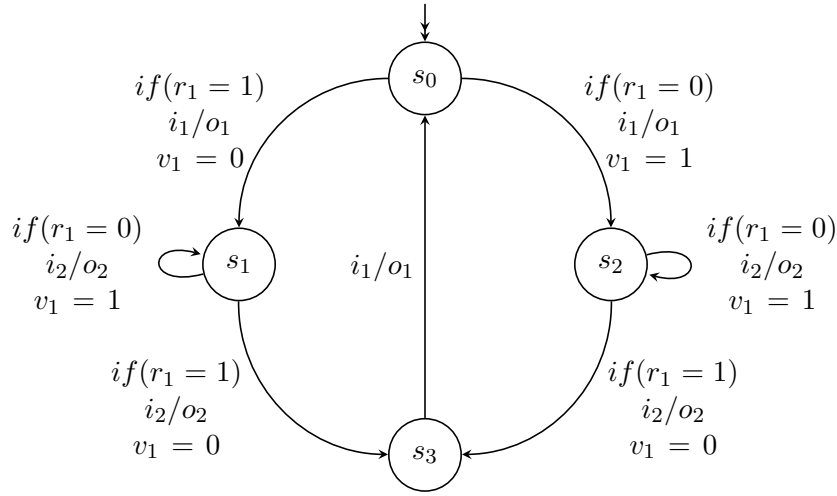
The given definition allows us to expand the FSM with:

- *Context variables:* A set of variables  $V$  whose values  $\mathbf{v}$  are accessible at any time of the execution and constitute the *configuration* of the machine,
- *Updating functions:* A function  $up$  that updates the values of the context variables based on the actual configuration of the machine and the input parameters of the transition,
- *Input and output parameters:* Both sets ( $R_i$  and  $R_o$  respectively) allow having a parametrized input without impacting the configuration of the machine, and
- *Predicates:* Also called *guards*. A set of functions  $P$  that insert conditions on the transitions to decide whether it should be executed or not. These elements are useful to avoid non-determinism where two transitions can be triggered under the same input.

An EFSM example is presented in Figure 3.2, which shows the graph representation of the machine  $M_2$  defined by:

- $\Sigma = \{s_0, s_1, s_2, s_3\}$ ,
- $V = \{v_1 \in \{0, 1\}\}$ ,
- $R = \{r_1 \in \{0, 1\}\}$ , and
- Some transitions of  $T$ , for instance:  $(s_0, i_1, if(r_1 = 1), \emptyset, o_1, v_1 = 1, s_1)$  or  $(s_2, i_2, if(r_1 = 0), \emptyset, o_2, v_1 = 1, s_3)$ .

The execution of this machine is not much more complicated as a traditional FSM. The execution starts at the state  $s_0$  with fixed values of the context variables, for example ( $v_1 = 1$ ). Then, the machine receives the input  $i_1$  parametrized with the value  $r_1 = 0$ , triggering the transition to the state  $s_2$  and set the value of  $v_1$  to 1. At that state, the machine receives the input  $i_2$  parametrized with the value  $r_1 = 1$ , causing the machine to move to the state  $s_3$  and setting the value of  $v_1$  to 0.

Figure 3.2: An EFSM  $M_2$ 

### 3.2.3 $l$ -equivalent of an Extended Finite State Machine

At this point, the EFSM have the potential to store the trace of states visited while executing the transitions. However, it is not clear to observe all the possible paths that lead from the initial state to another one by executing a fixed number of transitions. In order to do this, we require the derivation of an analogous model from the original EFSM.

The  $l$ -equivalent is a tree-shaped model that can be computed taking an EFSM as a base. The main goal when building this new model is to obtain a clear visualization of the paths with a previously fixed length  $l$  starting from the initial state. This is achieved by the following process:

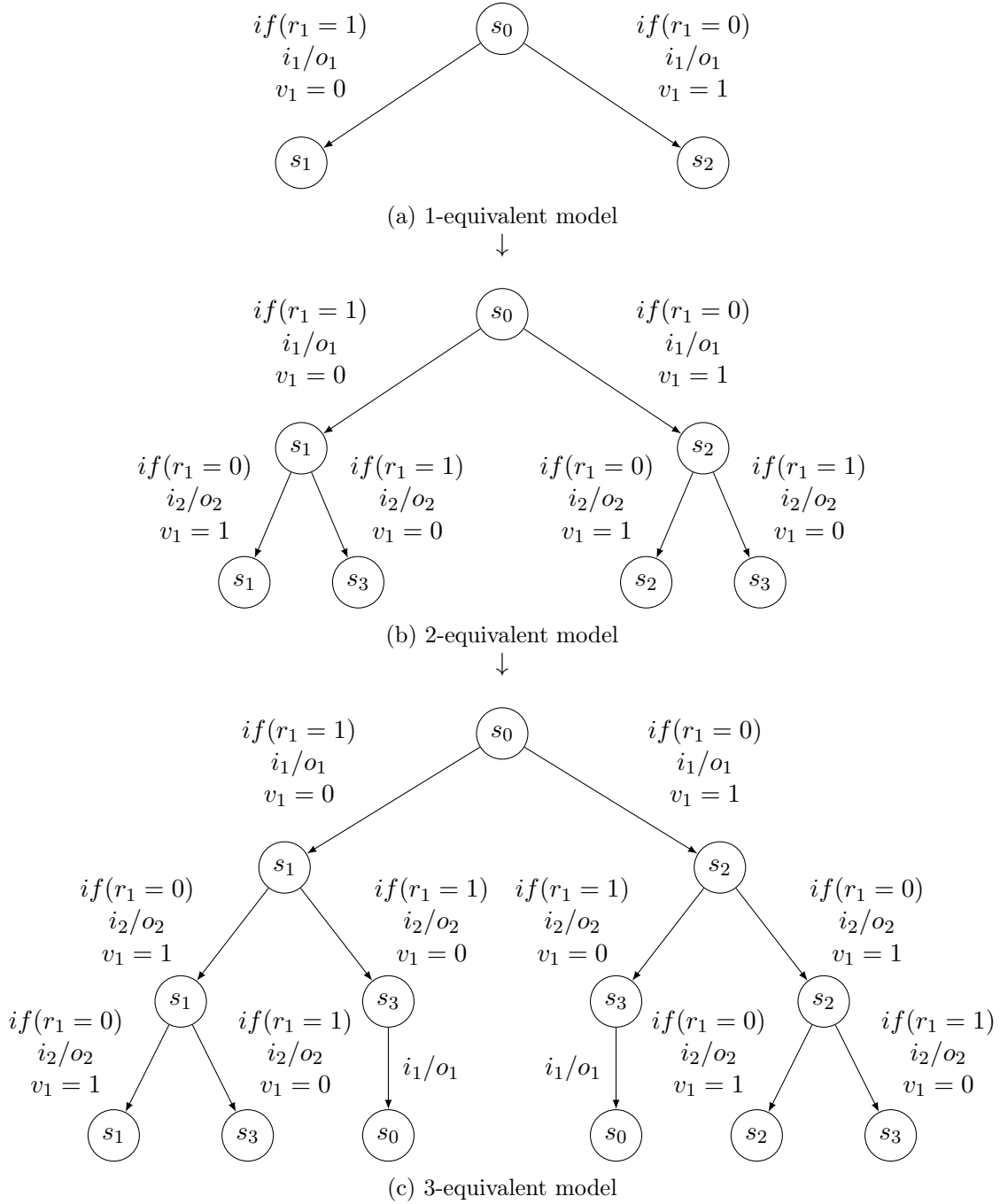
1. First, introduce *constraints* to all the loops on the machine in order to limit the maximum number of transitions to be executed.
2. Second, starting from the initial state of the machine (which is at level 0 of the  $l$ -equivalent), draw all the possible next states of the actual state in a lower level of the tree, *cloning* a state as many times as it can be reached by different transitions.
3. Third, repeat the last process starting at each one of the recently computed next states, until the desired length  $l$  is reached.

At the end of the process, the  $l$ -equivalent will have a tree structure whose length is exactly  $l$ . Within this model it is possible to notice exactly what are the choices from moving from one state to another while, at the same time, keep a memory about these decisions when arriving to the leaves of the tree.

In order to illustrate this process, Figure 3.3 shows the process of drawing the 1-, 2- and 3-equivalent forms of the  $M_2$  machine.

By observing the Figures 3.3a, 3.3b and 3.3c it is possible to notice that each  $l$ -equivalent form is built in an incremental way, starting from the previous model. In these figures, the first 1-equivalent was built by cloning the states  $s_0$ ,  $s_1$  and  $s_2$ , and the outgoing transitions of the state  $s_0$ . Using this derived model as a base, we introduced the constraint of executing the loops



Figure 3.3:  $l$ -equivalent forms of  $M_2$

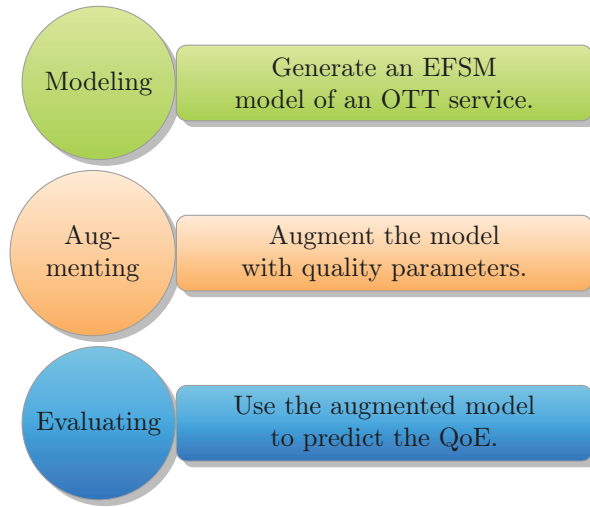


Figure 3.4: Overview of the QoE Evaluation Framework

only once, in order to meet the new length for the 2-equivalent. This model is then constructed by cloning the states  $s_1$  and  $s_2$  – which are children of  $s_1$  – and all the outgoing transitions of  $s_1$ . This step is repeated with the next states of  $s_2$ . Finally, the 3-equivalent is build similarly by changing the constraint of executing the loops up to 2 times, which allows computing the new next next states in an analogous way.

### 3.3 Quality of Experience Evaluation Framework

In Section 3.1 it was discussed the need of proposing a novel approach to evaluate the QoE, caused by the inclusion of business-related factor that has a temporal effect on the QoE. In this sense, it is required to start the development of this approach using an appropriate formalism that can provide accountability of business decision for future evaluations. The before-mentioned mathematical formalism of the EFSM can provide such feature intrinsically.

The proposed approach is therefore based on EFSM models in order to take advantage of the its traceability feature to fulfill this requirement. In addition, the extension capabilities of the EFSM allows including multiple types of parameters, including the traditional network-related parameters that are usually related with the QoE. Using these features, we propose a complete QoE evaluation framework based on the EFSM models.

The overview of the framework is presented in Figure 3.4, being conformed by three main sequential stages:

1. **Modeling:** The first step of this framework is the derivation of the base model of the OTT service that will be analyzed. The main goal of this step is to provide a model of the main functional requirements of the service;
2. **Augmenting:** Once the service in study has been modeled as an EFSM, it is required to identify the quality factors that will be considered into the QoE evaluation. These factors will be inserted into the model in order to expand the modeling capabilities with some non-functional requirements of the service, namely, quality-related requirements;

3. **Computation:** Finally, using the augmented model of the service, we will use the  $l$ -equivalent form in order to apply an appropriate QoE model. The latter must be able to correlate the quality factor that were previously introduced to a single QoE value.

The usage of trace models (EFSM and its  $l$ -equivalent form) allows having memory about the decisions of the users while they were using the service. We take advantage of this capability in order to consider the effect of business decisions in the expectations of the user and, therefore, in the QoE value of the service.

In the following subsections, the three stages of the framework will be explained in details, and a running will be presented in order to illustrate how does the proposed technique work.

### 3.3.1 Modeling an Over-The-Top Service

The QoE evaluation framework here exposed is based on the mathematical formalism of EFSM. In this sense it is required to build this model as a first phase of the whole methodology. To achieve this, it is required to consider the final objective of the whole evaluation framework in order to correctly formulate the scope of the proposed methodology.

As mentioned in Section 2 the QoE is a concept intrinsically subjective, which is influenced by many factors as, for example, technical, business, psychological, among others. All of the aforementioned elements impact on the expectations of the user and thus, on the value of the QoE, however, its final QoE rate still remains as a value give explicitly by the user. The presented approach captures this subjectivity by integrating the user into the elaboration of the model process.

---

#### Methodology Outline 1 Deriving an FSM to simulate the service functional requirements

---

*Step 1:* Generate a written description of the service. This can be either a description “in words” or a more formal description of the behavior (a flow chart, for example).

*Step 2:* Identify system states, by using the service description. As a convenient approach, the authors in [13] suggest the decomposition of the service (or a Web Application in their case), in order to identify logical blocks in the service, which are easier to model.

*Step 3:* Identify system transitions, inputs and outputs. In this case, user queries are the FSM inputs while the service responses are the FSM outputs.

---

This is done by directly analyzing the user-service interaction, aiming to build the OTT model using this interaction as the main guidelines to describe the behavior of the service. By analyzing this interaction, it is possible to ensure that only the decisions that directly impact the final user are considered into the QoE evaluation. This aids to simplify the model by suppressing factors that can be mistakenly consider as important on the QoE evaluation.

In order to help identify and specify the the user-service interaction, the proposed methodology suggests the creation of a written description of the behavior of the services as seen from the user. This document will aid the designer to correctly identify the primary element of the first EFSM model of the OTT service.

The above-mentioned process can be formalized in form of a methodology presented in the Methodology Outline 1. As mentioned in the step 2, Andrews suggests that the process of identifying the states of the machine can be achieved easier by decomposing the analyzed service into logical blocks. This approach is based on the “divide and conquer” idea, where smaller logical modules of an entire service are easier

---

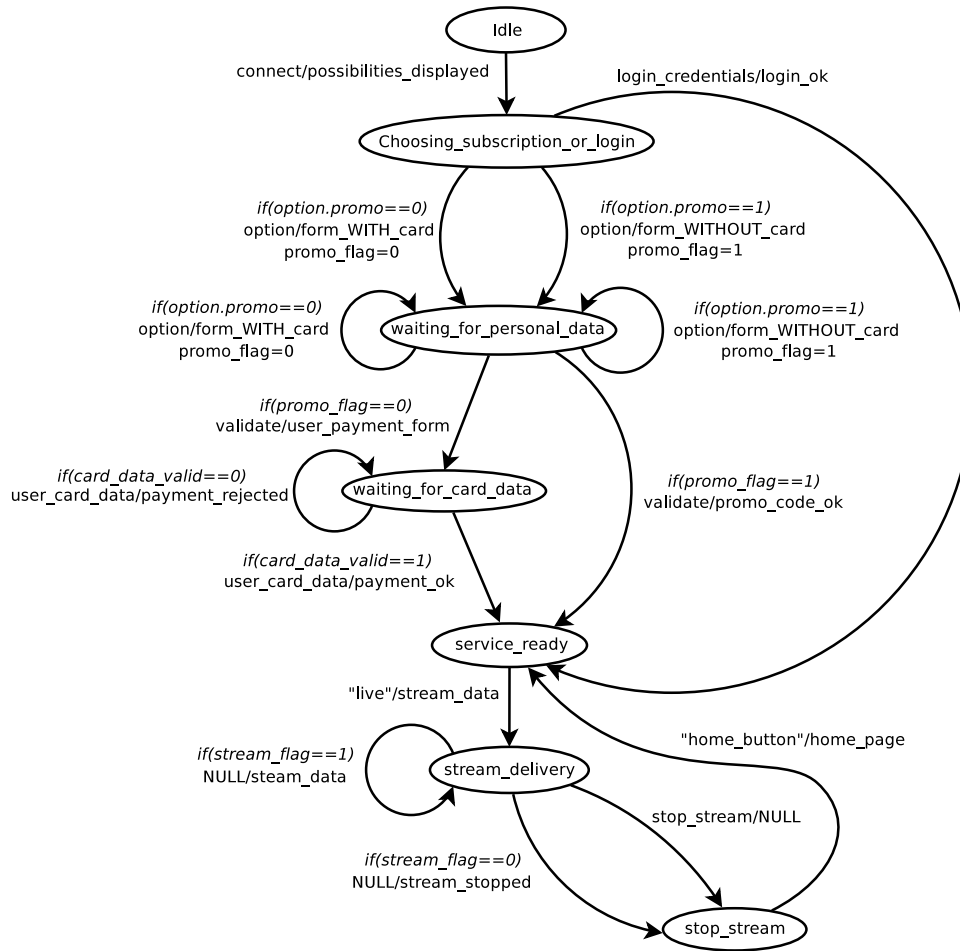


Figure 3.5: Initial EFSM model of the beIN Sports Connect service

to model by themselves. Then, all the big models are connected between them as if each one of them were a single component of the system.

Even when the proposed methodology seems to be a simple way to obtain the model, the literature does not offer an automated way of deriving this model. All of the approaches are *semi-automatic* in such a way that they are able to derive incomplete models that need to be validated manually, in order to detect possible invalid behaviors of underspecification of the service. Without going any further and in order to maintain the user's point of view of the analysis, it is recommended to use tools or methodologies that perform reverse engineering in order to derive a first approximation of the model (see, for instance [86]). In this sense, we ensure that the produced model has been derived starting from real use cases, thus confining the analysis to the variables that directly impact the user.

The proposed methodology is illustrated by applying it on a real OTT service. We selected the beIN Sports Connect service<sup>1</sup>, a sports television service traditionally offered as a premium pack for cable subscribers. In order to apply the methodology, we turned to its website and analyzed its behavior. This process ended with the writing of the description of the interaction, which describes how the service behaves against different interactions and what are the

<sup>1</sup><http://www.beinsportsconnect.fr>

responses of the platform. The obtained document with the entire description can be found in the Appendix A.

Using this description, we built the EFSM machine presented in Figure 3.5. The model presented captures the behavior of the beIN Sports Connect Service showing: (1) the states of the system while using the service, (2) the transitions between the states, and (3) the main inputs and outputs of the system for each transition. Additionally, it is possible to observe that this machine makes use of parametrized inputs (as, for example, both outgoing transitions from the “Choosing\_subscription\_or\_login” state) and even context variables (namely the “promo\_flag” and “stream\_flag” variables). All these elements make this model an EFSM rather than a simple FSM.

### 3.3.2 Augmenting the Model with Quality Indicators

Once the first version of the model has been derived, it is required to insert *quality indicators* into the machine. These indicators will aid us to keep track about how the user decisions impact on the final QoE. In order to achieve this goal, these indicators need to be inserted in the machine in such a way that the model itself can keep track of their values and how do they change. As states in Section 3.1, the EFSM models already provide traceability and extensibility features, which are to be used in order to augment the capabilities of the OTT model.

The proposed framework takes advantage of the *context variables* of the EFSM in order to insert the quality indicators aforementioned. In order to do this, it is needed to introduce some constraints to the quality indicators that we will analyze, that are mainly defined by the constraints of the context variables EFSM model:

- As defined in Section 3.2.2, the context variables must be valuatinal. In other words, an instance of a context variable of the machine needs has to have a fixed value at any time of the execution, and
- It is required to specify how every context variable value is updated at each transition. In other words, for each transition, it is required to specify which are the *updating functions* of the variables, even if this function is the identity for some transitions.

Considering these restrictions, we will impose the following constraints to the quality indicators that will be used in this framework:

- *Measurable*: Any quality indicator inserted must be able to be expressed in real values. The proposed framework is flexible enough to support both continuous and discrete variables as quality indicators.
- *Updatable*: At the same time, the values of the quality indicators need to be updated following functions previously established. To clarify the ideas, these values need to be updated under certain conditions. This specification can be easily done by means of a mathematical function.

Both constraints allows us to directly insert the quality indicators as context variables of the EFSM, process that will produce an augmented version of the OTT model. This is the main output of the second stage of the proposed framework: the OTT model augmented with quality indicators that will be used to analyze the QoE. This new procedure allows us to extend the previous methodology with the steps related with the augmentation of the machine, which are now show as steps 4 and 5 in the Methodology Outline 2.

As for the running example, we will go even further in order to simplify the model. We will limit the introduction of quality indicators to a finite number of discrete variables that will be used as context variables of the machine. This considerations will allow us go further when using the framework in order to statically analyze the OTT service. A deeper explanation for this simplification is provided in Section 6.2.

For the beIN Sports Connect service, three quality indicators were considered into the quality analysis:

The first one is an *objective* parameter. In Section 2.2.2.1 we stated that the most important factors that influence the QoE are related with the network quality itself, namely the QoS variables. The running example does not get rid of this fact, and we incorporate such factors into the analysis. Traditionally, these parameters might include (but are not limited to) the packet loss, the delay, the jitter and some others related with the stream itself like the video compression bitrate. However, beIN Sport Connect has the ability to periodically check the QoS of the network in order to determine if the quality of the network is good enough to keep the streaming or not, interrupting the service if the network parameters decay below levels that do not ensure certain quality level for the user. For modeling reasons, we take advantage of this feature and we model all these network-related parameters into a single one, called *stream\_flag*.

Secondly, a *subjective* parameter will integrate other psychological factors into the analysis. The QoE concept is, by its definition, a user-centered concept. In this way, it is also important to consider user centered opinions into the QoE evaluation. For the running example, we considered a single subjective quality indicator called *service\_confidence*, measuring the level of confidence the has with the service.

Finally, a *business-related* variable will enhance the analysis with the effect of monetary factors. The main goal of proposing a new approach to evaluate the QoE was the inclusion of of business-related parameters and analyze how do the business-related decisions of the user impact on the final QoE value. In this way, we consider a single business-related parameter, called *price\_premium*, defined as “the amount a customer will pay for the brand in comparison with another brand (or set of comparison brands) offering similar benefits.” [8]

For each one of the proposed quality parameter, we propose metrics in order to correctly measure and update their values as required by the framework:

- *stream\_flag* (objective parameter): This indicator will tell us if the stream is working or not, meaning that if the network requirements for the service are met or not. At any time, the self-monitoring capabilities of the service will re-compute the value of this value and decide to interrupt the service if necessary. We will also set this flag to **false** when the service has not started yet, or the user voluntarily decided to interrupt the streaming (as, for example, they decided to end watching the service). Given the nature of this variable, its value will be changed externally when the service performs the check of the network.

---

#### Methodology Outline 2 Augmenting the FSM with non-functional parameters

---

*Step 1:* Generate a written description of the service.

*Step 2:* Identify system states.

*Step 3:* Identify system transitions, inputs and outputs.

*Step 4:* Identify transitions where variables are updated. Once the variables under study are identified, it is possible to determine the transitions where their values are updated. An available QoE model for the parameters under study can be used or adapted for each one of the transitions involved.

*Step 5:* Augment the FSM with variables corresponding to the parameters under study and their updating functions.

---

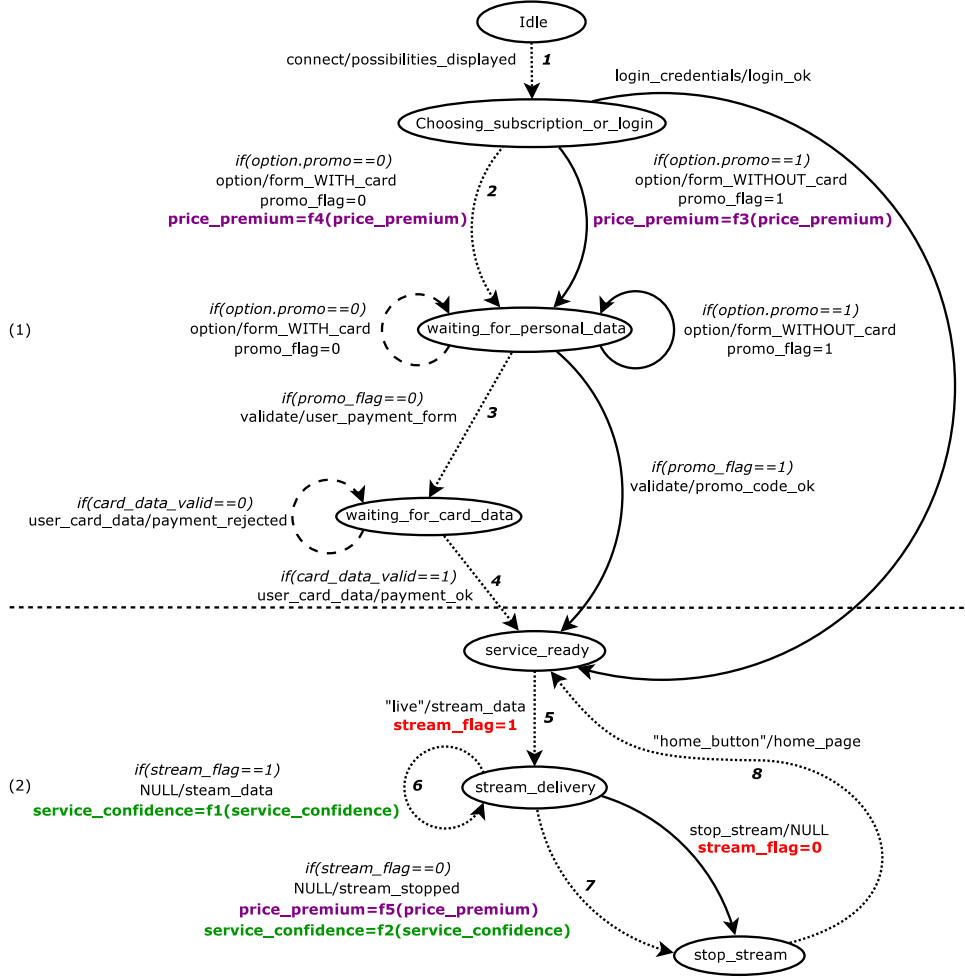


Figure 3.6: Augmented EFSM model of the beIN Sports Connect service

In addition, there exist two transitions where the value of this context variable must be updated: (1) when the machine moves from the state ‘service\_ready’ to ‘stream\_delivery’ setting its value to 1, which means that the service started the streaming of the video; and (2) when moving from the state ‘stream\_delivery’ to the state ‘stop\_stream’ under the input “stop\_stream” setting its value to 0, which means that the user voluntarily decided to stop the service (see Figure 3.6).

- *service\_confidence* (subjective parameter): The level of confidence of the user will be measured using a 3-values scale: 0 which means that the user has no confidence with the service; 1 meaning that the user is indifferent with the service; and 2 representing a user with high level of confidence with the service. As this value represents the confidence of the user with the service, we will update its value when the service either keeps streaming seamlessly or decides to stop the streaming – since the minimal network conditions are not met. Both cases are represented by two outgoing transitions from the ‘service\_ready’ state, being the first one represented by the loop over the same state and the second one represented by the transition to the state ‘stop\_stream’ guarded by the “stop\_stream” variable (see Figure 3.6). For each case, a unique updating function for the *service\_confidence* variable

is proposed, being one of them an increaser ( $f_1$  in Table 3.1) and the second a reducer ( $f_2$  in Table 3.1) of the value.

Table 3.1: Definitions of  $f_1$  and  $f_2$ , updating functions for *service\_confidence*.

Service_confidence( $s_c$ )	$f_1(s_c)$	$f_2(s_c)$
0	1	0
1	2	0
2	2	1

The proposed behavior is based on the fact that, on one hand, if the system “decides” to continue with the service, the user will increase their confidence with the service (updating function  $f_1$ ); on the other, if the system “decides” to stop the streaming, the user will lower their confidence with the service (updating function  $f_2$ ).

- *price\_premium*: (business-related parameter): In [8], Aaker recommends a scale of three values to measure this variable: 0 representing a user who is not a customer of the brand; 1 representing a “brand switcher” user; and 2 for users who are loyal customers of the brand. For this variable, three transitions were identified where its value must be updated: on both outgoing transitions from the state ‘choosing\_subscription\_or\_login’ to the state ‘waiting\_for\_personal\_data’; and on the transition from ‘stream\_delivery’ to the state ‘stop\_stream’ guarded by the “stop\_stream” variable (see Figure 3.6). For each transition, the functions  $f_3$ ,  $f_4$  and  $f_5$  are provided whose definition can be found in Table 3.2.

Table 3.2: Definitions of  $f_3$ ,  $f_4$  and  $f_5$ , updating functions for *price\_premium*

Price_premium( $p_p$ )	$f_3(p_p)$	$f_4(p_p)$	$f_5(p_p)$
0	1	1	0
1	1	2	0
2	2	2	1

The first two functions act as a raiser of the *price\_premium* with slight differences. In one hand,  $f_3$  raises the value for non customers of the brand going from 0 to 1 in case of the user is activating a promotional code, and leaving the value unchanged for other customer types. On the other,  $f_4$  raises the value for both non customers and brand switchers, going from 0 to 1 or 1 to 2 respectively when they select a subscription with engagement. In all other cases, this function leaves the value unchanged for other types of users.

All of these indicators were inserted as context variable of the model with their respective updating functions presented before. The augmented model can be seen in Figure 3.6 where the context variables are highlighted in red, green and purple for the objective, subjective and business-related respectively. In the same Figure, some transitions are marked with dots, and dashes, and some other are labeled with numbers from 1 to 8. This notation will be used and explained in the following Section.

### 3.3.3 Computation of the $l$ -equivalent model and the Quality of Experience

At this point, the OTT model captures both the functional – as seen from the point of view of the user – and the non-functional requirements of the OTT service that are relevant for the user. Even though the EFSM model has the ability to contain the aforementioned information, it



cannot be used to evaluate the QoE yet, since the contained configurations of the machine – and therefore, of the service – are not clearly displayed on the model. It is required to compute an alternative form of the model in order to clearly expose and analyze all the contained information of the model: the  $l$ -equivalent form.

### 3.3.3.1 Deriving the $l$ -equivalent of the EFSM

The main goal of this step is the expansion of the OTT model aiming to expand the information shown on the model. This process is performed by the computation of the  $l$ -equivalent form of the EFSM, which will now expose more information about the OTT model:

- *Paths of fixed length:* First, the  $l$ -equivalent unrolling process allows unveiling all the possible paths of a fixed length  $l$  starting from the initial state of the machine. This information is useful to keep track of all the decisions that the users took when reaching a terminal state by executing a fixed amount of transitions; and
- *Configurations of the machine:* Second, since it is possible to know which transitions were executed in order to arrive to a terminal state, it is possible to keep track about the context variables of the machine, updating their values when possible. In this sense, it is possible to know which is the final configuration of the machine at each terminal state, that is, the valuation of the context variables at the leaves of the tree.

In order to obtain this new information, this equivalent model can be computed by applying the process exposed in Section 3.2.3, i.e. by fixing the required length of the analysis and introducing the corresponding constraint to the model.

For the beIN Sports Connect example, we computed the  $l$ -equivalent in order to show how does the framework must be applied. In Figure 3.7 we show a single path of the 11-equivalent model of the machine presented in Figure 3.6, where the transitions of the paths are highlighted in dots. In the same Figure, the bold numbers on dotted transitions label them in order to easily identify the sequence of transitions that generates this path: 1–2–3–4–5–6–6–7–8–5–6.

The shown execution represents the usage of a new user, who is distinguished by the following history: (1) a completely new user who connects to the site to create a new account and select a paid subscription; (2) they do not make any mistake when selecting the subscription or introducing their card's data, therefore, the dashed loops in Figure 3.6 are not executed; (3) the customer starts using the service immediately; and (4) the service checks the status of the network three times, determining the network conditions are good enough to continue streaming the service in the first two instances, but deciding to stop the stream in the last one. Considering this interruption of the service, the user decided to go back to the home page and click on the “live” button, in order to start the streaming once again. Finally, the streams starts again, since the network conditions are now good enough to begin the streaming.

In the same Figure some states are marked from A to D, and the the valuation vector of the context variables is shown each time an updating functions changes the initial values specified at the “Idle” state.

### 3.3.3.2 Computing the Quality of Experience at terminal states

Once all the paths of the model were computed, all the required information to compute the QoE is available at the terminal states. It is possible then to proceed to compute its value. However, such calculations require the usage of a pertinent QoE model.

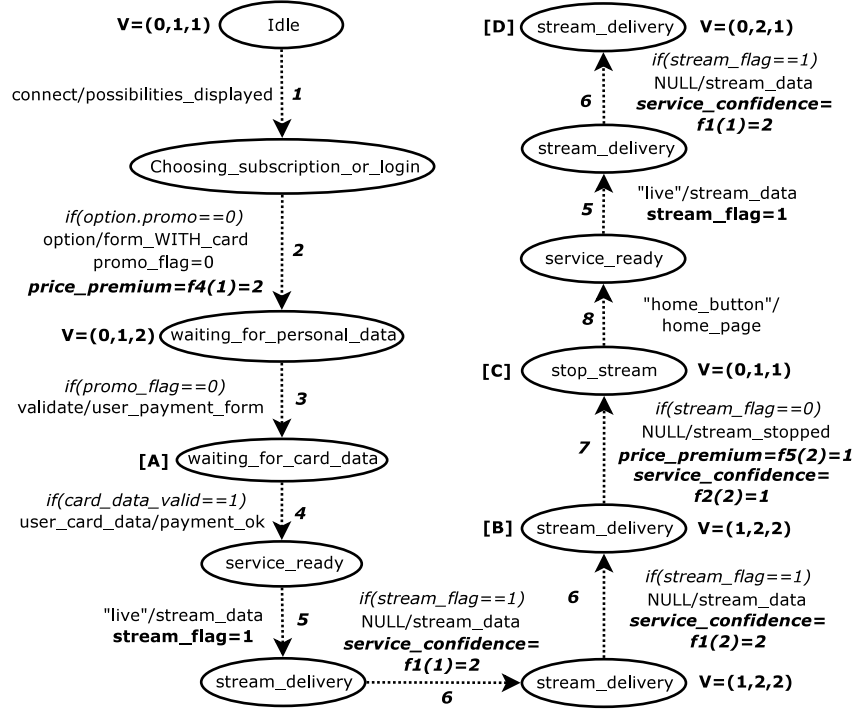


Figure 3.7: A single path of the 11-equivalent form

When the quality indicators were inserted in the EFSM model, we stated that the inserted variables are meant to be used to evaluate the QoE. This is done by applying a *QoE model* in order to correlate the values of the context variables to a single QoE value in the MOS scale. As stated in Section 2.2.2.1, the typical QoE models correlate one or more QoS parameters into a QoE value by using a mathematical function. In other words, the QoE model have the form:

$$QoE_{MOS} = f(QoS_1, QoS_2, \dots) \quad (3.1)$$

As the main goal of this work is to extend the domains of the Quality of Experience (QoE), we will also allow the introduction of other types of variables into the QoE model. In our particular case, we will introduce QoBiz into this analysis, which produces a more generalized version of the QoE model:

$$QoE_{MOS} = f(QoS_1, QoS_2, \dots, QoBiz_1, QoBiz_2, \dots) \quad (3.2)$$

In a more general way, the usage of EFSM as the underlying mathematical formalism does not restrict the introduction of any type of quality indicators. In this sense, the mathematical model to be used is restricted only by the inserted quality indicators, since they all the *indicators* have to be considered in the final evaluation of the QoE value. This fact was already generalized in [76], where QoE is understood as a function of  $n$  influence factors  $I_j$ :

$$QoE_{MOS} = \Phi(I_1, I_2, \dots, I_n) \quad (3.3)$$

With this information it is now possible to compute the final QoE value at the terminal stages of the  $l$ -equivalent model. At each one of this states, the configuration of the machine is known and it is possible to apply the QoE model using the valuation of the context variables.

Using this information we can expand event further the Methodology Outline 2 by adding the final steps in order to obtain the QoE values at the terminal states. The final whole process can be found in Methodology Outline 3.

In order to show how does the framework work when evaluating the QoE, a specific model will be used with the running example.

For the beIN Sports Connect Service, we considered three types of parameters that impact the QoE: objective, subjective and business-related. These indicators will be used in the linear combination QoE model proposed in [75]:

$$QoE = \sum_j \omega_j \cdot I_j \quad (3.4)$$

As stated in the same work, the coefficients – weights – of each quality indicator can be obtained by following the standard, recommended process by the ITU-T organization: “can be defined through an iterative process involving feedback surveys of QoE according to ITU-T recommendation.” [44], [75] In order to show how the framework should be used, the values of these weights will be fixed following the rules below:

- *Objective:* By definition, any OTT service uses a third-party infrastructure – the network – in order to deliver the content of the offered service. In this sense, the OTT provider does not have control over the network parameters in order to optimize the delivery of the content. For this reason, we selected a weight of 0.5 for the QoS-related variable “stream\_flag.”
- *Subjective:* A user who is confident with the service is likely to recommend the service to other potential customers of the service. For this reason, we consider that the “service\_confidence” variable is very important for the analysis, choosing a weight of 1.0 for it.
- *Business-related:* Finally, the loyalty level of the customer is directly related with the revenues of the business; loyal users spend more money on subscriptions. Therefore, this variable becomes critical when optimizing both the QoE and the revenues of the business. In this sense, we consider the weight of this value to be set to 1.0.

Using the proposed weights, the QoE generic linear model shown in Equation 3.4 can be expanded as follows:

$$QoE_{MOS} = 0.5 \cdot \text{stream\_flag} + 1.0 \cdot \text{service\_confidence} + 1.0 \cdot \text{price\_premium} \quad (3.5)$$

---

### Methodology Outline 3 Evaluating the QoE of the OTT service

---

*Step 1:* Generate a written description of the service.

*Step 2:* Identify system states.

*Step 3:* Identify system transitions, inputs and outputs.

*Step 4:* Identify transitions where variables are updated.

*Step 5:* Insert parameters as context variables and their updating functions.

*Step 6:* Unroll the machine to its  $l$ -equivalent. This is done by fixing the desired length  $l$  and inserting constraints in the loops of the EFSM. Then the states and transitions are expanded to form an acyclic directed graph.

*Step 7:* Apply a QoE model (function returning the QoE value) at final states. The approach proposed here does not force to use any fixed QoE model to predict the QoE, so any mathematical relation that maps the involved parameters into the QoE value can be used.

---

In Section 5.2.2 this generic linear model will be expanded in order to perform a deeper analysis when validating the whole QoE framework.

The model proposed in Equation 3.5 is applied at each terminal state of  $l$ -equivalent tree, using the configuration of the machine computed at those states. This procedure is explained using the Figure 3.7. As the  $l$ -equivalent model is built using an iterative procedure, any shorter path contained in the tree is also a path of a shorter  $l$ -equivalent model. Using this fact, it is possible to compute the QoE value at intermediate states in order to compare how does this value evolve when considering longer paths.

In Figure 3.7, the vector  $\mathbf{v} = (\text{stream\_flag}, \text{service\_confidence}, \text{price\_premium})$  shows the valuation of the context variables at the beginning of the path and each time a value changes. In addition, three intermediate states have been marked ( $A$ ,  $B$  and  $C$ ) in addition with the terminal one ( $D$ ). Each one of these states can be seen as a terminal state of the 3-, 7-, 8- and 11-equivalent model, which has been constructed following the iterative process described in the Methodology Outline 2. At each of the marked states, the QoE model specified in 3.5 was applied, using the computed configuration for the terminal state. The results of the computation can be seen in Table 3.3.

Table 3.3: Parameter values and QoE values at each state

State	$l$ -equivalent length	Parameter Vector $\mathbf{v}$	QoE value
A	3	(0,1,2)	3.0
B	7	(1,2,2)	4.5
C	8	(0,1,1)	2.0
D	11	(0,2,1)	3.0

### 3.4 Discussion

The results presented in Table 3.3 show the main utility of the proposed approach. By evaluating the QoE values at intermediate states of the whole path, it is possible to observe how these values evolve while unrolling the machine longer. With this information it is possible to analyze the whole path, in order to notice changes on the QoE value. These changes can be backtracked in order to find which transition lowers the quality value the most, which will lead to identify possible problems on the service model.

As for the running example, we can observe how the QoE value changes across the whole user history. When this user is introducing his card data (state  $A$ ), their QoE value is 3.0, which corresponds to an average value according to the MOS scale. This is due to the choice of a subscription type that requires engagement, which changes his price\_premium value from 1 to 2.

When the user starts to watch the stream, the service delivers the content seamlessly (state  $B$ ), which raises their service\_confidence from 1 to 2. This has an immediate impact on the QoE value, which goes up to 4.5 at this state.

However, in the next transition the service decides to cut service since the minimal network requirements are not met. This fact lowers their service\_confidence – from 2 to 1 – and their price\_premium – from 2 to 1, moving the machine to the state stop\_stream (state  $C$ ). In this scenario, the user has a QoE value of 2.0 which corresponds to a “bad” quality on the MOS scale. In this sense, the EFSM allows tracing back the origin of this reduction of the quality value, in order to draw conclusions about how the service behaves under certain scenarios.

Finally, the user decides to start the streaming once more. For this, they execute two transitions, going to state “service\_ready” and then to “stream\_delivery”. At this point, the OTT service checks the network conditions and decides to continue with the service, which raises the value of the service\_confidence from 1 to 2. At this point, the user has a valuation of  $v = (0, 2, 1)$ , corresponding to a QoE value of 3.0; a medium value for the quality.

This types of temporal analysis represents the main advantage of the proposed approach. By studying how do the QoE values evolve, it is possible to determine which are the transitions that highly impact on the quality experienced by the user. Knowing this crucial information is useful for the service provider in order to make improvements either to the offers or the service itself.

### 3.5 Conclusions

This chapter presented a novel QoE evaluation framework based on the mathematical formalism of the EFSM. The approach takes advantages of the extensibility and tracing capabilities of the EFSM in order to include the effect of business-related variables in to the QoE analysis.

In opposition to traditional QoE evaluation techniques, the proposed approach has the advantage of keeping track of the user’s decisions in order to consider its influence on their expectations. Using this feature of the tracing models, it was possible to include the impact of business variables – formally, monetary decisions – into the QoE analysis of an OTT service.

The proposed framework is formed by three main, incremental procedures. First, a model of the OTT service is elaborated from the point of view of the user, which guarantees the capturing the functional behavior of the service observed from the user user. This allows the module to capture the behavior of the service that potentially influence the QoE value. Second, the model is augmented by introducing *quality parameters* as context variables – and their respective updating functions – of the machine, extending the features of the model with ability of tracing the quality factor influencing the QoE. Finally, the EFSM is unrolled into its  $l$ -equivalent model, revealing all the possible paths – user scenarios – captured in the model. On these scenarios an appropriate QoE mathematical model is applied in order to obtain the final QoE value for each computed path.

This methodology has the advantage of considering the temporal effects of user’s decisions into the experienced quality by the user. In addition, the traceability features provided by the EFSM mathematical model allows identifying paths were low quality values are found and tracing back the transitions that highly impact the QoE value. With this information, the service provider can take decisions about the service itself, identifying possible flaws on the business decisions or technical parameters. This information is crucial to optimize both the investments made on infrastructure and the revenues of the OTT service.

---

---

# Chapter 4

## Implementation of the Quality of Experience Evaluation framework

### Contents

<b>4.1 The Montimage Monitoring Tool (MMT) . . . . .</b>	<b>43</b>
4.1.1 Architecture . . . . .	44
4.1.2 MMT Correlation Engine . . . . .	45
<b>4.2 MMT Extension with the Quality of Experience Evaluation Framework . . . . .</b>	<b>47</b>
4.2.1 <i>L</i> -equivalent Derivation Algorithm . . . . .	48
4.2.2 Quality of Experience Computation Algorithm . . . . .	49
<b>4.3 Applications of the Algorithms . . . . .</b>	<b>49</b>
<b>4.4 Conclusions . . . . .</b>	<b>50</b>

---

In the Chapter 3 the novel QoE evaluation framework was described in detail, including the three principal procedures required to apply the methodology. A further expansion of this research is the implementation of this methodology using a suitable tool, however it is required to find such tool in order to extend it with the proposed functionalities.

This Chapter presents an implementation of the proposed QoE evaluation framework using the Montimage Monitoring Tool (MMT), a modular, extensible monitoring tool used to verify properties of network protocols. Nevertheless, this tool is flexible enough to be used with other purposes than testing network protocols, making it suitable for implementing the proposed technique.

### 4.1 The Montimage Monitoring Tool (MMT)

The Montimage Monitoring Tool (MMT) is monitoring tool based on the DPI/DFI technique for verification and validation of network protocols and flows [87]. It provides real-time access to statistics about the traffic crossing over the network, applications communications, flows, and usage levels. These features make MMT a useful tool to enhance network security, monitor network performance and troubleshoot network operation.

Its working is based on the identification and generation of network events, which trigger the verification and testing features in order to detect operational, security and performance incidents or generate new events in the tool.

---

### 4.1.1 Architecture

The architecture of the MMT tool is organized in four main components, three of them developing incremental work in a sequential manner. The main organization of these components is presented in Figure 4.1.

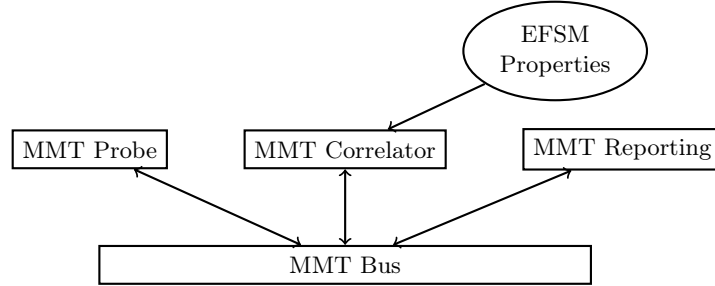


Figure 4.1: Architecture of the MMT solution.

In the Figure above we can observe how the main components of MMT interact between them. Each component has a specific function in the whole architecture, working together in a complimentary way:

- *MMT Bus*: This is the shared communication channel of all the components of the solution. This module is used to transmit messages and events from one component to another. This module has been implemented as an internal message service running on the loopback interface of the host machine. Other modules of the solution connect to this service and can subscribe to published messages on the channel.
- *MMT Probe*: This provides the core packet processing module to the tool. It was created as a C library that relies on DPI/DFI techniques in order to obtain information about the network or the application being analyzed. This module has the ability to inspect and analyze protocol fields values, QoS-related network parameters and Key Performance Indicator (KPI). In addition, this tool was designed to work both as an online monitoring tool or by analyzing structured information generated by the application or protocol under study – e.g. traces, logs. Finally, this module also includes a plugin architecture that allows the expansion of the capabilities in order to analyze new protocols and applications, as well as an open Application Program Interface (API) that allows the integration of third-party probes.
- *MMT Correlator*: Is the analysis engine based on formal properties to be tested, which are expressed as EFSM. This module checks the properties with the information extracted by the MMT probe engine, detecting operational and security issues or generating new events.
- *MMT Reporting*: This is the front end of MMT. The statistics and data generated by the Probe and Correlator modules are aggregated and shown using a graphical, web-based user interface. This modules is conceived to be customizable, that is, the user can select which data must be reported, as well as developing their own plugins to present more information.

The architecture described above and the capabilities of MMT make it a suitable option to be extended with the functionalities of the proposed framework. More in detail, the implementation takes advantage of three main characteristics of the Montimage Tool in order to expand its functionalities with new QoE prediction potential.

First, the probe module is capable of extracting the QoS parameters from the network, information that can be used as the QoS-related – objective – parameters. The framework can take advantage of the online monitoring capabilities of MMT in order to be used to predict further scenarios and their respective QoE value.

Secondly, the Correlator engine already provides support for EFSM properties, following the Javascript notation used by this module. In this way, it is possible to express the OTT model as an EFSM property of MMT. Later, the Correlator engine will use this specification in addition with the data extracted by the Probe module in order to apply the proposed methodology. Further details about this module will be provided in the next Section.

Finally, the Report unit can present the computed information to the user. In particular, the information related with the possible future scenarios and the QoE value is really valuable in order to take decisions with the analyzed service.

#### 4.1.2 MMT Correlation Engine

The MMT Correlation Engine is an independent module of the MMT solution in charge of processing and checking security and operational issues on protocols and services. This checking procedure is performed by testing formal properties expressed as EFSM, which are simulated and executed by the module.

It is implemented on framework *Node.js*, a Javascript runtime environment based on Google Chrome's V8 Javascript Engine. The base framework has been designed as an asynchronous, event-based Javascript runtime, allowing the execution of multiple concurrent requests without impacting the performance significantly<sup>1</sup>. This is provided by the non-blocking design for handling events; the programmer executes the node.js script by declaring callbacks that do not require a blocking function call in order to wait for the events – as traditional event-driven frameworks<sup>2</sup>.

This asynchronous nature of the events provided by Node.js is the main feature of which the MMT solution takes advantage in order to provide its functionality. Considering the underlying framework, the MMT Correlation Engine organizes its functionality following the architecture presented in Figure 4.2.

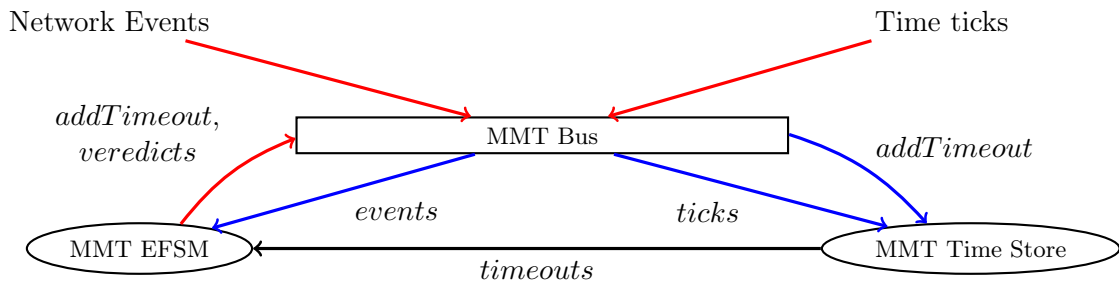


Figure 4.2: MMT Correlation Engine Architecture

<sup>1</sup><https://nodejs.org/en/>

<sup>2</sup><https://nodejs.org/en/about/>



The general architecture of the MMT Correlator engine is based on the event-driven capabilities of the Node.js framework. In Figure 4.2, red arrows represent the publication of events on the common bus, blue arrows denote subscription to the events and black arrows designate direct call between entities of the engine. Following this notation, the module works taking advantage of the published events and the flexibility of Node.js' callbacks.

In the upper part, two main event types are published on the MMT Bus module. First, MMT Probe analyzes the traffic – or the capture file – generating and publishing events with the information extracted from the analyzed packets. Second, time emulation is provided by programming a function invocation each second using Node.js API. This programmed invocation will publish the “tick” event on the MMT bus, emulating time effects on the machine which add support for EFSM with timeouts.

In the lower part of the Figure, the two main components of MMT Correlator engine are shown: MMT EFSM and MMT Time Store. The first one is an abstract class that is extended with the specification of the EFSM model representing the property to test. The second is the object that keeps track of the time and trigger timeouts on the machine if they are required. These two components are subscribed to their corresponding events on the MMT bus: (1) the EFSM class is subscribed to the events generated by the MMT Probe module – which trigger the transitions on the model – and (2) the Time Store class is subscribed to the “tick” events, in order to detect timeouts and take the corresponding actions with the machine.

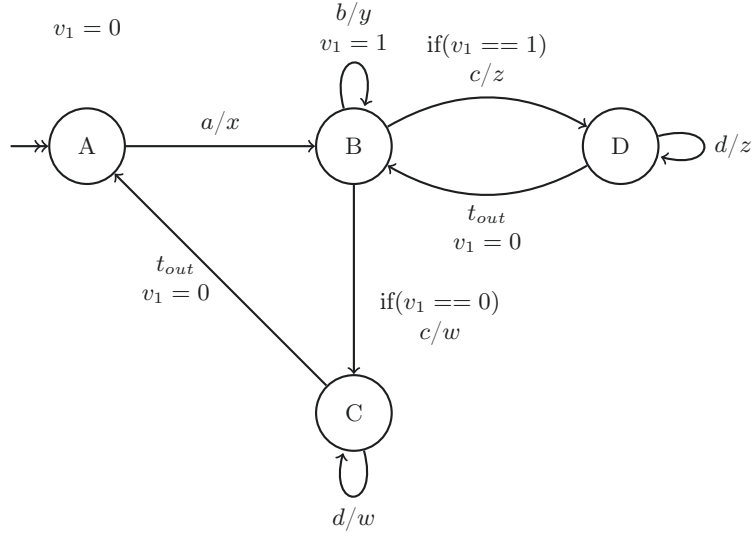
Setting up the timeouts is an action that must be triggered by the machine while emulating its execution. In order to do this, the machine publishes an “addTimeout” event on the MMT Bus with the information required to create the timeout. This information is captured by the MMT Time Store class – which is subscribed to this type of events –, triggering the required procedure to set up the timeout and start the timer. Once the timer has expired, the Time Store checks the execution state of the EFSM, and triggers the timeout if required by directly calling the handling function.

Finally, the MMT EFSM was introduced in order to test properties of the protocol or service under study. Once the property has been checked, a verdict must be emitted in order to inform the status of the check performed. This is done by the EFSM itself emitting a new event on the MMT bus, which will be later captured by the MMT reporting module in order to inform the user about the check result.

In order to exemplify how the MMT is used, we present a running example about how the machine in Figure 4.3 is expressed in the tool, which uses a single context variable  $v_1$ . In Appendix B.1 it is shown how this machine is expressed using the Javascript notation used with MMT Correlation engine. This specification is used to emulate its execution using the events generated by the MMT Probe module. In the same Javascript specification, we can observe that each transition is associated with a particular network event that will trigger it.

Finally, the MMT software also extends the capabilities of EFSM with timeouts, which allows the automatic trigger of a transition after a fixed amount of time has passed without changing the state of the machine. In Figure 4.3, the transitions from states  $D$  to  $B$  and from  $C$  to  $A$  are guarded by timeouts. These timeouts are expressed in lines 57 and 68 of the Appendix B.1 respectively.

---

Figure 4.3: EFSM  $M_3$ 

## 4.2 MMT Extension with the Quality of Experience Evaluation Framework

In order to take advantage of the features presented above, it is required to integrate the QoE evaluation framework presented in Chapter 3 into the MMT solution. The proposed framework can be summarized into the Methodology Outline 3 presented before, which can be decomposed into four main steps: modeling, augmenting,  $l$ -equivalent, and QoE computing.

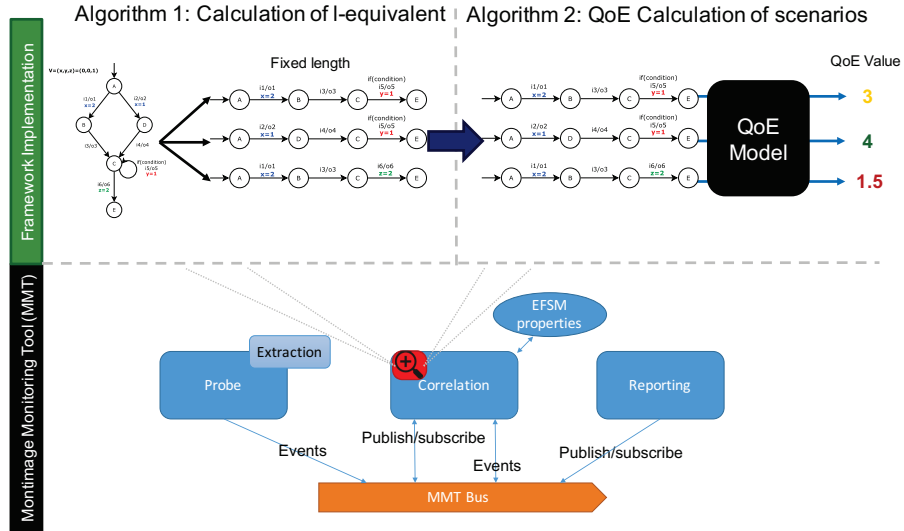


Figure 4.4: Architecture of the extension to MMT.

The first two steps of the procedure are related with the modeling procedure of the OTT service, therefore, they must be performed in a semi-automatic manner. However, and since the last two steps are mainly a transformation of the model and its use to compute the QoE value,

they can be executed automatically. In this sense, two main algorithms were developed in order to facilitate the generation of the  $l$ -equivalent model and the computation of the QoE for each computed path.

In the same way as the proposed methodology, both developed algorithms are meant to be used in an incremental way. The  $l$ -equivalent generation algorithm will compute the traces – user histories – using the Javascript specification of the OTT model. This will be the main output of this algorithm and at the same time, one of the inputs of the second. The QoE computing algorithm will take this traces as an input in addition with the pertinent QoE model that will be used to compute the quality value. This model will be expressed as a Javascript function that uses the values of the context variables in order to return a the QoE value at each state of the trace. This information will be the main output of the methodology, which can be encapsulated into new MMT events and sent to the MMT bus.

Finally, and in order to extend the functionalities of MMT with the QoE capabilities of the proposed framework, both algorithms were developed as an extension of the MMT Correlation engine, following the architecture shown in Figure 4.4.

#### 4.2.1 $L$ -equivalent Derivation Algorithm

The first algorithm is in charge of computing the  $l$ -equivalent model of the OTT EFSM. This is done by transversing the graph using the *Depth First Search (DFS)* technique [17], guaranteeing the construction of the path in a incremental, recursive way.

The pseudo-code of the procedure is shown in Algorithm 1. In the same way the original DFS, this is a recursive algorithm that will stop the calculations once the required length has been reached. This is the base case of the algorithm, where it simply returns a tuple with the state reached and the vector of context variables' values.

For the inductive step, the algorithm analyzes each outgoing transition of the actual state and determines, computing the final traces following a bottom-up procedure. First, it checks if the conditions to execute this transition stand or not. If so, the values of the context variables are updated using the corresponding updating

---

**Algorithm 1**  $l$ -equivalent generation

---

```

1: function BUILDPATHS(state, len, vars)
2:   if len ≤ 0 then
3:     return {(state, vars)}
4:   end if
5:   paths ← {}
6:   for all t in transitions[state] do
7:     step ← (state, vars)
8:     stand ← true
9:     for all c in t.conditions do
10:      stand ← stand && c(vars)
11:    end for
12:    if !conditionsStand then
13:      continue
14:    end if
15:    for all f in t.updateFunctions do
16:      for all var in vars do
17:        if var.name == f.name then
18:          var.value ← f.function(var.value)
19:        end if
20:      end for
21:    end for
22:    lowerPaths ← buildPaths(t.to, len − 1, vars)
23:    for p in lowerPaths do
24:      paths ← paths ∪ {[p, step]}
25:    end for
26:  end for
27:  return paths
28: end function

```

---

functions. Then, the procedure is

called recursively, starting from the ending state of the transition being analyzed, using the updated vector of the context variables and reducing the required length by 1. Finally, the partially computed traces are augmented with the actual configuration. This is done by concatenating a tuple to each partially computed trace, which contains the actual state and the context variables' values at this state. This last concatenation ensures the construction in a bottom-up manner, completing the process when the last recursive call – the last outgoing transition from the root node – has returned.

The result of applying this algorithm is a Javascript array that contains every computed path. Each trace is represented by an array of Javascript objects, representing each one of them a step of the trace. At each step, it is possible to find the state of the step and the values of the context variables at that state. As an example of this representation, in the Appendix B.2 we can find the Javascript representation of the computed paths of length 4 of the machine previously presented in Figure 4.3.

#### 4.2.2 Quality of Experience Computation Algorithm

The structure returned from this algorithm is then augmented with the information of the QoE values of each path. This computation is done by the Algorithm 2.

The second algorithm traverses all the previously computed paths, applying the QoE model expressed as a Javascript function. For each path, the algorithm identifies each step of the path and the values of the context variables at that state. Then, the given quality model is applied using the variables' values, obtaining the QoE at the analyzed step. The output of

---

#### Algorithm 2 QoE augmentation of the paths.

---

```

1: function EVALQOE(traces, qoeModel)
2:   results  $\leftarrow$  { }
3:   for all trace in traces do
4:     trf  $\leftarrow$  []
5:     for all s in trace do
6:       qoeVal  $\leftarrow$  qoeModel(s.vars)
7:       trf  $\leftarrow$  trf  $\cup$  (s.state, s.vars, qoeVal)
8:     end for
9:     results  $\leftarrow$  results  $\cup$  {trf}
10:  end for
11:  return results
12: end function

```

---

this algorithm has the same information as the Algorithm 1 that has been augmented with the field “QoE”. An example of this can be shown in the Appendix B.3, where a simplistic formula – adding 1 to the  $v_1$  variable – has been used to compute the “QoE” of the trace.

### 4.3 Applications of the Algorithms

As mentioned before, both algorithms presented above were implemented as an extension of the MMT Correlator engine. This feature allows the usage of the algorithms with the monitoring techniques provided by the MMT software.

In Figure 4.5 it is presented the architecture of a tool that uses the presented algorithms in order to predict the QoE of an OTT service under analysis. The proposed architecture takes advantage of the monitoring techniques provided by MMT Probe in order to draw conclusions and make recommendations to the service provider.

In the Figure it is presented the main workflow of the tool, showing the main inputs and outputs. As stated before, the proposed tool is implemented using the MMT solution as the underlying technology, which uses the MMT Probe module to analyze the live traffic of the

---

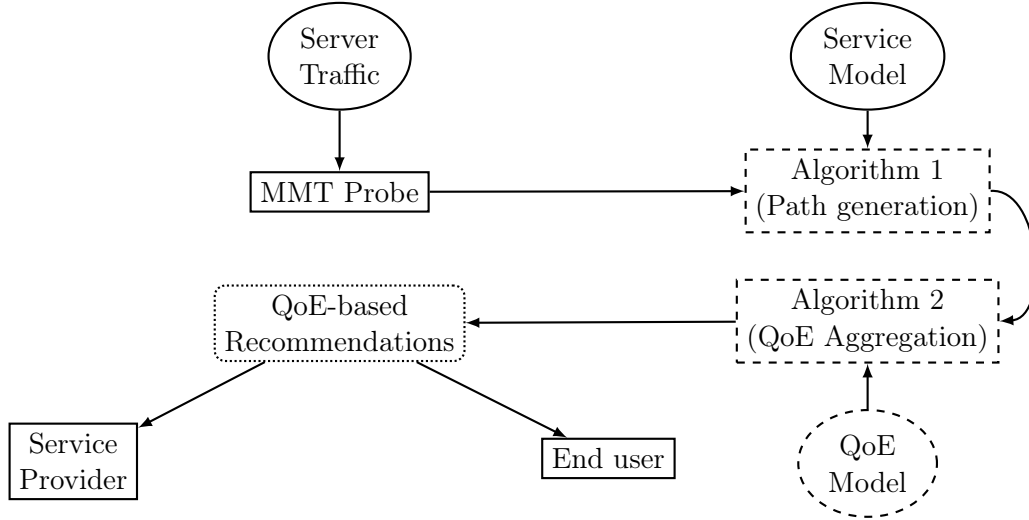


Figure 4.5: Architecture of the proposed monitoring tool.

server. By using DPI methodologies, it is possible to identify the values of the context variables introduced in the service model.

This extracted information, in addition with the Javascript representation of the OTT model, is the main input of Algorithm 1. With this data, the algorithm is capable of computing the future scenarios – also referred as paths – for the user under analysis for a fixed number of steps, including the future values of the context variables for each scenario.

Later, this information is augmented with their respective QoE values, which is computed by the Algorithm 2 also implemented in the MMT Correlation engine. This requires the usage of an appropriate QoE that maps the values of the context variables into a single quality value. In Section 5.2.2 it is discussed how this model can be elaborated by considering the effect of each context variable into the QoE value.

After applying both algorithms, the monitoring tool has computed the possible future scenarios for the user and their respective quality values. This information can be useful for both the service provider and the final customer of the OTT service.

On one hand, the analysis of the future paths can be used by the service provider in order to identify configurations where the user is unsatisfied with the service. This information can be used by the service provider in the sense of redirecting their efforts and resources aiming to improve the quality on these scenarios.

On the other, this type of analysis can also be useful for the user. By knowing the possible future configurations of the service and their respective QoE value, it is possible to build a recommendation system that takes this information into account and avoid the user choosing a configuration that it is known to have a low quality value.

## 4.4 Conclusions

In this chapter it was explained the implementation of the QoE evaluation framework presented in Chapter 3. This work was done by extending the capabilities of the MMT software, a monitoring tool designed to take advantage of the DPI/DFS techniques in order to test security and operational features of the protocol or stream under analysis.

The architecture of MMT solution – in particular the MMT Correlation Engine, written using the *Node.js* framework – allowed the implementation of the proposed framework as an extension of the API offered by this software. In particular, the implementation of the QoE framework benefits from two main features of the MMT Correlator Engine. Firstly, the usage of *Node.js* as the base technology, allowed representing EFSM using the a Javascript notation, which is a clear an simple way to specify data. Secondly, the module already provided the support for expressing the properties under testing by using the EFSM mathematical formalism, which reduced the implementation to be an extension of the existing software and expanded the capabilities of the whole MMT software.

Having this in mind, the implementation was formed by two main algorithms written to work with the MMT's EFSM. The first one is in charge of transforming the OTT model into its  $l$ -equivalent machine, exposing all the possible paths starting from the root node. The second algorithm uses the previously computed paths in order to compute its QoE value. This is done by the application of a suitable QoE model that maps the values of the context variables of the model into a single quality value.

Using both algorithms, a novel monitoring technique was proposed. This new approach takes advantage of the prediction capabilities of the QoE evaluation framework in order to compute the future scenarios – configurations – of the end user. With this information, the proposed prediction tool is able to anticipate possible bad quality scenarios for the customer, which can be used with two main purposes. On one hand, this information can be used by the service provider in order to identify possible unsatisfied users in order to focus the investments on this cases. On the other, the final user can benefit from this prediction by using the information in order to guide their choices while using the service.

---



# Chapter 5

## Quality of Experience Framework Validation

### Contents

<b>5.1</b>	<b>Implementation of an Over-The-Top Emulation Platform . . . . .</b>	<b>54</b>
5.1.1	Over-The-Top Emulation Platform . . . . .	54
5.1.2	Perturbed Video Generation Tool . . . . .	56
<b>5.2</b>	<b>Preliminary Refinements . . . . .</b>	<b>60</b>
5.2.1	OTT Model Enhancement . . . . .	60
5.2.2	Enhancement of the Quality of Experience model . . . . .	61
5.2.3	Evaluation of the Videos . . . . .	64
<b>5.3</b>	<b>Results and Model Validation . . . . .</b>	<b>66</b>
5.3.1	Context Variables Validation . . . . .	66
5.3.1.1	Business Parameter Refinement . . . . .	69
5.3.1.2	Subjective Parameter Refinement . . . . .	70
5.3.2	QoE Model Validation . . . . .	71
5.3.2.1	QoE Model Adjustment Discussion . . . . .	75
5.3.3	QoE Model Testing . . . . .	76
<b>5.4</b>	<b>Conclusions . . . . .</b>	<b>76</b>

In Chapters 3 and 4, the new approach for computing the QoE of an OTT service was introduced and implemented. At this state, the framework is ready to be used in combination with real OTT services, however, a step further needs to be taken before applying the approach. While the implementation of the framework is the last step to have a functional tool ready to be applied, some points of the proposed approach need to be validated before using them on real use cases.

The first element of the framework to be validated is the influence of the business variables. As stated in Chapter 2, the literature declare that business aspects can have an impact on the user's expectations and, therefore, in the perceived quality of the service. Despite the amount of works that recognize this influence, none of them clearly state a relationship between any business variable and the QoE value. Following this idea, it is required to evaluate the real effect of business decisions on the QoE as, for example, the effect of committing the user to the subscription for a fixed amount of time and their loyalty with the service. To achieve this goal, the same beIN Sports Connect Service model will be used, validating the introduced



business variable “price\_premium.” The main objective of this is to justify the introduction of the business variables as a real impact factor on the evaluation of the QoE.

The second element that requires validation is the QoE model. As the mathematical relationship between the context variables and the QoE value is the main element computing the final quality value, it is critical that this model needs to be appropriate for the service and the purposes of the tool. Being this said, the validation of this mathematical QoE model represents an important step for achieving the validity of the whole approach.

Having this goal in mind this Chapter presents the activities performed in order to validate the QoE evaluation framework previously presented. To achieve this, we follow the outlines recommended by the ITU organization [44]. First, an emulator of the analyzed platform – beIN Sports Connect – was implemented so that it mimics the behavior of the real OTT service, ask the user about the quality of the service and collect this information into a database. Second, a questionnaire was designed and added to the platform in order to validate the introduction of the business variables on the OTT model. Third, the videos used to test network impairments were prepared by using a specially developed tool in order to insert such perturbations. Then, the integrated platform was deployed in servers only accessible from a local network, in order to avoid further uncontrolled perturbations inserted by delivering content using Internet. Finally, the collected data were analyzed in order to draw the conclusions of the validation process, extracting information about the business variables and the QoE model used.

## 5.1 Implementation of an Over-The-Top Emulation Platform

The first step of the validation process is the implementation of an emulation platform that will be used as the base for showing impaired videos. The main goal of this step is having a close-to-reality platform that imitated the behavior of the real service under analysis; the beIN Sports Connect Service.

### 5.1.1 Over-The-Top Emulation Platform

In order to achieve the behavior of the real OTT service, two approaches can be followed. The first one is to analyze the service website directly, identifying which are the main steps that need to be accomplished in order to access the service. This approach is quite similar – if not identical – to the modeling process described in Section 3.3.1, hence, the service model already developed can be used to build such emulator.

In the particular case of the beIN Sports Connect service, we will use the model shown in Figure 3.6. By using the already developed model, the resulting emulator will only contain the functional requirements that are relevant for the undergoing analysis. In particular, the platform will mimic exactly the behavior – EFSM transitions – that can directly affect the QoE value, disregarding from the analysis the effects of unwanted factors.

By observing the Figure 3.6 and the written description presented in Appendix A, it is possible to specify the main features that the implemented simulation must have:

- *Home page:* The implemented service must have a home page, which acts as the presentation of the service.
- *Selection of subscription types:* In the home page, the emulator must offer the possibility to choose among the different types of subscriptions available. In particular, and as stated in the written description of the service, three types of subscriptions must be offered to

the user: no commitment subscription, 1-year commitment subscription and trial time activation.

- *Registration and payment forms:* The original beIN Sports Connect Service requires the registration of the user (asking personal information in addition with a valid email address) and the payment of the service (asking for credit card information if required).
- *Video stream delivery page:* After the registration and payment processes have been completed, the implemented service must be able to stream the videos using an HTML platform.

Considering these features, Figure 5.1 presents the architecture of the implemented simulator. It follows the same main behavior as the one represented in the EFSM from Figure 3.6, where some transitions have been extracted in order to simplify the implementation.

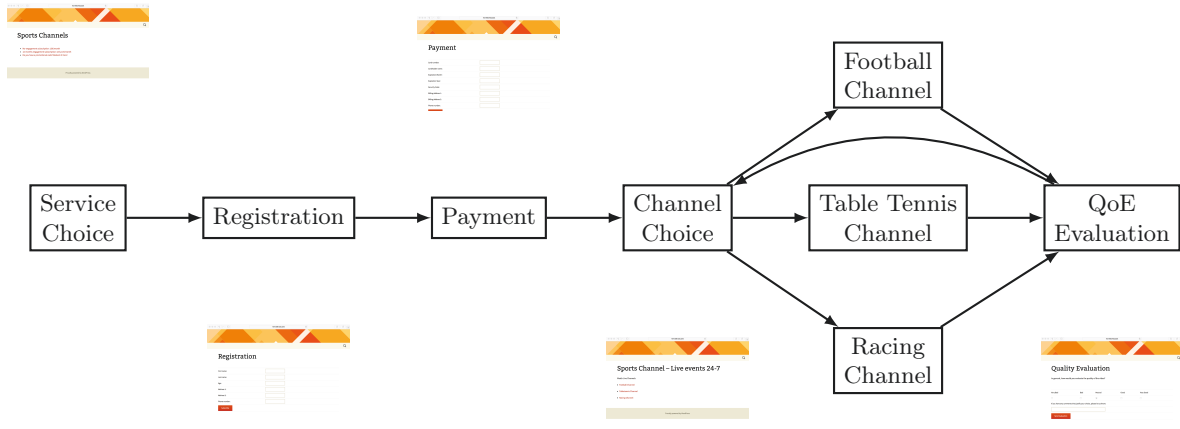


Figure 5.1: Architecture overview of the implemented OTT emulator.

As notes above, the main goal of this implementation is to provide a platform that mimics the main functionality of the beIN Sports Connect Service. Having this in mind, some transitions – as for example the loops on the registration and the payment states – have not been implemented in order to keep the model simple. In addition, the “stream\_delivery” state has been expanded in order to offer three types of contents rather than a single one: football, table tennis and racing. We implemented this in order to emulate the different live channels – with different contents – offered by beIN Sports. Finally, the “stop\_stream” state has been replaced by a QoE evaluation form, where the user can evaluate the video using the MOS scale and provide written comments if required.

This architecture was implemented using the Wordpress<sup>1</sup> blogging system, which allows an easy and flexible deployment of websites. In addition, this platform benefits from the HTML 5 technology, which offers native support for embedding videos on webpages<sup>2</sup>. In order to take advantage of this feature, the embedded videos need to be encoded using the h.264 video codec and encapsulated using the mp4 container. This ensures the visualization on most of the modern browsers and platforms.

<sup>1</sup><https://wordpress.org/>

<sup>2</sup>[http://www.w3schools.com/html/html5\\_video.asp](http://www.w3schools.com/html/html5_video.asp)

### 5.1.2 Perturbed Video Generation Tool

At this point, the platform is prepared to host the videos, however it is required to define which will be the disrupted scenarios that will be tested. In this sense, it is required to analyze the nature of OTT services in order to correctly define the variables that will be tested.

In Section 2.1 it was mentioned that OTT are – usually – multimedia services carried over third-party networks. These services are usually carried over the HTTP protocol, which runs over the TCP network protocol. This last protocol ensures the delivery of the data by providing techniques that acknowledges the reception of the data or retransmits the packet if it is required [61]. In order to test the appropriate network-related variables, it is required to consider such features when designing the test scenarios.

The literature propose multiple QoE models, being most of them expressed as a mathematical relationship. In particular, most of these models expose a mathematical formula that correlates network-related variables to QoE values, due the high level of influence of the QoS parameters into the perceived quality. However, in Section 2.2.1 we stated that these parameters can be extracted at two levels of the network.

On one hand, “lower”-level quality parameters are the ones that can be measured by observing the network flow on the routers. The extraction of this types of parameters does not require the use of DPI techniques, in other word, it is not required to know the content of the network packet, which does not interfere with the privacy of the user. Typical examples of these “low”-level parameters are the *packet loss*, the *packet jitter* (or *packet delay variation*) and the *network delay*.

On the other hand, it is possible to obtain more technical, QoS-related parameters that are related with the video flow but also directly impact the QoE. Such parameters require a deeper analysis of the network traffic in order to be computed, which justifies the usage of DFI and/or DPI techniques with the respective privacy trade-off. These variables are usually related with the multimedia flow, which can contain (but are not limited to) the *bitrate compression*, the *buffering time* and the *buffering frequency*.

The test platform was implemented using an HTML-based framework, and using the video playing capabilities offered by the HTML 5 technology. For this reason, all the data related with the video stream will be transmitted using the TCP transport protocol, which manages the effect “low”-level QoS parameters. Even when these variables can still be measured and considered into the QoE value, the use of TCP as the transport protocol generates effects on multimedia streams that makes the “high”-level parameters a better choice to consider.

First, the effect of packet loss is controlled by the retransmission feature of TCP. In case of packet loss on the network, TCP will retransmit the data when required, augmenting the apparent value of the delay of the network; since some packets need to be retransmitted, the client needs to wait for them in order to complete the receive window and make the data available to the multimedia application [61]. From the point of view of the user, they will experiment a buffering event followed by a continuation of the stream which will be delayed with respect to the original stream (as shown, for example in [23], [60]).

Secondly, delay introduces a retard on the arrival of the data to the client, which directly impacts the delay of the end played stream with respect to the source stream. In extreme cases of delay – or when the congestion of the network raises unexpectedly – the TCP protocol will adapt itself to this situation by changing the size of the reception window. In the first case, the user will see a small buffering event before continue the playback with the higher, expected delay. However, in a scenario where the amount of packets transversing the network rose, the

user will see a change on the video quality – if the video was encoded using the Scalable Video Coding (SVC) technique – or repetitive buffering events – if the video stream contains a single compressed bitstream – due the reduced bandwidth of the network [23].

In a similar way to the delay, the jitter has the effect of changing the delay of the the arrival of the packets to the client. In scenarios where the network jitter is high, the packets some packets will take too long to arrive to the client, causing retransmission of the data. This effect can be managed – usually automatically by the protocol – by augmenting the size of the TCP window if the bandwidth, which will allow TCP to wait longer for the packets. In both cases, the perceived effect by the user is an higher delay of the transmission with respect to the original source [23].

Finally, and as stated in [15], TCP, the cumulative effect of the three network parameters can be quite difficult to predict, but they “destroy the temporal relations shared between video frames and audio packets,” which is noticed as unsynchronized video and audio channels (as stated, for example in [46]).

The analysis exposed before allowed us to consider the “higher”-level parameters as the QoS-related parameters that will be tested with the videos. Formally, the perturbed videos will emulate impairments of *buffering* events and *bitrate* changes, due they are the most common perturbations experienced by the end user when watching H.264/AVC encoded video. This fact limits the variables that will be analyzed to the *encoding bitrate* of the video, the *buffering time* and the *number of buffering events*. Each one of these variables will be allowed to have fixed values following the next schema:

- *Video Bitrate Encoding*: 6Mbps, 3Mbps, 1500Kbps, 800Kbps and 400Kbps.
- *Buffering Time per Event*: 2, 3, 5, 8 and 20 seconds.
- *Total Buffering Events per Video*: 0, 1, 5 and 12.

In order to test the impact of these variables, a set of tests representing different scenarios of a real network was designed. In particular, the tests aim to cover scenarios with different available bandwidths, increasing and decreasing bandwidths, buffer preloading, packet loss and mixed perturbations. A complete list of the tested configurations is presented in Table 5.1.

In the Table 5.1, the test cases are highlighted according to the corresponding scenarios they represent:

- *No perturbation – Fixed bandwidth*: In red are shown the scenarios with a constant bitrate, simulating a networks that has a fixed amount of bandwidth available during the whole experiment.
- *Changing bandwidth*: In blue are represented six scenarios where the available bandwidth increases and decreases in different ways: slowly, at average speed and suddenly.
- *Initial buffering event*: In green are tested 4 cases of initial buffering time of the video. This case simulates the previous filling of the internal buffer usually used by OTT players.
- *Buffering and constant bitrates*: In orange are represented the cases where the congestion of the network is high enough to deplete the internal buffer of the player but not enough to lower the bitrate of the video. This is the case when the user forces certain video quality – for example while using YouTube – instead of letting the player choosing the optimal one.

Test Case	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5	Segment 6	Segment 7	Segment 8	Segment 9	Segment 10	Segment 11	Segment 12
Constantbitrate1	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Constantbitrate2	3M	3M	3M	3M	3M	3M	3M	3M	3M	3M	3M	3M
Constantbitrate3	1500K	1500K	1500K	1500K	1500K	1500K	1500K	1500K	1500K	1500K	1500K	1500K
Constantbitrate4	800K	800K	800K	800K	800K	800K	800K	800K	800K	800K	800K	800K
Constantbitrate5	400K	400K	400K	400K	400K	400K	400K	400K	400K	400K	400K	400K
Desc-bitrate1	6M	3M	1500K	800K	400K	400K	400K	400K	400K	400K	400K	400K
Desc-bitrate2	6M	1500K	400K	400K	400K	400K	400K	400K	400K	400K	400K	400K
Desc-bitrate3	6M	400K	400K	400K	400K	400K	400K	400K	400K	400K	400K	400K
Ascen-bitrate1	400K	800K	1500K	3M	6M	6M	6M	6M	6M	6M	6M	6M
Ascen-bitrate2	400K	1500K	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Ascen-bitrate3	400K	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Initial-freeze-time1	prebut2seg-6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Initial-freeze-time2	prebut3seg-6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Initial-freeze-time3	prebut5seg-6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Initial-freeze-time4	prebut8seg-6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Initial-freeze-time5	prebut20seg-6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M	6M
Freezing+bitrate1	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M	but2seg-6M
Freezing+bitrate2	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M	but3seg-6M
Freezing+bitrate3	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M	but5seg-6M
Freezing+bitrate4	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M	but8seg-6M
Freezing+bitrate5	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M	but12seg-3M
Freezing+bitrate6	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M	but13seg-3M
Freezing+bitrate7	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M	but15seg-3M
Freezing+bitrate8	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M	but18seg-3M
Freezing+bitrate9	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K	but2seg-1500K
Freezing+bitrate10	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K	but3seg-1500K
Freezing+bitrate11	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K	but5seg-1500K
Freezing+bitrate12	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K	but8seg-1500K
Freezing+bitrate13	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K	but12seg-800K
Freezing+bitrate14	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K	but13seg-800K
Freezing+bitrate15	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K	but15seg-800K
Freezing+bitrate16	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K	but18seg-800K
Freezing+bitrate17	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K	but2seg-400K
Freezing+bitrate18	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K	but3seg-400K
Freezing+bitrate19	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K	but5seg-400K
Freezing+bitrate20	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K	but8seg-400K
Freezing+bitrate21	but12seg-6M	but12seg-3M	but12seg-1500K	but2seg-800K	but2seg-400K	400K	400K	400K	400K	400K	400K	400K
Freezing+bitrate22	but13seg-6M	but15seg-3M	but15seg-1500K	but3seg-800K	but3seg-400K	400K	400K	400K	400K	400K	400K	400K
Freezing+bitrate23	but15seg-6M	but15seg-3M	but15seg-1500K	but5seg-800K	but5seg-400K	400K	400K	400K	400K	400K	400K	400K
Freezing+bitrate24	but12seg-6M	but15seg-6M	but15seg-6M	but8seg-6M	but20seg-6M	6M	6M	6M	6M	6M	6M	6M
Freezing+bitrate25	but12seg-3M	but15seg-3M	but15seg-3M	but8seg-3M	but20seg-3M	3M	3M	3M	3M	3M	3M	3M
Freezing+bitrate26	but2seg-1500K	but3seg-1500K	but5seg-1500K	but8seg-1500K	but20seg-1500K	1500K	1500K	1500K	1500K	1500K	1500K	1500K
Freezing+bitrate27	but2seg-800K	but3seg-800K	but5seg-800K	but8seg-800K	but20seg-800K	800K	800K	800K	800K	800K	800K	800K
Freezing+bitrate28	but2seg-400K	but3seg-400K	but5seg-400K	but8seg-400K	but20seg-400K	400K	400K	400K	400K	400K	400K	400K

Table 5.1: Video Test Scenarios.

- *Buffering and switching bitrates*: Finally, in purple are shown the cases that mix both buffering events and decreasing bitrates on the video. These cases represent a network highly congested where the initial quality cannot be maintained and the player decides to lower the quality in order to search for the optimal bitrate that does not significantly increases the congestion level of the network. This is the typical behavior of the TCP protocol.

In order to generate the perturbed videos that implement the proposed test cases, an automatic tool was developed in order to facilitate the generation of the final videos. In Figure 5.2, the main architecture of the video generation tool is presented.

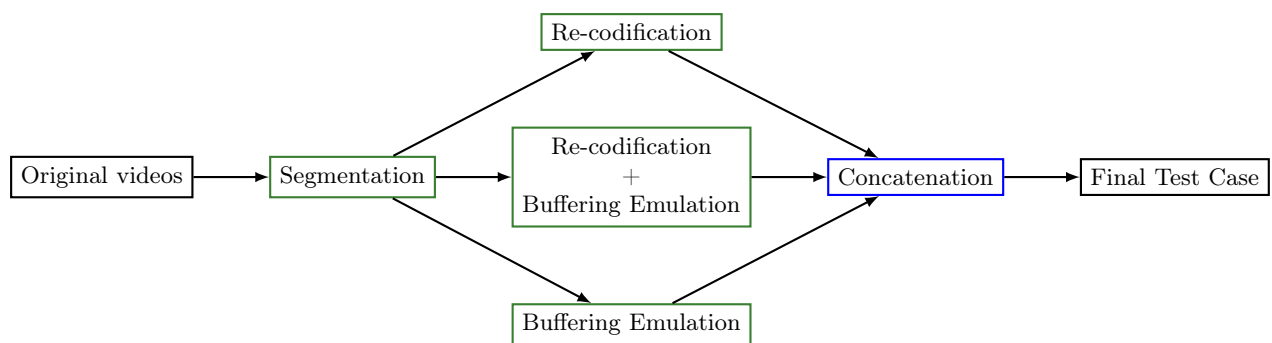


Figure 5.2: Architecture of the Video Generation Tool.

The tool makes use of the underlying library *ffmpeg*<sup>3</sup> in order to perform the different steps shown in Figure 5.2. In addition, these steps are separated in two scripts: “segment.sh” (shown in green) and “concat.sh” (shown in blue). The source code of both scripts can be found in Appendix C.

The first script dissects the original video in a number of segments passed as an argument of the script. Later, these segments are used to generate perturbed versions of these parts: first, the segments are re-encoded using the bitrates selected for the analysis; second, using the re-encoded parts, initial buffer times are simulated and inserted at the beginning of the initial segments of the video; third, using the values for the buffering times, video stallings are simulated and inserted at the end of the each segment, and; fourth, the mixed perturbations are generated, inserting freezing events at the end of each re-encoded part of the original video. The main output of this script is a set of video files that contains the perturbations that will be analyzed.

The second script uses these video files in order to assemble the final videos that will be embedded on the platform. This tool uses a Comma Separated Value (CSV) file which specifies the name of the case and which are the segments that will be used to generate the case. The Table 5.1 was generated from the real specification used by this tool. More in detail, the script analyses each line of the file, parsing the elements of the table and concatenating the corresponding segments using the *ffmpeg* tool.

<sup>3</sup><https://ffmpeg.org/>

## 5.2 Preliminary Refinements

Before deploying the platform and starting validation process of the framework, further refinements need to be done on the case of study. These modifications are composed of three main parts: the enhancement of the EFSM model, the enhancement of the mathematical QoE model used to compute the QoE and the evaluation of the quality of the videos generated.

### 5.2.1 OTT Model Enhancement

The first part of the validation process is the verification of the model, which comprises the features modeled in the EFSM and the impact of context variables on the expectations of the user and, therefore, on the perceived quality. The main goal of this procedure is to determine: (1) if the modeled – functional – features are in accordance with the expected behavior of the user, (2) whether the considered context variables impact the evaluation of the QoE or not, (3) the correctness of the context variables used with the beIN Sports Connect Service, (4) the correctness of the values used to measure each variable, and (5) the correctness of the updating functions for each parameter.

Once again, the validation process will be conducted for the context variables inserted in the beIN Sports Connect Service model, which acts as the running example on this research. In this model, three quality indicators were inserted into the model:

- *Subjective variable:* For the running example presented in Figure 3.6, the ‘service\_confidence’ variable was inserted to represent the subjective aspects of the QoE. This parameter represents the confidence level the user has with the service which is the semantics that will be tested in this section and what are the factors that impact on this confidence level.
- *Business variable:* In the beIN Sports Connect model, the business variable inserted was the ‘price\_premium’, which is an indicator of the extra price a customer is willing to pay for a brand, showing a loyalty level with the brand [8]. The validation process of this variable will be centered in measuring how the loyalty changes against different issues with the service.
- *Objective variable:* In the original model a single objective variable was inserted: the ‘stream\_flag’ variable. This parameter represents if the quality of the network – as computed by the stream player – is good enough to continue with the service or not. In this case, this variable will be expanded in order to have more control over the variables summarized into this single flag.

As mentioned before, a deeper analysis of the OTT model of the beIN Sports Connect requires an expansion of the ‘stream\_flag’ variable. This will be done by replacing the variable by other ones that can assist the analysis that will be performed. Naturally, the new variables inserted will be put with its respective possible values, transition guards and updating functions. Before going any further into the new expanded variables, it is required to have a deeper look about what does the ‘stream\_flag’ represent in the original model.

At the moment of modeling the beIN Sports Connect service in Section 3.3.2, the variable ‘stream\_flag’ was inserted in order to model a set of network-checking features performed by the player. These features are based on network parameters measurements in order to determine if they meet minimal level of quality to ensure the delivery of the service. Now that the

'stream\_flag' variable will be expanded, it is required to have a close look about which measurements are being performed to compute the value of this variable. However, the commercial nature of the service under analysis did not allow to know in details these characteristics.

In order to cope with these difficulties, the variables presented in Section 5.1.2 will be used to re-compose the 'stream\_flag' variable. This selection is based on a review of the literature, looking for the most influencing network parameters that impact on the QoE of the service. This is justified by the fact that the principal purpose of this parameter was to serve as an indicator to continue playing the video or not.

In summary, the 'stream\_flag' will be replaced with:

- *Video bitrate (variable 'bitrate')*: Measured in megabits (Mbps) or kilobits per second (Kbps). This variable directly represents the "quality" of the video from the point of view of the user. In a more technical way, it represents the minimal bandwidth required by the stream, which is directly related with the level of congestion of the network, namely the effective throughput of the TCP connection.
- *Buffering events length (variable 'buftime')*: Measured in seconds. The total length of the stalling events of the video stream represents the level of packet loss of the network and the available effective throughput of the connection.
- *Total number of buffering events (variable 'bufevnt')*: The last variable is the total number of freezing events. Connections with not enough bandwidth available and high packet loss percentages frequently require instances to fill the local buffers and continue the playing.

The newly inserted variables are in line with the effects simulated by the video generation tools previously described. With this change, the analysis will focus on "higher"-level objective parameters whose effect can be appreciated directly by the user of the OTT service modeled.

Considering the new variables presented before, the enhanced EFSM model for the beIN Sports Connect Service is shown in Figure 5.3.

Finally, the validation process of the features here presented was implemented by means of a survey formed by 15 questions. In order to guarantee that every user of the platform took the survey, this element was inserted before the main page was shown, ensuring that the answers are saved before the user starts using the service. The implemented survey can be found in Appendix D.

### 5.2.2 Enhancement of the Quality of Experience model

In Section 3.3.3.2 the QoE mathematical model applied with the framework must be in line with the quality factors inserted in the model. Considering the fact that the 'stream\_flag' variable was expanded by replacing it with three new ones, is it also required to adapt the QoE model to agree with the new parameters.

The original definition of the QoE model used is based on the Equation 3.4, which is already a particular instance of the more general model presented in [76]. The adaptation of the QoE model will keep the simplicity of the original one, but at the same time, it will provide a more close-to-reality approach to represent the impact of each factor on the final quality value.

The new QoE model will be based in the idea of splitting the 'stream\_flag' variable into the new variables, considering the different effects of the new parameters considered. In order to achieve this, it is required to know how each variable impacts the QoE, inserting the corresponding model for each indicator.



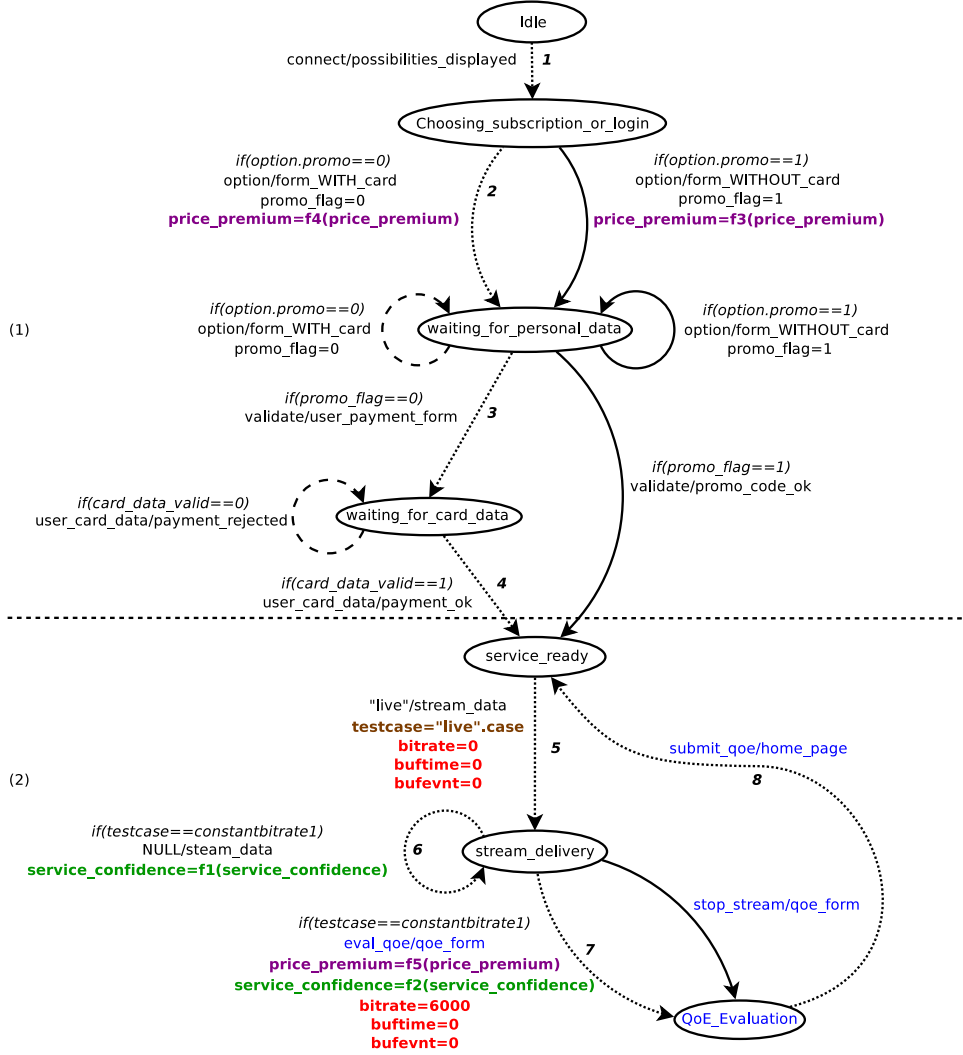


Figure 5.3: Expanded Model for the beIN Sports Connect Service

The first new variable inserted was the *bitrate encoding value*, which represents the “quality” of the video – low, mid or high resolution – from the point of view of the user. For this variable, the literature does not offer many conclusive studies that correlates the bitrate of the video and the MOS value of the QoE. However, it is possible to draw some conclusions about some studies that relate the acceptability rate of videos and their respective bitrate.

In [14] the authors analyze the influence of bitrates changes with the acceptability rate of the video, stating that small changes on the bitrate do not highly impact the QoE value of YouTube and Facebook videos. This fact is reaffirmed by Song et al. finding the same effects on more general video platforms whose content is distributed over mobile networks [79]

All the presented studies are in line with the work presented by Schatz et al., stating that typical logarithmic relationships can be found between QoE and application-related QoS parameters directly noticeable by the users [76]. Since the bitrate encoding is a variable that the user can notice directly, this parameter will be modeled by following a logarithmic relationship, i.e.:

$$MOS(B_r) = \alpha \cdot \ln(\beta \cdot B_r) + \gamma$$

where  $\alpha, \beta, \gamma \in \mathbb{R}$  are constants that need to be adjusted to the test running case of study.

In order to obtain a mathematical model that fits the bitrates of the running case of study, the values of  $\alpha$ ,  $\beta$  and  $\gamma$  will be adjusted considering: (1) minimal bitrates (400 Kbps in the implemented platform) should be associated with low QoE values; (2) high bitrates should map to high QoE values; (3) an “average” quality should be associated with an average QoE value. Considering these observations, the adjusted curve have the values  $\alpha = 1.3$ ,  $\beta = \frac{1}{1000}$  and  $\gamma = 2.4$ , whose plot is shown in Figure 5.4.

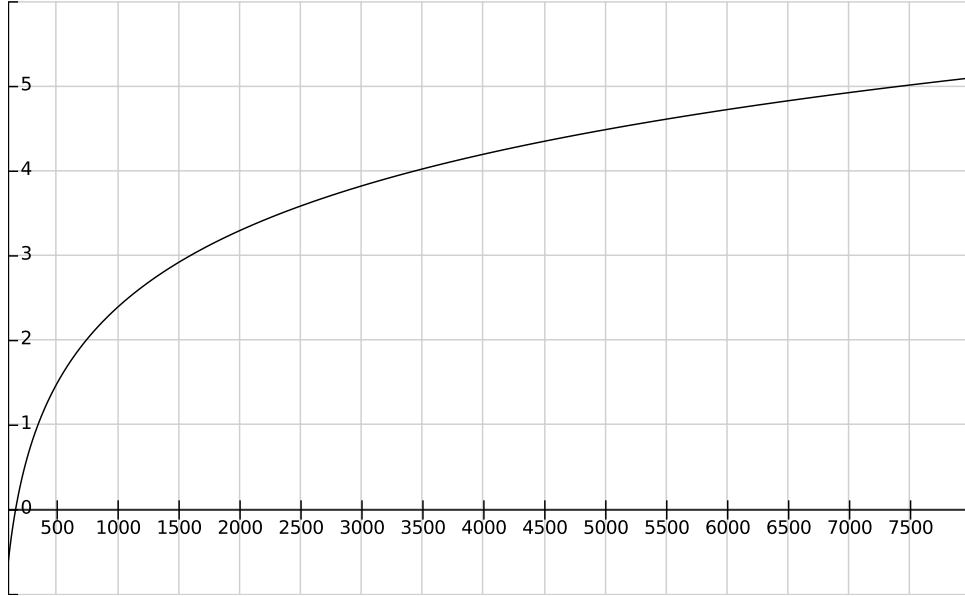


Figure 5.4: Plot of the adjusted impact of the bitrate on the QoE value.

The presented curve ensures the desired effect, granting QoE values near to 1 for 400 Kbps, 3 for 1500 Kbps, and 5 for 6000 Kbps. Additionally, this curve also ensures that initial changes on the bitrate are barely noticed by the user, but once the bitrate quality is already low, the QoE value lowers drastically. In summary, the bitrate QoE model that will be used is:

$$MOS(B_r) = 1.3 \cdot \ln \left( \frac{1}{1000} \cdot B_r \right) + 2.4 \quad (5.1)$$

Secondly, two variables related with the buffering times were inserted in the OTT model: the *buffering event length* and the *buffering frequency*. The effect of each variable has been studied in many ways. For example, in [34] the authors analyze the effect of stalling events (in length and frequency) for the well-known OTT service YouTube. In [24] the authors found that initial stalling – freezing – events do not impact significantly on the QoE, but the stalling events on other parts of the video do significantly impact the QoE of the videos. These observations are summarized in [77] stating that any stalling events should be avoided in order to maintain the perceived quality level. In this sense, in scenarios when bandwidth is a constraint, it is preferred to trigger a change on the bitrate encoding rather than letting the user experience buffering events.

All the exposed works find an exponential relationship between the stalling events on the video and the QoE value of the videos. This fact is confirmed in the general analysis of Schatz et al., who states that exponential relationships can be found between the QoE and QoS variables

that are not directly noticeable by the user [76]. As presented before, the stalling effects are mainly explained by congestion issues like packet loss or insufficient TCP throughput, which are, by definition, network-related QoS parameters. In this sense, an exponential QoE model will be used with this parameter, i.e.:

$$MOS(N, L) = 3.5 \cdot \exp(-(0.15 \cdot L + 0.19) \cdot N) + 1.5 \quad (5.2)$$

where  $N$  is the number of freezing events on the video,  $L$  is the length of each freezing event and  $N$  is the number of stalling events of the stream. This is the model proposed in [34] for the impact of stalling effects on YouTube (which runs over TCP connection) videos.

In summary, the expanded QoE model has the form:

$$QoE_{MOS} = \omega_{bitrate} \cdot MOS(B_r) + \omega_{stalling} \cdot MOS(N, L) + \omega_{subj} \cdot MOS_{subj} + \omega_{biz} \cdot MOS_{biz} \quad (5.3)$$

which now requires the proper adjustment of the weights. If we mix Equations 5.1 and 5.2 into 5.3 we have:

$$\begin{aligned} QoE_{MOS} = & \omega_{bitrate} \cdot \left[ 1.3 \cdot \ln \left( \frac{1}{1000} \cdot B_r \right) + 2.4 \right] + \\ & \omega_{stalling} \cdot \left[ 3.5 \cdot e^{-(0.15 \cdot L + 0.19) \cdot N} + 1.5 \right] + \\ & \omega_{subj} \cdot MOS_{subj} + \omega_{biz} \cdot MOS_{biz} \end{aligned} \quad (5.4)$$

The finding of the constants  $\omega_{bitrate}$ ,  $\omega_{stalling}$ ,  $\omega_{subj}$  and  $\omega_{biz}$  will be part of the validation process of the framework, which will now provide a proper QoE model to be used in real services. For this purpose, the recommended procedure proposed on [44] by the ITU will be used, which is based on adjusting the values of the weights from quality evaluations directly extracted from user. Finally, the required data to determine the weights will be extracted by using the emulation platform previously described in Section 5.1.

### 5.2.3 Evaluation of the Videos

The evaluation of the QoE values of the generated videos was done by using the same OTT emulation platform described in Section 5.1.1. To achieve this goal, it was required to insert small changes on the behavior of the original service, extending the capabilities of the platform with the collection of QoE-related data.

BeIN Sports Connect is a service conceived to deliver live streams of sports-related content. In this sense, the videos shown on the platform do not have a fixed length, which is not a desirable feature when evaluating the QoE. A first approach to fix this issue is limit the length of the shown videos to full sport matches, however the typical length of a full match does not allow to correctly evaluate the QoE value of the service. The ITU recommends different procedures for the evaluation of the quality, all of them using sequences no longer than 10 seconds [43]. This testing methods are not suitable for evaluating the quality of live services since they require the interruption of the service every 10 seconds in order to ask the user for the quality value. As a measure to tackle this issue, the proposed methodology was adapted to emulate a more close-to-reality scenario for live services.

First, the length of the videos presented was limited to 2 minutes. This limitation was imposed in order to provide test cases that were long enough to emulate a live service but, at

the same time, short enough to avoid a bias on the quality evaluation provided by the user. This sequences were segmented in 12 parts over which the impairments were emulated by the video generation tool presented in Section 5.1.2. The main objective of this modification is to generate perturbed video sequences suitable for emulating the behavior of a live OTT service.

Second, the original model presented in Figure 3.6 included a state called ‘stop\_stream’, which represented the state where either the service or the user has decided to conclude the stream of the service. In this work, the name and the semantics of this state was replaced with the ‘qoe\_evaluation’ state. These modifications included the change of the outputs of the incoming transitions – highlighted in blue in Figure 5.3 – which were changed from `NULL` to `qoe_form`. In addition to this, the platform was implemented to force the user to watch the entire video, who had to click on the link to go to the QoE evaluation form. This required modifications on the transition labeled as 7, which now has the input ‘eval\_qoe’. Finally, the input ‘home\_button’ of the outgoing from the new state ‘qoe\_evaluation’ has been changed to ‘submit\_qoe’ and is now parametrized with the variable ‘qoe\_value’.

Additionally, the expansion of the ‘stream\_flag’ variable required further modifications on transitions 5, 6, 7 and the outgoing one from ‘stream\_delivery’ with input ‘stop\_stream’. When the execution has reached the ‘service\_ready’ state, it means that the user is one step before watching a video. Once the user decides to watch one of the three possible channels by clicking on the “live” button, the system choses a random test case to be shown, which is represented as the ‘case’ parameter of the transition 5. This value is stored by the machine in the context variable *testcase* in order to keep track of the scenario tested.

The value of the test case is used in the transition 6, which has now expanded to a set of transitions all of them guarded by the value of the different test case. A single example of these transitions is shown in Figure 5.3 for the test case ‘ConstantBitrate1’. This distinction allows the introduction of different update functions for the different video scenarios in order to match how the changes of the quality affect the user’s QoE. Finally, once the video has finished, the transition 7 has to be executed, where the objective context variables are updated. When this transition is triggered, the value of the ‘bitrate’, ‘buftime’ and ‘bufevnt’ are set to the values they have in the last segment of the test case shown, preparing the machine for the ‘qoe\_evaluation’ state. All the aforementioned changes are reflected in Figure 5.3.

The introduced changes aim to allow the service to collect QoE-related data form the user about the presented video. In this sense, the new ‘qoe\_evaluation’ state is where the user is shown a form to evaluate the perceived quality of the video previously presented.

The evaluation of the QoE value is done by completing a form that shows 5 possible values for the video watched. This procedure is in line with the standard MOS scale followed by every perceived quality-related studies, procedure which makes part of the recommendation proposed by the ITU [40]. In addition, the form also shows a small text field where the user can leave any comment in order to justify their decision of the quality. A screenshot of the quality data collection form is shown in Figure 5.5

Once the prerequisites of the platform were met, the software was deployed in order to proceed with the experimental part of the validation process. In order to maximize the number of participants, the platform was deployed in two different research laboratories in France and Chile, being both of them network-related research laboratories.

157.159.100.224

## Quality Evaluation

In general, how would you evaluate the quality of the video?

Very Bad      Bad      Neutral      Good      Very Good

☐      ☐      ☒      ☐      ☐

If you have any comments that justify your choice, please let us know:

Send evaluation

Figure 5.5: Quality data collection form.

## 5.3 Results and Model Validation

### 5.3.1 Context Variables Validation

As mentioned before, the first objective of the validation process was the corroboration of both the context variables and the updating functions. To this end, the survey exposed in Appendix D was answered by the users of the emulation platform, recording their responses in a database.

A total number of 16 users provided their answers, and the results are summarized in Table 5.2.

By analyzing these data it is possible to draw some conclusion about the context variables and how they are updated.

First, it is possible to observe the high impact of the OTT services on the multimedia market: 50% of the surveyed users declare to use OTT services multiple times a week, 25% use them once a day, 13% use them multiple times a week, and 6% use them once a week or multiple times a month. In particular, 88% of the users use OTT services multiple times a week, showing the impact this marked has had on the users. In addition, the users of these services tend to keep using these services for long times: 94% of the users are customers since a year or more than a year. These two questions show the remarkable potential of the OTT services to catch the users and retain them as customers. The deployment of high bandwidth networks and the expansion of 4G mobile network have encouraged these raise.

In second place, three questions are related with the technical, noticeable parameters are present in the survey. On one hand, regarding the visual “quality” of the video, the answers are somehow equally spread across the highest values: 38% of the users prefer a DVD quality or better, 31% prefer HD video and 25% prefer Full HD streams. Despite this wide range of choice, it is possible to satisfy the 56% of the users by setting the video “quality” to HD.

On the other hand, the effect of the stalling events can also be analyzed from the survey.

Table 5.2: Summary of the validation answers

Question	Choices	Answers	Percentage
Do you use OTT Services?	Yes	16	100%
	No	0	0%
How often do you use OTT Services?	Multiple times a day	8	50%
	Once a day	4	25%
	Multiple times a week	2	13%
	Once a week	1	6%
	Multiple times a month	1	6%
	Once a month or less	0	0%
For how long have you been using OTT services?	Less than a week	0	0%
	Since a week or two	0	0%
	Since a month or two	1	6%
	Since half a year	0	0%
	Since a year	2	13%
	More than a year	13	81%
Do you consider yourself loyal?	Yes	11	69%
	No	5	31%
Do you like network checking feature?	Yes	11	69%
	No	5	31%
Does this feature impact your loyalty?	Yes	7	44%
	No	9	56%
Does it increase or decrease it?	Increase	6	37%
	Decrease	10	63%
Which video quality do you consider as satisfying?	6000 Kbps	4	25%
	3000 Kbps	5	31%
	1500 Kbps	6	38%
	800 Kbps	1	6%
	400 Kbps	0	0%
Which is a tolerable time for a freezing on the video?	2 seconds	5	31%
	3 seconds	8	50%
	5 seconds	3	19%
	8 seconds	0	0%
How many frezzings are tolerable in a 2-minutes video?	0	4	25%
	1	4	25%
	2	5	31%
	3	2	13%
	4	1	6%
	5	0	0%
Did you change OTT provider in the last two months?	Yes	1	6%
	No	15	94%
Do you feel committed with a minimal engagement time subscription?	Yes	6	37%
	No	10	63%
Does secure communications influence your trust or loyalty?	Yes	13	81%
	No	3	19%
Does it increases or decreases your loyalty?	Increases	13	81%
	Decreases	3	19%

Generally speaking 81% of the users can tolerate a 3 seconds freeze on the video in order to load new data on the player's buffer. We conjecture that this is the maximal length an OTT user can tolerate without significantly impact the QoE. This is in line with the model proposed by

Hoßfeld et al., stating that longer stalling events decrease faster the QoE value [34]. Regarding the number of stalling events a user can tolerate, 25% of the users prefer not experiencing any freeze event, however, 1 or 2 instances can still be tolerable for most of the users (81%). This result is also in line with the findings of Hoßfeld et al. showing that more buffering events heavily impact the QoE value [34].

In a similar way as the other dimensions, it is possible to draw some conclusions related with the business variables. First, 69% of the users consider themselves as loyal towards the service, which is in line with the fact that 94% of the total customers have not changes OTT provider in the last two months. Secondly, some important observations can be extracted by observing the answers related with the automatic checking feature of some OTT services, as it is the case of beIN Sports Connect. By observing the Figure 5.6 it is possible to extract important information about how the automatic check of the network feature impacts on the user loyalty.

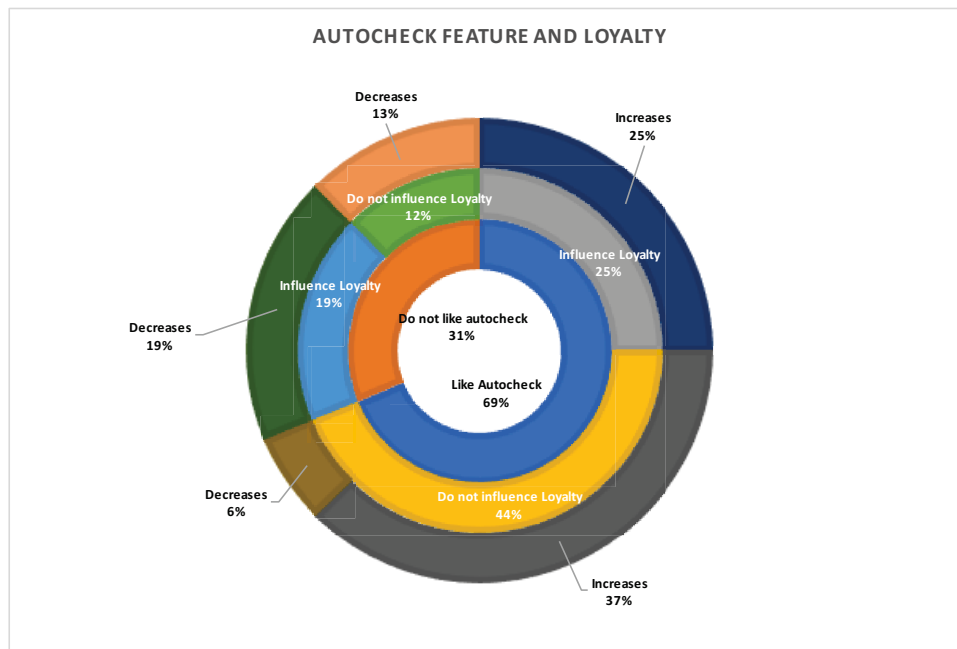


Figure 5.6: Network checking feature and loyalty.

On one hand, 69% of the users – 11 people – like the automatic check of the network conditions where 82% of those users – 9 people – consider themselves as loyal. In general, 44% of the users consider that this feature influence their loyalty and 56% consider not. A deeper analysis of these results show that 4 out of 11 people that are loyal consider that the network checking influence their loyalty, and it does *increase* their loyalty towards the service. From the other 7 people who consider themselves loyal, but do not like automatic check, 6 of them consider that this feature also *increases* their loyalty. These results are difficult to analyze, since it apparently contradicts the fact that the users do not like to experience stalling events, since these impairments are the most impacting events on the QoE [34]. In order to draw a valid conclusion, more users need to be surveyed in order to have a statistical relevance of the study.

On the other side, the user who dislike the automatic check feature, are equally spread about the influence of this characteristic on their loyalty, however, all of them believe that this behavior would decrease their loyalty with the service.

### 5.3.1.1 Business Parameter Refinement

Considering these observations a new  $M_{biz}$  indicator is introduced, which aims to measure the impact of the business variables – namely the subscription type and loyalty in form of the ‘price\_premium’ variable – into the QoE value. As before, the ‘price\_premium’ will be measured following the 3-values scale suggested by Aaker, which classifies the customer into three main groups: no customers, brand switchers and loyal users [8]. The classification of the user will use two data collected on the platform: the subscription type chosen by the user and their self loyalty consideration, process that will follow the logic summarized in Figure 5.7.

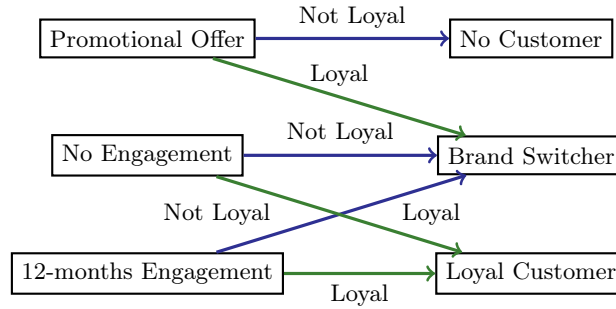


Figure 5.7: Mapping of the business-related context variables.

On the left side the subscription types are exposed, and they can be mapped to two different price premium categories. Green arrows represent the classification when the user considers themselves as loyal, while blue ones represent the opposite. As a way to exemplify this classification, a user who selected a subscription with no engagement and considers themselves as loyal, will be classified as a “loyal” user according to the ‘price\_premium’ scale.

Each one of the categories of the ‘price\_premium’ variable will be mapped into a QoE value following the logic shown in Table 5.3. The mapping tries to avoid assigning extreme QoE values, since they are not usually used when provided by real users.

Table 5.3: Mapping Price Premium categories to MOS values.

Price Premium Category	MOS Value ( $M_{biz}$ )	Meaning
No Customer	2	‘No customers’ can leave the brand easily.
Brand Switcher	3	‘Brand Switchers’ are neutral customers.
Loyal Customer	4	‘Loyal’ users are more tolerable with impairments.

Following the ideas extracted from the survey, it is possible to refine the definition of the updating functions that operate over the business parameter ‘price\_premium’. First, updating functions  $f_3$  and  $f_4$  in Figure 3.6 will be changed for a single new function  $f_{biz}^A(sub, loy) = M_{biz}$  that correctly assigns the  $M_{biz}$  variable according to the logic exposed in Figure 5.7.

In addition, by observing the results of the tolerable stalling events it is possible to determine that most of the users (more than 50%) consider a 3000 Kbps bitrate as “satisfiable”. Following these observations, the function  $f_5$  of Figure 3.6 will also be changed by a new function  $f_{biz}^B(M_{biz}, bitrate) = M_{biz}$ , which reflects this tolerance level. This function will act as a change of the user self-loyalty following the Table 5.4, which summarizes the effect of the video bitrate



on the user loyalty. These allocations are based on the fact that higher bitrates involve better “qualities” (usually HD-quality), while lower bitrates are considered unsatisfiable for most of the customers. Moreover, a neutral “quality” (referred as “DVD-quality”) will be associated with no changes on the loyalty, since a high amount of users (38%) consider it as satisfiable.

Table 5.4: Summary of the impact of video bitrate on user loyalty.

Video Bitrate (Kbps)	Old loyalty	New loyalty	Behavior
$\geq 3000$	No Yes	Yes Yes	Loyalty Increaser
1500	No Yes	No Yes	No changer
$< 1500$	No Yes	No No	Loyalty Diminishing

By mixing the behaviors of the MOS mapping of the price premium categories and the effect of the video bitrate, it is possible to provide new updating functions for the business-related variable. The definitions of these new functions  $f_{biz}^A$  and  $f_{biz}^B$  are given in Tables 5.5a and 5.5b.

Table 5.5: New definition of business-related updating functions.

(a) New definition of  $f_{biz}^A$ .

Subscription Type	Self Loyalty	$M_{biz} = f_{biz}^A(sub, loy)$
Promotional	Yes	3
	No	2
No Engagement	Yes	4
	No	3
Engagement	Yes	4
	No	3

(b) New definition of  $f_{biz}^B$ .

$M_{biz}^{old}$	bitrate	$M_{biz}^{new} = f_{biz}^B(M_{biz}^{old}, bitrate)$
2	$\geq 3000$	3
	1500	2
	$< 1500$	2
3	$\geq 3000$	4
	1500	3
	$< 1500$	2
4	$\geq 3000$	4
	1500	4
	$< 1500$	3

The values of  $M_{biz}$  will be computed using the data extracted from the emulation platform and the rules exposed here in order to find the weights of the adjusted QoE model.

### 5.3.1.2 Subjective Parameter Refinement

Finally, using the information related with the stalling events it is possible to extract conclusions about its impact on the subjective parameter. By observing the results of the applied survey in Table 5.2 it is possible to observe that most of the users – 81% can tolerate 1 or 2 stalling events in a two-minutes video, and each one of these events should not be longer than 3 seconds each one. Having this information in mind, the confidence level of the user with the service will be updated in order to model these observations.

First, the same three levels of confidence will be used, namely “not confident”, “neutral”, and “confident”. Following the same logic as for the business variable, each level will be mapped to a MOS value that will impact the final quality value of the user. The proposed mapping is shown in Table 5.6.

These categories will be used to provide an updated version of the ‘service\_confidence’ variable ( $M_{biz}$ ) and their respective updating functions. The updating functions  $f_1$  and  $f_2$  will be replaced for the new version  $f_{subj}$ . This function will consider the effect of the staling events in order to classify the user into a new confidence category, thus assigning a new QoE related with

Table 5.6: Mapping Service Confidence categories to MOS values.

Confidence Level	MOS Value ( $M_{subj}$ )	Meaning
Not confident	2	Customers that are not confident with the service demand more quality, thus evaluating similar technical conditions with lower QoE values.
Neutral	3	Neutral behavior that does not influence the QoE neither positively nor negatively.
Confident	4	Confident customers are more tolerant with the service, perceiving higher quality levels.

the confidence. The function will be applied using the values of total stalling events ('bufevnt') and the average length of the freezings ('avgbuftime') the following criteria according to the values of these variables. On one hand, the 'bufevnt' variable will have a positive effect on the confidence if the total number of events is 0, a neutral if they are up to 3 events, and a negative effect if there are more than 3. On the other hand, the 'avgbuftime' variable will affect the service confidence positively if the total buffering time is less or equal than 2 seconds, neutrally if the total time sums up to 3 seconds and negatively if the stalling time sums more than 3 seconds.

Using this description, the behavior of this function is specified in Table 5.7. This function also ignores cases that cannot be observed as, for example, 0 buffering events and a average buffering time more than 0 seconds.

In general terms, this function behaves considering the cumulative effects of both 'bufevnt' and 'avgbuftime', having each one a positive (increaser, in green), neutral (no-changer, in black) or negative (reducer, in red) effect. The symmetric relationship used to determine the cumulative effect is given in Table 5.8<sup>4</sup>.

All the changes to the updating functions describes here are presented in Figure 5.8.

The analysis of the responses given by the users allowed us to correctly adjust the updating functions and the valuation of the context variables, which have been already inserted in the updated OTT model shown in Figure 5.3. All the changes introduced here will be reflected on the model that will be validated using the QoE values extracted from the same platform.

### 5.3.2 QoE Model Validation

Using the same emulation platform it was possible to collect data related with the test videos shown to the users. The main goal of this procedure is the computation of the weights of the mathematical model presented in Equation 5.4. In order to simplify the research of these values, it is required to consider a few observations about the data collected and the proposed model.

The first component of the model considers the effect of the bitrate. This value can be directly obtained from the segments of the video, value that will be transformed into a single

<sup>4</sup>Since the relationship is symmetric, the missing values can be derived by symmetry.

Table 5.7: New definition of  $f_{subj}$ .

$M_{subj}^{prev}$	$bufevnt$	$avgbuftime$	$M_{subj}^{new} = f_{subj}(M_{subj}^{prev}, bufevnt, avgbuftime)$
2	0	$\leq 2$	3
		$2 < \cdot \leq 3$	-
		$> 3$	-
	$0 < \cdot \leq 2$	$\leq 2$	3
		$2 < \cdot \leq 3$	2
		$> 3$	2
	$> 2$	$\leq 2$	2
		$2 < \cdot \leq 3$	2
		$> 3$	2
3	0	$\leq 2$	4
		$2 < \cdot \leq 3$	-
		$> 3$	-
	$0 < \cdot \leq 2$	$\leq 2$	4
		$2 < \cdot \leq 3$	3
		$> 3$	2
	$> 2$	$\leq 2$	3
		$2 < \cdot \leq 3$	2
		$> 3$	2
4	0	$\leq 2$	4
		$2 < \cdot \leq 3$	-
		$> 3$	-
	$0 < \cdot \leq 2$	$\leq 2$	4
		$2 < \cdot \leq 3$	4
		$> 3$	3
	$> 2$	$\leq 2$	4
		$2 < \cdot \leq 3$	3
		$> 3$	3

Table 5.8: Cumulative effect of both variables.

Variable 1	Variable 2	Cumulative effect
↑	↑	↑
↑	-	↑
↑	↓	-
-	-	-
-	↓	↓
↓	↓	↓

MOS value using the logarithmic relationship ( $M_{B_r} = 1.3 \cdot \ln(\frac{1}{1000} \cdot B_r) + 2.4$ ). This value is then weighted by the corresponding factor  $\omega_{bitrate}$  that needs to be computed.

The second component of the model is the effect of the stalling events. The effect of these impairments is modeled by two main variables: the total length and the amount of freezing events of the videos. Both values are extracted by directly analyzing the video case shown to the user, values that are computed into a single value following the exponential relationship proposed ( $M_{L,N} = 3.5 \cdot \exp(-(0.15 \cdot L + 0.19) \cdot N) + 1.5$ ). This computed value is then weighted by the factor  $\omega_{stalling}$  that will be computed.

The third component of the QoE model is the effect of the subjective dimension. To this end, the valuation  $M_{subj}$  and their respective updating functions specified in Table 5.7 will be used. The subjective effect of the QoE is weighted by the factor  $\omega_{subj}$  that needs to be computed using the quality values obtained from the users.

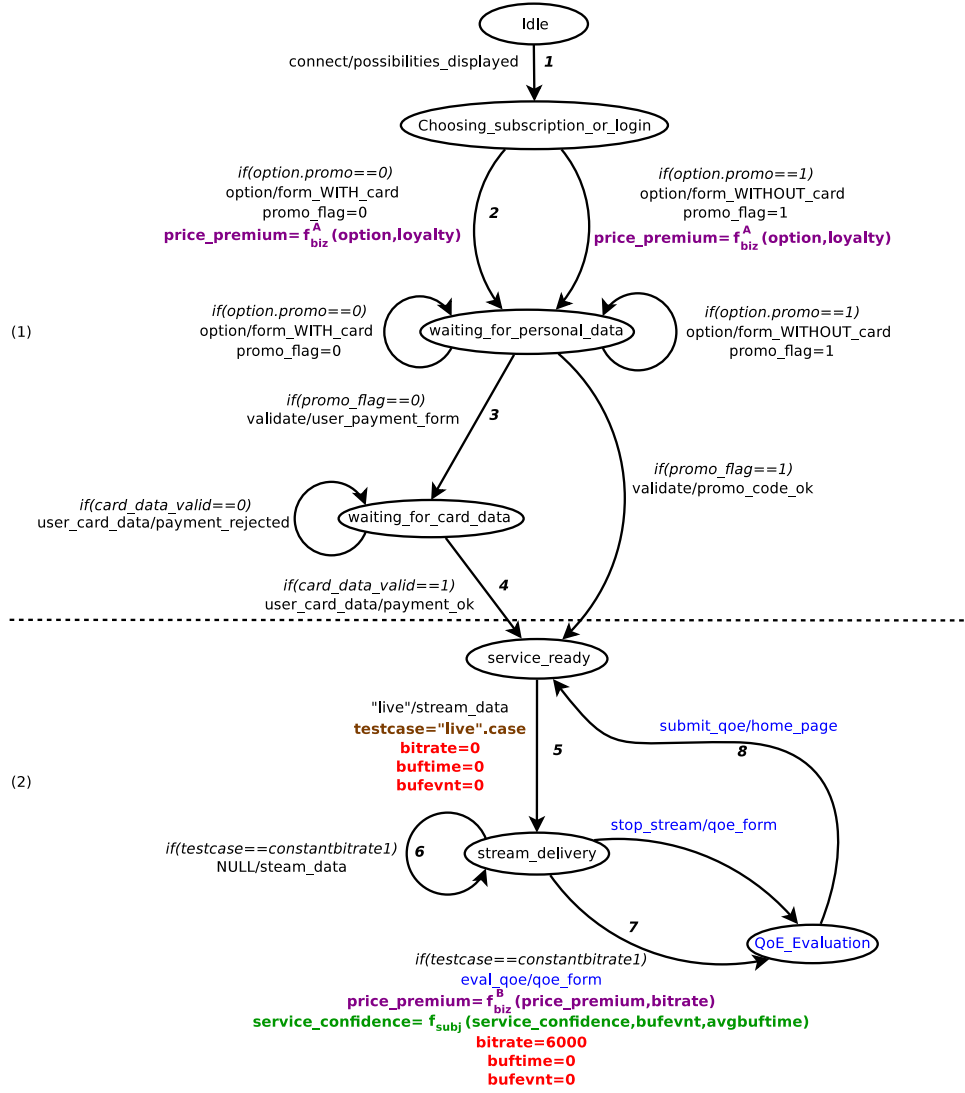


Figure 5.8: Updated OTT model of the emulation platform.

Finally, the fourth component is the effect of the business parameters. In this sense, the valuation given in the last Section will be used  $M_{biz}$ , using the validated updating functions of Table 5.5. The final component of this dimension of the overall QoE value is computed by weighting this  $M_{biz}$  value by the factor  $\omega_{biz}$ , which will be computed.

All the aforementioned constants will be computed using the data collected with the OTT emulation platform. This problem will be solved as a multivariable linear regression problem, which will adjust the values of the weights according to the training set that will be defined.

To this end, the  $n$  training entries will be organized as a matrix  $(\mathbf{A}|\vec{\mathbf{Q}})$  where  $\vec{\mathbf{Q}} = (Q^i)_{i=1..n}$  is the  $n$ -dimension vector of the QoE values obtained from the user, and  $\mathbf{A}$  is the matrix of composed by information extracted from the emulation platform having the form:

Table 5.9: Training set used to compute the weights of the model.

$M_{B_r}^i$	$M_{M_{L,N}}^i$	$M_{subj}^i$	$M_{biz}^i$	$Q^i$
4.72928731	1.50004418	2	4	1
3.828195975	1.5000002	2	4	1
2.927104641	1.504529077	2	4	4
1.208822049	5	4	3	4
4.72928731	1.50004418	2	4	1
1.208822049	1.642667714	2	2	1
1.208822049	1.802027553	3	2	2
4.72928731	5	4	4	4
4.72928731	1.504529077	2	4	1
1.208822049	5	4	2	2
1.208822049	5	4	2	2
1.208822049	1.802027553	3	2	2
2.927104641	1.504529077	2	4	4
4.72928731	1.5000002	2	4	2
2.109913383	1.5000002	2	2	1
4.72928731	2.371763566	2	4	2
2.927104641	1.50004418	2	3	1
1.208822049	5	4	3	4
3.828195975	1.501616912	2	4	1
4.72928731	1.509781748	3	4	4
4.72928731	1.5000002	2	3	1
3.828195975	1.501616912	2	4	3
2.109913383	1.50004418	2	3	2
1.208822049	5	4	3	2
1.208822049	1.53183347	2	3	2
2.109913383	1.509781748	3	2	2
2.927104641	5	4	4	5
2.109913383	5	4	3	3
1.208822049	5	4	3	1
1.208822049	5	4	2	3
2.927104641	1.5000002	2	4	2
3.828195975	1.504529077	2	4	4
2.927104641	1.504529077	2	4	2
4.72928731	2.867197424	2	4	4
2.109913383	1.509781748	3	3	1

$$\mathbf{A} = \begin{pmatrix} M_{B_r}^1 & M_{M_{L,N}}^1 & M_{subj}^1 & M_{biz}^1 \\ \vdots & \vdots & \vdots & \vdots \\ M_{B_r}^n & M_{M_{L,N}}^n & M_{subj}^n & M_{biz}^n \end{pmatrix}$$

where  $M_{B_r}^i$ ,  $M_{M_{L,N}}^i$ ,  $M_{subj}^i$  and  $M_{biz}^i$  represent the values obtained from the emulation platform from the  $i$ -ith test case used as training set.

A total number of 63 QoE evaluation responses could be retrieved from the databases of the deployments. Those registers were subdivided into two principal groups: the *training set* and the *testing set*. The former was used to perform a simple linear regression in order to find the values of the weights of the model. The latter will be used to test the derived QoE model, computing statistical metrics that aim to determine the correctness of the derived model. Table 5.9 shows the set of 35 entries used to compute the unknown weights. These entries were selected randomly from the extracted data.

The whole set of extracted data was loaded into a Microsoft Excel spreadsheet, in order to

use this tool to both compute the weights and validate the model. For the first goal, the Data Analysis Tool Pack of Microsoft Excel was used to perform the linear regression of the data. Table 5.10 shows the computed weights using the training set shown in Table 5.9.

Table 5.10: Computed weights extracted from the training set.

$\omega_{bitrate}$	$\omega_{stalling}$	$\omega_{subj}$	$\omega_{biz}$	$R^2$
-0.188359297	0.048513756	0.618778736	0.969467803	0.350191

Having the values of Table 5.10 in mind, it is possible to reconstruct the QoE model using Equation 5.4 and the values in Table 5.10:

$$\begin{aligned}
 QoE_{MOS} = & -0.188359297 \cdot \left[ 1.3 \cdot \ln \left( \frac{1}{1000} \cdot B_r \right) + 2.4 \right] + \\
 & 0.048513756 \cdot \left[ e^{-(0.15 \cdot L + 0.19) \cdot N} + 1.5 \right] + \\
 & 0.618778736 \cdot M_{subj} + \\
 & 0.969467803 \cdot M_{biz}
 \end{aligned} \tag{5.5}$$

### 5.3.2.1 QoE Model Adjustment Discussion

From the linear regression performed it is also possible to extract some statistical indicators. In Figure 5.9 is shown the distribution of the residuals from the adjustment.

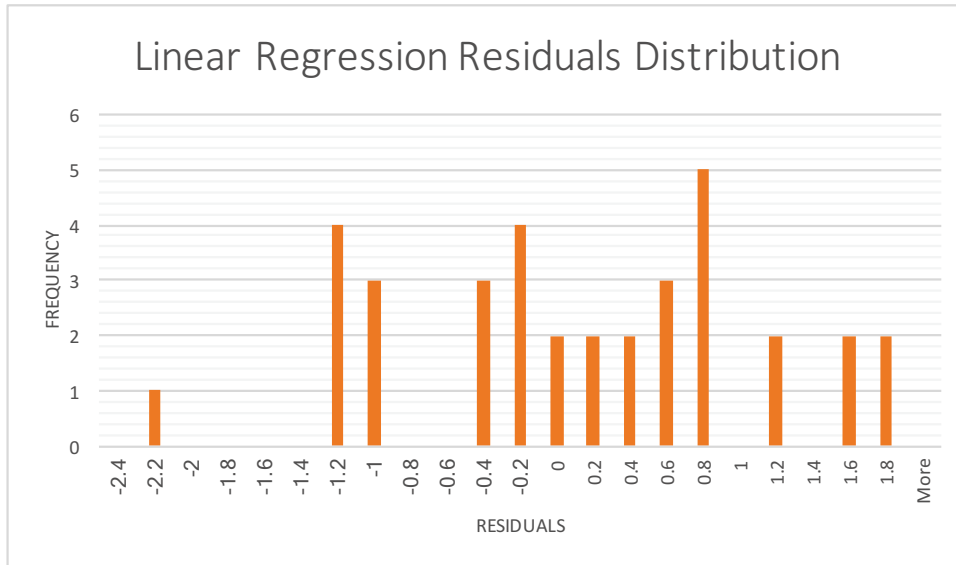


Figure 5.9: Linear Regression Residuals Histogram.

By observing the residuals, it is possible to observe that most of the predicted values are over estimated by the model, being most of them contained in the range  $[-0.4, 0.8]$ . Despite the fact that the adjustment of the curve has a R-square value of 0.350191, the MOS values predicted by the adjusted model present a tolerable difference in comparison with the values assessed by the users. This observation needs to be verified by using the rest of the entries before extracting any conclusion of the model.

### 5.3.3 QoE Model Testing

The rest of the assessed QoE entries were used to test the computed model shown in Equation 5.5 in order to determine its correctness. This was done by comparing the values assessed by the user with the predicted ones obtained by using the data extracted from the emulation platform and the Equation 5.5. Then obtained values were compared by measuring the difference between these them, obtaining a residual of the prediction. In Figure 5.10 is shown the distribution of these differences.

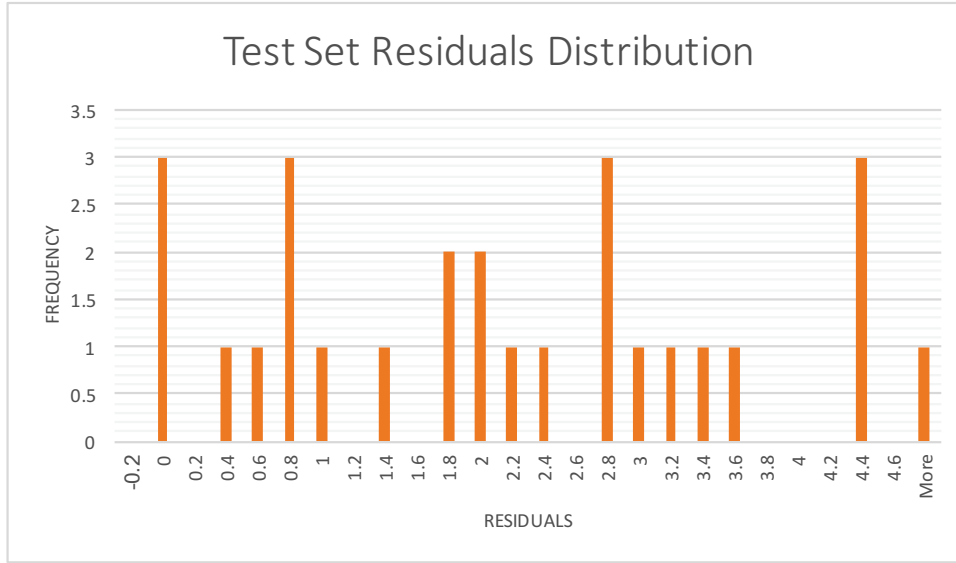


Figure 5.10: Test Set Residuals Histogram.

The figure shows that in general terms, the model does not predict the QoE values correctly. This can be observed by the fact that most of the residuals are contained in the range [1.4, 3.6]. This shows that, in general terms, the adjusted model overestimates the QoE value, reaching up to 4 levels – a difference between “good” and “poor” according to the MOS value. The involved test cases are usually related with low quality components due to bitrate and stalling events, but high quality due to the subjective and business-related parameters. This fact is reaffirmed by the computed weights shown in Table 5.10, which show high values for these two components, assigning high importance to the subjective and business parts of the model.

## 5.4 Conclusions

Despite the efforts of proposing a close-to-reality QoE model, the adjusted model needs to be refined even more. The test set showed that further research needs to be done in terms of integrating business and subjective parameters into the QoE analysis. Despite that the usage of simple models to map both subjective and business parameters into QoE values, they are not useful in real environments since their effects on the quality are far from being completely understood.

This observation was stated by Reiter et al. in [70], the QoE is a multidisciplinary concept that requires a holistic analysis. However, the impact analysis of human dimensions is still an open challenge on the perceived quality field.

In addition, the inclusion of context impact factors – such as the business-related parameters – can aid to generate QoE models that are useful to the service providers. These models will allow them to develop monitoring tools that play a double role; on one hand, by monitoring the quality of the end user and, on the other hand, optimize the revenues of the OTT service provider.

Without going any further, the work presented in this Chapter states the bases of the building process of such QoE models, following standardized procedures and using real-users data. It is advised to follow this approach once the required advances on business and perception fields provide the corresponding models. The inclusion of such correct, complete models would allow correct the proposed model and re-adjust it in order to have a more realistic QoE prediction model.

---





---

# Chapter 6

## Static Analysis of an Over-The-Top Service

### Contents

<b>6.1</b>	<b>Introduction</b>	<b>79</b>
<b>6.2</b>	<b>Static Analysis Algorithm</b>	<b>80</b>
<b>6.3</b>	<b>Analysis of a Real Over-The-Top Service</b>	<b>82</b>
6.3.1	Experimental Configuration	82
<b>6.4</b>	<b>Static Over-The-Top Model Analysis</b>	<b>84</b>
6.4.1	Linear Quality of Experience Model	84
6.4.1.1	Discussion	84
6.4.2	Expanded Quality of Experience Model	85
6.4.2.1	Discussion	87
<b>6.5</b>	<b>Conclusion</b>	<b>88</b>

---

### 6.1 Introduction

At this point, the proposed QoE evaluation framework has been useful to monitor the value of the QoE. This was done by using the proposed approach as an extension of the MMT software by implementing two algorithms that compute the  $l$ -equivalent model and their respective QoE value. However, it is also possible to use the same framework in order to statically analyze an OTT service.

In the next sections, it will be shown how the QoE evaluation framework presented in Chapter 3 can be used in order to generate a statistics about the different scenarios. This required the formulation of a new algorithm that is able to compute and classify all the possible configurations contained in the OTT model. Considering this information it is possible to draw conclusions about the service under analysis, including the characterization of the quality perceived by the customers.

The new functionality presented in this Chapter aims to complement the information to the service providers, in order to let them know about the expected QoE of the service in early stages of the development.

---

## 6.2 Static Analysis Algorithm

As presented in Chapter 3, the QoE evaluation framework was designed in order to provide a formal model of the OTT service user analysis. By using EFSM, the framework has the ability to capture functional and non-functional requirements of the service, which allows to focus the analysis on the features of the service that impact the QoE value.

This framework was used to generate two algorithms that were implemented in order to extend the functionalities of the MMT software – a monitoring tool. Such implementation extended the capabilities of MMT with the QoE computation functionalities based on EFSM and integrating the effects of business-related parameters into the evaluation of the perceived quality. However, the proposed framework can also be used to extract conclusions about the service before its release.

The EFSM are conceived to be standalone model, which retains the information about how the system behaves and what are the possible scenarios for the user. These scenarios are represented as *configurations* of the machine, which represent a possible state of an active user at any point of the execution of the machine. More formally, following the notation of Section 3.2 a *EFSM configuration* is defined as:

$$c = (\sigma, \mathbf{v}) \quad (6.1)$$

where  $\sigma \in \Sigma$  is a state of the EFSM and  $\mathbf{v} \in D_V$  is the valuation vector of the context variables of the machine. This tuple gathers two principal parts of the information used to completely identify the context the user: first, the state in the machine describes in which stage of the execution the user is; and second, the values of the context variables keep a track of the past decisions that took the user to the actual state. By analyzing these data, it will be possible to extract valuable information about the model of service and make corrections if necessary. However it is possible that the number of configurations contained in the machine might be difficult to handle computationally. It is required then to provide a further analysis about the ability of the models to contain information.

By following the whole methodology proposed in the Outline 3 on page 40, it is possible to have a preliminary estimation of the amount of configurations contained in the model. The first estimation can be done with the non-augmented EFSM obtained after applying the first three steps. At this point, the OTT model is principally formed by a set of states, which limits the amount of configurations to the number of sets. More formally, the configuration tuples will have the form  $C = \{c, c = \{(\sigma, \emptyset)\}, \sigma \in \Sigma\}$ , and the amount of configurations will be given by the cardinality of the states set, i.e.  $|C| = |\Sigma|$  since there are no more elements on the tuple.

Once the OTT model is augmented, the introduction of finite, discrete quality indicators increases the number of configurations contained in the machine, basically since now the context variables have to be considered in the analysis. Following the same notation of Section 3.2, let  $V$  be the set of context variables names,  $\mathbf{v} = (v_i)$  be the vector of context variables, and  $v_i \in V$  a single context variable. Considering these definitions, the configurations will now have the form  $C = \{c, c = (\sigma, (v_i)), \forall i \in \{1..|V|\}, \sigma \in \Sigma\}$ . In other words, the previous definition considers now each possible value of each context variable. With this definition in mind, the theoretical number of configurations contained in the model is given by

$$|C| = |\Sigma| \cdot \prod_{i=1}^{|V|} v_i \quad (6.2)$$

This amount represents the number of possible combinations of all the values in the tuple and it represents the maximal number of configurations contained in the OTT model. However the transitions of the model and the guards – predicates – of the model allow reducing this number to a fixed number of *reachable* configurations from the initial state. Due the semi-automatic nature of the modeling procedure, it is not possible to provide an analytical solution about how many of these configurations are reachable for a generic model. It is possible though to compute this amount by using the same QoE evaluation methodology.

The last fact requires the introduction of a third algorithm in order to effectively compute all the reachable configurations by emulating the execution of the machine. The designed procedure is presented in Algorithm 3

This procedure is a modification of the Algorithm 1 in order to emulate an execution of the machine. This algorithm is based on the DFS strategy in order to transverse the EFSM structure while determining the possible transitions to be executed.

Algorithm 3 is based on a recursive strategy in order to compute the configurations contained on the model. The procedure starts with an initial valuation of the context variables and uses it to start the computation. At each step, the algorithm computes the values of the guards – conditions – in order to discover which are the transitions that will be executed and the respective next states, given the values of the context variables. For each discovered transition, the procedure uses the updating functions in order to determine their value for the next state. Then it applies the same procedure recursively on each of the previously computed next states. Finally, when the desired length of the analysis is reached, the algorithm adds the state and the vector of the context variables to the set of reachable configurations of the model.

In the following sections this algorithm will be used in order to experimentally analyze the beIN Sports Connect service, showing how the QoE evaluation framework can be in early stages of the development of the service.

---

**Algorithm 3** Computation of the total number of configurations

---

```

function COMPUTECONFS(len, state, vars, confs)
  if len ≤ 0 then
    return confs = confs ∪ {(state, vars)}
  end if
  for all t in transitions[state] do
    stand ← true
    for all cond in t.conditions do
      stand ← stand && cond(vars)
    end for
    if !conditionsStand then
      continue
    end if
    for all f in t.updateFunctions do
      for all var in vars do
        if var.name == f.name then
          var.val ← f.func(var.val)
        end if
      end for
    end for
    computeConfs(t.to, len − 1, vars)
  end for
end function

```

---

### 6.3 Analysis of a Real Over-The-Top Service

The aforementioned Algorithm 3 was introduced in order to analyze the model of any OTT service and provide an analysis of the possible QoE values of the service before it has been released. In particular, the algorithm aims to provide support in the following aspects:

- *Possible future user scenarios:* The proposed algorithm has the ability to compute the possible configurations of the users retained in the model. With this information, it is possible to characterize all the possible scenarios that a user can face, including the respective values of the context variables.
- *Early QoE analysis:* Having all the scenarios computed, their respective QoE values can be computed by using a proper QoE model on them. In this sense, it is possible to unveil all the possible QoE values that the user might experience when the service is launched. This information can be crucial at initial stages of the development in order to identify either flaws on the general model or potential single scenarios that require to be improved.
- *Classification of user scenarios:* Once the values of the QoE are known, it is possible to perform a statistical classification of them. This procedure aims to provide an overview about how the quality values are spread across the scenario in order to estimate in advance the number of customers that can be considered as “satisfied”. By applying the appropriate criteria to this analysis – in the sense of considering certain quality level as “satisfactory” – it is possible to optimize the investment and, therefore, the revenues of the service.
- *Empirical depth analysis of the model:* In the previous sections, the QoE evaluation framework was used to propose an algorithm that predicts the future scenarios using the  $l$ -equivalent model. This evaluation required the calculation of this new form, called the unrolling process. It is not possible to know in advance which is the optimal length used to compute the  $l$ -equivalent model, however the static analysis algorithm can aid the search of this optimal value.

#### 6.3.1 Experimental Configuration

In order to show how this algorithm can be used, the proposed approach will be used with the beIN Sports Connect model. To achieve this goal, the algorithm will be applied to the service model presented in Figures 3.6 and 5.3, using the two QoE models presented in Equations 3.5 and 5.4 respectively, i.e., using both the linear and the expanded QoE models.

With this configurations, the number of configurations contained in the model will be computed using lengths of the  $l$ -equivalent model from 2 to 18, computing their respective QoE value. Each scenario will be classified according to the QoE value they have, which will give an overview about how the quality values are spread across all the scenarios contained in the model.

For the simulation, an initial valuation of the context variables was used, in order to represent a typical new customer who is accessing the service. Considering that each QoE model have slightly different context variables, this user was represented by using a set for each experiment performed. Table 6.1 presents the initial valuations of the context variables for each case.

In both linear and expanded models, the values of the business and subjective variables – ‘price\_premium’ and ‘service\_confidence’ – represent, respectively, a user that has a neutral opinion with the service and considered to be a brand switcher. For the case of the linear model,

Table 6.1: Initial Valuation of the context variables.

	Business	Subjective	stream_flag		
Linear Model	1	1	0		
Expanded Model	1	1	0	0	0
	Business	Subjective	Bitrate	Total buffer time	Buffer frequency

the single objective variable ‘stream\_flag’ was set to 0, representing that, at the initial state of the machine, the service has not started yet. Analogously, the set of expanded context variables – Bitrate, Total buffer time and Buffer frequency – are all set to 0 for the same reasons before described.

The aforementioned conditions do not allow to completely test the OTT models, since some guards of the model depend on external changes of the context variables. In order to achieve an exhaustive examination, it is required to test the transitions guarded by these conditions, which requires a further effort on the experimental configuration.

In Figure 3.6, the transition labeled as 6 and 7 are guarded by the value of the ‘stream\_flag’ variable. In the model, changes on this variable were triggered externally by the service itself, which periodically checks the conditions of the networks and emits a verdict deciding whether the minimal conditions to stream are met or not. Since the static analysis of the EFSM model cannot depend on variables whose values are computed on runtime, it is required to relax this constraint in order to statically compute its value. Two approach can be followed to achieve this.

On one hand, it is possible to simply suppress the guards and assign a fixed value to the variable. This simple method has the disadvantage of not modeling some realistic scenarios: either the user will watch the stream forever (‘stream\_flag’ = 1) or they will never watch the video (‘stream\_flag’ = 0).

A more suitable solution, on the other hand, is to provide a methodology to change the value automatically when a condition is met. This condition will be set considering the effect for the user of this variable on the service. As stated before, the ‘stream\_flag’ is a variable that models if the technical parameters of the network are below a minimal threshold. In this sense, if the network does not meet this requirement, the user will observe a failure on the service. It is possible to say that, from the point of view of the user, the ‘stream\_flag’ variable models the eventual failures of the service. For this reason, the guards of related with this variable will be modified in order to follow two scenarios that try to emulate service failures:

- *Service failure emulation after a fixed time:* Once the emulation has arrived to the state ‘stream\_delivery’, the ‘stream\_flag’ variable will be kept on 1 for two checking cycles, emulating a failure on the service after a fixed amount of time.
- *Service failure emulation after a random time:* Instead of changing the variable after a fixed amount of checking, the machine will determine the value of the ‘stream\_flag’ variable in a random basis: each time the machine arrives to the ‘stream\_delivery’ state, the machine will move to ‘stop\_stream’ with probability 0.5, and keep the stream with the same probability. For this, the Javascript random number generator will be used, which returns random value following a uniform distribution.

As for the expanded beIN Sports Connect model, the introduction of test cases helps the simulation of failures in the sense of determining the conditions of the objective parameters.

When the starts using the service by clicking the “live” button (transition 5 in Figure 5.8), a specific test case is selected. Each one of these test cases follow its specification given in Table 5.1, which shows the respective bitrate and buffering events at each segment of the 2-minutes video. Following these data, it is possible to either directly extract the value of the objective variable – as for the ‘bitrate’ – or derive its value – as for the case of “buftime” or “bufevnt” variables, representing the sum of the lengths and the count of the buffering events respectively.

Considering that the users were asked to watch complete sequences of 2-minutes videos, the variables are assigned after the last segment, i.e once the machine moves to the ‘QoE\_Evaluaiton’ state, transition labeled with 7 in Figure 5.8. In this Figure, the assignation of the variables is done using a fixed test case – ‘constantbitrate1’ – in order to show how does the model look. Generally speaking, there are as many transitions 6 and 7 as test cases are defined, opening the possibility to use different updating functions on other context variables such as the ‘service\_confidence’ or the ‘price\_premium’. The main idea behind this design is to open the possibility to insert different impact models depending on the test case. This feature is particularly useful for the subjective and business parameters when formal, mathematical models of these parameters could be provided by the literature.

Finally, the user has the possibility to return to the ‘service\_ready’ state once they provided their QoE value. At this point, the user is able to watch a new video on the platform, which would select a new test case. In order to simplify the analysis, a constraint will be introduced on the general behavior: the user will experience the same test case as the first video watched. This restriction will allows us to control what are the impairments experienced by the user,

## 6.4 Static Over-The-Top Model Analysis

Once the experimental setup was defined, the model was put under test by using an implementation of the Algorithm 3 on the MMT correlator engine. The next sections present the results obtained from the runnings.

### 6.4.1 Linear Quality of Experience Model

The implementation of Algorithm 3 was executed using the original OTT EFSM – with both failure emulations logics – and the linear QoE models. The obtained results for the running are shown in Tables 6.2.

#### 6.4.1.1 Discussion

Tables 6.2a and 6.2b show the results for the failure emulation after a fixed and a random time respectively. In both scenarios it is possible to observe that, as the depth of the analysis – length of the  $l$ -equivalent – grows, so does the number of configurations. This behavior can be observed until a depth of 11 and 12 is reached, where the amount of configurations of the machine stalls at 54. Even when the analysis goes deeper on the machine, no more configurations can be reached on both failure emulation scenarios.

By analyzing the average distribution of the QoE values, we observe that for both scenarios, no configurations with QoE below 2 can be observed when the length of the  $l$ -equivalent is low. This fact can be easily explained by the fact that the analysis is not deep enough to consider the effect of service failures in the QoE evaluation. However, at short lengths of the analysis it is possible to observe the effect of including subjective and business-related parameters into

Table 6.2: Configurations and QoE distribution for the original model.

(a) Failure emulation at fixed time							(b) Failure emulation at random time						
Length	Avg. number of conf.	Average QoE distribution					Length	Avg. number of conf.	Average QoE distribution				
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]			[0,1]	[1,2]	[2,3]	[3,4]	[4,5]
2	3	0	0	2	1	0	2	3	0	0	2	1	0
3	7	0	0	4	3	0	3	7	0	0	4	3	0
4	12	0	0	6	6	0	4	12.3	0.3	0	6	6	0
5	17	0	0	7	10	0	5	18.3	0.9	0.4	7	10	0
6	24	0	0	8	14	2	6	28.1	1.7	2.4	8	14	2
7	41	4	9	9	15	4	7	36.9	2.7	6.2	9	15	4
8	48	5	12	10	15	6	8	42.9	3.9	8.1	9.9	15	6
9	51	6	12	12	15	6	9	48.1	5.4	10.9	10.8	15	6
10	53	6	12	14	15	6	10	52.5	5.8	12	13.7	15	6
11	54	6	12	15	15	6	11	53.7	6	12	14.7	15	6
12	54	6	12	15	15	6	12	54	6	12	15	15	6
13	54	6	12	15	15	6	13	54	6	12	15	15	6
14	54	6	12	15	15	6	14	54	6	12	15	15	6
15	54	6	12	15	15	6	15	54	6	12	15	15	6
16	54	6	12	15	15	6	16	54	6	12	15	15	6
17	54	6	12	15	15	6	17	54	6	12	15	15	6
18	54	6	12	15	15	6	18	54	6	12	15	15	6

the experiment: instead of observing a flat distribution with high QoE values, we can observe “fair”<sup>1</sup> QoE values ranging from 2 to 4. We attribute this distribution to the effect of the past experiences of the user (modeled as the ‘service\_confidence’ variable) and the loyalty of the user with the brand (modeled as the ‘price\_premium’ variable), values that can change depending on the type of subscription chosen.

As the analysis goes deeper on the model, more configurations are reached and configurations with low QoE values start to appear, which show the effects of the emulated service failures. However, even when the analysis goes deeper, the number of configurations that can be considered as “low quality” scenarios do not grow considerably. This fact is supported by the normal distribution of the QoE values with an average of 3 once the maximum number of configurations has been reached.

### 6.4.2 Expanded Quality of Experience Model

The expanded OTT EFSM of Figure 5.8 and its respective enhanced QoE model of Equation 5.5 were implemented in the MMT tool.

From the test suite shown in Table 5.1, a set of 7 were selected to compute the configurations contained in the model:

1. Constantbitrate3: A test case that shows an constant, average bitrate according with the data assessed by the users.
2. Desc-bitrate1: This test represents a link with growing congestion, limiting more and more the available bandwidth for the streaming.
3. Ascen-bitrate1: On the contrary, this test represents a link that becomes less congested, augmenting the available bandwidth.
4. Freezing+bitrate11: This test case introduces 5 seconds stalling events every 20 seconds, with a constant average bitrate.

<sup>1</sup>According with the MOS scale.



Table 6.3: QoE distribution and configurations for simple impairments test cases.

(a) Constant Bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	2	4	4	0
6	13	0	0	0	3	6	4	0
7	15	0	0	0	4	7	4	0
8	16	0	0	0	4	8	4	0
9	16	0	0	0	4	8	4	0
10	16	0	0	0	4	8	4	0
11	16	0	0	0	4	8	4	0
12	16	0	0	0	4	8	4	0
13	16	0	0	0	4	8	4	0
14	16	0	0	0	4	8	4	0
15	16	0	0	0	4	8	4	0
16	16	0	0	0	4	8	4	0
17	16	0	0	0	4	8	4	0
18	16	0	0	0	4	8	4	0

(b) Descending Bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	2	4	4	0
6	15	0	0	0	3	8	4	0
7	22	0	0	0	4	10	8	0
8	27	0	0	0	4	11	10	2
9	31	0	0	0	4	11	13	3
10	34	0	0	0	4	13	14	3
11	35	0	0	0	4	14	14	3
12	35	0	0	0	4	14	14	3
13	35	0	0	0	4	14	14	3
14	35	0	0	0	4	14	14	3
15	35	0	0	0	4	14	14	3
16	35	0	0	0	4	14	14	3
17	35	0	0	0	4	14	14	3
18	35	0	0	0	4	14	14	3

(c) Ascending Bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	0	6	4	0
6	15	0	0	0	0	9	6	0
7	20	0	0	0	3	10	7	0
8	24	0	0	0	6	11	7	0
9	27	0	0	0	7	13	7	0
10	30	0	0	0	7	14	9	0
11	31	0	0	0	7	14	10	0
12	32	0	0	0	7	14	11	0
13	32	0	0	0	7	14	11	0
14	32	0	0	0	7	14	11	0
15	32	0	0	0	7	14	11	0
16	32	0	0	0	7	14	11	0
17	32	0	0	0	7	14	11	0
18	32	0	0	0	7	14	11	0

5. Freezing+bitrate17: A test case with shorter stalling events – 2 seconds long each one – but using a link with less bandwidth available.
6. Freezing+bitrate21: This case introduces short stalling events with descending bitrate.
7. Freezing+bitrate24: Finally, this case uses stalling events that are each time longer with a fixed, average bitrate.

The selection of the test cases was done in order to analyze how does the amount of configurations change with different scenarios usually experienced by the user of two principal types: simple (changing bitrate) and mixed (changing bitrate and stalling events). In addition, the QoE model will allow analyzing how does the QoE values are distributed, showing how different impairments can impact this distribution. Finally, in the same way as the experiments performed with the linear QoE model, the length of the computed  $l$ -equivalent will vary from 2 to 18 (except in the case of ‘Freezing+bitrate11’), in order to analyze how does the number of configuration changes as long as a deeper analysis is performed.

The obtained results for each one of the selected test cases are shown in Tables 6.3 and 6.4 for the simple and mixed perturbations respectively.

Table 6.4: QoE distribution and configurations for mixed test cases.

(a) Freezing Bitrate: 5 seconds stalling and constant bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	2	4	4	0
6	15	0	0	0	5	6	4	0
7	22	0	0	2	7	9	4	0
8	30	0	0	5	10	11	4	0
9	37	0	0	9	13	11	4	0
10	46	0	0	12	19	11	4	0
11	55	0	0	17	23	11	4	0
12	62	0	0	21	26	11	4	0
13	68	0	0	24	29	11	4	0
14	74	0	0	27	32	11	4	0
15	80	0	0	30	35	11	4	0
16	85	0	0	32	38	11	4	0
17	88	0	0	33	40	11	4	0
18	89	0	0	33	41	11	4	0
19	89	0	0	33	41	11	4	0
20	89	0	0	33	41	11	4	0

(b) Freezing Bitrate: 2 seconds stalling and low, constant bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	0	6	4	0
6	15	0	0	0	0	9	6	0
7	20	0	0	0	1	12	7	0
8	24	0	0	0	1	16	7	0
9	27	0	0	0	1	19	7	0
10	30	0	0	0	1	22	7	0
11	33	0	0	0	1	25	7	0
12	36	0	0	0	1	28	7	0
13	39	0	0	0	1	31	7	0
14	42	0	0	0	1	34	7	0
15	45	0	0	0	1	37	7	0
16	47	0	0	0	1	39	7	0
17	48	0	0	0	1	40	7	0
18	48	0	0	0	1	40	7	0

(c) Freezing Bitrate: 2 seconds stalling and descending bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	2	4	4	0
6	15	0	0	0	3	8	4	0
7	22	0	0	0	4	10	8	0
8	27	0	0	0	4	11	12	0
9	31	0	0	0	4	11	16	0
10	34	0	0	0	4	13	17	0
11	35	0	0	0	4	14	17	0
12	35	0	0	0	4	14	17	0
13	35	0	0	0	4	14	17	0
14	35	0	0	0	4	14	17	0
15	35	0	0	0	4	14	17	0
16	35	0	0	0	4	14	17	0
17	35	0	0	0	4	14	17	0
18	35	0	0	0	4	14	17	0

(d) Freezing Bitrate: 2 to 20 seconds, growing stalling events and average, constant bitrate.

Length	Avg. number of conf.	Average QoE distribution						
		[0,1]	[1,2]	[2,3]	[3,4]	[4,5]	[5,6]	6+
2	2	0	0	0	0	1	1	0
3	4	0	0	0	0	2	2	0
4	7	0	0	0	0	4	3	0
5	10	0	0	0	2	4	4	0
6	15	0	0	0	3	8	4	0
7	22	0	0	0	4	10	8	0
8	27	0	0	0	4	13	10	0
9	31	0	0	0	4	16	11	0
10	32	0	0	0	4	17	11	0
11	33	0	0	0	4	18	11	0
12	35	0	0	0	4	20	11	0
13	36	0	0	0	4	21	11	0
14	36	0	0	0	4	21	11	0
15	36	0	0	0	4	21	11	0
16	36	0	0	0	4	21	11	0
17	36	0	0	0	4	21	11	0
18	36	0	0	0	4	21	11	0

### 6.4.2.1 Discussion

Generally speaking, it is possible to observe the effect of the adjusted QoE model remarked in Section 5.4: the mathematical relationship that computes the QoE value seems to be too optimistic, at the point of computing values even out of the MOS scale (over 5). By observing a little more in detail these cases, it is possible to find that these configurations are characterized by states where the values of the technical variables are 0, which introduce errors in the computation of the model. Despite this, the fact that the proposed model needs further refinement is reaffirmed with this analysis.

Regardless of the rectification of the model, it is possible to analyze the results from the point of view of the static analysis of the OTT service. First, it is possible to observe the same stabilization of the number of configurations observed in the with the linear model. In these scenarios, it is possible to observe that this stabilization occurs at different lengths, which remarks the fact that this length does not completely depend on the EFSM model of the service, but also on the specific impairments experienced by the user. Having this in mind, it is important to define a suitable test case that represents the impairments most experienced by the user. Such

correct definition will allow analyzing the distribution of QoE values in close-to-reality scenarios.

More in detail, the single-impairment test cases seem to have a number of configurations based on the scenario tested. For example, the constant bitrate stabilizes early (at length of 8) and have less configurations than the ascending or descending bitrates scenarios. This is simply explained that the bitrate variable can take a wider range of values than the constant bitrate, which generates more possible combinations of variables' values. In addition, for both ascending and descending bitrates, the distribution of the QoE values are similar, showing more values with 'fair' quality in the case of ascending bitrates. This can be explained by the "first impression" effect: an initial low bitrate impacts the business and subjective variables, whose values are not recovered once the service gets better.

In the case of mixed impairments, the number of total configurations take longer to stabilize, since more possible values of the variables related with the stalling events introduce more possible combinations. In the case of Table 6.4a (5 seconds stalling and constant bitrate), the amount of configurations stabilize at length of 18 (which explains the extra two executions of length 19 and 20). In this particular case, more configurations are distributed with a QoE value between 2 and 4, which shows that this particular test case needs to be revised carefully and avoided by the service provider.

In the case of 2 seconds stalling events and low constant bitrate, it is possible to observe that there are more configurations contained in the model, however they are associated with lower QoE values. This fact is in line with the literature, remarking that stalling events should be avoided. By observing changing the bitrate impairments to a descending behavior (from Table 6.4b to 6.4c), the number of configurations QoE drops, but more scenarios are classified with better quality. It is possible to conjecture that this behavior is due to the fact that the combination of buffering events and a constant reduction the visual quality of the video is seen by the user as an effort of the service to keep the streaming despite the reduced network capabilities.

Finally, the effect of buffering event is remarked in the last case, shown in Table 6.4d. By comparing this case with the one shown in Table 6.4b, it is possible to observe that the configurations have lower QoE values in general terms, due the fact that the stalling events grow in length as the time passes. These test cases confirms the fact that it is preferable to lower visual quality than stop the streaming to refill the buffer on the client's player.

## 6.5 Conclusion

This Chapter presented an extension of the proposed QoE evaluation framework: an static analysis of an OTT service. The methodology presented in this Chapter represents a powerful tool for the service provider to detect flaws on the model and the service before the release to the public.

The proposed methodology is applied to the beIN Sports Connect running example, analyzing both models and QoE models introduced before: the initial EFSM in combination with the linear QoE model, and the refined OTT model with the adjusted QoE prediction formula. In both cases we can observe the main advantage this type of analysis proposes: the possibility of visualize in advance the QoE scenarios of an OTT service and identify possible cases where the QoE is particularly low. The proposed algorithm allows analyzing the EFSM model statically and predict which will be the expected QoE values for the users once the service is released.

Finally, it is important to remark the stabilization of the number of computed configurations contained in the machine. Since this number stays steady at a certain point, it is not necessary to perform a deeper analysis. In this sense, we conjecture that this value is the optimal length

---

of the  $l$ -equivalent in order to perform online analysis using the first two algorithms members of the framework. The drawback of this analysis is the computation of this limit: it is not possible to analytically compute this limit beforehand, but it is possible to determine it in advance in a empirical basis using the approach shown in this Chapter.

---



---

# Chapter 7

## Conclusion

### Contents

<b>7.1 Perspectives . . . . .</b>	<b>93</b>
-----------------------------------	-----------

---

The works organized and presented in this document describe the proposed approach to integrate the business variables in the Quality of Experience analysis. As stated in Section 2.3, business decisions can directly impact the functional and non functional requirements of an OTT service, being the latter usually the factors that business decision impact the most.

Business decisions usually include, but are not limited to, monetary decisions about the service, such as the amount of money invested on operational costs (in other words, the quantity spent on running the service), the investment done on multimedia content and the price of the service in form of subscription values. All these variables directly influence the expectations of the user, having different impacts on the expected quality of the service. These precedents establish the arguments of the inclusion of such variables when computing the quality value for OTT services. In addition, the novel proposal here introduced represents one of the first efforts to incorporate the effect of the business decisions into the analysis of the QoE.

The ensemble QoE evaluation framework presented in this work is formed by four main contributions presented in different conferences and workshops in a incremental basis: a mathematical formalism used to model OTT services, its respective implementation as an extension of a suitable monitoring tool, the validation of the whole whole approach – including the implemented algorithms, and the proposal of a static analysis methodology for OTT models.

The first contribution represents the base of the whole framework, providing the mathematical formalisms required to apply the evaluation framework. In this approach, any OTT service is modeled by using EFSM, where the functional and some non-functional requirements of the service are captured in the model. This first version is then enhanced with quality indicators in order to obtain an augmented model of the OTT service which now captures all the desired quality variables. Later, this model is used to predict the future scenarios of the user by computing the  $l$ -equivalent form of the model, a tree-shaped expanded version of the model which now shows the decisions a user took in order to arrive to a final configuration. This last fact is the main advantage of using such mathematical formalism for modeling OTT services: it allows keeping track about all the decisions of the user during their usage and how these decisions impact on the modeled quality indicators and, therefore, on the final QoE value.

Before providing a closer-to-market solution based on the proposed approach, it is imperative

---

to consider the nature of the three main stages of the framework: modeling, augmentation, and computation. Since both the modeling of an OTT service and its augmentation with quality indicators is a semi-automatic process out of the scope of the presented work, it was not possible to provide an automated procedure to derive such model. In fact, deriving formal models – as, for example, EFSM models – from natural language specifications is still an open challenge in reverse engineering. However, assuming the model of the service is given, it is possible to automatize the last process, that is the computation. This is done in two principal sub-stages: computation of the  $l$ -equivalent form and its augmentation with the QoE values.

The second contribution of this work is formed by the implementation of the aforementioned approach, composed by two principal algorithms:  $l$ -equivalent computation and QoE augmentation of the traces. The first procedure is based on the DFS approach, which transverses the OTT model in a recursive way, computing at each step the values of the context variables of the machine. Following this strategy, this algorithm computes all the possible paths – also called “traces” – of length  $l$  starting from the initial state of the model. This information is then used by the second algorithm, which computes the respective QoE values using a proper quality model. This model is, in general, a mathematical relationship that maps the values of the quality indicators to a single QoE value for the trace.

At this point, the framework is composed by the methodology to model an OTT service and the algorithms that process this model in order to predict the QoE values of different users. In order to provide a proper corroboration of the proposed approach, it was required to perform the validation of the algorithms by applying the framework to real users and obtain real QoE values that will be compared with the predicted ones.

The third contribution of this work represents the validation process of the proposed framework. To this end, a three-parts methodology was applied having in mind the elements of the framework that require validation. First, in order to show how the framework can be used for real OTT services, an emulator of the running example was built. This simulator was built in order to keep the same functional requirements – in other words, the same behavior – modeled on the EFSM of the running example, but inserting a QoE evaluation used to collect the user’s quality value. This emulation platform represented the base where the test videos were inserted. These video clips were generated having in mind the most common perturbations that can be experienced by the user: the bitrate value, the video freezing frequency and the freeing length. Having these variables in mind, different video test cases were generated and inserted in the emulation platform. Lastly, the introduction of the appropriate business variables requires the validation from the user, in order to ensure that the selected variables inserted in the model impact the quality of the user. This goal was achieved by the implementation of a questionnaire that every user of the platform needed to answer before using the emulated service. The aforementioned platform was deployed in two different laboratories in order to maximize the amount of surveyed users.

The data extracted was used having two main goals in mind. On one hand, the survey implemented at the beginning allowed stating that business variables do have impact on the QoE of OTT services. On the other, the QoE values obtained from the user allowed creating a novel QoE mathematical model that considers the business variables into the computation. This was achieved by adjusting the weights of a linear combination model, which is also useful to determine the impact factor of the business variables. Despite the efforts of adjusting the model, further research is required in order to obtain the real impact of subjective and business variables into the QoE. These advances will allow applying the proposed methodology once more and obtain a holistic, closer-to-reality QoE model. By developing this type of model will allow the

implementation of monitoring tools in form of, for example, an extension of the MMT software, providing a complete solution for QoE assessment. The whole set of technologies proposed and validated would be ready to be used on real life services in order to take advantage of their predictions features.

Finally, the proposed framework was also used to perform offline, static analysis of the OTT model in order to detect potential flaws on the model in early stages of the development of the service.

The fourth and last contribution is formed by the proposal of an analysis algorithm for OTT models based on EFSM. This procedure allows computing the contained configurations in the model, i.e. it is capable of predicting which are the possible scenarios a user can encounter when using the service. This information is later complemented by computing the respective QoE value using the proper quality model. This approach was implemented as another extension of the MMT software, using one of the previously-mentioned algorithms as a base. The idea behind this implementation is to transverse the model by computing, at the same time, which are the values of the context variables, reducing the amount of configurations from the theoretical ones to the achievable ones. In this way, the algorithm determines which are the user scenarios where the service provider should focus, eliminating redundant information and aiding to optimize the investments on the service.

The information returned by the algorithm is useful when designing an OTT service in order to have an idea about how do the QoE values will distribute across the different scenarios. With this information the service provider can detect flaws in the model (if the QoE values are mostly low) or better business opportunities.

The exposed approach to compute the Quality of Experience has been designed having in mind the requirements of the actual industry. Every day new actors try to take advantage of the revenues of Over-The-Top services actually dominated by a few big actors. In order to fix this, the works presented here provide the market a close-to-release set of technologies that can be used to improve the quality experienced by the users.

By providing such technologies, this work aims to provide a complete monitoring solution that will aid the actual and new actors to maximize both the quality of the services they offer, and the revenues of the business. In the same way, the aforementioned framework will aid new actors to easily deploy new services by providing tools that will lower the enter barriers to the market. The technologies here presented are were though having in mind an easy deployment of new services, which promotes the enlargement of the actual offer and strengthening the competence in a market that needs to be enriched.

## 7.1 Perspectives

The proposed QoE evaluation framework and its implementation presented in the previous sections allows the development of an online monitoring tool. However, it is still open the challenge to collect the data required by the framework in order to compute the QoE value. In this sense, the MMT software also provides a solution for this goal.

As mentioned in Section 4.1.1, the MMT software is composed of three principal components: *MMT Reporting*, *MMT Correlator* and *MMT Probe*. The latter was conceived as a DFI/DPI solution that analyzes the network traffic in order to extract data from the network traffic or stream under analysis. It is possible to take advantage of this capability in order to provide the



missing information.

The MMT Probe module was designed with a plugin API that can be used to extend the functionalities of the software. This API can be used to analyze the OTT stream and the Internet traffic between the user and the service in order to identify critical data for the framework. In particular, it is required to develop this analyzer in order to extract the values of the context variables previously inserted in the model. The challenge arises when trying to develop a generic extracting solution to meet any OTT model. Even when the efforts of developing such solution can lead a to an near to all-purpose extractor, it might not be possible to develop a solution flexible enough.

To understand this, it is required to think about the core of the QoE evaluation framework: the OTT model. This EFSM machine is built by following a semi-automatic process, where the quality of the obtained model strongly depend on the modeler's ability. Once the functional model is obtained, the augmentation process takes place, where the relevant quality impact factors must be chosen. This last procedure is a design decision that needs to be taken by the system modeler, making it impossible to be predicted in advance.

For this reason, both the OTT model and the value extractor need to be specified at the same time, in order to provide a piece of software that can meet the data requirements that the QoE evaluation framework has. In summary, the extractor needs to be designed having in mind the data that the model loaded in the MMT Correlator engine requires.

In addition to the extraction of the required data, the QoE framework also needs to present the data to the operator of the service. To this end, the MMT solution provides its Reporting module, which can be used to present the QoE values to the monitor entity. This module is based on a web user interface, and it has also been designed following an extensible approach. Both features open the possibility to easily deploy a monitoring web module that shows the predicted QoE values for the users of the platform.

Such solution will allow the OTT operator to keep track of the user's decisions and their QoE values, showing the best and worst case scenarios as predicted by the algorithms members of the framework.

Both proposed implementation will conclude the works presented here, providing a complete, integrated solution ready to be deployed on real OTT services.

---

# Bibliography

- [8] D. A. Aaker, “Measuring brand equity across products and markets”, *California management review*, 38, no., p. 103, 1996.
  - [9] Adobe Inc., *Video learning guide for flash: progressive and streaming video*, Accessed: 15 November 2016, Feb. 2011. [Online]. Available: [http://www.adobe.com/devnet/flash/learning\\_guide/video/part02.html](http://www.adobe.com/devnet/flash/learning_guide/video/part02.html).
  - [10] —, *Supported codecs for flash player*, Accessed: 15 November 2016, Feb. 2016. [Online]. Available: <https://helpx.adobe.com/flash/kb/supported-codecs-flash-player.html>.
  - [11] A. Ahmad, A. Floris, and L. Atzori, “Qoe-aware service delivery: a joint-venture approach for content and network providers”, in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, Jun. 2016, pp. 1–6. DOI: 10.1109/QoMEX.2016.7498972.
  - [12] M. Alreshoodi and J. Woods, “Qoe prediction model based on fuzzy logic system for different video contents”, in *2013 European Modelling Symposium*, Nov. 2013, pp. 635–639. DOI: 10.1109/EMS.2013.106.
  - [13] A. A. Andrews, J. Offutt, and T. R. Alexander, “Testing web applications by modeling with fsms”, *Software & Systems Modeling*, 4, no., pp. 326–345, 2005, ISSN: 1619-1374. DOI: 10.1007/s10270-004-0077-7. [Online]. Available: <http://dx.doi.org/10.1007/s10270-004-0077-7>.
  - [14] P. Casas, A. Sackl, S. Egger, and R. Schatz, “Youtube amp; facebook quality of experience in mobile broadband networks”, in *2012 IEEE Globecom Workshops*, Dec. 2012, pp. 1269–1274. DOI: 10.1109/GLOCOMW.2012.6477764.
  - [15] Z. Chen, S.-M. Tan, R. H. Campbell, and Y. Li, “Real time video and audio in the world wide web”, in *Fourth International World Wide Web Conference*, Boston, 1995, pp. 333–348.
  - [16] I. S. Consortium, *Internet host count history*, Accessed: 27 October 2016. [Online]. Available: <https://ftp.isc.org/www/survey/reports/current/>.
  - [17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, third. The MIT Press, 2009.
-

- 
- [18] N. Côté and J. Berger, “Speech communication”, in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Cham: Springer International Publishing, 2014, pp. 165–177, ISBN: 978-3-319-02681-7. DOI: 10.1007/978-3-319-02681-7\_12. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-02681-7\\_12](http://dx.doi.org/10.1007/978-3-319-02681-7_12).
  - [19] E. S. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, “A framework for qos-based routing in the internet”, RFC Editor, RFC 2386, Aug. 1998. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc2386.txt>.
  - [20] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service”, *IEEE Network*, 24, no., pp. 36–41, Mar. 2010, ISSN: 0890-8044. DOI: 10.1109/MNET.2010.5430142.
  - [21] L. Fitzpatrick, “Brief history youtube”, *Time Magazine*, no., 2010.
  - [22] M. Fransman, “Evolution of the telecommunications industry into the internet age”, *Jets Paper-University of Edinburgh Institute for Japanese European Technology Studies*, no., 2000.
  - [23] J. Frnda, M. Voznak, and L. Sevcik, “Impact of packet loss and delay variation on the quality of real-time video streaming”, *Telecommunication Systems*, 62, no., pp. 265–275, 2016, ISSN: 1572-9451. DOI: 10.1007/s11235-015-0037-2. [Online]. Available: <http://dx.doi.org/10.1007/s11235-015-0037-2>.
  - [24] M. N. Garcia, D. Dytko, and A. Raake, “Quality impact due to initial loading, stalling, and video bitrate in progressive download video services”, in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, Sep. 2014, pp. 129–134. DOI: 10.1109/QoMEX.2014.6982308.
  - [25] A. Ghezzi and M. Dramitinos, “Towards a future internet infrastructure: analyzing the multidimensional impacts of assured quality internet interconnection”, *Telematics and Informatics*, 33, no., pp. 613–630, 2016, ISSN: 0736-5853. DOI: <http://dx.doi.org/10.1016/j.tele.2015.10.003>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0736585315001239>.
  - [26] A. Gill *et al.*, *Introduction to the theory of finite-state machines*. McGraw-Hill New York, 1962, vol. 16.
  - [27] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “Youtube traffic characterization: a view from the edge”, in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’07, San Diego, California, USA: ACM, 2007, pp. 15–28, ISBN: 978-1-59593-908-1. DOI: 10.1145/1298306.1298310. [Online]. Available: <http://doi.acm.org/10.1145/1298306.1298310>.
  - [28] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, “Quality of service terminology in ip networks”, *IEEE Communications Magazine*, 41, no., pp. 153–159, Mar. 2003, ISSN: 0163-6804. DOI: 10.1109/MCOM.2003.1186560.
  - [29] W. Green, B. Lancaster, and J. Sladek, “Over-the-top services”, *Pipeline*, 4, no., 2006.
  - [30] H. Habibi Gharakheili, A. Vishwanath, and V. Sivaraman, “Perspectives on net neutrality and internet fast-lanes”, *SIGCOMM Comput. Commun. Rev.*, 46, no., pp. 64–69, Jan. 2016, ISSN: 0146-4833. DOI: 10.1145/2875951.2875962. [Online]. Available: <http://doi.acm.org/10.1145/2875951.2875962>.
-

- 
- [31] A. Hamam, M. Eid, A. El Saddik, and N. D. Georganas, “A fuzzy logic system for evaluating quality of experience of haptic-based applications”, in *Haptics: Perception, Devices and Scenarios: 6th International Conference, EuroHaptics 2008 Madrid, Spain, June 10-13, 2008 Proceedings*, M. Ferre, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 129–138, ISBN: 978-3-540-69057-3. DOI: 10.1007/978-3-540-69057-3\_14. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-69057-3\\_14](http://dx.doi.org/10.1007/978-3-540-69057-3_14).
- [32] W. C. Hardy, *QoS: Measurement and Evaluation of Telecommunications Quality of Service*. New York, NY, USA: John Wiley & Sons, Inc., 2001, ISBN: 0471499579.
- [33] R. Hastings, *Internet tolls and the case for strong net neutrality*, Accessed: 10 November 2016. [Online]. Available: <https://media.netflix.com/en/company-blog/internet-tolls-and-the-case-for-strong-net-neutrality>.
- [34] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, “Internet video delivery in youtube: from traffic measurements to quality of experience”, in *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*, E. Biersack, C. Callegari, and M. Matijasevic, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 264–301, ISBN: 978-3-642-36784-7. DOI: 10.1007/978-3-642-36784-7\_11. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-36784-7\\_11](http://dx.doi.org/10.1007/978-3-642-36784-7_11).
- [35] W. Ilsa GODLOVITCH, T. Bas KOTTERINK, W. J. Scott MARCUS, T. Pieter NOOREN, T. Jop ESMEIJER, and T. Arnold ROOSENDAAL, “Over-the-top (otts) players: market dynamics and policy challenges”, European Parliament’s Committee on the Internal Market and Consumer Protection, Study, Dec. 2015, Accessed: 14 November 2016. [Online]. Available: [http://www.europarl.europa.eu/RegData/etudes/STUD/2015/569979/IPOL\\_STU\(2015\)569979\\_EN.pdf](http://www.europarl.europa.eu/RegData/etudes/STUD/2015/569979/IPOL_STU(2015)569979_EN.pdf).
- [36] N. Inc., *About netflix*, Accessed: 28 October 2016. [Online]. Available: <https://media.netflix.com/en/about-netflix>.
- [37] D. Isenberg, “Rise of the stupid network”, *Computer Telephony*, 5, no., pp. 16–26, 1997.
- [38] ISO/IEC, “Information technology – dynamic adaptive streaming over http (dash) – part 1: media presentation description and segment formats”, International Organization for Standardization, Standard, 2014.
- [39] ITU, “Terms and definitions related to quality of service and network performance including dependability”, International Telecommunication Union, Recommendation E.800, Aug. 1994.
- [40] —, “Methods for subjective determination of transmission quality”, International Telecommunication Union, Recommendation P.800, Aug. 1996.
- [41] —, “New definitions for inclusion in recommendation itu-t p.10/g.100”, International Telecommunication Union, Recommendation P.10 Amendment 2, Jul. 2008.
- [42] —, “Quality of telecommunication services: concepts, models, objectives and dependability planning – terms and definitions related to the quality of telecommunication services”, International Telecommunication Union, Recommendation E.800 Amendment 2, Sep. 2008.
- [43] —, “Subjective video quality assessment methods for multimedia applications”, International Telecommunication Union, Recommendation P.910, Apr. 2008.
-

- 
- [44] —, “Estimating end-to-end performance in ip networks for data applications”, International Telecommunication Union, Recommendation G.1030, Feb. 2014.
  - [45] —, *Key ict indicators for developed and developing countries and the world (totals and penetration rates)*, Accessed: 27 October 2016. [Online]. Available: [http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2014/ITU\\_Key\\_2005-2014\\_ICT\\_data.xls](http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2014/ITU_Key_2005-2014_ICT_data.xls).
  - [46] J. I. Khan and R. Y. Zaghal, “Jitter and delay reduction for time sensitive elastic traffic for tcp-interactive based world wide video streaming over abone”, in *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, Oct. 2003, pp. 311–316. DOI: 10.1109/ICCCN.2003.1284187.
  - [47] P. Le Callet, S. Möller, A. Perkis, *et al.*, “Qualinet white paper on definitions of quality of experience”, *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, no., 2012.
  - [48] A. Leon-Garcia and I. Widjaja, *Communication Networks*, 2nd ed. New York, NY, USA: McGraw-Hill, Inc., 2004, ISBN: 007246352X, 9780072463521.
  - [49] F. Limbach, “Cooperative service provisioning with OTT players: An explorative analysis of telecommunication business models”, International Telecommunications Society (ITS), 25th European Regional ITS Conference, Brussels 2014 101388, Jun. 2014. [Online]. Available: <https://ideas.repec.org/p/zbw/itse14/101388.html>.
  - [50] V. Menkovski, G. Exarchakos, and A. Liotta, “Machine learning approach for quality of experience aware networks”, in *2010 International Conference on Intelligent Networking and Collaborative Systems*, Nov. 2010, pp. 461–466. DOI: 10.1109/INCOS.2010.86.
  - [51] S. Möller and A. Raake, “Motivation and introduction”, in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Cham: Springer International Publishing, 2014, pp. 3–9, ISBN: 978-3-319-02681-7. DOI: 10.1007/978-3-319-02681-7\_1. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-02681-7\\_1](http://dx.doi.org/10.1007/978-3-319-02681-7_1).
  - [52] M. S. Mushtaq, B. Augustin, and A. Mellouk, “Empirical study based on machine learning approach to assess the qos/qoe correlation”, in *2012 17th European Conference on Networks and Optical Communications*, Jun. 2012, pp. 1–7. DOI: 10.1109/NOC.2012.6249939.
  - [53] A. Odlyzko, “Paris metro pricing for the internet”, in *Proceedings of the 1st ACM Conference on Electronic Commerce*, ser. EC '99, Denver, Colorado, USA: ACM, 1999, pp. 140–147, ISBN: 1-58113-176-3. DOI: 10.1145/336992.337030.
  - [54] Opera Software, *A call for video on the web*, Accessed: 15 November 2016, Feb. 2007. [Online]. Available: <http://people.opera.com/howcome/2007/video/>.
  - [55] T. O'Reilly, *What is web 2.0*. "O'Reilly Media, Inc.", 2009.
  - [56] A. Perkis, P. Reichl, and S. Beker, “Business perspectives on quality of experience”, in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Cham: Springer International Publishing, 2014, pp. 97–108, ISBN: 978-3-319-02681-7. DOI: 10.1007/978-3-319-02681-7\_7.
  - [57] A. Petrenko, S. Boroday, and R. Groz, “Confirming configurations in efsm testing”, *IEEE Transactions on Software Engineering*, 30, no., pp. 29–42, Jan. 2004, ISSN: 0098-5589. DOI: 10.1109/TSE.2004.1265734.
-

- 
- [58] S. Pfeiffer, "Audio and video elements", in *The Definitive Guide to HTML5 Video*. Berkeley, CA: Apress, 2011, pp. 9–48, ISBN: 978-1-4302-3091-5. DOI: 10.1007/978-1-4302-3091-5\_2. [Online]. Available: [http://dx.doi.org/10.1007/978-1-4302-3091-5\\_2](http://dx.doi.org/10.1007/978-1-4302-3091-5_2).
- [59] J. Pokhrel, B. Wehbi, A. Morais, A. Cavalli, and E. Allilaire, "Estimation of qoe of video traffic using a fuzzy expert system", in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Jan. 2013, pp. 224–229. DOI: 10.1109/CCNC.2013.6488450.
- [60] T. Porter and X. H. Peng, "An objective approach to measuring video playback quality in lossy networks using tcp", *IEEE Communications Letters*, 15, no., pp. 76–78, Jan. 2011, ISSN: 1089-7798. DOI: 10.1109/LCOMM.2010.110310.101642.
- [61] J. Postel, "Transmission control protocol", RFC Editor, STD 7, Sep. 1981. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc793.txt>.
- [62] A. Poutanen, "Triple play services", *Innovation in Telecommunications*, no., p. 39, 2006.
- [63] A. Raake, *Speech quality of VoIP: assessment and prediction*. John Wiley & Sons, 2007.
- [64] A. Raake and S. Egger, "Quality and quality of experience", in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Cham: Springer International Publishing, 2014, pp. 11–33, ISBN: 978-3-319-02681-7. DOI: 10.1007/978-3-319-02681-7\_2. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-02681-7\\_2](http://dx.doi.org/10.1007/978-3-319-02681-7_2).
- [65] S. Ramachandran, "Netflix to pay comcast for smoother streaming", *The Wall Street Journal*, no., Feb. 2014.
- [66] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo, "The logarithmic nature of qoe and the role of the weber-fechner law in qoe assessment", in *2010 IEEE International Conference on Communications*, May 2010, pp. 1–5. DOI: 10.1109/ICC.2010.5501894.
- [67] P. Reichl, P. Maille, P. Zwickl, and A. Sackl, "A fixed-point model for qoe-based charging", in *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, ser. FhMN '13, Hong Kong, China: ACM, 2013, pp. 33–38, ISBN: 978-1-4503-2183-9. DOI: 10.1145/2491172.2491176. [Online]. Available: <http://doi.acm.org/10.1145/2491172.2491176>.
- [68] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience", *Telecommunication Systems*, 52, no., pp. 587–600, 2013, ISSN: 1572-9451. DOI: 10.1007/s11235-011-9503-7.
- [69] P. Reichl and P. Zwickl, "The economics of quality of experience: recent advances and next steps", *Invited Paper, IEEE COMSOC MMTC E-Letter*, 10, no., 2015.
- [70] U. Reiter, K. Brunnström, K. De Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank, "Factors influencing quality of experience", in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Cham: Springer International Publishing, 2014, pp. 55–72, ISBN: 978-3-319-02681-7. DOI: 10.1007/978-3-319-02681-7\_4. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-02681-7\\_4](http://dx.doi.org/10.1007/978-3-319-02681-7_4).
- [71] H. Rifaï, S. Mohammed, and A. Mellouk, "A brief synthesis of qos-qoe methodologies", in *2011 10th International Symposium on Programming and Systems*, Apr. 2011, pp. 32–38. DOI: 10.1109/ISPS.2011.5898880.
-

- 
- [72] T. S.A., *A digital manifesto: an open and safe internet experience for all*, Accessed: 28 October 2016, Jan. 2014. [Online]. Available: <https://www.telefonica.com/en/web/public-policy/digital-manifesto/>.
  - [73] A. Sackl, S. Egger, P. Zwickl, and P. Reichl, "The qoe alchemy: turning quality into money. experiences with a refined methodology for the evaluation of willingness-to-pay for service quality", in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, Jul. 2012, pp. 170–175. DOI: 10.1109/QoMEX.2012.6263846.
  - [74] A. Sackl and R. Schatz, "Evaluating the influence of expectations, price and content selection on video quality perception", in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, Sep. 2014, pp. 93–98.
  - [75] J. Sandoval, A. Ehijo, A. Casals, and C. Estevez, "New model and open tools for real testing of qoe in mobile broadband services and the transport protocol impact: the operator's approach", in *2013 IEEE Latin-America Conference on Communications*, Nov. 2013, pp. 1–5. DOI: 10.1109/LatinCom.2013.6759831.
  - [76] R. Schatz, T. Hoffeld, L. Janowski, and S. Egger, "From packets to people: quality of experience as a new measurement challenge", in *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*, E. Biersack, C. Callegari, and M. Matijasevic, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 219–263, ISBN: 978-3-642-36784-7. DOI: 10.1007/978-3-642-36784-7\_10. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-36784-7\\_10](http://dx.doi.org/10.1007/978-3-642-36784-7_10).
  - [77] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoffeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming", *IEEE Communications Surveys Tutorials*, 17, no., pp. 469–492, Firstquarter 2015, ISSN: 1553-877X. DOI: 10.1109/COMST.2014.2360940.
  - [78] J. Shaikh, M. Fiedler, and D. Collange, "Quality of experience from user and network perspectives", *annals of telecommunications - annales des télécommunications*, 65, no., pp. 47–57, 2010, ISSN: 1958-9395. DOI: 10.1007/s12243-009-0142-x. [Online]. Available: <http://dx.doi.org/10.1007/s12243-009-0142-x>.
  - [79] W. Song, D. Tjondronegoro, and M. Docherty, "Saving bitrate vs. pleasing users: where is the break-even point in mobile video quality?", in *Proceedings of the 19th ACM International Conference on Multimedia*, ser. MM '11, Scottsdale, Arizona, USA: ACM, 2011, pp. 403–412, ISBN: 978-1-4503-0616-4. DOI: 10.1145/2072298.2072351. [Online]. Available: <http://doi.acm.org/10.1145/2072298.2072351>.
  - [80] J. Sujata, S. Sohag, D. Tanu, D. Chintan, P. Shubham, and G. Sumit, "Impact of over the top (ott) services on telecom service providers", *Indian Journal of Science and Technology*, 8, no., 2015, ISSN: 0974 - 5645. [Online]. Available: <http://www.indjst.org/index.php/indjst/article/view/62238>.
  - [81] A. Testolin, M. Zanforlin, M. D. F. D. Grazia, D. Munaretto, A. Zanella, M. Zorzi, and M. Zorzi, "A machine learning approach to qoe-based video admission control and resource allocation in wireless systems", in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, Jun. 2014, pp. 31–38. DOI: 10.1109/MedHocNet.2014.6849102.
-

- 
- [82] I. Van Beijnum, *Netherlands becomes world's second "net neutrality" country*, Accessed: 14 November 2016, Oct. 2012. [Online]. Available: <http://arstechnica.com/tech-policy/2012/05/netherlands-becomes-worlds-second-net-neutrality-country/>.
- [83] A. Van Moorsel, "Metrics for the internet age: quality of experience and quality of business", in *Fifth International Workshop on Performability Modeling of Computer and Communication Systems, Arbeitsberichte des Instituts für Informatik, Universität Erlangen-Nürnberg, Germany*, Citeseer, vol. 34, 2001, pp. 26–31.
- [84] M. Varela, L. Skorin-Kapov, and T. Ebrahimi, "Quality of service versus quality of experience", in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Cham: Springer International Publishing, 2014, pp. 85–96, ISBN: 978-3-319-02681-7. DOI: 10.1007/978-3-319-02681-7\_6. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-02681-7\\_6](http://dx.doi.org/10.1007/978-3-319-02681-7_6).
- [85] S. Wahlmueller, P. Zwickl, and P. Reichl, "Pricing and regulating quality of experience", in *Next Generation Internet (NGI), 2012 8th EURO-NGI Conference on*, Jun. 2012, pp. 57–64. DOI: 10.1109/NGI.2012.6252165.
- [86] N. Walkinshaw, R. Taylor, and J. Derrick, "Inferring extended finite state machine models from software executions", *Empirical Software Engineering*, 21, no., pp. 811–853, 2016, ISSN: 1573-7616. DOI: 10.1007/s10664-015-9367-7. [Online]. Available: <http://dx.doi.org/10.1007/s10664-015-9367-7>.
- [87] B. Wehbi, E. M. de Oca, and M. Bourdellès, "Events-based security monitoring using MMT tool", in *Fifth IEEE International Conference on Software Testing, Verification and Validation, ICST 2012, Montreal, QC, Canada, April 17-21, 2012*, 2012, pp. 860–863. DOI: 10.1109/ICST.2012.188.
- [88] T. Wheeler, *Fcc chairmain to tom wheeler: this is how we will ensure net neutrality*, Accessed: 14 November 2016, Feb. 2015. [Online]. Available: <https://www.wired.com/2015/02/fcc-chairman-wheeler-net-neutrality>.
- [89] S. Winkler, "Video quality measurement standards-current status and trends", *Proceedings of ICICS 2009*, no., 2009.
- [90] E. Woollacott, *Europe votes for net neutrality in no uncertain terms*, Accessed: 14 November 2016, Apr. 2014. [Online]. Available: <http://www.forbes.com/sites/emmawoollacott/2014/04/03/europe-votes-for-net-neutrality-in-no-uncertain-terms/>.
- [91] T. Wu, "Network neutrality, broadband discrimination", *Journal of Telecommunications and High Technology Law*, no., p. 141, 2003. DOI: 10.2139/ssrn.388863.
- [92] YouTube, *About youtube*, Accessed: 27 October 2016. [Online]. Available: <https://www.youtube.com/yt/about/>.
-





## Appendix A

# beIN Sports Connect Service Description

## A.1 Presentation

BeIN Sports Connect is offered as an OTT service that allows the users to watch beIN Sports France channels. These streams provide access to sports content such as matches of Brazil 2014 world cup, 2015 Champions League, French Rugby league, Wimbledon tennis tournament etc.

Traditionally, beIN Sports channels are being offered as a premium channel pack paid in addition to a basic cable subscription, giving access to two main channels (beIN Sports 1 and 2) and 8 other signals used to stream special events when the first two are already being used.

In addition to this traditional schema, beIN Sports offers the access to the same content (all 10 channels) through Internet, without the need of a basic cable subscription. Even for the subscribers who watch beIN Sports with a TV-cable subscription, beIN Sports Connect must be paid separately if the user wants to access the content in other screens than his/her TV. However, beIN Sports Connect allows new users to watch beIN Sports content without a cable operator, with the restriction of being able to watch the channels through Internet and using a browser (for PC/Mac) or the specific application (for iPhone/Android devices).

## A.2 Service Options

BeIN Sports Connect offers its customers 3 types of subscriptions:

- Subscription with a one-year engagement, offered with a price of 5,5€ during the first two months and 11€ from the third month until the end of the subscription, which may not be less than a year.
  - Subscription without engagement. As this schema is more flexible for the user (in the sense that the service can be suspended at any moment), it is offered with the slightly higher price of 12€ per month.
  - “Discovery” Subscription. If the user has a code to try beIN Sports Connect for a several days, this subscription has to be chosen in order to activate the promotional code. When using this option, the subscription process does not request the user to register any payment method, however, the user can “upgrade” the subscription (to a paid one) once the promotional offer has ended.
-

### A.3 Technical Details

In help page of the service, the user can find some technical specifications of the service offered:

- The service is accessible through 3G, 4G and WiFi networks. It is recommended to use a WiFi network in order to get the optimal quality.
- The image quality depends on the Internet connection of the user. In other words, the stream is capable to adapt to the network conditions by degrading itself if optimal conditions are not met. As explained in the next section, the streaming can be watched in a small HTML player or in full screen. Between these two playing options, just the second one has the ability to offer the content with HD quality if the user bandwidth is enough.

### A.4 Service Operation

The main functionality of beIN Sports connect can be divided into 2 main phases:

- Service registration
  - The user can access the site directly by <https://beinsportsconnect.fr/index.html> or by clicking the link on the beIN Sports main site <http://www.beinsports.fr/>.
  - The user clicks the button “S’abonner”, which displays an HTML form showing three steps to complete the registration. These three stages are completed in the same form, clicking a validation button once the user has completed all the steps.
  - The first stage is selecting which plan does the user want, selecting one of the offers presented in before.
  - After selecting the plan, the user is prompted to fill his personal data: first name, last name, birthday, email (twice for confirmation), create a password, his address, postal code, city and the country. This last information is important because this service is only available for Monaco, France and its territories outside Europe.
  - In the final step, the user must select his payment method (Visa or MasterCard are the unique accepted methods). The payment method is offered to the user only if he selects the subscription that requires a payment, i.e. any subscription but “offre découvert”. In addition, the user must accept some conditions related to the subscription, such as the storage of the user’s card information in a secure third-party server, the general conditions of sell and select if whether the user wants to receive beIN Sports offers or not.
  - The service now redirects the user to a third party site<sup>1</sup> in order to process the payment. In this new site, the user must provide the data of his credit card in order to be stored in a secure server. This storage is required to renew the service each month in the case of the paid plans.
  - If the payment is rejected, an account for the user is created in the beIN Sports Connect site, but it is not associated with any subscription plan, which does not allow the user to access the content of the service. Later, the user can log into his account and pay for any of the subscription plans offered

---

<sup>1</sup><http://www.ogone.com>

---

- 
- Once the payment has been accepted (following the protocols established by user's bank) the site logs the user in automatically, allowing him to use the service as soon as the payment was made.
  - Service utilization
    - A user that already has a subscription may reach the site by beIN Sports main site <http://www.beinsports.fr/> or directly in <https://beinsportsconnect.fr/index.html>.
    - To access the content, the user must log into his account by clicking the button “Se connecter”. This will show a form asking for the user's credentials (login email and password) that were created during the subscription process. The form also offers the ability to remember the session, allowing the user to avoid the login screen each time he visits the site. Once the credentials are filled in the form, clicking on the “Connexion” button logs the user in.
    - Once the user is logged in, the site will load immediately the content being streamed in beIN Sports 1 and it will be presented in a small player contained in the webpage. In this state, the user has the ability to control some playing options like pause or stop the reproduction, control the volume, or watch it in full screen.
    - If the user decides to stop watching the service, he can simply close the page at any moment, or perform a more “graceful” exit by exiting full screen mode (if necessary), stopping the player, logging out and quitting the web browser.
-



# Appendix B

## Javascript notation for the Extended Finite State Machine $M_3$

### B.1 Represenation of the Extended Finite State Machine $M_3$

```

{ id: "M3",
  hascontext: true,
  logdata: true,
  onCreate: function() {},
  onDelete: function() {},
  events: ['a', 'b', 'c', 'd', 'timeout.tout'],
  states: [{ id: 'A' }, { id: 'B' }, { id: 'C' }, { id: 'D' }],
  contextvars: [{varname: 'v1', startval: 0}],
  transitions: [
    { from: 'A',
      to: 'B',
      event: 'a'
    },
    { from: 'B',
      to: 'B',
      event: 'b',
      updatefunctions: [{
        varname: 'v1',
        fct: function(oldval){
          return 1;
        }
      }]
    },
    { from: 'B',
      to: 'D',
      event: 'c',
      conditions: [{
        fct: function(active_state, evt, msg, opt){
          if(active_state.contextvariables["v1"].value === 1)
            return true;
          return false;
        }
      }]
    },
  ],
}

```

---

{ from: 'B',	35
to: 'C',	36
event: 'c',	37
conditions: [{	38
fct: function(active_state, evt, msg, opt){	39
if(active_state.contextvariables["v1"].value === 0)	40
return true;	41
return false;	42
}	43
}]	44
},	45
{ from: 'C',	46
to: 'C',	47
event: 'd'	48
},	49
{ from: 'D',	50
to: 'D',	51
event: 'd'	52
},	53
{ from: 'D',	54
to: 'B',	55
event: 'timeout.tout',	56
actions: [{fct: mmt.startTimer, opts: {timeout: 2000, name: 'tout'}}],	57
updatefunctions: [{	58
varname: 'v1',	59
fct: function(oldval){	60
return 0;	61
}	62
}]	63
},	64
{ from: 'C',	65
to: 'A',	66
event: 'timeout.tout',	67
actions: [{fct: mmt.startTimer, opts: {timeout: 2000, name: 'tout'}}],	68
updatefunctions: [{	69
varname: 'v1',	70
fct: function(oldval){	71
return 0;	72
}	73
}]	74
}	75
]	76
}	77

---

## B.2 Computed paths for the Extended Finite State Machine $M_3$

---

[ [ { state: 'A', event: 'a', vars: [ { name: 'v1', value: '0' } ] },	1
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: '0' } ] },	2
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: 1 } ] },	3
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: 1 } ] },	4
{ state: 'B', event: null, vars: [ { name: 'v1', value: 1 } ] },	5
[ { state: 'A', event: 'a', vars: [ { name: 'v1', value: '0' } ] },	6
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: '0' } ] },	7
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: 1 } ] },	8
{ state: 'B', event: 'c', vars: [ { name: 'v1', value: 1 } ] },	9
{ state: 'D', event: null, vars: [ { name: 'v1', value: 1 } ] },	10
[ { state: 'A', event: 'a', vars: [ { name: 'v1', value: '0' } ] },	11
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: '0' } ] },	12
{ state: 'B', event: 'c', vars: [ { name: 'v1', value: 1 } ] },	13
{ state: 'D', event: 'd', vars: [ { name: 'v1', value: 1 } ] },	14
{ state: 'D', event: null, vars: [ { name: 'v1', value: 1 } ] },	15
[ { state: 'A', event: 'a', vars: [ { name: 'v1', value: '0' } ] },	16
{ state: 'B', event: 'b', vars: [ { name: 'v1', value: '0' } ] },	17
{ state: 'B', event: 'c', vars: [ { name: 'v1', value: 1 } ] },	18
{ state: 'D',	19
event: 'timeout.tout',	20
vars: [ { name: 'v1', value: 1 } ] },	21
{ state: 'B', event: null, vars: [ { name: 'v1', value: 0 } ] } ] ]	22

---

## B.3 Augmented paths for the Extended Finite State Machine $M_3$

---

[ [ { state: 'A',	1	[ { state: 'A',	21
event: 'a',	2	event: 'a',	22
vars: [ { name: 'v1', value: 0 } ],	3	vars: [ { name: 'v1', value: 0 } ],	23
qoe: 1 },	4	qoe: 1 },	24
{ state: 'B',	5	{ state: 'B',	25
event: 'b',	6	event: 'b',	26
vars: [ { name: 'v1', value: 0 } ],	7	vars: [ { name: 'v1', value: 0 } ],	27
qoe: 1 },	8	qoe: 1 },	28
{ state: 'B',	9	{ state: 'B',	29
event: 'b',	10	event: 'b',	30
vars: [ { name: 'v1', value: 1 } ],	11	vars: [ { name: 'v1', value: 1 } ],	31
qoe: 2 },	12	qoe: 2 },	32
{ state: 'B',	13	{ state: 'B',	33
event: 'b',	14	event: 'c',	34
vars: [ { name: 'v1', value: 1 } ],	15	vars: [ { name: 'v1', value: 1 } ],	35
qoe: 2 },	16	qoe: 2 },	36
{ state: 'B',	17	{ state: 'D',	37
event: null,	18	event: null,	38
vars: [ { name: 'v1', value: 1 } ],	19	vars: [ { name: 'v1', value: 1 } ],	39
qoe: 2 } ],	20	qoe: 2 } ],	40

---



---

[ { state: 'A',	41 [ { state: 'A',	61
event: 'a',	42   event: 'a',	62
vars: [ { name: 'v1', value: 0 } ],	43   vars: [ { name: 'v1', value: 0 } ],	63
qoe: 1 },	44   qoe: 1 },	64
{ state: 'B',	45 { state: 'B',	65
event: 'b',	46   event: 'b',	66
vars: [ { name: 'v1', value: 0 } ],	47   vars: [ { name: 'v1', value: 0 } ],	67
qoe: 1 },	48   qoe: 1 },	68
{ state: 'B',	49 { state: 'B',	69
event: 'c',	50   event: 'c',	70
vars: [ { name: 'v1', value: 1 } ],	51   vars: [ { name: 'v1', value: 1 } ],	71
qoe: 2 },	52   qoe: 2 },	72
{ state: 'D',	53 { state: 'D',	73
event: 'd',	54   event: 'timeout.tout',	74
vars: [ { name: 'v1', value: 1 } ],	55   vars: [ { name: 'v1', value: 1 } ],	75
qoe: 2 },	56   qoe: 2 },	76
{ state: 'D',	57 { state: 'B',	77
event: null,	58   event: null,	78
vars: [ { name: 'v1', value: 1 } ],	59   vars: [ { name: 'v1', value: 0 } ],	79
qoe: 2 } ],	60   qoe: 1 } ],	80

---

# Appendix C

## Video Generation Tool

### C.1 segment.sh

```
#!/bin/bash 1
# First argument, number or segments and case 3
if [ -z "$1" ] 4
then 5
    echo "No number of segments supplied" 6
    exit 1 7
fi 8
9
if [ -z "$2" ] 10
then 11
    echo "No case supplied" 12
    exit 1 13
fi 14
15
# Initialize variables 16
CASE="$2" 17
BASEDIR=/home/notts/vids 18
SEGMENTS=$BASEDIR/segments-$CASE 19
ORIGFOLD=$BASEDIR/orig 20
ORIGNAME=2min-$CASE.mp4 21
22
VIDEOCODEC=libx264 23
AUDIOCODEC=libfaac 24
25
if [ ! -d "$SEGMENTS" ]; then 26
    mkdir $SEGMENTS 27
fi 28
29
length=$(( 120 / $1 )) 30
31
echo "Each segment length = $length s." 32
33
# FIRST: Segment the video 34
echo "Segmenting original video" 35
for (( i=0; i < $1; i++ )) 36
```

---

```

do
    start_time=$(( $i * length ))
    #echo "start_time = $start_time"
    min_start=$(( $start_time / 60 ))
    sec_start=$(( $start_time % 60 ))
    echo "Generating segment $i"
    ffmpeg -i $ORIGFOLD/$ORIGNAME -ss 00:00:min_start:$sec_start -t 00:00:$length -codec:v
        $VIDEOCODEC -codec:a $AUDIOCODEC $SEGMENTS/segment-$i-orig.mp4 #&>
        $SEGMENTS/segment-$i-orig.specs
    echo "Done"
done
echo "Segmentation done"

for (( i=0; i < $1; i++ ))
do
    # SECOND: Re-encode the segments
    for b in 6M 3M 1500k 800k 400k
    #for b in 256k
    do
        echo "Compressing segment $i in $b"
        ffmpeg -i $SEGMENTS/segment-$i-orig.mp4 -codec:v $VIDEOCODEC -b:v $b -codec:a
            $AUDIOCODEC $SEGMENTS/segment-$i-$b.mp4 #&>$SEGMENTS/segment-$i-
            $b.specs
        echo "Done"
    done
    echo "Bitrate reencoding done"

    # THIRD: Insert buffering events at the beginnig of the segment
    for t in 2 3 5 8 20
    do
        suffix='seg'
        echo "Adding $t seconds of buffering at the beginnig of segment $i"
        #screenshot image of the first frame
        echo "Extracting first frame"
        ffmpeg -ss 00:00:00 -i $SEGMENTS/segment-0-6M.mp4 -frames:v 1 out1.jpg #>/dev/
            null

        #Generate Silent image mp4
        echo "Generating silent MP4 of $t sec."
        ffmpeg -f lavfi -i aevalsrc=0 -loop 1 -i "out1.jpg" -t "$t" -c:v $VIDEOCODEC -crf 23 -
            pix_fmt yuv420p -c:a $AUDIOCODEC "freeze_frame.mp4" #>/dev/null

        #concat freeze frame and original video
        echo "Concatenating silent MP4 and video"
        suffix='seg'
        ffmpeg -i "freeze_frame.mp4" -i "$SEGMENTS/segment-0-6M.mp4" -filter_complex "
            [0:0] [0:1] [1:0] [1:1] concat=n=2:v=1:a=1 [v] [a]" -map "[v]" -map "[a]" -c:v
            $VIDEOCODEC -crf 23 -pix_fmt yuv420p -c:a $AUDIOCODEC "$SEGMENTS/
            segment-$i-prebuf$t$suffix-6M.mp4" #&>$SEGMENTS/segment-$i-prebuf$t-$i-6M.
            specs

        rm out1.jpg freeze_frame.mp4
    done

```

---

---

```

#FOURTH: Insert buffering events
for f in 2 3 5 8 20
do
    echo "Adding $f seconds of buffering at the end of segment $i"
    suffix='seg'
    # Get the last frame of the segment
    framenummer='ffprobe -show_streams -select_streams v $SEGMENTS/segment-$i-orig.
    mp4 2> /dev/null | awk -F= '/^nb_frames/ { print $2-1 }'
    ffmpeg -i $SEGMENTS/segment-$i-orig.mp4 -vf select='eq\((n,$framenummer\) -vframes
    1 out1.jpg

    #Generate Silent image mp4
    echo "Generating silent MP4 of $f sec."
    ffmpeg -f lavfi -i aevalsrc=0 -loop 1 -i "out1.jpg" -t "$f" -c:v $VIDEOCODEC -crf 23 -
    pix_fmt yuv420p -c:a $AUDIOCODEC "freeze_frame.mp4" #>/dev/null

    #concat freeze frame and original video
    echo "Concatenating silent MP4 and video"
    suffix='seg'
    ffmpeg -i "$SEGMENTS/segment-$i-orig.mp4" -i "freeze_frame.mp4" -filter_complex "
    [0:0] [0:1] [1:0] [1:1] concat=n=2:v=1:a=1 [v] [a]" -map "[v]" -map "[a]" -c:v
    $VIDEOCODEC -crf 23 -pix_fmt yuv420p -c:a $AUDIOCODEC "$SEGMENTS/
    segment-$i-buf$f$suffix.mp4" #&>$SEGMENTS/segment-0-prebuf$i$suffix-6M.specs

    rm out1.jpg freeze_frame.mp4
    echo "Done"
done

echo "Insertion of buffering events done"

#FIFTH: Mixed perturbations.
for b in 6M 3M 1500k 800k 400k
do
    for f in 2 3 5 8 20
    do
        echo "Adding $f seconds of buffering at the end of segment $i, with $b bitrate"
        suffix='seg'
        # Get the last frame of the segment
        framenummer='ffprobe -show_streams -select_streams v $SEGMENTS/segment-$i
        -orig.mp4 2> /dev/null | awk -F= '/^nb_frames/ { print $2-1 }'
        ffmpeg -i $SEGMENTS/segment-$i-orig.mp4 -vf select='eq\((n,$framenummer\)
        -vframes 1 out1.jpg

        #Generate Silent image mp4
        echo "Generating silent MP4 of $f sec."
        ffmpeg -f lavfi -i aevalsrc=0 -loop 1 -i "out1.jpg" -t "$f" -c:v $VIDEOCODEC -
        crf 23 -pix_fmt yuv420p -c:a $AUDIOCODEC "freeze_frame.mp4" #>/dev/
        null
    
```

---

---

```

#concat freeze frame and original video 123
echo "Concatenating silent MP4 and video" 124
suffix='seg' 125
ffmpeg -i "$SEGMENTS/segment-$i-orig.mp4" -i "freeze_frame.mp4" - 126
    filter_complex "[0:0] [0:1] [1:0] [1:1] concat=n=2:v=1:a=1 [v] [a]" -map "[v]" -
    map "[a]" -c:v $VIDEOCODEC -b:v $b -crf 23 -pix_fmt yuv420p -c:a
    $AUDIOCODEC "$SEGMENTS/segment-$i-buf${suffix}-$b.mp4" #&>
    $SEGMENTS/segment-$i-buf${suffix}-$b.specs 127

rm out1.jpg freeze_frame.mp4 128
echo "Done" 129
done 130
done 131
echo "Mixed perturbations done" 132
done 133
done 134

```

---

## C.2 concat.sh

---

```
#!/bin/bash
# First argument, file specifying how to concat the videos
if [ -z "$1" ]
then
    echo "No testcases file supplied"
    exit 1
fi

if [ -z "$2" ]
then
    echo "No case supplied"
    exit 1
fi

if [ -z "$3" ]
then
    echo "No number of segments supplied"
    exit 1
fi

# Initialize variables

function join { local IFS="$1"; shift; echo "$*"; }

CASE="$2"
BASEDIR=/Users/diego/videos-journal
SEGMENTS=$BASEDIR/segments-$CASE
FINAL_DIR=$BASEDIR/vids-$CASE

OLDIFS=$IFS
COLNUMBER=$3
cases[0]="test";
counter=0

if [ ! -d "$FINAL_DIR" ]; then
    mkdir $FINAL_DIR
fi

# Preload the content of the CSV
while IFS=$OLDIFS read line || [ -n "$line" ]; do
    IFS=',' read -a array <<< "$line"
    for index in ${!array[@]}
    do
        indice=$(( counter * COLNUMBER + index ))
        cases[indice]=${array[index]}
    done
    counter=$(( counter + 1 ))
done < "$1"

# Iterate over all cases
for (( i=0; i<counter; i++ ))
do
    tempfiles=()
    casename="lala"

```

---

---

```

for (( j=0; j<COLNUMBER; j++ ))
do
    index=$(( i * COLNUMBER + j ))
    if [ $j -eq 0 ]; then
        echo "Processing case ${cases[index]}"
        casename=${cases[index]}
        echo "Filename = ${cases[index]}-$CASE.mp4"
    else
        tempfiles[$j]="temp$j.ts"
        echo "Generating piece $j"
        # echo "segment-$(( $j - 1 ))-${cases[index]}.mp4"
        ffmpeg -i $SEGMENTS/segment-$(( $j - 1 ))-${cases[index]}.mp4 -c copy -bsf:v
            h264_mp4toannexb -f mpegts temp$j.ts #&>/dev/null
        echo "Done"
    fi
done

echo "Concatenating the pieces"
LIST="$(join \ | ${tempfiles[@]})"
ffmpeg -f mpegts -i "concat:$LIST" -c copy -bsf:a aac_adtstoasc $FINAL_DIR/$casename-
    $CASE.mp4 #&>$FINAL_DIR/$casename-$CASE.specs
echo "Done"

rm -f temp*

done

exit 0

```

---

# Appendix D

## Validation Survey

1. Do you use any Over-The-Top (OTT) services (like Netflix, Spotify or YouTube)?
    - ☐ Yes
    - ☐ No
  2. How often do you use OTT service?
    - ☐ Multiple times a day
    - ☐ Once a day
    - ☐ Multiple times a week
    - ☐ Once a week
    - ☐ Multiple times a month
    - ☐ Once a month or less
  3. For how long have you been using OTT services?
    - ☐ Less than a week
    - ☐ Since a week or two
    - ☐ Since a month or two
    - ☐ Since half a year
    - ☐ Since a year
    - ☐ Since more than a year
  4. Do you consider yourself as a loyal user of OTT services?
    - ☐ Yes
    - ☐ No
  5. Some Video On Demand (VOD) OTT services have the ability to automatically check the network conditions and stop the video stream if the network conditions are not good enough to continue. Do you like this particular feature on OTT services?
    - ☐ Yes
    - ☐ No
  6. Does this feature influence your loyalty with the service?
    - ☐ Yes
    - ☐ No
  7. If so, does it increase or decrease your loyalty?
    - ☐ It *increases* my loyalty.
    - ☐ It *decreases* my loyalty.
-



8. Which video quality do you consider satisfiable for this type of video services?
- ☐ 6000 Kpbs (Full HD quality)
  - ☐ 3000 Kpbs (HD quality)
  - ☐ 1500 Kpbs (DVD quality)
  - ☐ 800 Kpbs (Standard TV quality)
  - ☐ 400 Kpbs (Low TV quality)
9. When watching video OTT services, sometimes the video stops in order to load more data. How long do you think it is a tolerable time to wait for this new data to arrive?
- ☐ 2 seconds
  - ☐ 3 seconds
  - ☐ 5 seconds
  - ☐ 8 seconds
10. Considering the length given in the last question, how many events like this can you consider tolerable in a 2-minutes video?
- ☐ 0 events
  - ☐ 1 events
  - ☐ 2 events
  - ☐ 3 events
  - ☐ 4 events
  - ☐ 5 events
11. There exist various OTT providers offering similar services. Did you change of provider during the last 2 months?
- ☐ Yes
  - ☐ No
12. If an OTT service you use offers a paid subscription with a minimal engagement time, do you feel committed with the service during this period?
- ☐ Yes
  - ☐ No
13. When paying for your OTT service subscription, the corresponding transaction can be sometimes secure (using secure channels to communicate with your bank). Does this feature influence your trust and loyalty with the service?
- ☐ Yes
  - ☐ No
14. If yes, does this increase or decrease your loyalty towards the service?
- ☐ It *increases* my loyalty
  - ☐ It *decreases* my loyalty
15. What other features (except for engagement) influence your loyalty and/or trust towards the service?

---

---

# Appendix E

## Version française abrégée

### E.1 Chapitre 1 : Introduction

#### E.1.1 Motivation

Internet est devenu une plateforme effective et fiable afin d'offrir une valeur ajoutée aux utilisateurs. Son taux d'adoption a augmenté ainsi que la variété des dispositifs qui le supportent, permettant aux utilisateurs d'avoir une connexion active sur d'autres appareils que les postes de travail. De ce fait, un nouveau marché de distribution de contenu s'est ouvert en se basant sur deux caractéristiques : l'expansion continue du réseau qui permet d'offrir des services à un nombre croissant d'utilisateurs ; et l'implémentation de liens haut débit qui permettent d'offrir des services extensibles, qui peuvent s'adapter aux différentes conditions du réseau. Cependant, il faut analyser une apparente contradiction dans le dernier postulat.

Malgré la capacité d'expansion d'Internet, il a été conçu comme un réseau « meilleur effort », c'est-à-dire, la distribution des données et son intégrité ne sont pas garanties par le réseau. Cette caractéristique pose des défis aux différents services comme la téléphonie et la télévision qui ont été conçus pour réserver certaine quantité de ressources pour maintenir la communication, ce qui permet de gérer facilement la *qualité* de ces services.

Pendant que les entreprises commençaient à offrir des services sur Internet, la garantie de la qualité a commencé à être un vrai défi : les services – surtout multimédia – doivent se battre pour des ressources qui sont toujours occupées, interfèrent avec d'autres services et produisent scénarios de « mauvaise qualité » qui ne répondent pas aux attentes de l'utilisateur. Des études sur les effets des différentes détériorations du réseau sur les services multimédia peuvent être trouvés dans la littérature (cf. les articles [15], [23], [46], [60]).

Même si l'état du réseau représente le facteur le plus important de la qualité d'utilisation par l'utilisateur, elle n'est pas la seule. Le vrai défi pour les services qui utilisent Internet comme réseau de distribution est d'assurer à la fois une qualité de service et les revenus économiques. En utilisant Internet comme réseau de distribution de contenu, les fournisseurs de services baissent les coûts de déploiement du service, au prix d'une perte du contrôle de la procédure de distribution. Il faut noter ici que les réseaux de distribution de données sont, usuellement, gérés par les sociétés de télécommunications, qui n'obtiennent pas de revenus pour livrer les contenus des fournisseurs multimédia. Cette situation augmente les coûts opérationnels du réseau, leur laissant le pouvoir de limiter l'usage de ces services alternatifs qui réduisent leurs revenus. Trouver une convention entre les deux marchés est la question principale à résoudre.

---

Considérant tous ces éléments, il faut assurer un haut niveau de qualité même sur des scénarios avec de hautes perturbations réseau, tout en visant d'améliorer le niveau de satisfaction de l'utilisateur et les chiffres d'affaires. Le travail présenté dans cette thèse a pour objectif de mettre à disposition des fournisseurs de services, la technologie qui leur permettra de surveiller leurs services, optimiser leurs efforts opérationnels et livrer un service de haute qualité à l'utilisateur.

### E.1.2 Problématique

Au début de l'ère Internet, le concept de *qualité* avait été lié aux métriques réseau, par exemple le taux de perte de paquets ou le délai. En considérant les types de services implémentés sur l'Internet de cet ère, ces métriques suffisaient à garantir le fonctionnement des services. L'ère du *Web 2.0* introduisait le contenu enrichi sur Internet sous forme de contenu interactif, audio et vidéo sur les sites. Ce changement a rendu les sites web plus dynamiques et plus exigeants pas seulement pour les terminaux des utilisateurs, mais aussi pour le réseau qui transporte ces contenus. En considérant les quantités des données et leur nouvelle nature, les paramètres de Quality of Service (QoS) peuvent impacter les services multimédias de manières imprévisibles. De ce fait, la QoS ne représente pas une bonne métrique pour prédire les effets sur les flux multimédia, car ils dépendent fortement d'autres métriques objectives de haut niveau ou même le type de contenu.

Pour faire face à ces défis, il a été nécessaire de trouver une métrique plus étendue, de manière à intégrer la qualité depuis le point de vue de l'utilisateur. En premier temps, le concept de la *QoS Perçue* a été adaptée à cet effet, qui a évolué vers le concept de la *Quality of Experience (QoE)*. Même lorsque ce concept reconnaît l'approche multidisciplinaire de la qualité, son analyse a été focalisé sur l'étude des relations entre la QoS et la QoE, c'est-à-dire, la qualité *expérimentale* de l'utilisateur. Les travaux présentés dans cette thèse visent à récupérer la nature holistique de la QoE en proposant une étude qui intègre les paramètres d'affaires dans l'analyse de la QoE.

### E.1.3 Objectifs et planification de la thèse

La nature multidisciplinaire de la QoE demande une proposition d'approche multidisciplinaire pour analyser sa valeur. Les travaux décrits dans cette thèse présentent une analyse holistique de l'impact des paramètres objectifs et d'affaires sur la QoE sous forme de quatre contributions principales.

Le Chapitre 2 présente un état de l'art. Dans un premier temps, une brève histoire des services Over-The-Top (OTT) sera présentée, en analysant sa définition, l'innovation qu'ils ont apporté à l'écosystème d'Internet et les principaux conflits entre les fournisseurs OTT et les sociétés télécom. Puis, seront présentés les concepts QoS et QoE, dont la base de ce dernier est le résultat de l'évolution de la QoS via l'intégration de la dimension perceptuelle de la qualité. Ce dernier concept propose une approche holistique pour l'évaluation de la qualité du point de vue de l'utilisateur, qui vise à inclure l'intégration de paramètres d'affaires dans l'évaluation. Pour cet effet, la dernière section présente le concept de la Quality of Business (QoBiz), montrant les efforts effectués pour intégrer le domaine d'affaires sur l'analyse de la QoE.

L'intégration de paramètres d'affaires sur l'analyse de la QoE exige d'introduire une nouvelle approche pour l'analyser. A ce sujet, le Chapitre 3 présente le système compréhensif d'évaluation de la QoE pour les services multimédia implémentés sur Internet. Ce système utilise le modèle mathématique des « Extended Finite State Machine (EFSM) » pour représenter l'effet des décisions de l'utilisateur sur la QoE. Ensuite, ce chapitre montre comment utiliser le modèle d'un service OTT et un modèle de la QoE appropriée pour faire le calcul de la valeur de la QoE.

Afin de fournir au marché du monitoring un ensemble d'outils prêts à utiliser, le Chapitre 4 présente la mise en œuvre du système théorique présenté antérieurement. Cette implémentation profite des capacités de monitoring et d'émulation EFSM de l'outil MMT, sous forme de deux algorithmes qui ont été développés comme extension du module MMT Correlator. Le premier algorithme effectue l'analyse du modèle EFSM en calculant les traces contenues dans les machines. Le deuxième algorithme complète le premier en faisant le calcul des valeurs respectives de la QoE pour chaque trace calculée préalablement. Cette mise en œuvre peut bénéficier des capacités DPI/DPI de MMT pour obtenir les valeurs des paramètres réseau et d'affaires qui sont alimentés dans les deux algorithmes. En poursuivant cet objectif, ce chapitre montre le design de l'outil de monitoring qui profite de la technologie développée pour faire le calcul de la valeur de la QoE des scénarios futurs d'un utilisateur particulier.

Le prototype mis en œuvre sur l'outil MMT doit être validé avant d'être utilisé sur des services réels. Le Chapitre 5 présente cette étape de validation, qui comprend aussi l'ajustement d'un modèle de la QoE approprié à ces effets. Cet étape est accomplie en utilisant une plateforme d'émulation d'un service OTT qui poursuit un double objectif. D'une part, la plateforme a été conçue pour déterminer l'impact des paramètres d'affaires sur la QoE sous forme d'un questionnaire soumis à l'utilisateur. D'autre part, la plateforme a été utilisée pour collecter des valeurs de la QoE de vidéos qui émulent des détériorations du réseau. Cette information a été utilisée pour élaborer un modèle de la QoE qui montre la relation entre les paramètres qui ont été considérés dans l'analyse.

Le chapitre 6 présente une extension des technologies déjà présentées sous forme d'analyse statique à un stade précoce du développement. En considérant que le modèle EFSM a été conçu depuis le point de vue de l'utilisateur, il a la capacité de retenir tous les scénarios que l'utilisateur peut expérimenter. De ce fait, il est possible d'utiliser le système pour étudier la distribution des valeurs de la QoE. Cette approche utilise un troisième algorithme pour calculer tous les possibles scénarios de l'utilisateur et la valeur de la QoE.

Finalement, le chapitre 7 présente les conclusions générales et les perspectives des technologies proposées et mises en œuvre dans le cadre de ce travail.

## E.2 Chapitre 2 : État de l'art

### E.2.1 Les services Over-The-Top (OTT)

#### E.2.1.1 Le commencement : les télécommunications et les médias audiovisuels

Pendant les années 80 et le début des années 90, les besoins de communication des utilisateurs ont été satisfaits par les services télécoms ; la transmission des données par la voix se faisait en utilisant le téléphone, pendant que les transmissions de documents se faisait par la poste ou via le fax [22]. La technologie que supportait ces deux technologies était chère et peu abondante, et ainsi, contrôlée principalement par un nombre limité de sociétés de chaque pays [37]. De plus, ces réseaux ont été développés en utilisant des hypothèses qui limitaient l'extensibilité du réseau.

En même temps, la distribution de contenu multimédia a été faite soit par des moyens physiques ou soit via des réseaux qui utilisaient des hypothèses similaires à celles des réseaux téléphoniques. Malgré une interconnexion quasi nulle entre les réseaux, les deux services sont toujours offerts par les mêmes sociétés télécom, qui ont fait l'effort de déployer l'infrastructure nécessaire pour supporter les deux chaînes de distribution.

Ces caractéristiques ont marqué l'existence de deux marchés d'investissement pour les parties

---

intéressées : la téléphonie et la distribution de contenu multimédia. L'équilibre de ces deux marchés a été affaibli par l'avènement d'Internet qui proposait un réseau plus flexible pour distribuer une variété de types de contenu en même temps [37].

### E.2.1.2 L'ère d'Internet et l'ascension des services basés sur Internet

La propulsion d'Internet comme un nouveau réseau de distribution plus flexible a fait revoir les paradigmes d'implémentation des réseaux déjà existants. Dans le cas du service téléphonique, le réseau est en charge de faire la plus grande partie du travail – en réservant les ressources nécessaires pour établir la communication [37] – pendant que les terminaux effectuent uniquement la codification des données transportées [48], où la voix est codifiée en utilisant des codes spécifiques. Toutes ces hypothèses rendent le déploiement du réseau téléphonique très coûteux et exigeant pour intégrer de nouveaux services [48].

Par contre, Internet tend à inverser ces hypothèses en proposant un réseau où les données sont le principal concept. Cela permet de relaxer les hypothèses sur les types de données transportées, ce qui présente beaucoup d'avantages : (1) en déplaçant la logique du réseau aux terminaux, il est possible de laisser au réseau ce qu'il doit faire : la transmission des données [48] ; (2) les données savent où elles doivent être livrées, ce qui permet de relaxer la dépendance entre les types de données et le réseau [48] ; et (3) en utilisant des signaux digitaux pour la transmission des données, il est possible de transporter plusieurs services sur le même réseau [37]. Toutes ces caractéristiques ont permis de définir cette nouvelle technologie comme un *réseau stupide* [37].

Sa première implémentation a été réalisée par l'organisme américain ARPA pour le *projet ARPANET*. Cette mise en œuvre – utilisée au début pour des objectifs militaires et plus tard pour la recherche – permettait la communication entre hôtes qui utilisaient la même technologie sous-jacente, cependant son développement était focalisé sur l'interconnexion de réseaux, concept nommé *internet* [22], [48], [61]. Pour cet effet, le protocole TCP a été proposé pour permettre l'abstraction de différents réseaux physiques qui utilisaient la même logique sous-jacente : les paquets.

En visant cette flexibilité du nouveau réseau, les entreprises commençaient à déplacer leurs technologies vers ce nouveau réseau, grâce aux faibles coûts de transmission de données [37]. De même, la flexibilité proposée par Internet permet d'ajouter le service téléphonique à ce réseau, service qui voulait être bloqué par les sociétés télécoms [37]. Cette dispute représente un précédent dans la guerre actuelle sur la distribution de contenu.

### E.2.1.3 L'« invasion » du domaine de distribution de contenu

Les compagnies télécoms voyaient en les services sur Internet une potentielle menace pour leurs chiffres d'affaires. De ce fait, ils essayaient de prendre le contrôle du marché d'Internet en participant à cette nouvelle opportunité d'affaires. Au début, le service d'Internet était offert sous forme de connexions de bas débit en utilisant le réseau téléphonique [22], [48]. Ensuite, ils ont augmenté leur offre avec des services télévisuel intégrés aux services téléphoniques et d'Internet. Cette stratégie a permis l'amélioration des capacités de l'Internet publique et de son adoption à la maison [45].

Ces améliorations de la disponibilité et des capacités d'Internet ont rendu la transmission des données plus complexes [55]. Des nouvelles entreprises ont donc vu une opportunité d'affaires sur ce réseau.

Au mois de mai 2015, le site web *YouTube* a été lancé comme une plateforme de téléchargement et de partage de vidéos en utilisant Internet [21], [92]. La distribution de contenu

multimédia via Internet avait été prévue en 1997 par Isenberg dans son célèbre article « *The Rise of the Stupid Network* » [37]. En même temps, d'autres entreprises de services multimédia commençaient à émerger, c'est le cas de Netflix et Skype, sociétés qui représentent respectivement la compétence directe de services télévisuel et de téléphonie. Le développement de nouvelles technologies pour la compression et le transport de vidéos a permis le déploiement de ces types de services appelés *OTT*. Sa définition dans [29] permet de les caractériser comme « services transportés sur Internet qui offrent de la valeur ajoutée aux utilisateurs sans partager les revenus avec les compagnies télécoms traditionnelles » [29].

Ces caractéristiques laissent aux fournisseurs d'Internet – les entreprises télécoms – comme les compagnies en charge de munir un produit de remplacer le reste de leur offre. Dans cette situation, les compagnies télécoms commençaient à voir leurs revenus diminuer par les nouvelles entreprises OTT qui commençaient à occuper « le domaine de distribution de contenus ». Il est temps pour eux de réagir à cette menace et de réinventer leur offre au risque de perdre la bataille contre la nouvelle compétence.

#### E.2.1.4 La guerre de la « compétence injuste »

Dans cette nouvelle situation, les entreprises télécoms ont réagi de la même manière par rapport à la menace initiale du service ToIP : ils ont essayé de bloquer l'utilisation des services OTT [37]. Leur demande est basée sur une « compétence injuste » de la part des fournisseurs multimédia, qui utilisaient ces réseaux pour livrer leurs services sans payer le transport de données [80]. Les télécoms ont tenté d'expliquer cette réaction à cause de la grande quantité de données requis par les services OTT, ce qui augmente les coûts opérationnels du réseau [49].

La dispute mentionnée a forcé les entreprises télécoms à prendre des mesures agressives contre les OTT : (1) baisser le débit des flux des services OTT, (2) exiger un paiement aux fournisseurs OTT pour utiliser les réseaux, ou (3) bloquer complètement les flux liés aux services OTT. L'exemple le plus connu de ce type de mesures confrontait Comcast – fournisseur d'Internet américain – et Netflix, ce dernier a réglé un paiement à Comcast pour éviter la réduction du débit des flux multimédia.

Ces types de mesures ont ouvert la discussion sur les différents rôles de chaque entreprise dans l'écosystème d'Internet. Les fournisseurs d'Internet doivent-ils se munir d'un accès au réseau quelles que soient les données de l'utilisateur ? Doivent-ils se munir d'un accès « différencié » pour chaque type de contenu ? Sur ces questions, le concept de la « neutralité du réseau » peut guérir les blessures de guerre.

#### E.2.1.5 Le traité de la « neutralité du réseau »

Certaines compagnies télécoms ont déjà commencé à bloquer ou limiter l'accès aux services OTT. Dans l'Union Européenne, l'opérateur hollandais KPN avait bloqué l'accès aux services VoIP et de messagerie texte en 2011. Ce fait a initié le débat sur la neutralité du réseau, étant les Pays Bas le premier pays en implémenter une loi de neutralité de réseau [35], [82].

En termes générales, un réseau neutre doit permettre d'accéder à Internet sans restriction, sans différenciation de contenu et sans perturber la qualité de l'accès arbitrairement [91]. Ces définitions permettent la distribution de contenu par Internet, mais ils permettent aussi l'accès au contenu illégal. De ce fait, quelques pays ont initié une discussion sur la neutralité du réseau de manière à trouver des implémentations qui s'adaptent aux besoins locaux, par exemple, aux États Unis [30], [88] et aux quelques pays de l'Union Européenne [30], [90].

---

Malgré les efforts de limiter le pouvoir des entreprises télécoms, la neutralité du réseau reste une discussion ouverte. Certains pays ont commencé son implémentation, il y a bien d'autres où le manque d'une loi effective permet aux ISPs de dégrader l'accès aux services complémentaires ce qui peut faire baisser leurs revenus.

## E.2.2 Les concepts de qualité

Tous les services qui utilisent Internet comme moyen de distribution de données, doivent considérer que cette technologie a été conçue comme un réseau de « meilleur effort ». De ce fait, l'assurance de la *qualité* de ces services représente toujours un défi, car le concept de *qualité* n'est pas encore clair sur un réseau comme Internet.

### E.2.2.1 Qualité de Service (QoS)

Le premier effort d'une définition de qualité peut être extrait de la définition du réseau donnée par Isenberg, qui établit les données comme le concept central du réseau [37]. De cette manière, le concept de la qualité doit mesurer la capacité du réseau de livrer les messages. Cette idée a été conservée dans la définition de la *Quality of Service (QoS)*, qui est définie par différentes organisations (l'ITU [39] et l'IETF [19]) comme l'ensemble des métriques qui permettent d'établir des « niveaux de service ». La principale différence entre les deux définitions a été montrée par Gozdecki et al. en [28], qui expose la différence entre la *qualité évaluée* (le niveau minimal de qualité acceptée par l'utilisateur, influencé par facteurs économiques et d'autres), la *qualité inhérente* (celle qui montre la capacité du réseau de transporter les données) et la *qualité perçue* (le niveau de qualité perçue subjectivement par l'utilisateur et qui intègre ses attentes) [28].

Malgré l'effort de clarifier les concepts de qualité, les chercheurs ont utilisé le concept de la QoS comme synonyme de la performance du réseau et, de cette manière, du concept de la qualité inhérente [84].

Finalement, l'inclusion des éléments subjectifs et économiques par moyen des concepts de qualité perçue et évaluée respectivement a donné l'opportunité d'analyser le concept de la qualité d'un point de vue plus profonde et holistique : la *Quality of Experience (QoE)*.

### E.2.2.2 Qualité d'expérience (QoE)

Le concept de la QoE a été conçu à partir de la qualité perçue qui « refléchit l'expérience de l'utilisateur. Elle est influencée par les attentes de l'utilisateur qui sont comparées à la performance observée du service » [28]. Cette définition présente la base sur laquelle Van Moorsel a défini *Quality of Experience*) comme « des métriques liées à l'expérience de l'utilisateur » en remarquant qu'elle « peut être influencée par des éléments subjectifs » [83]. Cette caractérisation établie clairement la nature subjective de la QoE, cependant elle ne peut être pas utilisée comme une définition formelle.

En 2011, la communauté Qualinet a donné une définition plus formelle du concept : « (La QoE est le) degré de plaisir de l'utilisateur d'un service. Dans le contexte d'un service de communication, elle est influencée par le contexte, le réseau, le dispositif, l'application, les attentes et buts de l'utilisateur, et le contexte d'utilisation » [47]. De plus, cet approche expose la nature multidisciplinaire de la QoE, qui peut être influencé par différents *Facteurs d'Impacts (IF)* : [47], [70].

- *IF humaines* : L'ensemble des caractéristiques liées à l'être humain. Caractéristiques comme le contexte démographique ou économique, la composition psychique ou l'état émotionnel sont des exemples de ces indicateurs [70].
- *IF de contexte* : Les facteurs qui décrivent l'environnement de l'utilisateur [47]. Dans cette catégorie, il est possible d'identifier les contextes *physique*, *temporal*, *social*, et *économique* [47].
- *IF de système* : Finalement, les propriétés et caractéristiques qui déterminent la qualité technique du service [47]. Ces indicateurs sont usuellement liés au concept de la QoS déjà présenté et peuvent être classifiés selon leur liaison au *contenu*, *média*, *réseau* et *dispositif* [70].

Malgré la nature subjective de la QoE, l'ITU a défini une échelle pour mesurer les niveaux de qualité subjective. Cet instrument est appelé *échelle Mean Opinion Score (MOS)*, et elle définit 5 niveaux de perceptions présentées dans la Table 2.1. Cette échelle a été adoptée pour évaluer la qualité des différents services multimédia, ce que l'a transformée en la métrique par défaut pour estimer la qualité expérimentée.

Finalement, la littérature propose de différents modèles mathématiques qui mettent en relation les différents IF avec sa correspondante valeur de la QoE. Ces modèles sont classifiés, principalement en deux gros groupes :

- *Relations exponentielles* : Ces relations sont basées sur le fait que des petits changes sur les IF impactent fortement sur la valeur de la QoE quand sa valeur est haut. Au contraire, quand la valeur de la QoE est déjà bas, grands changements de ces IF ne provoquent que des petits changements sur la qualité [20].
- *Relations logarithmiques* : Ces relations sont basées sur la *loi Web-Fechner* qui caractérise des « petits changes notables » sur la perception sont liées à une portion fixe du stimulus appliqué [66].

Ces modèles sont principalement utilisés pour l'élaboration des outils de monitoring qui permettent de prédire la valeur de la QoE.

### E.2.2.3 Le concept de la Qualité d'affaires (QoBiz)

Il a été mentionné que la QoE peut être influencée par différents facteurs. Van Moosel a noté ce fait, en analysant l'importance des décisions d'affaires sur la qualité des services basés en Internet [83]. Dans ce travail, Van Moosel propose une approche intégrale pour comprendre la qualité des services d'Internet qui intègre trois concepts : la QoS (dimension technique), la QoE (dimension subjective) et la *Qualité d'affaires (QoBiz)*. Ce dernier concept est défini comme « métriques mesurées en termes d'argent », étant directement liée à la dimension d'affaires du service. L'analyse de ces trois dimensions vise à maintenir le fonctionnement de l'écosystème de services complet, de manière de garantir, au même temps, une correcte administration du réseau, des revenus pour les investisseurs et un haut niveau de qualité pour l'utilisateur [56], [69].

La définition munie par Van Moosel peut être interprétée depuis deux points de vue. Du côté du fournisseur du service, garantir un haut niveau de qualité aux utilisateurs impacte directement les revenus de la compagnie [49], [74], [80], [83]. Du côté de l'utilisateur, le prix du service a été

---



identifié comme l'un des facteurs les plus important qui impactent les attentes et, de ce fait, la QoE [67], [69].

La littérature a analysé théoriquement la relation entre les schémas de prix et la qualité de service, de façon à munir les fournisseurs de services des recommandations pour les prix d'un service multimédia [73], [74]. De plus, Wahlmüller et al. ont simulé une approche basée sur le *Paris Metro Pricing (PMP)* pour surveiller plutôt la QoE que la QoS [85]. Finalement, cette étude a été enrichi par Ahmad et al. en proposant une approche qui encourage la collaboration entre les ISP et les OTT [11].

Malgré les efforts d'analyse de la QoE d'un point de vue holistique, son analyse est toujours un problème ouvert à cause des nombreux facteurs qui impactent sa valeur. Ce travail vise à fournir le marché avec un ensemble de systèmes d'évaluations et d'outils pour l'intégration des paramètres d'affaires sur l'évaluation de la QoE.

### E.3 Chapitre 3 : Système d'évaluation de la QoE basée sur les machines à états finis étendues

L'inclusion des paramètres d'affaires dans l'analyse de la QoE nécessite la proposition d'un nouveau système d'évaluation capable d'être :

- *Extensible* : De manière à intégrer les paramètres d'affaires sans interférer avec les autres aspects de la QoE.
- *Traçable* : Les décisions d'affaires pris par l'utilisateur impactent principalement ses attentes. De ce fait, le système doit être capable de maintenir un registre des décisions économiques historiques de l'utilisateur afin de les considérer au moment de l'évaluation de la QoE.

Le formalisme mathématique de base choisi pour ce but sont les machines aux états finis étendues EFSM. La prochaine section montre ses caractéristiques principales ainsi que son fonctionnement.

#### E.3.1 Préambule – Machines aux états finis étendues

Le modèle mathématique de base s'appelle *Machine à états finis (FSM)*. Ce modèle a été conçu comme un groupe d'*états* (étant l'un parmi eux l'*état initial* et une autre l'*état actuel*), un groupe de *transitions* (qui change l'état actuel de la machine, et sont activés suivant un événement produit). Formellement, une FSM est défini par la tuple  $M = (I, \Sigma, O, \delta, s_0)$  [26] où  $I$  est l'ensemble de données acceptées par la machine,  $\Sigma$  l'ensemble d'états,  $O$  l'ensemble des sorties de la machine,  $\delta : \Sigma \times I \rightarrow \Sigma \times O$  la fonction de transitions et  $s_0$  l'état initial. Un exemple d'une FSM est montré dans la Figure 3.1.

Ce formalisme peut être enrichi de manière à inclure plus d'éléments à modéliser. Les *Machines à états finis étendues (EFSM)* sont le modèle augmenté avec des *paramètres d'entrée et de sortie*, des *variables de contexte*, et des *opérations* et des *gardes* définis sur les variables de contexte. Afin de donner une définition formelle, considérons les ensembles associés à une FSM et laissons :  $R$  et  $V$  être deux ensembles disjoints de paramètres de noms de variables ;  $R_i \subseteq R$  et  $D_{R_i}$  les ensembles de paramètres d'entrée et leurs évaluations respectives pour chaque  $i \in I$  ;

$R_o$  et  $D_{R_o}$  les ensembles de paramètres de sortie et leurs évaluations respectives pour chaque  $o \in O$ ; et  $D_V$  l'ensemble de évaluations des variables de contexte  $\mathbf{v}$ . Considérant ces définitions une EFSM est la tuple  $M = (\Sigma, T)$  [57] où  $\Sigma$  est l'ensemble d'états et  $T = \{(\sigma, i, P, op, o, up, \sigma')\}$  est l'ensemble de transitions de la machine où  $\sigma, \sigma' \in \Sigma$  sont les états initial et final,  $i \in I$  est le donnée d'entrée de la transition,  $o \in O$  le donnée de sortie,  $P : D_{R_i} \times D_V \rightarrow \{True, False\}$  est la fonction gardien de la transition,  $op : D_{R_i} \times D_V \rightarrow D_{R_o}$  est la fonction des paramètre de sortie, et  $up : D_{R_i} \times D_V \rightarrow D_V$  est la fonction de mise à jour de variables de contexte. Un exemple de ces machines étendues  $M_2$  est montrée dans la Figure 3.2.

### E.3.1.1 $L$ -équivalent d'une EFSM

Les EFSM ont la capacité de garder la trace des états visités par l'utilisateur, mais en utilisant ces modèles il n'est pas clair quels sont tous les possible chemins qu'un utilisateur peut suivre à partir de l'état initial. Afin d'atteindre ce but, il faut transformer le modèle en le modèle  $l$ -équivalent, version qui peut être construite avec le processus suivant :

1. Introduire des restrictions sur les boucles, de manière de limiter le nombre d'exécutions de ces transitions.
2. A partir de l'état initial, dessiner les prochains états dans un nouveau niveau, clonant les états si besoin.
3. Répéter le pas antérieur en utilisant les nouveaux états dessinés récemment comme le début jusqu'à ce que la longueur souhaitée soit atteinte.

A la fin du processus, le  $l$ -équivalent aura la forme d'un arbre de hauteur  $l$ . Dans la Figure 3.3 il est montré le 1-, 2 et 3-équivalents de la machine  $M_2$ , ainsi que les pas itératifs pour le calcul de ces modèles.

### E.3.2 Système d'évaluation de la QoE

L'utilisation des EFSM permet de formuler une nouvelle approche pour évaluer la valeur de la QoE conformé par 3 étapes :

1. **Modélisation** : Cet étape permet d'obtenir un modèle EFSM formel du service OTT qui représente les requis fonctionnels du service ;
2. **Augmentation** : Le modèle obtenue doit être augmenté avec l'insertion d'indicateurs de qualité, ceux qui introduisent des requis non-fonctionnels dans le modèle ;
3. **Calcul** : Finalement, le modèle est utilisé pour calculer la valeur de la QoE par l'application d'un modèle de qualité approprié.

Les sections suivantes exposent les détails de chaque étape et montrent son application à un service OTT réel.

---

### E.3.2.1 Modélisation d'un service OTT

De manière à intégrer le point de vue de l'utilisateur, l'analyse est faite de manière à modéliser le comportement du service comme vu par l'utilisateur. L'analyse produite ne considérera ainsi que les caractéristiques du service qui impactent directement sur les attentes de l'utilisateur et donc, sur la QoE du service.

Pour y arriver, une étape intermédiaire est requise : la création d'une description du fonctionnement du service qui est construit pour analyser l'interaction entre l'utilisateur et le service. Cette description permet d'identifier les éléments principaux du modèle EFSM à dessiner et, en particulier, les états, les entrées et sorties, et les transitions de la machine, ainsi que quelques gardes associés à ces transitions.

Le processus mentionné peut être formalisé en grandes lignes dans le Plan de Méthodologie 1. Cet approche a été appliqué sur un service OTT réel : beIN Sports Connect. La description intermédiaire écrite peut être trouvée dans l'Annexe A, celle qui a été utilisée comme la base pour élaborer le modèle présenté dans la Figure 3.5.

### E.3.2.2 Enrichissement du modèle avec les indicateurs de qualité

La EFSM obtenue doit être augmenté avec l'introduction d'« indicateurs de qualité ». Ces indicateurs seront introduits sous la forme de *variables de contexte* de la machine, de manière à les utiliser en combinaison avec la traçabilité des EFSM afin de retenir l'effet des décisions de l'utilisateur dans l'évaluation de la QoE.

Cependant, il est nécessaire de limiter le type de variables qui pourront être considérées. Dans le cas du système d'évaluation présenté ultérieurement, les indicateurs de qualité doivent les caractéristiques suivantes :

- *Mesurable* : Chaque paramètre introduit doit être mesurable de manière de représenter des valeurs réelles. Le système proposé est capable de supporter variables discrètes ainsi que continues.
- *Révisable* : En même temps, leurs valeurs doivent être mises en jour au moment d'exécuter les transitions qui représentent des actions de l'utilisateur remarquables. Ces mises à jour peuvent être exprimées en forme de fonctions mathématiques.

L'insertion de ces variables – et ses fonctions respectives de mis à jour – produira une version augmentée du modèle qui peut être utilisée pour évaluer la QoE. Il est alors possible d'étendre la méthodologie avec les étapes d'augmentation du modèle, ce qui peut être trouvé dans le Plan de Méthodologie 2.

Cette approche a été appliquée sur le cas d'étude de beIN Sports Connect, en introduisant 3 types de paramètres :

- *Objectif* : Les paramètres objectifs sont les facteurs les plus importants qui impactent la QoE. De ce fait, nous avons introduit le paramètre '*stream\_flag*', une variable binaire qui représente si les conditions minimales du réseau sont satisfaites ou non.
- *Subjectif* : L'inclusion des paramètres subjectifs est faite par l'introduction de la variable '*service\_confidence*'. Ce variable est mesuré en utilisant une échelle de 3 niveaux qui représentent les différents niveaux de confiance de l'utilisateur.

- *Paramètre d'affaires* : L'innovation de l'approche est l'inclusion des paramètres d'affaires dans l'analyse de la QoE. Pour atteindre ce but, le paramètre '*price\_premium*' a été introduit comme une manière de mesurer « la quantité qu'un utilisateur est prêt à payer pour une marque par rapport aux autres qui offrent un bénéfice similaire » [8]. Cette variable est mesurée en utilisant une échelle de trois niveaux qui représentent la fidélité de l'utilisateur à la marque.

Les trois paramètres – et leurs fonctions respectives de mise à jour – ont été introduits dans la EFSM de beIN Sports Connect, produisant le modèle montré dans la Figure 3.6.

### E.3.2.3 Computation du modèle *l*-équivalent et de la QoE

Le modèle élaboré doit être transformé avant de faire le calcul de la QoE. La machine doit de ce fait être transformée dans son modèle *l*-équivalent, de manière à montrer plus clairement l'information contenue dans le modèle : les *traces* de longueur fixe et les *configurations finales* de chaque trace. Cette transformation est faite par application du processus expliqué dans la Section E.3.1.1.

Le processus a été appliqué sur la machine de beIN Sports Connect pour calculer le 11-équivalent. Dans la Figure 3.7 une seule trace de cette machine est montré, celle qui sera utilisée ultérieurement pour l'analyser.

Une fois que toutes les traces ont été calculées, il est nécessaire d'appliquer un relation mathématique approprié pour calculer la valeur de la QoE. De ce fait, il est possible de compléter les étapes du système d'évaluation proposé, montrant la méthodologie complète dans le Plan de Méthodologie 3.

Dans le cas d'étude, nous avons appliqué une combinaison linéaire des indicateurs de qualité – montré dans l'Équation 3.5 – à chaque état terminal du *l*-équivalent, de manière à montrer l'utilisation de l'approche. Dans la Table 3.3 les valeurs de la QoE sont montrées pour chaque état nommé dans la Figure 3.7. Il est possible d'observer comment la valeur de la QoE évolue en différents états, en particulier, comment la valeur de la QoE baisse quand un problème survient sur le flux (états *B* à *C*). Ensuite, la QoE remonte sa valeur une fois que l'utilisateur a essayé de redémarrer le service (états *C* à *D*).

### E.3.3 Conclusions

Le principal avantage de l'approche proposé est l'inclusion d'effets temporeux des décisions de l'utilisateur sur la QoE. De plus, la traçabilité des EFSM permet d'identifier les traces avec une valeur de la QoE bas, et de revenir sur les pas de manière à trouver la transition à l'origine d'une valeur basse de la qualité. Cet information est importante au moment d'optimiser les inversions et les revenus du service OTT.

## E.4 Chapitre 4 : Implémentation du système d'évaluation

Le système d'évaluation présenté dans la Section E.3 est prêt à être mis en œuvre, cependant son implémentation nécessite un outil de base afin de faciliter son déploiement rapide sur le marché.

### E.4.1 L'outil de monitoring de Montimage (MMT)

L'outil de monitoring de Montimage MMT est un outil qui permet la vérification et la validation de protocoles Internet ainsi que des flux [87]. Son architecture est montrée dans la Figure 4.1, composée principalement de 4 modules :

- *MMT Bus* : Canal de communication commun des trois modules de MMT. Les événements et messages émis par les autres modules sont transportés en utilisant ce canal.
- *MMT Probe* : Module d'analyse de réseau. Ce module utilise les techniques d'*analyse paquets/flux en profondeur (DPI/DFI)* afin d'extraire les données nécessaires pour de la vérification de propriétés sur les protocoles ou flux analysé.
- *MMT Correlator* : Module d'analyse de propriétés. Ce module utilise les données extraites pour le module Probe afin de vérifier des propriétés exprimées sous forme de EFSM. Ces machines émettent des verdicts qui déterminent la conformité du protocole de la propriété testée.
- *MMT Reporting* : Module de rapport des verdicts à l'utilisateur. Ce module utilise une interface web pour montrer les résultats et verdicts des propriétés testés.

La mise en œuvre du système d'évaluation de la QoE sera faite sous forme d'extension du module Correlator de MMT. La capacité de DPI/DFI de MMT permet de compléter l'approche proposée afin d'extraire les valeurs des indicateurs de qualité et de les fournir directement au modèle OTT dans le module Correlator. De plus, ce module supporte déjà des propriétés exprimées sous forme de EFSM, en utilisant la notation Javascript. Un exemple de la spécification d'une EFSM pour MMT Correlation est montré dans l'Annexe B.1 pour la machine de la Figure 4.3. Finalement, le module Reporting peut être utilisé pour montrer les résultats de l'évaluation à l'utilisateur ou montrer l'information additionnelle en relation avec l'analyse du modèle.

### E.4.2 Extension de l'outil OTT avec le système d'évaluation de la QoE

En analysant les étapes du Plan de Méthodologie 3, nous pouvons noter que les 5 premières étapes impliquent la modélisation et l'augmentation du modèle EFSM, ce qui ne peut pas être automatique. Cependant, les deux derniers – le calcul du *l*-équivalent et le calcul de la QoE – peuvent être accomplis automatiquement. De ce fait, l'implémentation du système d'évaluation sera basée sur la mise en œuvre de deux algorithmes sous forme d'extension à l'outil de Montimage, dont l'architecture est montrée dans la Figure 4.4. Ces algorithmes feront les calculs ci-mentionnés en utilisant la machine du service OTT comme principale entrée de l'approche.

#### E.4.2.1 Algorithme de calcul du *l*-équivalent

Le premier algorithme fait le calcul du *l*-équivalent du modèle du service. Son pseudo-code est montré dans l'Algorithme 1 et il est basé sur une approche de parcours en profondeur DFS [17].

Le fonctionnement de l'algorithme est caractérisé par deux comportements principaux : (1) le cas de base est attendu lorsque la longueur requise est égal à 0, l'algorithme renvoie une tuple contenant l'état de la machine et les valeurs des variables de contexte; et (2) le cas général représente l'invocation de l'algorithme avec une longueur différente de 0, donc le processus doit calculer les transitions qui seront exécutés en calculant la valeur des gardiens de chaque transition, de mettre à jour les valeurs des variables du contexte – en utilisant les fonctions

respectives, et invoquer récursivement la fonction en utilisant le prochain état comme état initial, les nouvelles valeurs des variables de contexte et une longueur 1 pas moins long. L'algorithme augmente les traces partielles des invocations récursives avec l'état actuel et les valeurs des variables avant de renvoyer ce résultat à l'invocation précédente.

L'exécution de cet algorithme calculera l'ensemble de traces de longueur désiré ainsi que les valeurs des variables de contexte pour chaque état de chaque trace.

#### E.4.2.2 Algorithme de calcul de la QoE

Les traces renvoyées par l'algorithme antérieurement seront augmentées par le deuxième algorithme montré dans l'Algorithme 2.

Cette implémentation itère sur chaque trace en appliquant le modèle de la QoE. Pour chaque trace, l'algorithme itère sur chaque état calculant la valeur de la QoE par l'application du modèle de qualité et les valeurs des variables de contexte à chaque état.

La sortie principale de cet algorithme est la même information que l'algorithme précédent, sauf qu'il est maintenant augmenté avec les valeurs respectives de la QoE.

#### E.4.3 Application des algorithmes

Les algorithmes présentés dans la Section E.4.2 ont été dessinés afin d'être complétés avec les capacités de monitoring de MMT. La Figure 4.5 montre l'architecture de l'outil qui prédit la QoE en utilisant le système d'évaluation proposé.

La Figure montre les principales entrées et sorties de l'outil, ainsi que le fonctionnement de l'ensemble des éléments. Le module MMT Probe examine le flux du service OTT de manière à extraire les valeurs des variables de contexte et l'état actuel de l'utilisateur. Cette information est utilisée par le premier algorithme pour calculer les futurs scénarios de l'utilisateur sous forme de traces du  $l$ -équivalent. Ces traces sont fournies au deuxième algorithme – ainsi que le modèle de qualité – pour calculer les valeurs de la QoE de chaque scénario calculé. Finalement, cette information est présentée au fournisseur du service afin d'identifier les scénarios de basse qualité. Ces recommandations peuvent aider le fournisseur à optimiser ses efforts – et investissements – et à améliorer la qualité générale du service OTT.

### E.5 Chapitre 5 : Validation du système d'évaluation de la QoE

La technologie proposée est prête à être utilisée avec des services OTT réels, cependant deux éléments principaux de l'approche doivent être validés avant de les déployer.

D'une part, il est nécessaire de déterminer l'effet réel d'inclusion des paramètres d'affaires, en particulier, l'inclusion du paramètre « price\_premium ».

D'autre part, le deuxième algorithme fait le calcul de la valeur de la qualité en utilisant un modèle de la QoE approprié. De ce fait, il faut trouver ce modèle qui met en relation les valeurs des indicateurs de qualité avec la valeur finale de la QoE.

#### E.5.1 Mise en œuvre d'une plateforme d'émulation OTT

##### E.5.1.1 Plateforme d'émulation OTT

La nature privée et commerciale du cas d'étude présentée – beIN Sports Connect – ne permet pas d'analyser l'approche en utilisant le même service modélisé. De ce fait, un simulateur du

service a été construit de manière à simuler les principales fonctionnalités du service réel, en utilisant comme base le modèle de la Figure 3.6.

La Figure 5.1 montre l'architecture du simulateur mis en œuvre, ce qui conserve les pages principales du service original : page d'accueil, sélection de l'abonnement, enregistrement et paiement, et page d'affichage des vidéos. De manière à adapter le service aux requis de la validation, des modifications ont été introduits par rapport au modèle original de la Figure 3.6 : les boucles de paiement et de sélection de l'abonnement n'ont pas été implémentées, l'état « *stream\_delivery* » a été remplacé avec trois autres de manière à offrir trois types de contenu, et l'état « *stop\_stream* » a été changé par un état d'évaluation de la QoE de manière à obtenir les valeurs de la qualité expérimenté par l'utilisateur.

Cette plateforme a été implémenté en utilisant la technologie de blogging *Wordpress*, ce qui permet d'utiliser HTML 5 afin d'insérer des vidéos dans les pages web.

### E.5.1.2 Outil de génération des vidéos perturbées

De manière à simuler des scénarios de perturbation réseau, il faut considérer la nature du service qui est en train d'être analysé. En générale, les paramètres objectifs – liées aux perturbations du réseau – peuvent être classifiés dans deux groupes. D'une part, les paramètres de « bas niveau » peuvent être mesurés en observant le flux multimédia dans les routeurs. C'est le cas du *taux de perte de paquets*, le *jitter* ou le *délai* du réseau. D'autre part, les paramètres de « haut niveau » sont plutôt liées aux métadonnées du flux multimédia, ceux qui impliquent l'analyse en profondeur du flux. De ce fait, ces paramètres sont liés au flux de vidéo, par exemple le *taux de compression (bitrate)* de la vidéo, le *temps de gel* de la vidéo ou la *fréquence des événements de gel*.

En considérant la technologie HTML 5 de la plateforme OTT mise en œuvre, les paramètres de « haut niveau » seront testés sur les vidéos. En effet, les paramètres de bas niveau peuvent impacter de manières peu prédictibles sur les flux TCP, ce qui peut rendre encore plus complexe l'analyse de la qualité. En particulier, trois paramètres seront analysés : le *taux de compression (bitrate)* de la vidéo, le *temps de gel* de la vidéo ou la *fréquence des événements de gel*.

Afin de simuler des détériorations communes du réseau, 5 scénarios ont été définis : *sans perturbations* ; *taux de compression variable* ; *événement de gel initial* ; *événements de gel et taux de compression fixe* ; et *événements de gel et taux de compression variable*.

Considérant ces types de détériorations, un total de 44 scénarios ont été créés, tous montrés dans la Table 5.1.

Finalement, de manière à générer les vidéos détériorés pour les insérer dans la plateforme de simulation, un outil automatique a été développé, dont l'architecture est montrée dans la Figure 5.2. Cet outil profite de la librairie *ffmpeg* pour segmenter les vidéos, simuler des perturbations dans chaque segment et, finalement, reconstruire la vidéo détériorée en utilisant les segments perturbés.

## E.5.2 Raffinements préliminaires

### E.5.2.1 Amélioration du modèle OTT

Le but principal de cette amélioration est de déterminer si le modèle du service représente le service sur analyse, et si les variables de qualité insérées impactent sur valeur de la QoE. Ces dernières variables ont été introduites sur le modèle peuvent être classifiées dans trois groupes :

*Variable Subjectif* (Le « service\_confidence »), *Variable d'affaires* (Le « price\_premium »), et *Variable objectif* (Le « stream\_flag »).

Ce dernier indicateur sera remplacé par trois autres de manière à mettre le modèle en concurrence avec les scénarios de détérioration du réseau simulés sur les vidéos. De ce fait, la variable « stream\_flag » sera substitué par :

- *Le taux de compression (variable « bitrate »)* : Mesuré en Megabits (Mbps) ou Kilobits (Kbps) par seconde. Cette variable représente la « qualité » de la vidéo comme vue depuis le point de vue de l'utilisateur.
- *La longueur des événements de gel (variable « buftime »)* : Mesuré en secondes. Cette variable note la longueur totale des événements de gel expérimentés par l'utilisateur.
- *Le nombre total des événements de gel (variable « bufevnt »)* : Cette variable représente la quantité d'événements de gel vus par l'utilisateur.

Les variables ici présentées ont été introduites sur le modèle OTT, produisant un modèle amélioré qui est montré sur la Figure 5.3.

Finalement, la validation du modèle et des variables introduites – y compris celles qui ont été présentées dans ce Section – a été faite en utilisant une enquête de 15 questions. Ce questionnaire a été introduit tout au début de l'expérimentation pour qu'il soit rempli par l'utilisateur avant d'évaluer la plateforme.

### E.5.2.2 Amélioration du modèle de la QoE

Dans la Section E.3.2.3, il est mentionné que le modèle de la QoE appliqué doit être en ligne avec les indicateurs de qualité insérés dans le modèle. De cette manière, le modèle de la QoE doit être amélioré grâce à l'expansion de la variable « stream\_flag ».

Initialement, le modèle comprenait une combinaison linéaire des variables de qualité (voir Équation 3.4). Dans la version améliorée, cette idée de base sera maintenue, cependant chaque partie du modèle général sera extraite soit de la littérature ou soit des réponses de l'enquête inséré dans la plateforme. Cette idée donne une nouvelle forme au modèle général, exposée dans l'Équation 5.3.

Pour le cas des variables objectifs, la littérature offre des études en relation avec l'effet de chaque dimension sur la QoE, ceux qui seront utilisées sous forme de composants du modèle générale.

Dans [14], les auteurs proposent un modèle logarithmique qui met en relation l'effet des changements sur le taux de compression de la vidéo avec la valeur de la QoE. Ce modèle sera utilisé comme base pour en adapter un nouveau aux valeurs du bitrate en analyse. Le nouveau modèle adapté peut être trouvé dans l'Équation 5.1.

De façon similaire, dans [34] les auteurs proposent un modèle qui intègre simultanément les effets de la longueur et de la fréquence des événements de gel de la vidéo. Cette relation exponentielle est montrée dans l'Équation 5.2. Finalement, il est possible de reconstruire la forme générale du modèle de la QoE en fusionnant chaque partie proposée dans la structure générique (voir Équation 5.4).

Il faudra donc trouver les valeurs des coefficients  $\omega_{bitrate}$ ,  $\omega_{stalling}$ ,  $\omega_{subj}$  et  $\omega_{biz}$  en utilisant la plateforme de simulation afin d'obtenir les valeurs des variables de contexte et les valeurs de la qualité qui représentent. En utilisant ces données il sera possible de calculer les coefficients pour adapter le modèle au cas d'étude.



### E.5.2.3 Évaluation des vidéos

Afin d'utiliser la plateforme mise en œuvre pour évaluer la qualité des vidéos générées, il faut introduire des modifications sur le modèle pour utiliser les outils présentés.

Le service modélisé a été conçu pour montrer un flux multimédia en direct, ce qui ne peut être implémenté sur la plateforme qui simule le service. De ce fait, le flux en direct a été substitué par les vidéos générées, ayant 2 minutes de longueur. De plus, le modèle original de la Figure 3.6 comprend un état « stop\_stream » qui a été remplacé pour un nouvel état « qoe\_evaluation » de manière de montrer à l'utilisateur le formulaire pour collecter sa valeur de la qualité (Voir la Figure 5.5). Ce changement comprend aussi la modification des transitions autour de cet état pour montrer correctement le formulaire. Finalement, l'expansion de la variable « stream\_flag » comprenait la modification de quelques transitions autour de l'état « stream\_delivery » de manière à tracer correctement les valeurs des nouveaux paramètres insérés. Tous les changements ci-présentés peuvent être observés sur la Figure 5.3.

Les changements introduits permettaient le déploiement de la plateforme dans deux endroits – laboratoires de recherche en France et Chili, ce qui cherche à maximiser la quantité d'utilisateurs qui évaluent les vidéos.

## E.5.3 Résultats et validation du modèle

### E.5.3.1 Validation des variables de contexte

Un total de 16 utilisateurs donnaient leurs réponses à l'enquête initiale. Les résultats de ce questionnaire sont montrés dans la Table 5.2.

En analysant ces résultats il est possible d'extraire quelques conclusions :

- Les services OTT impactent de manière importante les utilisateurs ; la plupart d'entre eux déclarent les utiliser depuis longtemps et fréquemment.
- La plupart des utilisateurs préfèrent des vidéos de « haute qualité » (HD) ou meilleur.
- En générale, les utilisateurs peuvent tolérer des événements de gel jusqu'à 3 secondes, temps distribué en 2 événements maximum.
- La plupart des utilisateurs se considèrent comme des clients fidèles, et la plupart d'entre eux aiment que le service soit capable de surveiller automatiquement l'état du réseau, adaptant le flux si nécessaire. Cependant, les utilisateurs déclarent que cette caractéristique *augmente* leur fidélité.

En utilisant ces résultats, il est possible de raffiner le paramètre d'affaires ( $M_{biz}$ ) ainsi que ses fonctions de mis à jour ( $f_{biz}^A(sub,loy) = M_{biz}$  et  $f_{biz}^B(M_{biz},bitrate) = M_{biz}$ ). En premier temps, la relation entre les niveaux de price premium et ses valeurs respectives de la QoE est faite en utilisant l'échelle de la Table 5.3, sa catégorisation initiale suit la relation montrée sur la Figure 5.7, et les mises à jour des catégories, dont la logique est montrée sur les Tables 5.5a et 5.5b.

De plus, le paramètre subjectif ( $M_{subj}$ ) a été raffiné ainsi que ses fonctions de mis à jour. L'échelle de la Table 5.6 met en relation les niveaux de confiance avec leur valeur respective de la QoE. Ces catégories – ainsi que la valeur de la qualité associée – sont mises à jour par la fonction  $f_{subj}$ , dont la définition est montrée dans la Table 5.7.

L'analyse des réponses de l'enquête initiale nous ont permis de formuler ces définitions et de les appliquer dans le modèle de la Figure 5.8.

### E.5.3.2 Ajustement du modèle de la QoE

L'ajustement du modèle de la QoE – le calcul des coefficients de l'Équation 5.4 – sera fait par une régression linéaire sur les données. Ce processus demande le calcul des valeurs  $M_{B_r}$ ,  $M_{L,N}$ ,  $M_{biz}$  et  $M_{subj}$  en utilisant les valeurs des variables de contexte et la qualité pour les scénarios évalués par les utilisateurs.

Ce calcul donne un tableau complet (composé par 63 entrées) qui sera utilisé pour calculer la régression linéaire. Cet ensemble a été divisé en deux groupes principaux : le groupe d'entraînement et le groupe d'essai. Le premier peut être observé dans la Table 5.9. Ces données ont été chargées sur Microsoft Excel afin d'employer l'Utilitaire d'Analyse pour faire la régression linéaire. Une fois calculé, les résultats obtenus sont montrés sur la Table 5.10. De ce fait, il est possible de reconstruire le modèle de la QoE ajusté au cas d'étude sur l'Équation 5.5.

La Figure 5.9 expose les différences entre les valeurs de la QoE calculées par le modèle et les valeurs données par les utilisateurs sur les données d'ajustement. Dans la plupart des cas, le modèle prédit une valeur plus haute (entre 0.4 et 0.8 niveaux), malgré le coefficient d'ajustement du modèle ( $R^2 = 0.350191$ ).

### E.5.3.3 Test du modèle de la QoE

La deuxième partie des données a été utilisée pour tester le modèle. La Figure 5.10 montre les résiduels entre le modèle des valeurs des utilisateurs. En générale, le modèle prédit une valeur plus haute – entre 1, 4 en 3, 6 niveaux – qui peut faire une différence significative au moment de classer le scénario.

### E.5.4 Conclusions

Malgré les efforts à fournir un modèle de la QoE valide, les coefficients obtenus et les tests réalisés montrent que plus de recherche doit être faite sur ce sujet. Particulièrement, il faut trouver l'effet des paramètres subjectifs et d'affaires sur les valeurs de la QoE, ce qui aidera à construire un modèle plus précis et adapté au service analysé.

## E.6 Chapitre 6 : Analyse statique d'un service OTT

Le système d'évaluation proposé peut être aussi employé pour analyser la quantité d'information contenue dans le modèle EFSM du service. Cependant, ce type d'analyse demande l'introduction d'un nouvel algorithme.

### E.6.1 Algorithme d'analyse statique

Les EFSM ont été dessinées comme un modèle indépendant, ce qui a la capacité de retenir l'information – sous forme de *scénarios* de l'utilisateur – du service qu'il modélise. Chacun de ces scénarios est représenté par une *configuration* de la machine, composée de l'état et des valeurs des variables de contexte :  $c = (\sigma, \mathbf{v})$ . Théoriquement, le nombre total maximal des configurations contenues sur une machine est donnée par :

$$|C| = |\Sigma| \cdot \prod_{i=1}^{|V|} v_i \quad (\text{E.1})$$

c'est-à-dire, le nombre total de combinaisons possibles d'états et ses valeurs respectives des variables de contexte. Cependant, seulement une partie de ces configurations est accessible depuis l'état initial.

Afin de calculer toutes les configurations contenues dans la machine, il faut introduire un troisième processus au système d'évaluation, montré dans l'Algorithme 3. Cet algorithme est une modification du processus qui calcul le  $l$ -équivalent, sauf qu'il ajoute la configuration trouvée à l'ensemble une fois que l'exécution arrive au cas de base.

Cet algorithme a été appliqué sur le cas d'étude de beIN Sports Connect afin de montrer comment le système d'évaluation peut être appliqué même aux étapes précoces de développement du service.

### E.6.2 Analyse d'un service OTT réel

L'objectif de cet analyse est d'aider le fournisseur de services à réparer des possibles défauts sur le service en observant les possibles valeurs de la qualité. En particulier, l'algorithme est capable de :

- Calculer des scénarios possibles de l'utilisateur.
- Analyser les valeurs de la QoE qui sont contenues dans le modèle.
- Classifier les scénarios par ses valeurs de la qualité.
- Analyser en profondeur le modèle du service, en déterminant la longueur du  $l$ -équivalent optimal pour l'analyse.

#### E.6.2.1 Configuration expérimentale

Pour montrer l'application de l'approche, elle a été appliquée aux deux EFSM des Figures 3.6 et 5.3, en utilisant respectivement les modèles des Équations 3.5 et 5.4. Pour chaque expérience, le nombre de configurations contenues – et ses valeurs respectives de la qualité – a été calculé pour les  $l$ -équivalents de longueur 2 à 18. Finalement, une évaluation initiale des variables de contexte est requise, comme montrées dans la Table 6.1.

La configuration décrit ci-dessus ne permet pas une vérification correcte des EFSM et des modèles de la QoE, car les conditions d'expériences ne permettent pas de simuler des pannes sur le service. De ce fait, il faut définir des scénarios qui simuleront les pannes du service pour chaque type de modèles.

D'une part, le modèle « original » – c'est-à-dire la Figure 3.6 et l'Équation 3.5 – intègre la capacité de mesurer l'état du réseau de manière automatique, ce qui ne permet pas d'exécuter la transition qui détecte les mauvaises conditions du réseau. Pour simuler ces pannes, deux scénarios ont été définis : (1) une panne simulée après une période de temps fixe, c'est-à-dire une exécution de la transition de panne après un nombre fixé d'exécutions de la transition pour maintenir le flux ; et (2) une panne simulée de manière aléatoire, où la transition pour terminer le flux est exécuté avec une probabilité de 0,5.

D'autre part, pour le modèle « validé » il est possible d'utiliser les mêmes scénarios définis pour les vidéos, car ils définissent les détériorations du réseau expérimentés par l'utilisateur. De ce fait, un sous-ensemble de la totalité des cas a été sélectionné : « Constantbitrate3 », « Desc-bitrate1 », « Ascen-bitrate1 », « Freezing+bitrate11 », « Freezing+bitrate17 », « Freezing+bitrate21 », et « Freezing+bitrate24 ». Ces scénarios ont été choisis de manière à représenter les détériorations les plus expérimentés par les utilisateurs.

### E.6.3 Analyse statique du modèle d'un OTT

#### E.6.3.1 Modèle linéale de la QoE

L'Algorithme 3 a été exécuté avec la machine de la Figure 3.6 et le modèle de l'Équation 3.5. Les résultats sont montrés sur les Tables 6.2 pour chaque scénario de panne simulé.

En générale, le nombre de configurations contenues dans le modèle augment en même temps que la longueur du  $l$ -équivalent. Cependant, il est possible d'arriver au nombre maximal de configurations contenues dans la machine.

En observant la distribution des valeurs de la QoE, il est possible de voir l'absence des scénarios avec une mauvaise valeur de la qualité quand la profondeur de l'analyse n'est pas suffisante pour simuler les pannes. Une fois que l'analyse devient plus profonde, les valeurs de la qualité commencent à montrer une distribution normale, cependant, une fois arrivé à la quantité maximale de configurations, aucune nouvelle configuration est ajoutée à l'analyse.

#### E.6.3.2 Modèle validé de la QoE

De la même façon par rapport au modèle linéaire, l'Algorithme 3 a été exécuté avec la machine de la Figure 5.3 et le modèle de l'Équation 3.5. Les résultats de ces expériences sont montrés dans les Tables 6.3 et 6.4.

Dans cette analyse, il est possible d'observer le même effet de surestimation des valeurs de la qualité. Cette simulation renforce le besoin de faire encore plus de recherches sur le modèle validé.

Malgré cet effet, la même stabilisation du nombre des configurations est observée dans tous les cas testés, sauf que cet effet est survenu sur différentes longueurs du  $l$ -équivalent, dépendant du cas sous analyse. Ce fait montre l'importance de définir correctement l'ensemble des cas d'essai afin de caractériser la QoE des scénarios les plus proches du réels.

### E.6.4 Conclusions

L'extension proposée dans cette section permet de faire une analyse statique du modèle EFSM d'un service OTT. Il a été conçu pour détecter des défauts dans le modèle – et donc dans le service – en étapes précoces du développement.

L'approche a été appliqué sur les modèles d'un service réel, permettant d'analyser les scénarios contenus sur le modèle et, de plus, la distribution des valeurs de la QoE. Cette dernière expérimentation permet d'avoir une vue générale des valeurs de la qualité que l'utilisateur va expérimenter une fois le service soit accessible.

Finalement, l'analyse peut être utilisé aussi afin de trouver la limite de la longueur du  $l$ -équivalent, point à partir d'où le calcul le plus profond n'ajoute plus d'information à l'analyse du service.

## E.7 Chapitre 7 : Conclusions et perspectives

Les travaux présentés dans ce document visent à introduire les décisions d'affaires dans l'analyse de la QoE. Comme a été montré dans la Section E.2.2.3 les décisions d'affaires – en particulier, les prix du service – peuvent impacter les attentes de l'utilisateur et, de ce fait, la qualité expérimentée. La nouvelle approche proposée dans ce document vise à fournir le marché du

---

monitorage de service avec un ensemble de méthodologies et d'outils – en quatre contributions – qui intègrent les dimensions d'affaires dans l'évaluation de la QoE.

La première contribution est la proposition d'un nouveau système d'évaluation de la QoE. Ce système utilise le formalisme mathématique des EFSM pour maintenir une trace des décisions de l'utilisateur, afin d'intégrer ces décisions dans la prédiction de la qualité. L'approche est formée par 3 étapes incrémentales (modélisation, extension et calcul) qui permettent d'élaborer un modèle fonctionnel du service OTT, de l'étendre avec des indicateurs de qualité et, finalement, de l'utiliser pour calculer la valeur de la qualité en différents scénarios.

Afin de mettre en œuvre l'approche proposée, il est impératif d'analyser la nature des étapes établis par le système d'évaluation. Les deux premières étapes impliquent la modélisation et l'insertion des indicateurs de qualité dans le modèle, décisions qui ne peuvent pas être prises de manière semi-automatique, cependant, ce n'est pas le cas de la dernière étape : le calcul. De ce fait, la deuxième contribution présente la mise en œuvre des deux algorithmes qui font le calcul de la QoE en utilisant l'approche proposée.

De manière à fournir un ensemble d'outils complet, les technologies proposées doivent être validées avant d'être commercialisées. La troisième contribution présente les travaux relatifs à cette validation qui suivent une méthodologie composée par 3 parties. De manière à montrer la faisabilité de l'approche, un simulateur d'un service OTT a été implémenté, simulant le même comportement par rapport au service original et récupérant des valeurs de la QoE expérimentés par l'utilisateur. Cette plateforme a été utilisée comme base pour montrer des vidéos perturbées qui simulent des détériorations communes du réseau. Finalement, la même plateforme a été utilisée pour valider l'insertion des paramètres d'affaires dans l'analyse de la QoE, par le biais d'un questionnaire qui devait être rempli directement par l'utilisateur.

Les données collectées ont été utilisées avec un double objectif. D'un côté, le questionnaire introduit a permis d'établir le vrai impacte des paramètres d'affaires sur la QoE. D'un autre côté, les valeurs de la qualité fournies par les utilisateurs ont été employées pour la création d'un modèle de la QoE qui prend en compte différents éléments, les paramètres d'affaires parmi eux. Malgré les efforts d'ajuster un modèle robuste, encore plus de recherche est requis de manière à déterminer le vrai impacte des paramètres subjectifs et d'affaires sur la QoE. Ce développement permettra de fournir au marché un ensemble d'outils prêt pour son déploiement dans le domaine du monitorage des services OTT.

Finalement, la quatrième contribution était composée d'une extension de l'approche sous la forme d'une analyse statique du modèle d'un service OTT. En utilisant une modification d'un des algorithmes implémentés, il est possible de calculer la quantité d'information – le nombre de configurations – contenue dans le modèle du service. Cette information peut être augmentée avec les valeurs de la QoE afin d'analyser sa distribution, ayant une vue générale de tous les états possibles qu'un utilisateur peut expérimenter. Ce type d'analyse aide les fournisseurs de service à détecter des défauts sur le service même à des étapes précoces de développement.

Les travaux ici présentés ont été développés en considérant l'état actuel du marché de monitoring et de l'Internet. Chaque jour de plus en plus d'entreprises commencent à offrir ses services par Internet, et le monitoring de la qualité expérimenté par l'utilisateur reste toujours un problème ouvert.

L'approche et sa mise en œuvre présentées visent à fournir aux fournisseurs OTT un ensemble d'outils prêts à être utilisés afin de maximiser les revenus de l'entreprise ainsi que la qualité expérimentée par l'utilisateur.

### **E.7.1 Perspectives**

Le système d'évaluation a été conçu pour être complété avec les autres modules de MMT, en particulier, MMT Probe. De manière à profiter des capacités de DPI/DFI de ce module, le développement d'une extension de ce module est nécessaire, afin d'extraire les valeurs des variables de contexte de l'EFSM.

Cependant, l'analyse proposée par le système d'évaluation demande l'introduction d'indicateurs de qualité qui doivent être sélectionnés par l'analyste. De ce fait, le développement de l'extension du module MMT Probe doit être faite en même temps que le processus d'augmentation, de manière à fournir une pièce de logiciel qui satisfait les besoins du modèle produit.

De plus, l'approche exposée doit aussi présenter les données produites – les valeurs calculées de la QoE – au fournisseur de service. Pour ce but, le module MMT Reporting doit être étendu de manière à montrer les données produites par les algorithmes décrits.

Les deux extensions des travaux décrits permettront à l'opérateur du service de maintenir une trace des décisions de l'utilisateur et la valeur respective de la QoE. De plus, il sera possible de montrer le meilleur et le pire scénario prévu par les algorithmes. Finalement, ces extensions finiront l'intégration des outils et des méthodologies présentés, qui seront donc prêts à être déployés sur des services OTT réels.







**Titre :** Analyse et influence des paramètres d'affaires sur la qualité d'expérience des services Over-The-Top

**Mots clés :** Services Over-The-Top (OTT), Qualité d'Expérience (QoE), Qualité de Service (QoS), Qualité d'Affaires (QoBiz), Machines Étendues à États Finis (EFSM), Modèles Formels

**Résumé :** A l'époque où l'Internet est devenu la plateforme par défaut pour offrir de la valeur ajoutée, des nouveaux fournisseurs de services multimédia ont saisi cette opportunité en définissant les services Over-The-Top (OTT). Cependant, Internet n'étant pas un réseau de distribution fiable, il est nécessaire de garantir de haut niveau de Qualité d'Expérience (QoE), ainsi que les revenus des Fournisseurs de Services d'Internet (ISP) et des OTTs.

Le travail présenté dans ce document va au-delà de l'état de l'art, en proposant une solution qui prend en compte cet objectif. Les principaux apports qui y sont présentés peuvent être synthétisés en quatre contributions.

En premier lieu, l'inclusion des paramètres liés aux modèles d'affaires dans l'analyse de la QoE a demandé un nouveau cadre pour calculer la QoE d'un service OTT. Ce cadre est basé sur le formalisme mathématique des Machines Étendues à États Finis (EFSM), ce qui profite de deux avantages des EFSMs : les traces des machines suivent les décisions de l'utilisateur et les variables du contexte utilisés comme indicateurs de qualité, seront utilisées ultérieurement pour computer la QoE.

La deuxième contribution consiste à mettre en œuvre deux algorithmes. Le premier fait le calcul d'une forme équivalente, ayant la forme d'un arbre qui représente les

traces de la machine. Le deuxième utilise les traces et fait le calcul de la QoE pour les états terminaux de chaque trace. Les deux algorithmes peuvent être utilisés comme base d'un outil de monitoring capable de prévoir la valeur de la QoE d'un utilisateur. De plus, une mise en œuvre concrète de ces deux algorithmes comme une extension de l'Outil de Monitoring de Montimage (MMT) est aussi présentée.

La troisième contribution présente la validation de l'approche avec un double objectif. D'une part, l'inclusion de paramètres du modèle d'affaires est validée et on détermine leur impact sur la QoE. D'autre part, le modèle de la QoE proposé est validé par la mise en œuvre d'une plateforme d'émulation d'un service OTT qui montre des vidéos perturbées. Cette implémentation est utilisée pour obtenir des valeurs estimées par utilisateurs réels qui sont utilisés pour dériver un modèle approprié de la QoE.

La dernière contribution se base sur le cadre donné et fournit un analyse statique d'un service OTT. Cette procédure est réalisée par un troisième algorithme qui calcule la quantité des configurations contenues dans le modèle. En analysant à l'avance tous les scénarios possibles qu'un utilisateur peut rencontrer, le fournisseur des services OTT peut détecter des défauts dans le modèle et le service à une stade précoce du développement.

**Title:** Analysis and Influence of Business Parameters on Quality of Experience for Over-The-Top Services

**Keywords:** Over-The-Top Services (OTT), Quality of Experience (QoE), Quality of Service (QoS), Quality of Business (QoBiz), Extended Finite States Machines (EFSM), Formal Models

**Abstract:** At a time when the Internet has become the de facto platform for delivering value, the new multimedia providers took advantage of this opportunity to define Over-The-Top (OTT) services. Considering that Internet is not a reliable distribution network, it is necessary to ensure high levels of Quality of Experience (QoE) and revenues for Internet Service Providers (ISP) and OTTs. The work presented in this dissertation goes beyond the state of the art by providing a solution having this goal in mind. The main contributions presented here can be summarized in four main points.

First, the inclusion of business-model related parameters in the QoE analysis required a new framework for calculating the QoE of an OTT service.

This framework is based on the Extended Finite States Machine (EFSM) mathematical formalism, which takes advantage of two features of the EFSMs: (1) the traces of the machines that keep track of the user's decisions and; (2) the context variables used as quality indicators, correlated later with the QoE.

The second contribution is the design and the implementation of two algorithms. The first computes the *l*-equivalent, a version in the form of a tree of the model

that exposes the traces of the machine. The second uses the traces and computes the QoE at the final stages of each trace. Both algorithms can be used to design a monitoring tool that can predict a user's QoE value. In addition, a concrete implementation is given as an extension of the Montimage Monitoring Tool (MMT).

The third contribution presents the validation of the approach, having two objectives in mind. On the one hand, the inclusion of business-model related parameters was validated by determining the impact of such variables on the QoE. On the other hand, the proposed QoE model is validated by the implementation of an OTT emulation platform showing disrupted videos. This implementation is used to obtain QoE values evaluated from real users, values used to derive an appropriate QoE model.

The last contribution uses the framework to perform a static analysis of an OTT service. This is done by a third algorithm that computes the amount of configurations contained in the model. By analyzing in advance all the possible scenarios a user can face – and their respective QoE, the OTT provider can detect flaws in the model and the service from the early stages of development.