Modeling and optimization of the quality of customer experience (QoE) of data services on the mobile network. Application to video streaming

Mohamed Bouzian

École doctorale Sciences & Technologies de l'information et de la Communication
Unité de recherche : I3S

Thèse de doctorat

Présentée en vue de l'obtention du
grade de docteur en Informatique
de
l'UNIVERSITE COTE D'AZUR

par
Mohamed Bouzian

Modeling and optimization of the quality of customer experience (QoE) of data services on the mobile network. Application to video streaming

Dirigée par Guillaume Urvoy-Keller

Soutenue le 20 Juillet 2017
Devant le jury composé de :

André-Luc	Beylot	Professeur, IRIT/ ENSEEIHT	Rapporteur
Tijani	Chahed	Professeur, Telecom SudParis	Rapporteur
Lin	Chen	Maître de conférence, LRI/ Paris Sud	Examinateur
Mustapha	Bouhtou	Directeur scientifique, Orange Labs	Superviseur
Taoufik	En-Najjary	Ingénieur de recherche, Orange Labs	Superviseur

Modeling and optimization of the quality of customer experience (QoE) of data services on the mobile network. Application to video streaming

---

Mohamed Bouzian

July, 2017
Version: 0.2.1

University of Nice Sophia-Antipolis

I3S laboratory

Documentation

Modeling and optimization of the quality of customer experience (QoE) of data services on the mobile network. Application to video streaming

Mohamed Bouzian

1. Examiner Lin Chen, senior lecturer,
Department of Computer Science
LRI/ Paris Sud

1. Reviewer Andre-Luc Beylot, Professor
Department of Network and Telecommunications Engineering
IRIT/ENSEEIHT

2. Reviewer Tijani Chahed, Professor,
Department of Networks and Telecommunication Services
Telecom SudParis

1.Supervisor Taoufik En-Najjary, Research engineer
Orange Labs

2.Supervisor Mustapha Bouhtou, Scientific Director
Orange Labs

3.Supervisor Guillaume Urvoy-Keller, Professor
Department of Networking
I3S/UNS

July, 2017

Mohamed Bouzian
Modeling and optimization of the quality of customer experience (QoE) of data services on the mobile network. Application to video streaming
Documentation, July, 2017
Reviewers: Andre-Luc Beylot, Professor and Tijani Chahed, Professor,
Supervisors: Taoufik En-Najjary, Research engineer and Mustapha Bouhtou, Scientific Director

University of Nice Sophia-Antipolis
I3S laboratory
Grand Chateau B.P. 2135 06103 CEDEX 2, 28 Avenue Valrose
06000 and Nice

Abstract

In recent years, mobile devices have become more powerful in terms of computing power, memory, size and screen quality. These improvements have greatly stimulated demand for multimedia services including video streaming. Moreover, customers are increasingly demanding in terms of the quality rendered on this type of service. In addition, a bad experience of video streaming has a great impact on the perception that customers have of the operator. Optimizing the quality of experience (QoE) of customers on the services of video streaming is thus a major competitive challenge.

In this thesis, we are interested in modeling and optimizing the QoE of the streaming services taking into account the usages and the mobility of the clients. In particular, we are interested in two strategies for delivering content on the mobile network: the Fast Caching strategy and the On-Off strategy. We develop analytical models that allow us to calculate major indicators of the QoE for the streaming service that are: i) starvation, which drives operators to send more video content to customers and ii) the loss due to abandoning the video playback, which pushes them to send video content in a strictly reasonable manner. An optimal balance is to be found.

We also propose QoE oriented strategies by maintaining a good level of QoE over the entire network and optimizing the QoE indicators of streaming services.

Abstract (Frensh)

Ces dernières années, les terminaux mobiles sont devenus plus performants en termes de puissance de calcul, de mémoire, de taille et de la qualité de l'écran. Ces améliorations ont fortement stimulé la demande de services multimédias notamment la vidéo streaming. Par ailleurs les clients sont de plus en plus exigeants quant à la qualité rendue sur ce type de service. En plus une mauvaise expérience de la vidéo streaming a un grand impact sur la perception qu'ont les clients de l'opérateur. Optimiser la qualité d'expérience (QdE) des clients sur les services de la vidéo streaming est donc un enjeu concurrentiel majeur.

Dans cette thèse, nous nous intéressons à modéliser et optimiser la QdE des services streaming en tenant compte des usages et de la mobilité des clients. En particulier, nous nous intéresserons à deux stratégies de délivrance des contenus sur le réseau mobile : la stratégie ''Fast Caching" et la

stratégie ‘’On-Off”. Nous développons des modèles analytiques qui nous permettent de calculer des indicateurs majeurs de la QdE pour le service streaming qui sont : i) la famine, qui pousse les opérateurs à envoyer plus de contenu et ii) la perte due à abandonner la lecture d’une vidéo, qui pousse ces derniers à envoyer du contenu vidéo d’une manière strictement raisonnable. Un équilibre optimal est à trouver.

Nous proposons aussi des stratégies de gestion de la QdE en maintenant un bon niveau de QdE sur la totalité du réseau et en optimiser les indicateurs de la QdE du service streaming.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisors, Prof. Guillaume Urvoy-Keller, Mr. Taoufik En-Najjary, Mr. Mustapha Bouhtou and Mrs Lucile Sassatelli for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Besides my supervisors, I would like to thank the rest of my thesis committee: Prof. Lin Chen, Prof. Tijani Chahed, and Prof. Andre-Luc Beylot, for their insightful comments and encouragement which incented me to widen my research from various perspectives.

My sincere thanks also goes to Mms Marie-Françoise Colinas who provided me an opportunity to join their team as a PhD student.

I thank my colleagues for the stimulating discussions, and also for all the fun we have had in the last four years. Also I m grateful to all my friends for supporting me during all these years.

To my life-coach, my late father: because I owe it all to you. Many Thanks!

Last but not the least, I would like to thank my family: my mother, my sister, my two brothers and my nieces for supporting me spiritually throughout writing this thesis and my life in general.

Contents

1 Introduction **1**

1.1 Motivation . . . 3
1.2 QoE concept and definition . . . 3
1.2.1 QoE Influence factors and metrics . . . 3
1.2.2 QoS and QoE . . . 5
1.3 QoE and video streaming . . . 9
1.3.1 Video Streaming . . . 9
1.3.2 QoE Modeling and Monitoring in Video Streaming. . . . 10
1.3.3 QoE measurement in the context of video streaming . . . 14
1.3.4 Video Quality models with unreliable transport mechanisms . . . 16
1.3.5 Video Quality models with reliable transport mechanisms . . . 17
1.3.6 Streaming strategies . . . 19
1.4 Thesis outline . . . 20
1.5 Conclusion . . . 21
1.6 Introduction (French) . . . 21

2 State the of art **25**

2.1 Models for computing the QoE . . . 26
2.2 QoE and network criteria . . . 28
2.2.1 QoE and user's dynamic . . . 28
2.2.2 QoE and resource management . . . 29
2.2.3 QoE and pricing . . . 30
2.3 Inspiring works and first experiments . . . 31
2.3.1 Modeling user video's QoE in literature . . . 31
2.3.2 QoE, streaming downloading strategies and energy consumption . . . 34
2.3.3 Simulating QoE indicators on different streaming strategies . . . 36
2.4 Conclusion . . . 41

3 Evaluation and optimization of QoE of On-Off strategy **43**

3.1 On-Off vs Fast Caching (FC) . . . 43
3.1.1 Fast Caching streaming strategy . . . 44
3.1.2 On-Off streaming strategy . . . 44
3.1.3 On-Off vs Fast Caching: advantages and disadvantages . . . 45
3.2 Analytical Markovian model for QoE metrics . . . 47
3.2.1 Network model and hypotheses . . . 48
3.2.2 Users and transitions distribution: $(\pi_i)_{0\leq j\leq K-1}$ and $(\nu_i)_{1\leq j\leq K-1}$ . . . 50
3.2.3 Prefetching delay distribution . . . 51
3.2.4 Starvation probability . . . 52

3.3 Model validation . . . 57
3.4 Starvation probability: On-Off vs. Fast caching . . . 58
3.5 QoE optimization: starvation probability vs start-up threshold . . . 58
3.5.1 QoE indicators . . . 59
3.5.2 Goal function . . . 60
3.5.3 Optimization of the goal function: results . . . 64
3.6 Conclusion . . . 65

4 Evaluation and optimization of QoE of On-Off strategy: Loss due to abondment 67
4.1 From starvation phenomenon to loss due to abandonment . . . 67
4.2 Network model . . . 68
4.3 Loss due to the abandonment . . . 70
4.3.1 Probability distribution of buffered data $Z(q,t)$ for the On-Off streaming strategy . . . 71
4.3.2 Probability distribution of buffered data $Z(q,t)$ for Fast Caching streaming strategy . . . 74
4.3.3 Model result and simulation . . . 75
4.3.4 Case where every user can abandon the video playback . . . 77
4.4 Optimization: starvation vs wasted bytes . . . 78
4.5 Conclusion . . . 81

5 Enforcing QoE at the base station 83
5.1 QoE for streaming video: from the study of existing models to proposing new strategy 84
5.2 Linear modeling of the QoE . . . 86
5.2.1 Modeling of the base station and main client variables . . . 86
5.2.2 variables controlling client status . . . 88
5.2.3 Filling and updating clients' buffer . . . 91
5.2.4 Capacity constraint . . . 92
5.3 Objective function . . . 92
5.3.1 Formulation of the optimization problem . . . 93
5.4 Simulations . . . 94
5.5 Conclusion . . . 100

6 Conclusion and perspectives 103
6.1 Conclusion and perspectives (French) . . . 104

Bibliography 107

1 Introduction

> *We cannot solve our problems with the same thinking we used when we created them.*
>
> **— Albert Einstein**

The next generation of mobile networks is being deployed to carry and assure a high quality when delivering media data from service providers to customers. The operators always try to deliver information data in an optimal manner while ensuring a good user perception. Their goal is to provide network services that users want to use in the most cost and resources efficient way and by ensuring a positive experience to the end users. This relationship between cost, resources efficiency and user perception of a system or service is still relatively poorly understood. Operators and service providers are switching their focus from quality of service (QoS) of the network access to user perception and satisfaction, which is the network quality from the user's perspective. A high end-to-end quality becomes the customer's highest requirement that must be guaranteed by telecommunication operators and service providers. This is known as quality of experience (QoE). It becomes commonly used to measure the customer satisfaction. The QoE is not only a customer indicator but a concept of evaluation of the quality of the network services from end-user, to ensure a good quality over all the communication chain (e.g. service provider-information network-customer). In fact, operators and service providers need to improve and ensure a satisfactory level of QoE towards their customers. To reach this goal, they encountered challenges such as measuring and monitoring this user's perception quality.

In previous years, telecommunication operators used the notion of quality of service as a concept for improving their abilities to send data over access networks efficiently. Their delivery strategies was always focused on connecting network elements, connection elements capabilities and maintenance. This approach allows operators to determine a path for every information flow with efficient resources. The QoS parameters were, traditionally, end-to-end delay, loss rate and bandwidth. To achieve a good performance, a QoS-based delivery system may also try to optimize other criteria such as network utilization and carried load. These criteria can influence the delivery strategy, but satisfying all user's requirement is not easy. Therefore, most strategies and approaches were network oriented and only considered the service-level agreement between operators and service providers. However, this perspective is no longer optimal with the emergence of new types of content on the web and especially high quality multimedia content. Users are watching and downloading high quality videos and broadcasting their own videos on the Internet. These QoS sensitive data flows impose stringent constraints on the delivery system that are far more challenging than the agreements

between operators and services providers. Thus, the interest of both service provider and operators in providing a satisfactory delivery to achieve user requirement becomes more important than just considering network parameters and maintenance.

As a consequence, QoE has become a key metrics for the operators. The International Telecommunication Union (ITU) defines the QoE as the overall acceptability of an application or service, as perceived subjectively by the end user. It is a measurement of how well a service satisfies the user's requirement and all end-to-end system performance and quality. QoE includes influence factors such as the customer state (e.g. happy, motivated, predisposed...), the ergonomics of the system (e.g. functionality, complexity...) and the context or environment in which the service will be experienced.

Given the above context, we focus, in this thesis, on how to deal with this concept of Quality of Experience. More precisely, in the context of multimedia services and specially streaming video, we will tackle the problem of QoE for streaming video services. In the rest of this chapter, and in order to understand more the QoE concept, we first, present the definition of QoE and our motivation to work on it. We then discuss the factors influencing the QoE and, after, see how to model, monitor and measure the QoE in a video streaming context. The chapter ends with the plan of the rest of the thesis. Table 1.1 provides a list of acronyms (and their definitions) used extensively in the manuscript.

Abbreviations	**full words**
ACK	acknowledge
CDN	Content Delivery Network
DASH	Dynamic Adaptative Streaming over HTTP
HTTP	Hyper Text Transfer Protocol
iOS	IPhone Operating System
IPTV	Internet Protocol TeleVision
JPEG	Joint Photographic Experts Group
LTE	Long Terme Evolution
MPEG	Moving Picture Experts Group
MCDN	Mobile Content Delivery Network
QoS	Quality of Service
QoE	Quality of Experience
IP	Internet Protocol
IF	Influence factor
RTSP	Real Time Streaming Protocol
RTP	Real Time Protocol
RTCP	Real-time Transport Control Protocol
TCP	Transmission Control Protocol
XML	eXtensible Markup Language
KPI	Key Performance indicators
3GPP	3rd Generation Partnership Project

Tab. 1.1: List of abbreviations

1.1 Motivation

Nowadays, the customers have a wide range of choice of services and products proposed by operators and service providers. Consequently, the competition between operators is becoming more and more fierce. Hence, the end user is today in a strong position: selecting the best between different competing operators and providers. Apart from the pricing system of the different providers, which is a useful decision-making aid for users, the customer's choices are also influenced by the expected and experienced quality. So, the interest of operators and services providers in how end users perceive quality has increased. Consequently, the QoE has been defined and used to help operators and service providers to take into account customer satisfaction. In the next sections, we will clarify what QoE is and how important it is before discussing the QoE in the context of streaming video.

1.2 QoE concept and definition

Today, the customer perception plays a major role in all communicating systems. The quality perceived by the end users when using a service is called the QoE. For example in streaming video, data passes through many processing phases before delivering to end user (acquisition, processing, coding, transmission, decoding...). The quality perceived by the end user may be affected by each of these phases.

In the past years, service providers ensured only the QoS notion, which measures and guarantees characteristics of service from the service providers perspective. The QoS takes into account only the characteristics of the components of a system and not the end user perception. In contrast to quality of service (QoS), QoE describes the overall performance of a network from the user's perspective. QoS is a component of QoE, which means that a network featuring a high level of QoS can result but not assure a high QoE.

QoE represents aspects related not only to subjective perception, but also customer behavior and needs. Thus, the best QoE evaluations are obtained by sampling a large number of user's perception. But the question is, how to evaluate and improve QoE? To tackle this issue, let us, first, detail the QoE metrics.

1.2.1 QoE Influence factors and metrics

Different factors can influence QoE in the context of communication services and applications. QoE can be subject to a range of various and complex correlated factors. In [Boo], the author defines three sets of influence factors: human, context and system IFs. The human IFs refer to any human users's characteristic. This characteristics can describe the economic background, or the cleint's emotional state [ETI12]. The contexts IFs are factors encompassing the user's environment (physical context, evolution and birth environment, and technical and information context) [ETI12]. System IFs refer to characteristics that determine the technical quality of a product a service, or application [ETI12]. The systems IFs can be divided in three categories: network, content and device influence factors. The network IFs are related to the transmission of data over a network [Boo]. The most popular

networks IFs are: delay, jitter, bandwidth, loss, error rates and throughput. The QoE depends also on the nature of service: the impact of some IFs can be more threatening than others if the proposed service is interactive and not passively consumed [Boo]. For examples, streaming video is more passive consumption compared to video conferencing, in which delay may have a big impact. In the context of streaming video, the influence of loss and bandwidth is quite different. Bad network's conditions can result in video starvation (freezes in video) without losing content.

The content IFs encompass all factors related to the nature of data. Content type requires some system properties. For example, details and motions in a video scene is important while the bandwidth and dynamic range are the major factors that can affect QoE for audio information (i.e mucisal and voice data flow) [Boo].

The media related system IFs refer to media technical factors such as encoding rate, screen resolution, frame rate, sampling rate and data flow synchronization. To deal with the problematic of limited resources, we use compression technics, which can be either lossless or lossy. This second compression technics ensures higher compression rates at the cost of quality. For instance, the most used lossy coders are JPEG, and MPEG4/AVC H.264 for image and video compression.

The device IFs refer to the terminal equipment. The characteristics of this received system, such as capacity, will have an impact on the QoE. For example, if the user equipment have a low resolution, most of the original image details might be lost.

QoE is difficult to compute and predict due to its subjective aspect. However, we should find a way to measure it to be able to evaluate the overall product or service quality. We need a a measurement strategy to evaluate QoE as realistically as possible. One solution to achieve this goal is to determine the factors that influence client perception, at the beginning, and try to measure and evaluate this factors. For example in the context of video streaming, the first famous influence factor is video starvation. In fact, video starvation ruins completely user perceived quality since users are not happy if a videos freezes during playback. This is known in the literature by the top-down approach [Sol+07]. The idea here is to define at the beginning the influence factors and then, generate operating requirements based on that.

We can define two categories of expectations for the initial phase (the beginning): reliability and quality. The quality in this case is typically the QoS and application software issues. The reliability factors includes:

- -service access time: anywhere;
- -service accessibility: anytime;
- -service availability: setup time of a service;
- -service retainability (continuity of service connection).

The QoE factors include:

- -en to end response delay: delay resulting of transmitting delay(at MAC layer), Propagation delay and queuing Time of a packet;

- -quality of the session: encompasses all the quality interactions between two terminal devices during a connection to exchange data ;

- -bit rate variations: variation of the number of bits that are conveyed or processed per unit of time;

- -service response time: time between the end of an order requested to a smart system and the beginning of a response of the same system.

In the literature, these factors represens the Key Performance indicators (KPIs). The KPIs are evaluated in order to compute a score for each product or service. It is important to note that some KPIs will be irrelevant in some situations while being the most significant in another cases. For example, service response time is very important for interactive gaming while it have no big impact in web browsing applications.

With the huge amount of video transmitted across the Internet infrastructure, video QoE is of large interest to users, service providers or telecommunication operators. As a consequence, video services have received a lot of attention by research and development activities over the past years. Next, we will detail more the relation between QoS and QoE. Finally, we will focus on how to monitor and measure the QoE in the context of video streaming.

1.2.2 QoS and QoE

The concept of QoS has been tackled for more than a decade [REC11; REC01]. It includes essentially the technical aspects on service quality.

QoS is defined as "ability of the network to provide a service at an assured service level" [Sol+07]. The QoS can also be defined as "capability of a network to provide better service to selected network traffic described by the following parameters: delay and jitter, loss probability, reliability, throughput and delivery time" [Mar+07]. From this definition, we can see directly that QoS is a technical concept which has almost no meaning to end user. Figure 1.1 shows the QoE diagram. We remark directly that QoS is a component of QoE.

Compared to QoS, the concept of QoE refers directly to user perception about the quality of a service, product or network. It can be expressed directly by customer feelings like "excellent service", "good service" or "bad service". In [Fie+10], Fiedler and al. give a schematic relation describing the impact of QoS problems on QoE. In Figure 1.2, the x-axis is the QoS disturbance and the y-axis is the QoE value, which is described by a score scale from 1 to 5 (1 is the worst case and 5 is the best). We remark that the QoE can be broken down into three areas which are separated by *threshold 1* and *threshold 2*. Here is the analysis of these three areas:

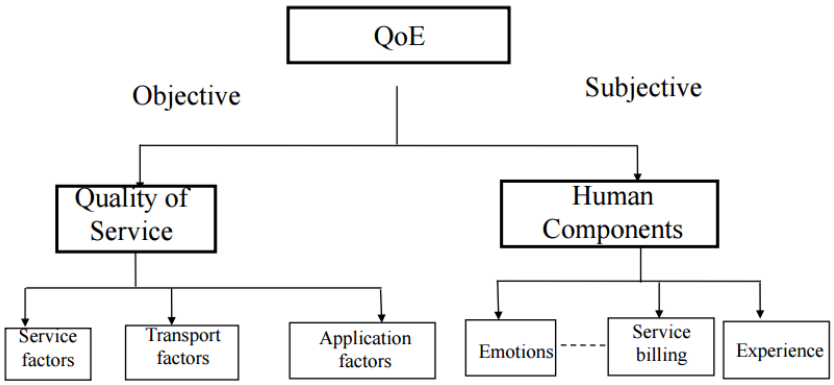


Fig. 1.1: diagram linking QoE and QoS.

- Area of constant QoE: the QoS in this area is good enough to insure a good and constant QoE. A little variation by disturbing the QoS does not affect the QoE value.

- Area of decreasing QoE: in this area and when QoS disturbance exceeds *threshold 1*, the QoE value starts to decrease. Then, the higher the QoS disturbance increases, the higher the QoE value decreases. Also, a little additional QoS disturbance might have a considerable impact on the QoE since it decreases exponentially.

- Area of unacceptable QoE: After *threshold 2*, the quality perceived becomes unacceptable. The dashed line reflects the possibility that a user leave the service.

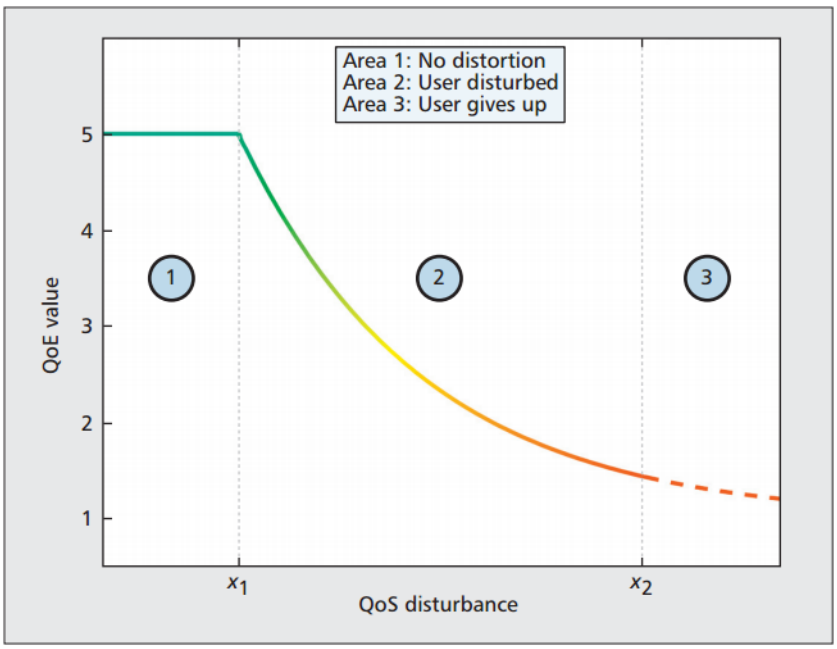
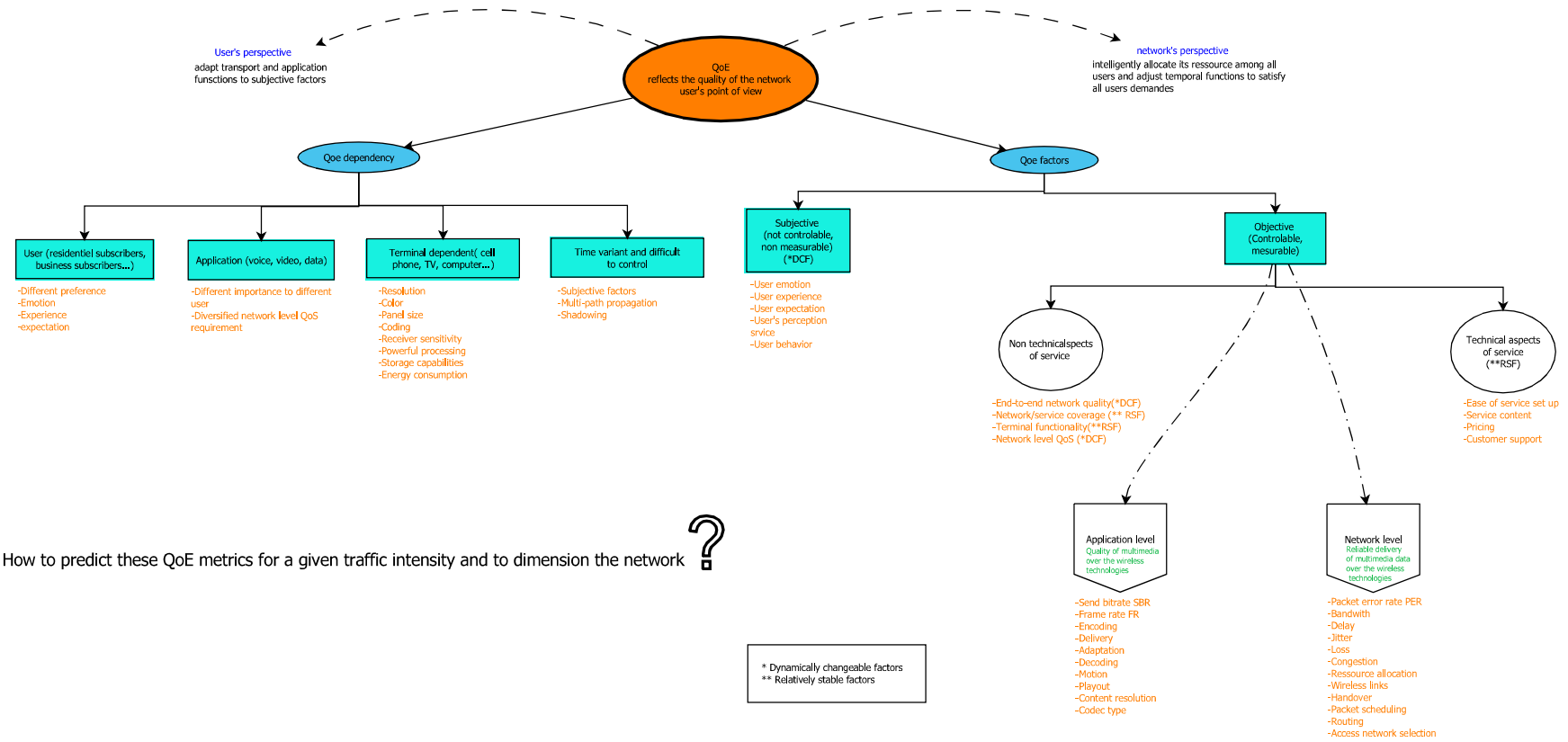


Fig. 1.2: General mapping curve between QoS and QoE.

Concerning the relation between QoE and QoS in Figure 1.2, the service providers and operators should detect the instant of *threshold 1* and find and propose some solutions to improve the quality as long as the quality is dropping into the decreasing area. The goal is two-fold: solve the quality

problem degradation and avoid the churn problem (i.e customers leaving the operator). Therefore, the QoE is the most important indicator for a telecommunication company to properly manage and serve their customers. The diagram in the next page sums up all QoE factors and dependency.



How to predict these QoE metrics for a given traffic intensity and to dimension the network ?

In order to guarantee a good QoE, apart from understanding the relation between the QoS and QoE and especially in the context of multimedia services, we need to model and monitor the QoE in streaming video. The next section discusses this issue.

1.3 QoE and video streaming

Video content accounted for 64 % of all the world's internet traffic in 2014 [Ti]. According to a new report from Cisco, by 2019, online video will be responsible for four-fifths of global Internet traffic. This shows that the streaming traffic does not cease to increase and represents almost all the traffic exchange on the Internet. This is the reason why operators are more focused on streaming service which constitutes the context of this thesis. Next in this section, we define streaming video before moving to how to model and monitor the QoE in this streaming service. After that, we will discuss video streaming techniques.

1.3.1 Video Streaming

A video stream is a data flow sent, after compression, from a service provider (i.e Youtube, Dailymotion, Vimeo...) and displayed by the end user in real time. The major idea behind streaming video is that user does not have to wait to download all file or streaming data to play it. Instead, the media content is sent in a continuous stream of packets and is played as it arrives at the client terminal (i.e mobile, tablet, laptop...). To play data stream and watch a video, the user needs a player, which is a program that uncompresses and sends video data to the display and audio data to speakers. A player can be part of a browser or a downloaded application. Generally when a mobile device requests a video, an interchangeability is created between the transport layer and the application layer of the OSI model [Wik13] which is represented in Figure 1.3.

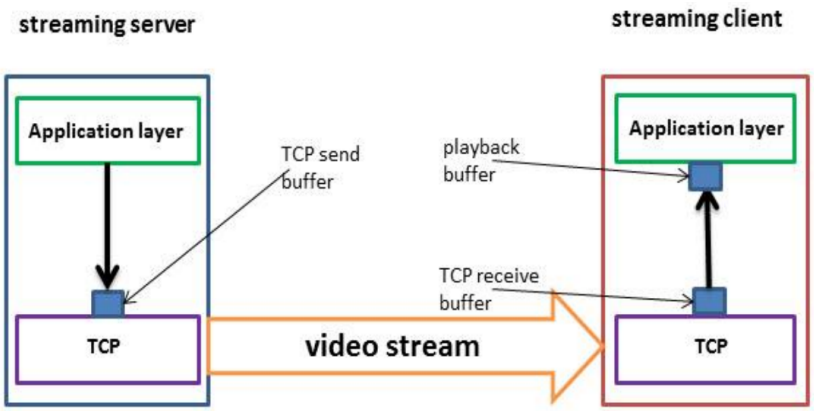


Fig. 1.3: Interplay of TCP and applications via buffers.

Streaming video is stored in a service provider server and sent to the client when she requested it.

Next, we will see an overview of modeling and monitoring the QoE in the context of streaming video.

1.3.2 QoE Modeling and Monitoring in Video Streaming.

In this section, we give an overview of QoE models that were proposed in literature, for video streaming services. Song and al. [Son+12] studied and mapped the impacts of user, system and context IFs at four elements of the mobile video chain: mobile user, mobile device, mobile network, and mobile video services. Their model can be exploited by video vendors to design a User-centric strategy that aims to improve user's experience. As defined in section 1.2, QoE takes into account all the End to End (E2E) quality : users or clients, terminal devices, user network, core and access network. Different from QoS, the QoE is affected by other factors. In [ZA11], Zhang and Ansari define two categories of these factors: subjective and objective factors. The first group is for the non-controllable factors and are conducted to obtain information on the quality of multimedia services using opinion scores, while the objective factors can be controlled and are used to estimate the network performance. It includes the technical indicators such as network quality, network coverage and terminal properties, and non-technical indicators such as service or product content, pricing.

As previously mentioned, QoE is a large concept which is difficult to evaluate, monitor and measure since it needs to take into account a wide number of QoE IFs. In order to provide good QoE evaluating models, we should consider a large number of IFs and not just some of IFs: it may be insufficient. Consequently, knowledge of the major IFs for a given product or service, from QoE models, can be the best input for a QoE monitoring model. A telecom operator must keep in mind two major purposes while monitoring QoE: i) the threshold under which the QoS perceived bu users becomes unacceptable and ii) the degree of influence of each IF on the QoE. Agboma and Liotta [AL08] explain how the QoE can be captured in a statistical model that correlates the QoS factors with QoE factors and the degree of influence of each one of these parameters. Authors group parameters that affect the QoE of multimedia content in two groups: application QoS parameters(i.e resolution, frame rate, codec type,...) and Network QoS parameters(i.e bandwidth, delay, packet loss, jitter,...). The paper aims to maximize QoE while reducing network costs and ressources. It shows how, practically, we can control the application QoS parameters to manage the QoE. Depending on the types of service (news, music video, football...), the end user is more influenced by some network parameters such as encoding rate, frame rate. For example, the quality of a video perceived by the end user is more influenced by the encoding bitrate. The model will allow operators to anticipate the QoE by allocating network resources accordingly.
Generally, the goal of QoE models is to help operators and service providers to understand, evaluate and then monitor all the end-to-end prameters. When discussing QoE monitoring, we distinguish between two approaches: network-based approach and client side approach [SK13]. Next, we will see how to monitor QoE while operating on network or user side.

Monitoring at the client side

While monitoring the QoE at the client, processed data can be divided in five categories:

- end user data: it includes all objectives measures such as user motivation, user mouse and keyboard clicks;

- device data: it includes all technical aspects of the terminal equipment such as screen resolution;
- application data: it includes all parameters such as response time and ergonomics;
- network data: it includes core and access network configuration and characteristics such as bandwidth, loss and jitter.

Monitoring the QoE, especially, at the client side is the best way to serve better the end user. In fact, it provides the best indications about the perceived quality to serve with efficiency customers. We should just find a way to provide the network, service provider or application with QoE parameters feedback to be able to adjust, control, and optimize the QoE. In [Rao+11], Rao shows that the streaming strategies vary with the type of application (web browser or mobile application) and the type of container (Silverlight[1], Flash or HTML[2]) for video streaming. In[Nam+13], authors analyze the behavior of Youtube's and Netflix's network traffic (HTTP adaptive bit-rate streaming video[3]) while watching the videos on mobile devices (iOS and Android) over wireless networks (Wi-Fi, 3G and LTE). The actual traffic depends on several factors like the type of equipment, the application in use and also network conditions. Experiments show that the quality depends, highly, on "hardware" of the customer's equipment when he asks a video. In [Liu+], Liu and al. compare and analyze the behavior and the performance when using, respectively, Android and iOS devices to access Internet streaming services. They found that, for Android devices, the downloading of a video on Youtube would be paused when the total buffer size reaches a treshold of about 20 MB. When the to-be-played data is less than 4 MB, the equipment resumes the downloading. Note that to keep the connection with the server when the downloading is paused, the device downloads a PageSize (64 KB) of data every 15 seconds. They found that Android devices uses a fixed amount of memory for the playout buffer compared to iOS devices that always adjust the playout buffer dynamically at runtime. They found also that, Android devices always download the exact amount of data corresponding to the video size, while iOS devices always try to keep as much video data as possible in the buffer for user's experience. In DASH[4] services, the QoE indicators such as the format of the reports of QoE and protocols and QoE measurements is activated at the user terminal in MPEG and 3GPP normalization [Sto11]. Note that generaly, DASH over HTTP provides the possibilty to control the streaming connection to end customer. In 3GPP standarization, the client can report the following QoE metrics: HTTP request/response transactions, representation switch events, average throughput, initial playout delay, buffer level and play list [Che+15a]. The reported documents are formated as an eXtensible Markup Language (XML). In 3GPP LTE systems and to optimaly manage the network resources and improve the QoE perceived by the end user, it is important to adopt a QoS delivery strategy for DASH services [OS12]. Figure 1.4 shows a possible manner of Policy and Charging Control (PCC) architecture for delivering with good performance, in terms QoS/QoE, end-to-end data for DASH services [SK13]. As mentionned in [OS12], the current 3GPP policy architecture supports only QoS service adaptation and delivery for RTSP-based adaptive streaming services and so the need for new strategies for HTTP adaptive streaming services.

[1] An application framework for writing and running rich Internet applications.
[2] Standard markup language used to create web pages.
[3] See section 1.3.5
[4] Technique that enables high quality streaming of media content over the Internet delivered from conventional HTTP web servers. It is based on the preparation of the content in different qualities and throughput, cut into segments of short duration (a few seconds). Each of these segments is made available individually by means of an exchange protocol, which is generally the HTTP protocol

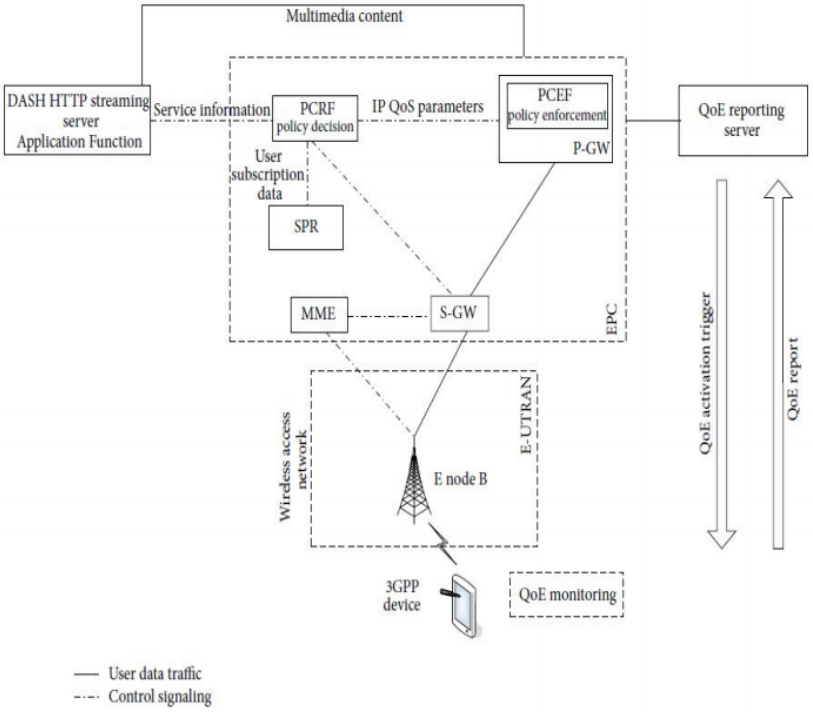


Fig. 1.4: Example of possible PCC architecture performing end-to-end QoS/QoE delivery for DASH services [SK13].

On the other hand, other studies focused more on collecting QoE key metrics at the client side. In [Ket+10b] Ketyk'o and al. have presented an approach of implementing a QoE measurement tool for the Android systems by collecting objective and subjective parameters. The objective parameters are linked to QoS monitor components implemented on an Android terminal. It includes packet loss rate, video and audio jitter and the duration of the session to a specific network type. Also, subjective parameters are linked to a Monitor component (deployed on the Android equipment). It includes picture quality, sound quality, a test of users's ratings of content, playback fluidity and speed of downloading data.

Monitoring in the network

In [Hoß+12], authors compare two approaches to monitor the QoE on YouTube. A novel passive monitoring in network tool for YouTube is presented. This work aims at detecting and measuring video frezees by approximating the video buffer state. The main idea is to compare the playback times of video frames and the time-stamps of received video flow [SK13].

In the context of MEDIEVAL project (MultiMEDia transport for mobile Video Applications), Amram et al. [Rao+11] have designed a new dynamic transport architecture for the next generation mobile networks adapted to video streaming. The idea is the transport optimization of video delivery achieved through a QoE oriented design of networking as well as the integration of content delivery networks techniques. In fact, their new architecture aims at providing optimized video traffic in the

mobile operator's core network through intelligent caching and cross-layer interactions, with the objective of reducing the load on the operator's backbone while providing a satisfactory QoE to the users. To reduce the load in the operators mobile core network, they opted to implement "Mobile CDN". They use MCDN nodes that cache popular video files and seek to optimize the number and location of caches to ensure a satisfactory level of performance at a sufficiently low cost.

To maximize the QoE experienced by video users, operators must ensure: (i) resource allocation and negotiation from the core network to the wireless access, and (ii) optimal handover decisions by interacting with the mobility entities. Hoque and al. [Hoq+13] developed and implemented a tool called EStream that reduces energy consumption in the case of TCP traffic in wireless streaming for Youtube and Dailymotion. It reduces energy consummation by 65, 50-60 and 35 % for, respectively, WIFI, 3G and LTE. In fact, EStream is a cross layer mechanism that, at the network layer, checks the TCP acknowledgements received from the streaming clients to identify client TCP receive buffer status: this is called Fast Start Streaming Period, when traffic is sent using maximum bandwidth and the goal is to look for ACK packets corresponding to a burst, and report the playback buffer status; window size zero indicates that client buffer is full . EStream starts, thereafter, an other period in which it selects the burst size at the application layer based on these client receive buffer status in order to reduce energy: based on the informations gathered before, EStream seeks an optimal burst interval in which a client can wait without its buffer running dry.
Xu and al. [Xu+13] have another vision to monitor and control QoE in network. They model and propose a closed formula for computing the starvation probability (which is a factor of the QoE) when wathcing a video from dailymotion for example. Their study is based on the fact that the flow dynamics (arrival or departure of one or more flow at the base station) is the fundamental reason of playback starvation (buffer underflow). Using an approach based on solving linear partial differential equations, modeling the infinitesimal variation of buffer playout, they derive a closed form solution of the starvation probability and thereafter, the distribution of starvation events. In the same context, they model, in [Xu+11], the playout buffer as an M/M/1 queue. By setting the number of packets to download, they use the Ballot theorem or a recursive approach to derive the number of buffer starvations. The Ballot theorem approach gives an explicit result while the recursive approach does not, but it is more general because it allows obtaining the number of buffer starvations with non-independent and identically distributed (i.i.d.) arrival process.
Parandeh-Gheibi and al. [Rao+11] study fundamental rate-delay-reliability trade-offs. They consider the interruption (starvation) probability as well as the number of initially buffered packets as the two most IFs on QoE. They characterize the optimal tradeoff between these IFs as a function of system parameters such as the packet arrival rate and file size, for different channel models. In the first model, they assume packets arrive according to independent Poisson processes from multiple servers or peers and deterministic departures. They show that for arrival rates slightly larger than the play rate, the minimum initial buffering required to achieve a certain level of starvation probability remains bounded as the file size grows. In the second model, they consider channels with memory, which can be modeled using Markovian arrival processes, and characterize the optimal trade-off curves for the infinite file size case.

From what we have seen, monitoring at the network or customer side are both important to understand and improve the QoE. So, it is clear that the works considering these two visions (monitoring at the network and client side) will bring more added value.

Monitoring Combining network and client side

Due to the dominance of YouTube traffic in video streaming, Staehle and al. [Sta+11] have proposed a monitoring tool: YOMO. The tool detects a YouTube video and determines its buffered playtime. YOMO runs at the client side and parses all incoming TCP flows. Consequently it recognizes each flow containing Flash Video data by detecting the header of a Flash Video file. Thereby, YOMO can detect and anticipate a QoE degradation that is, video freezes. The only considered factor influencing the Youtube QoE is the interruption of the video playback: it is the key IF in this case. The tool communicates the video freeze information to the network advisor by computing and visualizing the time between the amount of video already downloaded and the current instant of the video, and checks if it drops below an critical threshold. The particularity of YoMo is its ability to predict the freezes instant in advance. Consequently, YoMo offer a possibility to network operator to react before QoE degradation and then avoid a bad customer experience.
Ketyk'o and al. [Ket+10a] present a measurement approach of QoE for mobile video streaming in a 3G network. The measurement concept here is based on collecting data, by a sampling method, that combines objective and subjective metrics to evaluate the QoE for customers [SK13]. Parameters such as audio and video packet loss and jitter are obtained at the server-side, while the received signal strength indication is measured at the client side. The subjective assessment parameters have been conducted in two phases: (i) presage questionnaire and (ii) usage period during which customers use the mobile application in six different environments: indoor and outdoor, on a train or at bus and finally at home or work. This study has shown clearly that the QoE for video streaming is widely affected by the QoS and also by the context. In addition, the authors propose some linear functions for modeling the technicalaspects of QoE and confirmed that the emotional satisfaction was the major QoE aspects for the asked users [SK13].

1.3.3 QoE measurement in the context of video streaming

From the discussion above, we can say that modelling and monitoring QoE is a tough task due to the various factors affecting QoE as well as methods of collecting data especially in video streaming services. According to [SK13], the main challenges can be summarized in this following four questions: (i) What to collect?; (ii) Where to collect?; (iii) When to collect?; and (iv) How to collect?

Firstly, we should determine data to gather. The what or which data to gather is determined by the QoE metrics list which directly depends on the service or product type and context. The decision regarding data to collect is the starting point for any QoE monitoring or measurement tool, approach or strategy [Sta+11]. Secondly, we should choose a location where to gather data is also a major issue in the QoE evaluation process. The QoE assessement may include different data collected from the base stations, access or core networks, or from the service servers, content, or cloud. In adittion, the obtained parameters may be derived from both side: application or network level, or a combination of them. Moreover the collected data needs, generally, to be sent to a QoE system able to make optimal decisions for QoE parameters. Consequently, the transfer of data to a QoE enforcing entity needs to be treated. Thirdly, we should determine when to collect data: (i) before the product is developed; (ii) after the product is developed and not delivered; or (iii) after the product is delivered. Additionally, the frequency of monitoring and measuring data needs to be

considered. Finally, the where and when above problematics determine how to acquire data. The QoE monitoring approach and tools requires computational operations. Consequently, computational degree of complexity and terminal battery life need to be taken into account in all this process [SK13].

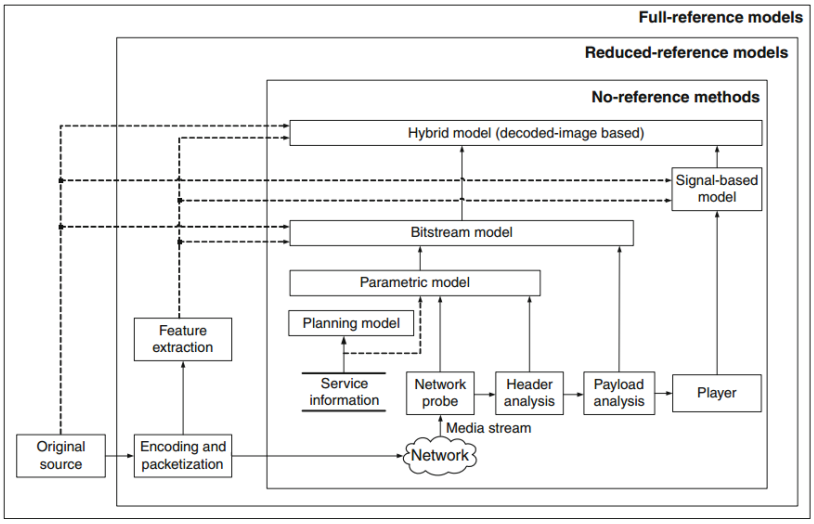


Fig. 1.5: Categorization of video QoE evaluatin algorithms [MR14].

These questions allow us to review the problem in a different way. In fact, to assess and monitor the QoE, algorithms and technics can be categorized based on the type and amount of data and information gathered. The quality models can be categorized in terms of [MR14] p. 278:

- the amount of original information they use: No-Reference (NR)(i.e the model can not have use the original information; Reduced-Reference (RR) (i.e the model can use information extracted from the original one, box "Feature extraction" in Figure 1.5)); and Full-Reference (FR)(i.e the model can use the original information ("Original source" in Figure 1.5)),

- the type of information used to predict the quality: Signals ("signal-based model" in figure 1.5) and/or transmission-related indicators ("Parametric model" in Figure 1.5) and/or bitstream information ("Bitstream model" in Figure 1.5).

- the indicator in which they take into account the explicit modeling of the user visual system.

In order to deliver video data, there are two possibilities. The first one is to use HTTP to distribute video chunks and use TCP for controlling intermediate congestion. Another possibility is to use an RTP container over UDP where congestion detection is done by exploiting RTCP messages.

In the next section, we will discuss the video quality models for streaming in two cases:(i) with unreliable transport mechanisms (i.e User Datagram Protocol (UDP)) , and (ii) with reliable transport mechanisms (i.e Transmission Control Protocol (TCP)) [MR14].

1.3.4 Video Quality models with unreliable transport mechanisms

This section summarizes the video measurements and degradations that are encountered in the case of streaming with unreliable transport. Various models types are presented: packet-header-based models for network planning and monitoring, bitstream-based models, pixel-based models and hybrid models [MR14] p.280.

Coding and Packet Error Degradations for video streaming

While encoding a video, three types of frames may be used: "I-frames," "P-frames" and "B-frames". The impact of packet loss on user perceived quality depends on the type of the frame in which the loss happens. On the one hand, we can predict the P-frames and B-frames from previous I-frames and P-frames. On the other hand, wa cannot predicte I-frames because they are an intra-coded frames and so do not rely on other frames. Therfore, when a loss occurs on an I-frame or a P-frame, it spreads till the next I-frame. Otherwise, when a loss occurs on a B-frame, the loss spreads till the next I-frame or P-frame[MR14] p.280.

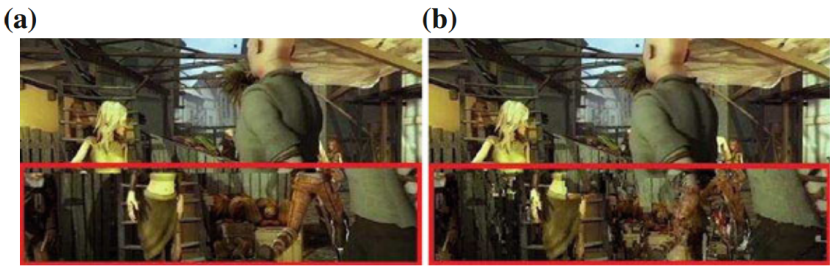


Fig. 1.6: Impact of packet loss when one slice per frame is used. a Loss occurred in the current frame. b Loss propagated from previous frames [MR14]p.281.

In addition, the impact of packet loss on perceived quality depends on the concealment of the packet loss used by the decoder. One packet loss may cause the loss of all the corresponding area and also the pixel area of the affected slice if we applied slicing as packet loss concealment (see red rectangle in Figure 1.6).

Also, The packet loss may yield to freezes images (see the red rectangle area of figure 1.6). In fact, when loss occurs, content from the previous frame is spread to next frames when loss occurs and so a blocking artifact.

Bitstream Based Models for predicting video quality

The objective of bitstream-based models is to predict quality of a video using the encoded video while transmitted over the network (see Figure 1.6). These models analyses the video bitstream without the reconstruction of the pixel information. These bitstream models are interesting and

offer the possibility to monitor the video quality at any point on the network (network-based monitoring) [MR14]p.283.

Analyzing the encoded video on the packet headers allows us to get additional information at the frame level. This give us some first indication and characteristics of the quality of the video (see [MR14]page 283 for more details).

In this section, we gave examples of models for QoE video in the case of an unreliable protocol (they use an RTP container over UDP where congestion detection is done by exploiting RTCP messages). For more details on the models discussed, you can refer to [MR14]chapter 19. The next section discusses the video QoE models for streaming in the case of reliable transport mechanisms, which is the dominant case in the Internet and the scenario considered in this thesis.

1.3.5 Video Quality models with reliable transport mechanisms

In this section, we present two models of video streaming over HTTP: Progressive Download and Adaptive Streaming Models.

Progressive Download Models

In this model, the content provider will encode its video and distribute it with a normal web server. Customers need to know the URL of the video file to watch the video, and could start watching a video while the the video is being downloaded.

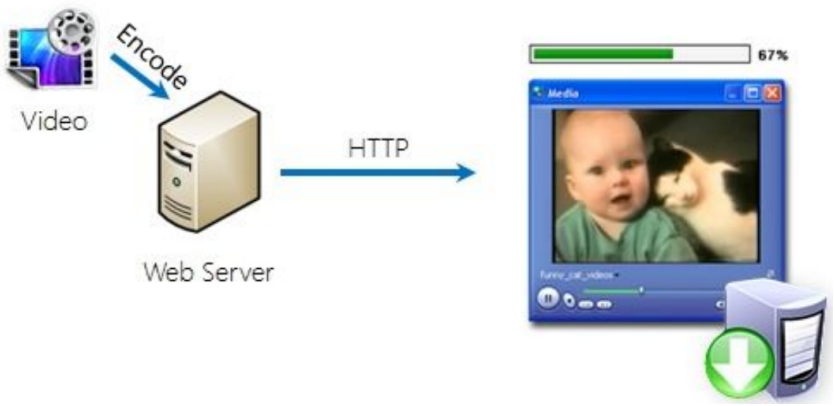


Fig. 1.7: Progressive Download Model.

In [26], authors propose a model for rebuffering[5] degradations for an audiovisual QoE. The model takes as inputs the number of starvation[6] events and the average length of a single starvation event.

[5]This is a phenomenon that occurs just after a video freeze. In fact, when a video freezes, user waits a few seconds, to accumulate/buffer content in its buffer, before seeing the video playback restart

[6]It refers to video freeze. When the played data reaches the downloaded data, playback stops. This phenomenon is know in literature by "starvation phenomenon"

The model is based on the idea of "different resolution for a video sequence". The video durations were on average 5.6 minutes and up to 15 minutes (i.e Youtube videos). Another model has been proposed in the ITU-T P.1202.1 Recommendation [P.112]. It captures the impact of rebuffering degradations on the quality for low video resolutions (up to HVGA). The base idea of this model is the ratio between the rebuffering duration and the total sequence duration (i.e it includes the rebuffering duration). The last model in this subsection is ITU-T P.1202.1 model. It has been developed based on subjective laboratory tests results. The model was tested and validated on sequences of 16 s and up to 30 s of video. We can use the same model for longer sequences [MR14].

Adaptative Streaming Models

Those second models are proposed to overcome the phenomenon of rebuffering (wait before an amount of data has been downloaded after a starvation event) and ensure smoother flow video. It is based on breaking the video content into segments of small durations. Each of these video segments is made available at different bitrate and the client selects the highest segment, in terms of quality, depending on network conditions (see Figure 1.8). In fact, the adaptative client periodically requests a video sequence from the HTTP server, and then decodes and displays this sequence. Consequently, the client may change quality from between sequences depending on the available bandwidth. The only problem is that quality may change during the video.

In the previous section, we present some models estimating the QoE of video streaming in the context of laboratory testing. The viewing environment is different from real viewing conditions. Consequently, more aspects need to be taken into account to monitor QoE and more good assessment for QoE.

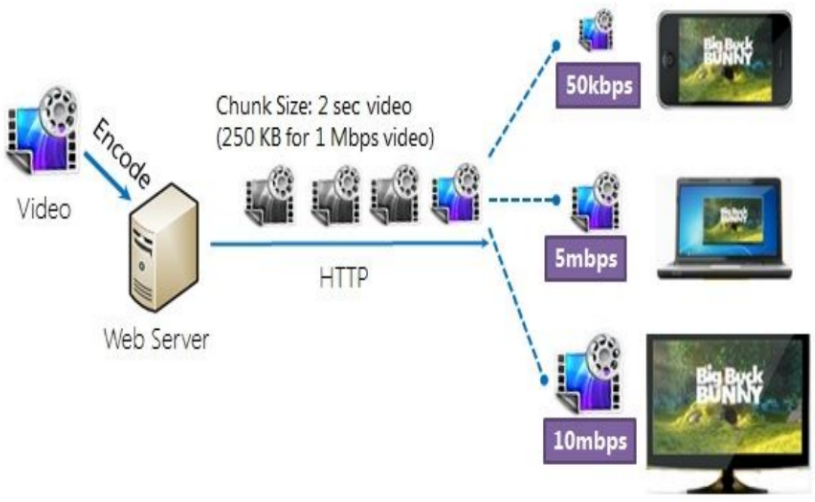


Fig. 1.8: HTTP Based Adaptative Streaming.

Next, we will see how streaming traffic is delivered from a service provider (i.e Youtube, Dailymotion...) to customer terminal (i.e smartphone, tablet..) before outlining our thesis plan.

1.3.6 Streaming strategies

Delivering HTTP streaming traffic to the end-user is an arduous challenge for operators as they must (i) ensure that viewers watch the video without buffering and at the highest picture quality possible (and so tracking the real user experience) and (ii) anticipate that user may abandon viewing early and so streaming the content gradually to the application buffer at the terminal device. These two challenges are contradictory as the former calls for an as fast as possible transfer to the end user while the latter pushes the content provider and the mobile operator to deliver at a given time instant only a bit more than what the user current viewing process requires. These challenge are reflected in the streaming strategies observable on the market, where traffic delivery might be controlled either by the server, the client or a mix of both. Popular strategies are [Hoq+]:

- Fast caching: the server delivers as fast as possible the video to the client;
- Rate throttling: the server starts with a period during which data is delivered as fast possible followed by a period where rate is capped to a limit above the playback rate;
- On-Off: the server pushes data as fast as possible to the client that regularly pauses, by emitting TCP zero advertised window or aborting the TCP connection, when its buffer is full enough and resumes transfer when the playback process has drained the buffer to a low value.

Next, we will restrict and detail the Fast-caching and On-Off strategies as they constitute two extreme cases in terms of QoE. To put it differently, if On-Off can offer similar QoE to Fast-caching, it can can also compete with Rate-throttling.

Figure 1.9 shows the variation of buffered data in seconds as a function of the playback time in seconds for different terminal devices and service providers.

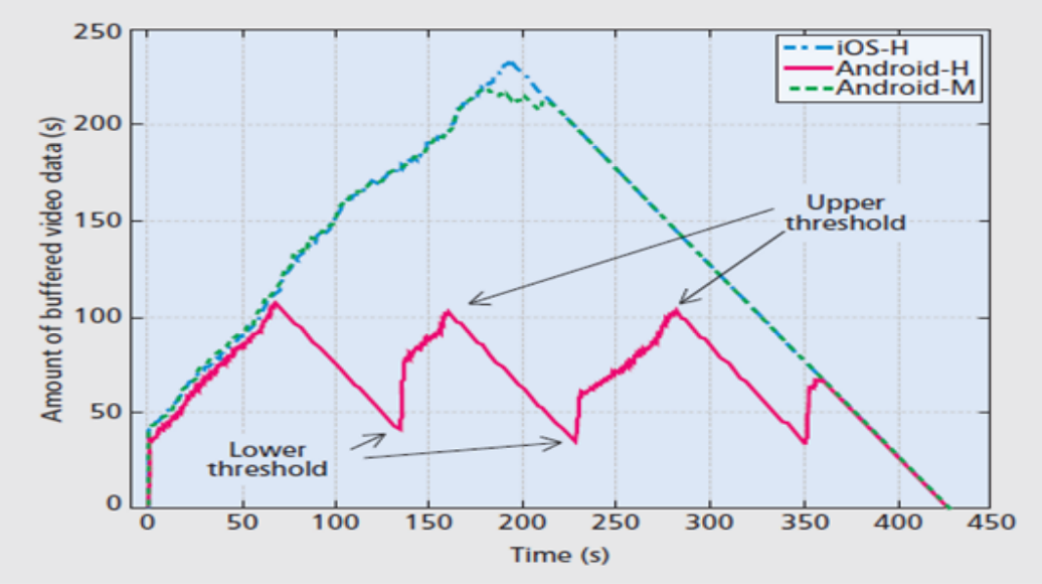


Fig. 1.9: Downloading strategies by the YouTube player for different types of terminals [RM+14].

Fast Caching streaming strategy

One of the most popular streaming strategy is Fast Caching strategy. In the beginning of this streaming strategy session, the terminal client prefetches content in an attempt to rapidly fill the playback buffer and reach the so-called start-up delay (i.e the duration between the time that a user initiates a session and the time that the playback start, in seconds). In the literature, this pre-phase is known as the prefetching period: it is commun to all downloading strategies. After this prefetching phase, it is the available bandwidth which defines at which rate the data is transmitted since the server sends the data at full available bandwidth. In other words, the video server delivers as fast as possible the video to the client and the client player reads data continuously from the receiver buffer. The downloading of the video continues during the viewing phase until the video has been fully downloaded. This downloading strategy requires no complex engineering at the server and the client.

On-Off streaming strategy

The second most popular streaming strategy is the on-off strategy. After a prefetching phase, the server starts to push data as fast as possible to the client that regularly pauses, by emitting TCP zero advertised window or aborting the TCP connection, when its buffer is full enough and resumes transfer when the playback process has drained the buffer to a low value. In fact, this strategy consists in maintaining the level of the buffered data (i.e downloaded and unread data) between a lower threshold and a high or upper threshold. A user starts the OFF phase when the total size of the unread data in the buffer exceeds the high threshold and the download pauses. When the total data size drops below a low threshold, the download resumes [Liu+] and the OFF phase ends. The difference between the time the download pauses and the time the download resumes is the OFF period duration. The ON period starts while the buffer reaches the lower threshold by resuming the download of the video and ends when the buffer reaches the high threshold. In Figure 1.9, we can see that the lower threshold is around 40 seconds of buffered data and the high/upper threshold is around 105 seconds.

The purpose of such buffer management is to prevent starvation situations, while avoiding unnecessarily filling the buffer if the client decides to abandon video playback. So the question is: is it the client or the server that decides on the values of low and high thresholds? and, what is the optimal values of low and high thresholds in order to ensures the best quality? To answer these questions, the authors in [Liu+] have studied the streaming strategies employed for HTTP streaming in the Internet. Among these strategies, we find the on-off strategy for Android terminals. They observed that the values of low and high thresholds may change from an Android terminal to another. In addition, they show that the values of these two thresholds remain constant during a streaming session.

1.4 Thesis outline

The rest of the thesis manuscript is organized as follows.

Chapter 2 In this chapter, we focus more on literature and related works. We first talk about the different works that attempt to model and optimize the QoE for streaming video services. Especially, we present the QoE metrics for streaming services. As this thesis focuses on QoE for streaming services delivery, we then present our first attempts to understand what happens while using FC or On-Off streaming strategies via a simulator that we developed.

Chapter 3 In this chapter, we take a mobile operator perspective and focus on the QoE when using fast-caching or on-off strategies. We develop an analytical model that takes into account users dynamics and allows us to derive the probability of starvation (frozen video) that the client may experience. We show that the QoE of the end user is equivalent under the fast-caching and on-off strategies and we explore the trade-off between two dimensions of QoE in the on-off case: the initial startup delay and the starvation probability.

Chapter 4 In this chapter, we develop an analytical model which takes into account users dynamics and allows us to quantify the loss in bytes due to users' viewing abandonment. Furthermore, we formulate a multi-objective optimization problem to find ON and OFF period durations that feature a good trade-off between loss due to abandonment and starvation probability.

Chapter 5 In this chapter, we tackle a linear modeling to seek the optimal delivery streaming data to a set of users connected to a base station under a users dynamics environment. Here the goal is more to propose a technical solution and a good algorithm that improve the general QoE for all users.

Chapter 6 In this chapter, some conclusions and the future works and challenges of the presented works are discussed.

1.5 Conclusion

In the context of a highly competitive telecom market, satisfying clients QoE has become a major goal of operators. The streaming video market is becoming increasingly complex, requiring a crucial and permanent effort from the providers of video content and the operators to ensure and maintain a good QoE for their users anywhere anytime, over any context, technology, network and across multiple mobile operator. Research in the field of video streaming is rich of models addressing issues related to QoE. In the next chapter, we present the related works and some experiments we developed in order to understand the QoE in the context of streaming video services. Especially, we develop a simulator to try to obtain preliminary numerical answers to our two main thesis questions (i) Does On-Off strategy affect and reduce the QoE performance? and (ii) Is there an optimal parametrization between the different QoE streaming service indicators ensuring good QoE?

1.6 Introduction (French)

La prochaine génération des réseaux mobiles a été déployée pour transporter et assurer une qualité élevée lors de la transmission des données multimédias des fournisseurs de services aux clients. Le but des opérateurs est de transmettre les données informatiques de manière optimale tout en

assurant une bonne perception de l'utilisateur. Leur objectif est de fournir des services réseau que les clients souhaitent utiliser de la manière la plus rentable et la plus efficace et en assurant une expérience positive quand à l'utilisateur finale. Cette relation entre le coût, l'efficacité des ressources et la perception par l'utilisateur d'un système ou d'un service est encore relativement mal comprise. Les opérateurs et les fournisseurs sont entrain de changer leur vision et se concentrent plus sur la satisfaction et la qualité perçue par l'utilisateur finale, qui est la qualité du réseau du point de vue de l'utilisateur, que sur la qualité de service (QdS) liée au réseau d'accès. Une bonne et haute qualité de bout en bout devient la plus haute exigence du client qui doit être garantie par les opérateurs de télécommunication et les fournisseurs de services. C'est ce que nous appelons la qualité de l'expérience (QdE). Elle est couramment utilisée pour mesurer la satisfaction du client. La QdE n'est pas seulement un indicateur du client, mais un concept d'évaluation de la qualité des services réseau de de point de vue l'utilisateur final, afin d'assurer une bonne qualité sur l'ensemble de la chaîne de communication (par exemple, fournisseur de services-informations réseau-client). En fait, les opérateurs et les fournisseurs de services doivent améliorer et assurer un niveau satisfaisant de la QdE envers leurs clients. Pour atteindre cet objectif, ils ont rencontré des défis tels que la mesure et le contrôle de cette qualité perçue par l'utilisateur.

Au cours des années précédentes, les opérateurs de télécommunication ont utilisé la notion de qualité de service comme un concept pour améliorer leurs capacités à envoyer des données sur des réseaux d'accès de manière efficace. Leurs stratégies de transmission des données étaient toujours axées sur l'interconnexion des différents éléments et équipements du réseau informatique, les capacités de ces éléments de connexion et la maintenance. Cette approche permet aux opérateurs de déterminer un chemin pour chaque flux d'information d'une manière efficace vis à vis des ressources disponibles. Les paramètres de la QdS étaient le retard de bout en bout, le taux de perte et la bande passante. Pour obtenir une bonne performance, un système de transmission basé sur la QdS peut également essayer d'optimiser d'autres critères tels que l'utilisation du réseau et la charge transportée. Ces critères peuvent influencer la stratégie de transmission. Toutefois, il n'est pas simple de satisfaire toutes les exigences de l'utilisateur. Par conséquent, la plupart des stratégies et des approches étaient axées sur le réseau et ne considéraient que l'accord au niveau service entre les opérateurs et les fournisseurs de services. Cependant et avec l'émergence de nouveaux types de contenu sur le Web et surtout du contenu multimédia de haute qualité, cette perspective n'est plus optimale. Les utilisateurs regardent et téléchargent des vidéos de haute qualité et diffusent leurs propres vidéos sur Internet. Ces flux de données sensibles à la QdS imposent des contraintes strictes sur le système de transmission qui sont beaucoup plus difficiles que les accords entre opérateurs et fournisseurs de services. Ainsi, l'intérêt du fournisseur de services et des opérateurs en fournissant une transmission satisfaisante des flux de données pour atteindre l'exigence de l'utilisateur, devient plus important que simplement considérer les paramètres du réseau et la maintenance.

Par conséquent, la QdE est devenue une mesure clé pour les opérateurs. L'Union internationale des télécommunications (UIT) définit la QdE comme l'acceptabilité globale d'une application ou d'un service, tel que perçu subjectivement par l'utilisateur final. Il s'agit d'une mesure de la qualité de satisfaction d'un utilisateur ainsi que toutes les performances et à la qualité du système de bout en bout. La QdE inclut des facteurs d'influence tels que l'état du client (par exemple heureux, motivé, prédisposé ...), l'ergonomie du système (par exemple, la fonctionnalité, la complexité ...) et le contexte ou l'environnement dans lequel le service sera expérimenté.

Compte tenu du contexte ci-dessus, nous nous concentrons, dans cette thèse, sur la façon de traiter ce concept de qualité d'expérience. Plus précisément, dans le contexte des services multimédias et surtout le streaming des vidéos, nous aborderons le problème de la QdE pour ces services de streaming des vidéos.

Le reste du manuscrit de thèse est organisé comme suit:

Chapitre ?? Dans ce chapitre, nous nous concentrons davantage sur la littérature et les travaux connexes. Nous discutons d'abord les différents travaux qui tentent de modéliser et d'optimiser la QdE pour les services du streaming de la vidéo. Particulièrement, nous présentons les métriques de la QdE pour les services de streaming. Comme cette thèse se concentre plus sur la QdE pour la transmission des données du streaming, nous présentons nos premières tentatives pour comprendre ce qui se passe lors de l'utilisation de stratégies de streaming "Fast Caching" ou "On-Off" via un simulateur que nous avons développé.

Chapitre 3 Dans ce chapitre, nous adoptons une perspective d'opérateur mobile et nous concentrons sur la QdE lors de l'utilisation de stratégies de streaming "Fast Caching" ou "On-Off". Nous développons un modèle analytique qui prend en compte la dynamique des utilisateurs et nous permet de dériver la probabilité de famine (vidéo gelée) que le client peut expérimenter. Nous montrons que la QdE de l'utilisateur final est équivalente dans les deux stratégies et nous explorons le compromis entre deux dimensions de la QdE dans le cas de la stratégie on-off: le délai de démarrage initial et la probabilité de famine.

Chapitre 4 Dans ce chapitre, nous développons un modèle analytique qui prend en compte la dynamique des utilisateurs et nous permet de quantifier la perte en octets due à l'abandon du téléchargement d'une vidéo par un utilisateur. En outre, nous formulons un problème d'optimisation multi-objectifs pour trouver les durées des périodes ON et OFF qui présentent un bon compromis entre les pertes due de l'abandon et de la probabilité de famine.

Chapitre 5 Dans ce chapitre, nous abordons une modélisation linéaire pour trouver la stratégie optimale de transmission des données pour un ensemble d'utilisateurs dynamiques connectés à une station de base. Ici, l'objectif est de proposer une solution technique et un bon algorithme qui améliore la QdE générale pour tout un ensemble d'utilisateurs.

Chapitre 6 Dans ce chapitre, nous discutons quelques conclusions ainsi que quelques travaux futurs et les défis des travaux présentés.

2

State the of art

> *Mathematicians may flatter themselves that they possess new ideas which mere human language is as yet unable to express.*
>
> — **James Clerk Maxwell**
> (Scientist)

In this chapter, I shall expose analysis of general works related to the QoE especially for streaming services. I will detail more the problematic and the closest with respect to our work. Finally, we will present results of some experimentations throw a simulator to prepare and introduce our QoE models presented in the next chapters and the solving approaches.

In the past years, quality requirements have increased a lot due to the arrival of new internet services such as the video on demand and streaming services. This led telecommunication operators to work more on understanding and improving the quality perceived by the end user. Generally, we qualify the quality of a service as "unacceptable" when this service is interrupted at the time of its use by the customer. For example, a quality of a streaming service is unacceptable when the stream freezes during playback. It is obvious that service interruptions is directly related to the routing of data (i.e packet loss, jitter,...). So, the question that arises is how to route data, from a service provider to the end user, in a way to avoid service interruptions? Contrary to what we may believe, the interruption of a service can come from the architecture of the network as well as from the properties of the terminal device or the services providers. In fact, both video providers and terminal devices impact the sending or downloading strategy. The Youtube server with iPhone 4S adopts the encoding rate[1] strategy while the same video server adopts the Fast Caching[2] strategy with Samsung Galaxy [Hoq+12]. Consequently, server of the provider and the terminal properties define the downloading strategy under which data will be delivered through the network. Operators try this to route data by maximizing the routing quality. The quality of a service can be seen as the global quality achieved by either terminal equipments, service provides and telecommunication operators. However, the quality can be assessed differently depending on our positioning in the communication chain (i.e service provider, customer or operator). Figure 2.1 shows the different perspectives from which we can view the quality. The end user may be happy if the proposed service is easy to use with good quality transfer and robustness. The service provider looks for ensuring different proposed qualities in order to satisfy end user under the network conditions and properties deployed by operator. Operators try to route data efficiently while satisfying user and service providers requirements.

[1] Streaming strategy in which the player reads data at a rate equal to the encoding server rate to send the data
[2] the server sends the data at full available bandwidth and the player reads data continuously from the receiver buffer

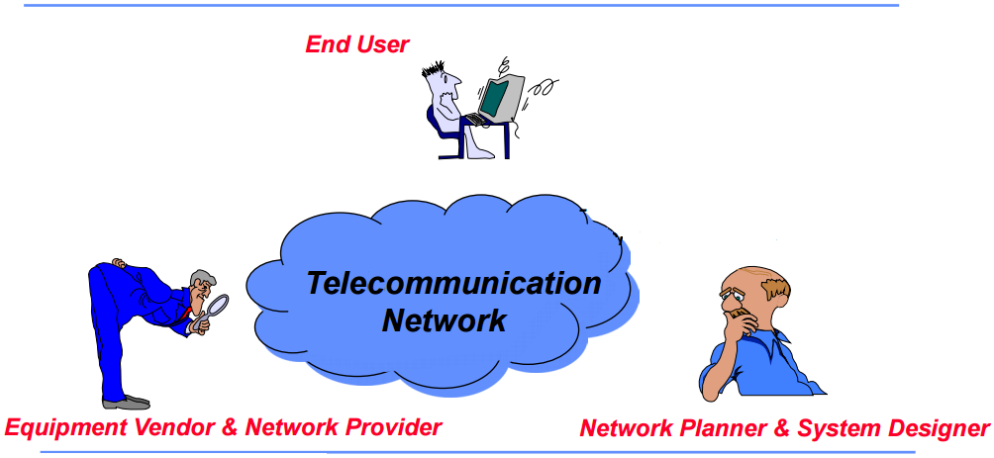


Fig. 2.1: Different perspectives of network quality.

Next in this chapter, we will highlight works related to the QoE especially for video streaming service. We will then present some preliminary experiments in order to understand the QoE for streaming video. Finally, we propose a section which will give a glimpse on what we will propose later in this manuscript.

2.1 Models for computing the QoE

This section reviews the work that attempted to formulate a mathematical expression of the QoE. We present some explicit formulas derived in literature, and analyze the factors taken into account in each model to compute the QoE.

In [Kim+08], the authors use traditional QoS parameters to measure the QoE. They first describe the correlation model between the QoE and the QoS and then, the authors studied the evaluation method of the QoE using QoS parameters. These parameters include jitter, delay, error rat, loss rate, signal success rate and bandwidth. The expression of the correlation between the QoE and the QoS is computed as follows:

$$QoE(QoS) = K(\frac{(e^{QoS-\alpha} + ep^{-QoS+\alpha})}{(e^{QoS-\alpha} + ep^{-QoS+\alpha} + \beta)} + 1) \tag{2.1}$$

With α the quality class of QoS of the network level. The QoE value measured is mapped to an existing Mean Opinion Score (MOS) grade with a scale of five points. The parameter β is computed according to the class of service. K is a constant vector mapping the user satisfaction with the used service. Finally, the authors explain how to use the QoS measured informations (i.e jitter, delay, error rat, loss rate, signal success rate and bandwidth) to extract the objective QoE using their model in Equation 2.1.

[Gon+09] proposes a model to measure the QoE based on five factors which are: retainability, usability, integrality, availability and instantaneousness. The authors presented an experience model

to evaluate the quality, by taking into account these factors. They use the model in a VoIP service. The important metrics were as follows:

- retainability: service interruption ration noted b;
- usability: usability of a service noted d;
- integrality: packet loss, jitter and delay noted a;
- availability: success ratio to access a service noted c;
- instantaneousness: the response time to access a service noted e.

To measure the QoE, the authors proposed the following formula:

$$QoE = \frac{1}{2} sin(\lambda)(ab + bc + cd + de + ca) \tag{2.2}$$

With λ a constant equal to 72 degree angle. The proposed model facilitate the measurement of the QoE while using a service in order to improve the QoE perceived by a VoIP service. In another approach, the authors in [MOE 08] tend to better evaluate the QoE using the correlation between QoE and QoS. They propose an adaptive learning model divided in three sub models:

- Model of user: encompasses user characteristics namely location, device preferences and learning goal;
- Model of a domain: concept of mapping data, semantic networks or concept graphs.
- Model of adaptation: this component connects the two previous cited models by using adaptive rules.

  The main elements of the QoE are the flow related aspects and the learning related aspects. Let QoL be the Quality of Learning (i.e learning strategy, feedback)and QoF the Quality of Flow (i.e interaction, technology used and emotions). The QoE is expressed as follows:

$$QoE = f(QoL(QoS), QoF(QoS)) \tag{2.3}$$

Here, the QoS is basically delay, jitter, packet loss. Hamam et al. [Ham+08] proposed a mathematical formula to compute the QoE. This formula is a weighted linear combination of the QoS and user experience (UE) where The UE is a linear combination of some user related metrics. The QoE is expressed as follows:

$$QoE = \eta.QoS + (1 - \eta).UE \tag{2.4}$$

With η is between 0 and 1 and controls the weight given to the QoS compared to the UE parameters.

In this section, we have presented different approaches and models to compute the QoE. These models are a linear combination of objective metrics (i.e QoS) and the correlation between the QoE and the QoS. However, the subjective factors have not been considered. It is obvious that that a QoE model should take also into account user's satisfaction. For example, the QoE while watching a video should be evaluated taking into account subjective aspect. To remedy this problem, researchers tend to change the method to estimate or calculate the QoE. In fact, instead of seeking the expression of the QoE, they try to assess and evaluate indicators of this QoE. Using an indicator to measure quality seems to be a good approach, especially if the indicator directly reflects a measure that the client may experience. It is certain that the indicator the QoE depends on the type of service. For example, in streaming video, no one can deny that freezes can be the best indicator to measure the quality experienced by the end user.

Before presenting the key works which were the starting points of our approaches and models presented in the following sections, we will survey the impact of user mobility, network resource management and pricing on the QoE.

2.2 QoE and network criteria

In this section we analyze the relation between the QoE and some network criteria that conducts our work. It is not easy to explain how network criteria impact the QoE. A lot of topics can be addressed. First, we talk about user's dynamics and QoE and then the network resource management. Finally, we present some approaches that link QoE and pricing.

2.2.1 QoE and user's dynamic

Mobility describes the movement of mobile users, and how their location change over time. The user's dynamic and mobility is an important issue that impacts the QoE. In fact, to establish a communication between, for example, a client and service provider, it is necessary to know where the client is located. When the client moves, there must be continuity of communication when she passes from one cell to another (commonly called handover).

In [Ber+08], the authors propose methods to evaluate the QoE under different mobility management strategy. They propose a series of test enabling the evaluation of the QoE in an heterogeneous wireless access network. This test reproduces a 3G heterogeneous network consisting of three radio access networks: GSM EDGE radio access network (GERAN), WLAN interfacing based on DiffServ and MPLS and universal terrestrial radio access network (UTRAN). User terminals are connected to routers. Authors test three types of handover:

- Intra ingress routers with vertical handover, where the handover is performed between base stations of different radio access network;

- Inter ingress routers with vertical handover in which the mobility management plays the major role with an ingress router change.

- Horizontal handover. The handover is made on the same RAN between base stations.

This tests led to depict the QoE of a user under test by MOS values. Authors map after this values to obtain the level of satisfaction, based on an objective metric model. On the other side in [CM09], the authors propose an algorithm, called smooth adaptive soft handover algorithm (SASHA), to handover with load balancing on different types of networks, using an analytic function for decision-making. Their study was focused on an heterogeneous network where feature terminals uses different access technologies. In order to maximize the QoE under users mobility, an efficient resource allocation method is required. The authors propose a Multimedia Mobility Management System (M3S) to route high quality multimedia data to end users at the application level. Consequently, M3S provides handover management and efficient resource allocation using the algorithm SASHA. SHASHA performs handover by transferring the load between different networks. The authors use a Multimedia streaming metric to quantify the impact of separated networks to the quality. This metric is expressed as follows:

$$QMS^i = w_1.QoS^i_{grade} + w_2.QoE^i_{grade} + w_3.Cost^i_{grade} + w_4.Peff^i_{grade} + w_5.UPreff^i_{grade} \tag{2.5}$$

Where i is the communication channel; $UPreff^i_{grade}$ evaluates the user's preference grade for the network used by channel i; QoE^i_{grade} evaluates the quality of video content perceived by user i. The quality grade for every channel i, QoE^i_{grade}, is determined by sharing the overall QoE according to the rate share of a channel and is expressed as follows:

$$QoE^i_{grade} = QoE_{overall} * RateShare^i \tag{2.6}$$

This approach aims to maximize the perceived quality while streaming multimedia content by deploying all available resources efficiently. M3S uses SASHA to dispatch the load over channels in order to deliver high quality multimedia data.

The two approaches detailed above show the impact of user mobility on the quality perceived by the end user. In the next section, we discuss the impact of resource management on the QoE.

2.2.2 QoE and resource management

Good resource management of the network is the best way to ensure and maintain the system quality. It has also a lot of impact on end user quality. In this section we present two approaches aiming to maximizing the QoE.

In [AL08], the authors present an approach aiming at maximizing the user perceived quality while minimizing network resources. Using this QoE management methodology, operators can predict and then anticipate the user experience and consequently allocate resources in an optimal manner.

To do that, the authors use a statistical model to predict qualitative indicators of a user from known quantitative indicators. The function used in the model is expressed as follows:

$$f_{km} = u_0 + u_1 X_{1km} + u_2 X_{2km} + ... + u_p X_{pkm} \tag{2.7}$$

With f_{km} is the predicted score for group k in case m; X_{ikm} is the value of predictors X_i for group k in case m; and u_i is coefficient for variable i.

In [Yam+07], the authors present a system for managing network resources to facilitate the control of resource access. This system is based on QoE. In fact, the authors propose a so called network resource management system (NetRM) that distributes and manages the network resources in order to satisfy the service request. In fact, for fixed mobile convergence (FMC) networks, the authors provide a service control scheme for RACF (Resource and Admission Control Functions) based on the end-user QoE predicted by the per-segment based observation method of the network QoS metrics using RTP/RTCP.

In [Az+14], Hatem Abou-zeid and al. present an energy-efficient optimization algorithm that aims at: 1) Minimizing the transmission requirements without causing streaming freezes; 2) minimize the power consumption of a Base Station (BS) by switching off the BS in deep sleep. They use the mixed-integer linear programming to optimize resource management in order to optimize the QoE.

In this section, we presented some literature works illustrating how we can manage network resources in order to maximize the QoE. The main difference between these three approaches is the function used to manage network resources. In each case, the authors implement an adaptive approach to allocate resources while satisfying network requirements. Next, we will see how the price of a service impacts the QoE.

2.2.3 QoE and pricing

In this section, we will investigate the relationship between QoE and pricing or billing. In other words, we discuss how pricing can impact the QoE. It is obvious that the price of a service can also be considered as a QoE parameter. To prove that, Takahashi and al. [Tak+08] studied an IPTV application while focusing on QoE and considering the billing of the proposed service. In fact, the QoE was presented as the satisfaction of the client requirements and expectations. Authors found that billing has a big impact on the expected quality by customers. In their works, they use the channel zapping time to measure user's satisfaction. For users who pay a basic IPTV package, accept a long channel zapping time while the premium IPTV (i.e more expensive package) clients were dissatisfied with the same channel zapping time. Consequently, the client's satisfaction depends strongly on the service billing and price. The authors propose a function to compute user's utility, noted U_s which is expressed as follows:

$$U_s = f(T_s, B_s) \tag{2.8}$$

Where T_s is a service quality metric (i.e startup time, channel zapping delay, picture quality...), and B_s is the service billing. f is is a weighted function of T_s and B_s. Thus, every client may have different user's utility, and so different level of satisfaction, even if they may have the same service characteristics and quality: depending on the price they pay.

In this section, we analyzed the relation between the QoE and network criteria or pricing. We saw how user's dynamic, resource management and pricing impact the QoE. These two first criteria are primary questions in this thesis. In fact, in streaming video service, all is about how much data user will receive. This data is directly attached to user's dynamics and resource management. In the next section, we will detail some works that constitute starting points to what we are presenting in this thesis.

2.3 Inspiring works and first experiments

In this section, we discuss two works that are closed to our work. In fact, we explain the techniques and methodology used in each one of these two works in details. After that, we present some first experiments that we run in order to understand, motivate and position our work. We finally present our simulator that reflects real client behavior and will be the reference to our models presented in the next chapters.

2.3.1 Modeling user video's QoE in literature

The quality of experience in streaming video services can, first, be summarized with two indicators:

- Video freeze: during the video playback, if the video hangs, it is said that there is a video freeze. In the literature, this phenomenon is known as video starvation.

- Start-up delay: the duration between the time that a user initiates a session and the time that the playback starts, in seconds.

The problem that deteriorates these two indicators is the lack of streaming data received by the terminal user equipment. In fact, let us assume that a user requests a video and plays it with a bitrate equal to x bits per second. it is obvious that if this user receives x+ϵ bits per second, she would never experience a video starvation. Consequently, all is about how much data a user receives per second. To this end, there is two approaches to tackle this problem of modeling: Packet level approach and flow level approach. In the next sections, we will analyze these two modeling techniques in order to compute QoE indicators for streaming video services.

Modeling QoE at packet level

In this approach, we are interested in a particular user who receives packets of streaming data sent by a server provider and routed through the access network. We assume that the buffer size is finite. Here in this packet level approach, the key assumption is the following:

- Packets arrive in the user terminal buffer following a predefined distribution or strategy, and the service rate of packets is also assumed to follow a predefined distribution.

Generally, we assume that packets arrive in the user terminal buffer following a Poisson process with intensity λ. The service rate of packets is assumed to be exponential with parameter γ. In [Xu+12], the authors model the buffer as an M/M/1 queue. The video consists of N consecutive arrivals of packets. The buffer is assumed to have a constant size denoted K. The goal is then to compute the distribution of the number of buffer starvations. When the client requests a video, the prefetching phase starts. During this phase, the user waits till she accumulates x bits of data [Xu+12], (i.e start-up threshold[3]) equivalent to start up delay. After this phase, the playback of streaming packets starts. When the terminal buffer is empty, starvation occurs, and the playback stops. The user enters then in a phase called the rebuffering phase. During this phase, the user waits until the received buffer accumulates a quantity x_1 of data and the packets are not played. In summary, different phases of this modeling approach are:

- the prefetching phase: terminal user device starts receiving packets following an arrival Poison process and the playback is not started yet;
- the playback phase: after the accumulation of start-up threshold packets, the playback starts;
- the starvation phenomenon: when there is no more data to read from the received buffer, the playback stops (i.e the video freeze);
- the rebuffering phase: the user restarts to accumulate data to reach the rebuuffering threshold x_1 without playing packets received during this phase;
- the end of video: the end of the connection is when the terminal receives all the N packets.

In other cases, the end of a video can be triggered by user if she decide to quit video during playback.

The prefetching phase is annoying to the end user. In fact, this phase should not be long so client can be patient and wait till the playback starts. The goal of [Xu+12] is then to optimize the QoE of streaming service, by analyzing the tradeoff between the start-up threshold and the starvation phenomenon.

In the next section, we present the flow level approach from literature in order to model the QoE for streaming video services.

[3] Quantity of data or packets accumulated during prefetching phase

Modeling QoE at flow level

Let C be the capacity of a base station. This capacity may depend on several network parameters. For example, the capacity C may depend on the distance between the end user and the base station. In this flow approach, the idea is to distribute the bandwidth of this base station on a group of users connected to the same base station on a an homogeneous cellular network. Consequently, we should have the information about users number and location to be able to route data to this end users. So, the starting point and big assumption on this approach is to model the dynamic of users. In fact, when the dynamic of users is fixed, we will be able to compute the throughput of each users at any time and then know how much data every user receive per second. Once known, we can see if users experience video starvations knowing their playback speed.

In [Xu+13], a work from which we are inspired, the authors start their modeling by setting a user's dynamic. First, they assume that the arrival of a user is following a Poisson process with intensity λ. They consider a wireless cellular network that can manage at most K simultaneously clients. They assume that all flows have a SNR and throughput. Users have the same playback speed called the Bitrate. Finally, they assume that video duration of users is the same and exponentially distributed with parameter θ. They model the arrival and departure of users as a Markov chain presented in figure 2.2.

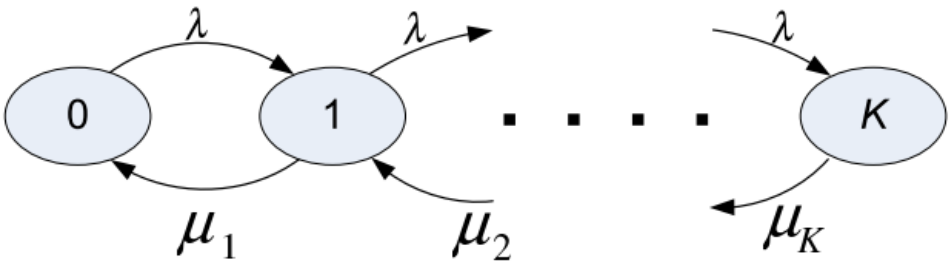


Fig. 2.2: Markov chain of user's dynamic.

From this model, they concentrate and track the performance of a tagged user while connecting to the system, and derive the probability of having a video freeze. In fact, they compute the starvation probability of a tagged user while he passed through different phase which are:

- Connection phase: the tagged user joins the system and requests a video with exponentially distributed duration;
- Prefetching phase: the tagged user receives data with a throughput $\frac{C}{(i+1).Bitrate}$ where i is the number of users connected to the system;
- Playback phase: the tagged user starts consuming data with a speed of Bitrate while continuing downloading data with the same speed as in Prefetching phase;
- Starvation phase: the tagged user may experience a starvation while reading data. This phenomenon is computed in probabilistic manner thanks to the stochastic model of user's dynamic.

- End of a video: when the tagged user downloads the entire video, she leaves the system.

The authors derive the starvation probability as a function of startup threshold.

This work was our first source of inspiration and the starting point of our modeling. In fact, we keep the same user's dynamic and we focus on a tagged user while donwloading a video with On-Off strategy (see Section 1.3.6). In the next section, we present another interesting work in which authors analyze the performance and the downloading strategies of some streaming devices, both Android and iOS devices; and especially On-Off streaming strategy for Android equipments.

2.3.2 QoE, streaming downloading strategies and energy consumption

In this section, we discuss two works that analyze the performance of donwloading streaming strategies. In the first one [Liu+], the authors conduct a study to analyze the behavior of the pair (service provider, terminal device) while downloading a video stream. In the second work [Mat+15], the authors propose an algorithm at packet level, using viewing statistics of users, to control wasted data due to user abandonment behavior.

Streaming strategies performance and QoE

In this section, we present and discuss the performance of a device while requesting a video from a video server through internet access [Liu+]. The authors analyze around 26,713,708 HTTP requests, the equivalent of 15,725 video and a total of 27.4 TB of video streaming traffic. By analyzing streaming traffic, the authors find that Android and iOS devices request video content differently (i.e requested data and buffer management) while requesting the same video. They found that iOS terminals consume more power because of redundant traffic which is not the case for Android terminals. The conclusions of the authors conclusions were the following:

First, they confirm that android terminals use a single HTTP video request to download a video from a video server while the iOS devices use, generally, multiple HTTP requests. Concerning the buffer management strategy, Android and iOS terminals use also different buffer management policies to download videos. iOS devices tend to download as fast as possible the whole video packets and then keep as much traffic as possible in the terminal buffer. On the other hand, Android devices set a so called high water threshold. When the buffered traffic reaches this threshold, the terminal pauses the downloading of the video. As the playback continues, it consumes traffic and the buffered data decreases. When this buffered data drops below a so called low water threshold, the downloading will be resumed. Figures 2.3 and 2.4 show respectively, the downloading approaches for an Android and iOS devices.

This work inspired us to analyze more and study the case of Android devices. Especially, we will work on On-Off strategy used by Android devices and see how the starvation phenomenon happens while under this streaming technique.

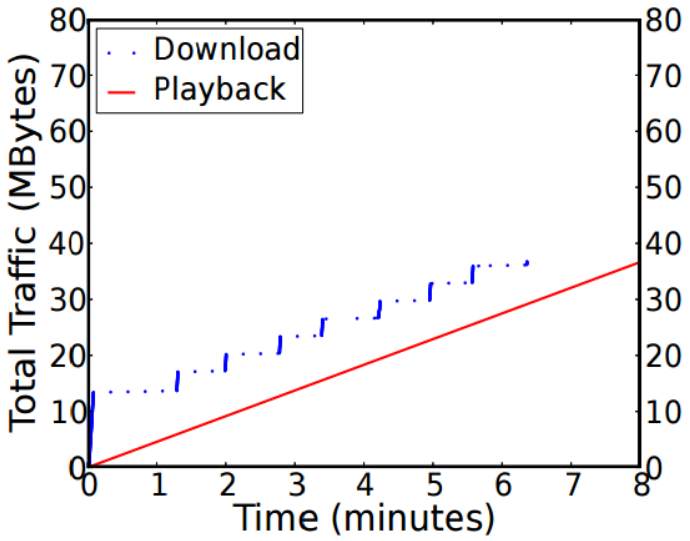


Fig. 2.3: Downloading strategy for an Android device.

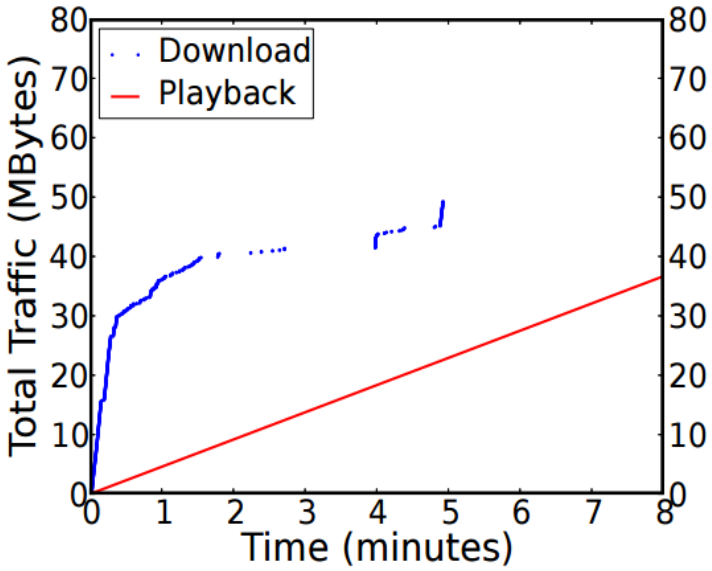


Fig. 2.4: Downloading strategy for an iOS device.

In the next section, we present a work in which the authors propose an algorithm at packet level that use viewing statistics of users, to control wasted data due to user abandonment behavior.

Wasted data and QoE

No one can deny that video data downloaded and not played, namely wasted streaming data, by users constitute one of the big problems that need to be taken into account when optimizing the QoE. In fact, when a user downloads data, it consumes battery energy. If this data is not played, it is a waste. However, it is not easy to know how much data to send to a terminal device in order to not waste energy if the user decides to stop the video and thus ensuring a smooth video playback (i.e without

video freeze). In [Mat+15], the authors propose an algorithm in order to manage this wasted traffic phenomenon. To do that, they study the tradeoff, in terms of energy, between video chunks sizes of the video content and tail energy, which is a fixed energy to download a chunk. In fact, if the chunks are small we have a lot of total tail energy. If the chunk is large chunks, the total tail energy is low but the risk of downloading traffic that end users will not view because of abandoning the video playback increases. To solve that, the authors propose to use statistical viewing of users to control wasted energy and anticipate video abandonment. First, they propose a formula to compute the mean energy due to the downloading of a single chunk expressed as follows:

$$E[E_{waste}(0, S_1)] = E_{tail}(T_1) + \frac{E[B_{waste}(0, T_1)]}{r_{dl}} * P_{rx}(r_{dl}) \tag{2.9}$$

Where:

- $E_{tail}(T_1)$ is the tail energy to download one chunk of size T_1,
- $E[B_{waste}(0, T_1)]$ is the mean wasted data if user decides to abandon the video playback,
- r_{dl} is the rate at which the video traffic is downloaded,
- $P_{rx}(r_{dl})$ is the power consumption while receiving streaming traffic at rate r_{dl}.

The authors applied then a recursive method to compute the wasted energy while downloading a video of multiple chunks and seek the optimal chunk size enabling to optimize wasted energy while anticipating their abandoning behavior.

In our work, we are not adopting a chunk approach but we are interesting on techniques used to model the abandoning behavior of end users. In our chapter 4, we will see how we quantify wasted data due to abandonment behavior especially for the On-Off streaming strategy.

Next, we present some experiments that we realized in order to understand more what happens in terms of QoE for streaming video service indicators (i.e starvation phenomenon and wasted traffic due to user abandon) while streaming with Fast Caching or On-Off strategy.

2.3.3 Simulating QoE indicators on different streaming strategies

In this section, we present simulations that we carried out at the beginning of the thesis in order to motivate our work, and understand better the impact of different streaming strategies on the QoE indicators. In the first section, we explain why we focus on QoE for streaming services and seek different streaming strategy that a pair (service provider, user terminal) can adopt. In the second section, we run some experiments in order to compare the QoE indicators for the two limit cases of streaming strategy (i.e Fast Caching and On-Off). In the last section, we present our developed algorithm that simulate real comportment of user while streaming with Fast Caching or On-Off strategy. We use our simulator in the rest of our work to compare our models with it.

Quality of experience and streaming service and strategies

As explained before, the QoE of services is a notion that encompasses the whole service chain from a service provider to end user. Particulary, the QoE depends on the type of service requested by the end user. Consequently, we sought to know what services are consumed by Internet users? And, what is the most dominant service?

These two questions represent a good starting point in understanding the share of use of each service in the overall volume of traffic exchanged over the Internet, as well as the service that dominates the flow of data on computer networks. Based on the findings of these two questions, we will probably be working on a service or some services to improve the overall QoE.

We captured an entire network traffic of a European mobile operator in March 2014. We have collected 800 Gigabytes of data, equivalent to 30 minutes of traffic in the day, and 30 minutes during the night. We used Tstat [Tst14] to select streaming traffic. Figure 2.5 shows the percentage of each service type for different network's type, in our data traffic. We remark that the most dominating services are:

- Cloud,
- Mail,
- Streaming,
- Download,
- Web.

In addition, we note that the traffic of the video streaming service, exchanged on the network, does not cease to increase over the years. In fact, video streaming represents almost half of traffic in network based in figure 2.5. Video streaming is the driver factor of traffic growth. [Mar15] confirms that 64 % is the percentage of video streaming traffic on the world's internet traffic, in 2014. In addition and according to [CIS], by 2019, streaming video services will be four-fifths of the overall Internet traffic. Figure 2.6 shows the predictions of the percentage of video streaming service traffic in 2019.

As streaming services more popular, clients becomes more QoE demanding and satisfying users is an utmost important matter for internet service providers (ISPs). Consequently, our attention was more to work on the optimization of the QoE for streaming video services.

In order to reach this objective, we should first know how data is delivered from a service provider to user terminal. Next, we will show results about a traffic analysis that we run in order to know how data is delivered from a video service provider to end user.

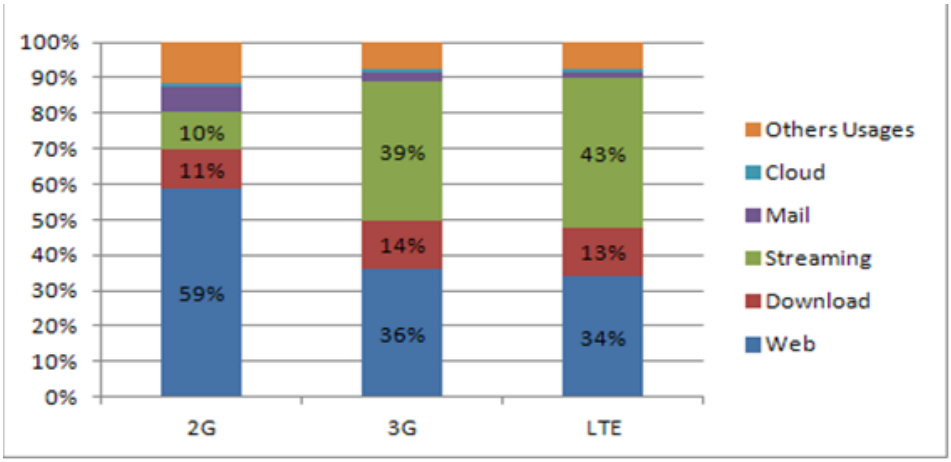


Fig. 2.5: The percentage of each service in the overall volume of traffic for different types of network.

Consumer Internet Video 2015–2020							
	2015	2016	2017	2018	2019	2020	CAGR 2015–2020
By Network (PB per Month)							
Fixed	27,011	34,978	45,134	57,656	73,413	90,239	27%
Mobile	1,756	3,138	5,378	8,607	13,295	19,668	62%
By Category (PB per Month)							
Video	22,344	29,046	38,297	50,596	67,423	86,704	31%
Internet video to TV	6,424	9,070	12,215	15,667	19,284	23,203	29%

Fig. 2.6: Consumer Internet streaming video services predictions for 2020.[Mar15]

Streaming delivery strategies

In this section, we seek different possible streaming strategies while streaming over the Internet. As explained in Section 1.3.6, the pair (service provider, terminal equipment) decides both on how streaming traffic is delivered. In fact, there exists different strategies to download streaming data over the internet. To confirm that, we run an experiment in order to seek streaming strategies used by some devices with different service providers. Nokia Lumia 800, iPhone 4S, HTC Desire C, Samsung (i.e YouTube, Vimeo, Dailymotion, Metacafe and Wat.tv. We analyzed collected data collected using Wireshark. The number of watched videos is 126 and the duration of these videos is from 5 to 80 minutes. Figure 2.7 shows the test results.

We remark that there are five different download strategies which are Fast Caching, On-Off, Throttling server, Encoding rate and the Dynamic Adaptive Streaming over HTTP (DASH).

Nowadays, the most used strategies are the Fast Caching and On-Off streaming strategies [Hoq+]. For this reason, we decided to work on the QoE while streaming a video for different strategies. As

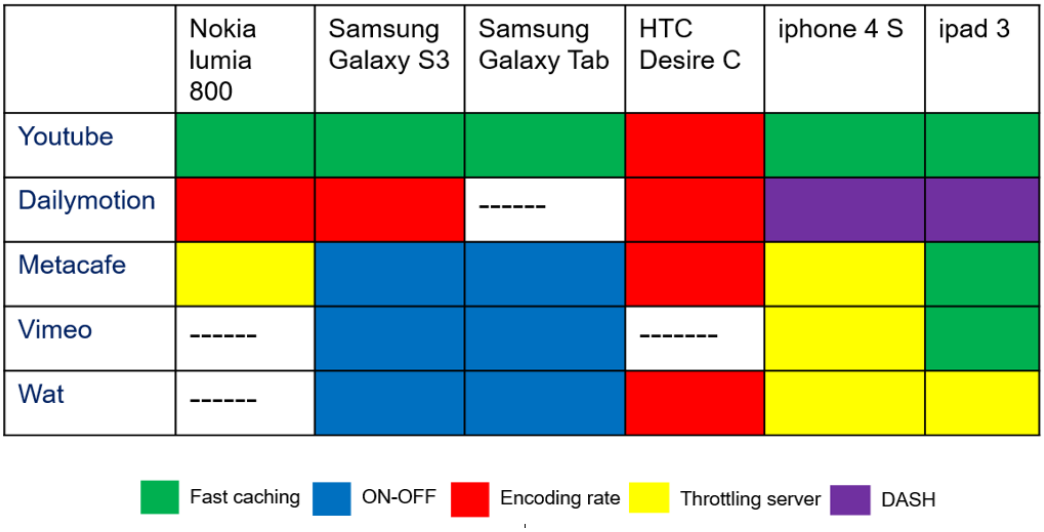


Fig. 2.7: Streaming strategies technical match between service provider and terminal device.

the Fast Caching strategy has been modeled and analyzed in [Xu+13], we decide to work on On-Off streaming strategies to first, compute the QoE indicators, for example starvation probability, and also to compare the performance of On-Off and Fast Caching strategies. In the next section, we show the result of an experiment we have carried out to compare these two streaming strategies.

QoE indicators

To compare the energy consumption between the ON/OFF and Fast Caching strategies, we simulated the behavior of these two techniques under realistic network conditions, using Matlab. We fixed the capacity of the network at 6 Mbps, the bitrate of different user devices at 480 Kbps, and the initial delay before starting the playback of a video at 5s. We suppose that the network can manage a maximum of 15 devices simultaneously, and we assume that the capacity is uniformly shared between user devices. We assume, on average, that there is one user requesting access to the network every 180 seconds. A user leaves the network when he finishes downloading the entire video. We concentrate on one tagged user and see what happens when he decides to watch only 25% of the video.

Figure 2.8 shows the bar-plots of fraction of downloaded data as a function of video duration, when a tagged user decides to watch only 25% of the video. it is clear that when the video length exceeds 10 minutes, the tagged user downloads around two times more data in Fast Caching then in On-Off. Consequently, the user consumes more energy in the Fast Caching strategy to download streaming data that has not, maybe, be played. To confirm that the On-Off strategy is optimal regarding the Fast Caching, under these conditions, we plotted histograms of the number of buffer starvations during a streaming session, for the On-Off and Fast Caching strategies. Figures 2.9 and 2.10 show the histograms of the two strategies under the same assumptions and parameters values as figure 2.8 for a 5 minutes video duration. We notice that the number of buffer starvations is, approximately, the same for the two strategies. Hence and in this case, the On-Off strategy is optimal because it allows the best energy consumption and buffer starvations trade-off.

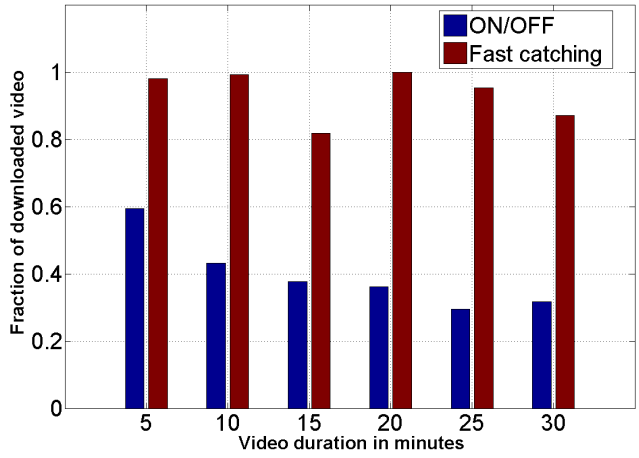


Fig. 2.8: Fraction of downloaded data while watching only 25 % of the video for On-Off and Fast Cching strategies.

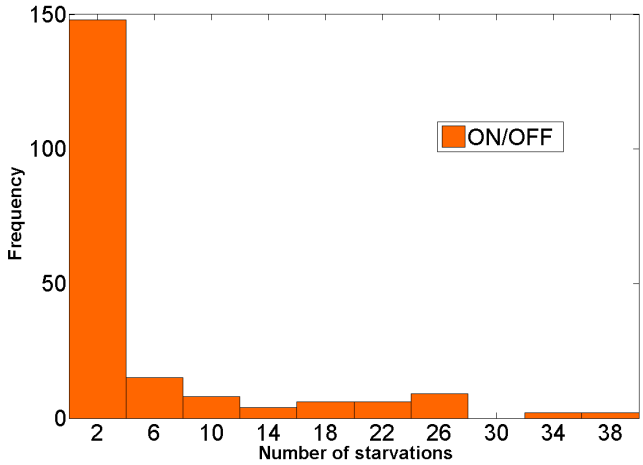


Fig. 2.9: histogram of the number of buffer starvations for ON/OFF strategy.

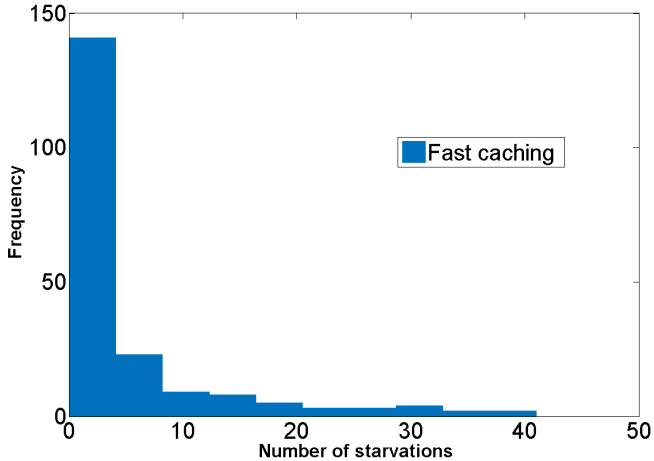


Fig. 2.10: histogram of the number of buffer starvations for Fast Caching strategy.

2.4 Conclusion

In this chapter, we first presented an overview of some QoE models to compute QoE indicators. We talked, afterwards, about the relationship between the QoE and network criteria. We also analyzed some inspiring works from the literature. Finally, we ran some experiments to compare QoE indicators for the Fast Caching and On-Off streaming strategies. These preliminary results form our motivation to work on the modeling and the optimization of the QoE and the simple simulator that we introduce will be the reference for our models in the next chapters.

3 Evaluation and optimization of QoE of On-Off strategy

> *The Initial Mystery that attends any journey is: how did the traveler reach his starting point in the first place?.*
>
> — **Louise Bogan**
> (Poet.)

Video is the dominating traffic over the Internet in terms of volume. Social and professional networks (i.e. Instagram, Facebook...) as well as video portals (i.e. Youtube, Dailymotion...) acted as catalysts for video content production. Our initial concern in this chapter is the evaluation and optimization of quality of experience. We focus on how to provide a smooth playback streaming service during a video streaming session. The first section is an overview of the two most popular streaming strategies: on-off and fast catching. In the second section, we take a mobile operator perspective and focus on the QoE when using fast-caching or on-off strategies. We develop an analytical model that takes into account user's dynamics and allows us to derive the probability of starvation (frozen video) that the client may experience. We show that the probability of starvation of the end user is equivalent under the fast-caching and on-off strategies and we explore, in the third section the trade-off between two dimensions of QoE in the on-off case: the initial startup delay and the starvation probability.

3.1 On-Off vs Fast Caching (FC)

Delivering http streaming traffic to the end-user is a daunting task for operators as they must (i) move from a model where QoS metrics such as loss rate and delay are monitored to tracking user experience through several QoE metrics (e.g., initial startup delay, starvation probability related to frozen video playback events) and (ii) anticipate that user may abandon viewing early. These two challenges are contradictory as the former calls for an as fast as possible transfer to the end user while the latter pushes the content provider and the mobile operator to deliver at a given time instant only a bit more than what the user current viewing process requires. These challenges are reflected in the streaming strategies observable on the market, where traffic delivery might be controlled either by the server, the client or a mix of both. Popular strategies are [Hoq+]: fast caching strategy in which the server delivers as fast as possible the video to the client; rate throttling strategy in which the server starts with a period during which data is delivered as fast as possible followed by a period where rate is capped to a limit above the playback rate and, On-Off strategy the server pushes data as fast as possible to the client that regularly pauses, by emitting TCP zero advertised window or aborting the TCP connection, when its buffer is full enough and resumes transfer when the playback process has drained the buffer to a low value.

In addition, energy consumption might be taken into account in the operator and client perspective [Sie+15].

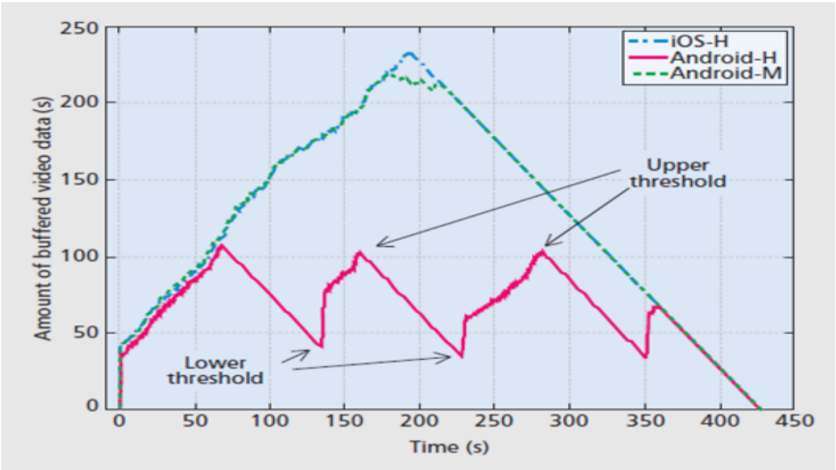


Fig. 3.1: Downloading strategies for different types of terminals [RM+14].

Ramos-Muñoz, et al. have studied and presented in [RM+14] the policy of downloading a video by the YouTube player on different types of terminals. Figure 3.1 shows the variation of buffered data in seconds as a function of the playback time in seconds. The strategy observed on some Android and IOS devices (Android-M and IOS-H - blue curve in Figure 3.1) corresponds to a Fast Caching strategy while Android-H devices (pink curve on Figure 3.1) feature an ON/OFF strategy in which the download stops when the playback buffer has reached a so-called high buffer threshold and resumes when the playback process has consumed a fixed amount of video (up to the so-called low buffer threshold). Next in this section, we will explain how these two popular streaming strategies work before conducting a comparison study between them.

3.1.1 Fast Caching streaming strategy

One of the most popular streaming strategy is Fast Caching strategy. In the beginning of this streaming strategy session, the terminal client prefetches content in an attempt to rapidly fill the playback buffer and reach the so-called start-up delay (i.e the duration between the time that a user initiates a session and the time that the playback start, in seconds). In the literature, this pre-phase is known as the prefetching period: it is commun to all downloading strategies. After this prefetching phase, it is the available bandwidth which defines at which rate the data is transmitted since the server sends the data at full available bandwidth. In other words, the video server delivers as fast as possible the video to the client and the client player reads data continuously from the receiver buffer. The downloading of the video continues during the viewing phase until the video has been fully downloaded. This downloading strategy requires no complex engineering at the server and the client.

3.1.2 On-Off streaming strategy

The second most popular streaming strategy is the on-off strategy. After a prefetching phase, the server starts to push data as fast as possible to the client that regularly pauses, by emitting TCP

zero advertised window or aborting the TCP connection, when its buffer is full enough and resumes transfer when the playback process has drained the buffer to a low value. In fact, this strategy consists in maintaining the level of the buffered data (i.e downloaded and unread data) between a lower threshold and a high or upper threshold. A user starts the OFF phase when the total size of the unread data in the buffer exceeds the high threshold and the download pauses. When the total data size drops below a low threshold, the download resumes [Liu+] and the OFF phase ends. The difference between the time the download pauses and the time the download resumes is the OFF period duration. The ON period start while the buffer reaches the lower threshold by resuming the download of the video and ends when the buffer reaches the high threshold. In Figure 3.1, we can see that the lower threshold is around 40 seconds of buffered data and the high/upper threshold is around 105 seconds.

The purpose of such buffer management is to prevent starvation situations, while avoiding unnecessarily filling the buffer if the client decides to abandon video playback. So the question is: is it the client or the server that decides on the values of low and high thresholds? and, what is the optimal values of low and high thresholds in order to ensures the best quality?

To answer these questions, the authors in [Liu+] have studied the streaming strategies employed for HTTP streaming in the Internet. Among these strategies, we find the on-off strategy for Android terminals. They observed that the values of low and high thresholds may change from an Android terminal to another.

In addition, they show that the values of these two thresholds remain constant during a streaming session. This problem of optimal choice of the durations of the on and off periods was the starting point of our reflection and analysis which constitutes this chapter 3. In fact, our approach was to model this on-off strategy in order to see and act on the on and off durations and thus to optimize the quality of experience for the on-off streaming strategy. Our work was among the first attempts to model the on-off strategy. Later in this chapter, we will see more details about this modeling. Just before starting our mathematical modelling of this problem, let's address the differences between the most two popular strategies: fast caching and on-off.

3.1.3 On-Off vs Fast Caching: advantages and disadvantages

A key reason to use the On-Off strategy is to limit the volume of traffic uselessly downloaded to the end user, ie., the part of the video downloaded but not watched by the user, either because of a lack of interest or because of bad network conditions. Figure 3.2 reports the distribution of video actually downloaded by all the client of a major European mobile operator. This figure was built out of 800 Gigabytes of data, equivalent to 30 minutes of traffic collected during the day time (non peak hour), and 30 minutes during the night (peak hour). We used Tstat [Tst14] to filter out streaming traffic. We observe from Figure 3.2 that 25% of sessions abandon very early the download of a video, and only 35 % of sessions download the entire video. The on-off strategy might thus enable to limit the amount of uselessly downloaded video traffic, which represents a financial and energy cost both for the end user and the access network.

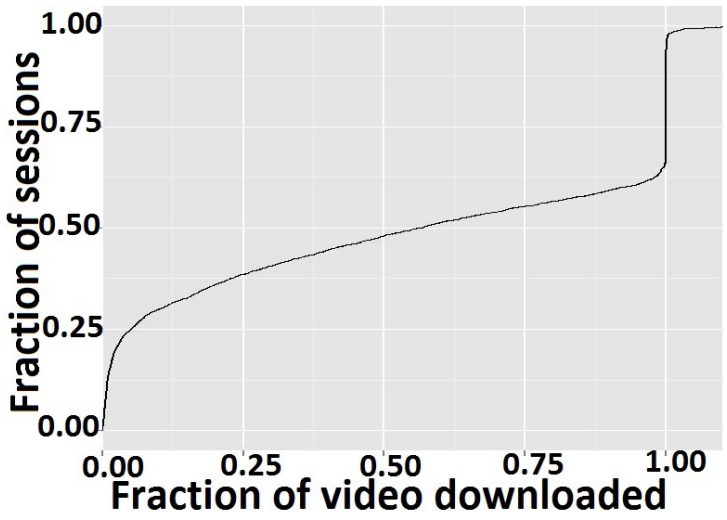


Fig. 3.2: Fraction of abandon during a streaming session.

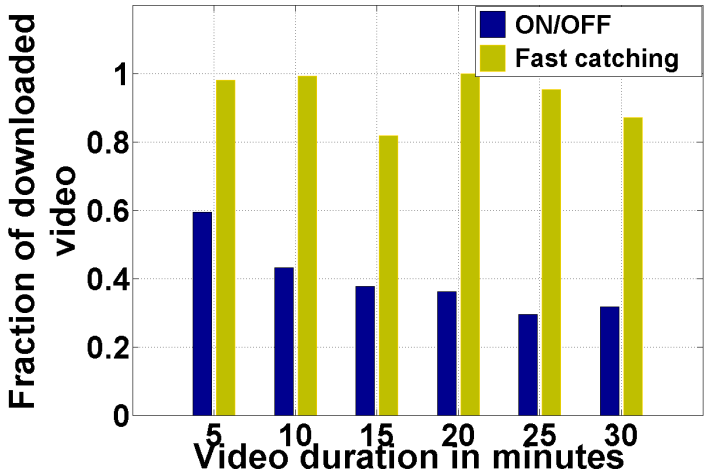


Fig. 3.3: Fraction of downloaded data while watching only 25 % of the video for ON/OFF and Fast Cching strategies.

As we have no way to derive the amount of traffic downloaded but not watched from our trace, we exemplify the typical gain we can expect through a simple simulation of a mobile cell with users that abort the video transfer when 25% of the video has been watched.

We consider a cell with a fixed capacity of 6 Mbps, a video bit rate of 480 Kbps and an initial startup delay (before starting the playback of a video) of 5s. We suppose that the cell can handle 15 devices maximum simultaneously, and we assume that the capacity is uniformly shared between user devices. Clients arrive according to a Poisson process with an average of 1 client every 180 seconds. A user other than the tagged user) leaves the network when she finishes downloading the entire video.

Figure 3.3 shows the bar-plots of the average fraction of downloaded data as a function of video duration, when a tagged user decides to watch only 25% of the video. As the video length increases, the fraction of downloaded video gets closer and closer to 25% in on-off while it decreases only

marginally with Fast caching. The percentage is higher for small videos as the amount of bytes downloaded represents a larger share of the video for smaller videos.

While the on-off strategy might be beneficial from a financial and energetic perspective, it is a viable solution only if the QoE perceived by the end user is not deteriorated when employing this approach.

This is reason why we develop the analytical model presented in the next section.

3.2 Analytical Markovian model for QoE metrics

Quality of experience (QoE) has received a significant attention from researchers. Agboma and Liotta [AL08] demonstrate how QoE can be captured in a statistical model that correlates QoS with QoE parameters. The authors show how, practically, one can control the application QoS parameters for QoE management. The prediction model shall allow operators to anticipate the user's experience and allocate network resources accordingly.

In [Hoß+12], a novel YouTube in-network monitoring tool is proposed as a passive network monitoring tool. This approach aims at detecting and measuring stalled events of the video playback by approximating the video buffer status through the comparison of the playback times of video frames and the time stamps of received packets. The pythomo tool [Jul+11] addresses the same problem using an active approach.

ParandehGheibi and al. [Par+11] developed a Markovian model for the delivery of a multimedia flow to a client over a wireless channel and further assuming that random linear coding can be used. While interesting from a theoretical perspective – to study the fundamental rate-delay-reliability trade-offs that exist in such a scenario – it is not directly applicable to the On-Off strategy that we consider in this paper.

The closest work to our analytical effort is [Xu+13]. In this paper, Xu and al. developed a Markovian analytical model for the fast-caching HTTP streaming strategy that permits the analysis of the initial startup delay and its impact on the starvation probability. In this this chapter, we extend this model to the On-Off scenario.

Our focus is exclusively on HTTP streaming traffic that already constitutes the majority of traffic in terms bytes with continuous expected growth worldwide in the near future (see e.g. Sandvine quarterly report – `https://www.sandvine.com/`).

We restrict our attention to the Fast-caching and On-Off strategies as they constitute two extreme cases in terms of QoE. To put it differently, if On-Off can offer similar QoE to Fast-caching, it can also compete with Rate-throttling. By our model, we aim at answering two key questions: (i) is the QoE achieved with On-Off strategy similar to that of fast caching, (ii) which trade-off exists between initial startup delay and the starvation probability experienced by the end user.

3.2.1 Network model and hypotheses

We consider a cell using admission control. If the network is saturated, any request for additional user access will be denied. A user joins the network, requests a video and leaves the network after completing the download of the entire video. Note that a streaming user cannot watch more than one video from his terminal device. At the client side, downloaded video packets are stored in the play-back buffer and played at a deterministic rate depending on the video encoding rate. The download and playback process define two critical phases in a streaming session: prefetching and playback period. The first one occurs when a streaming client launches the video. This is a period during which the download of the video is performed without playback has started. The playback period corresponding to the next phase, in which the video is viewed on the mobile device, and wherein the downloading of the video may optionally continue. In the Fast Caching mode, the video download continues during the viewing phase until the video has been fully downloaded. In contrast, in the On-Off mode, the download stops when the playback buffer has reached a so-called high buffer threshold and resumes when the playback process has consumed a fixed amount of video (up to the so-called low buffer threshold). In [Liu+13] authors shows that both strategies are in use by actual devices. iOS devices use fast caching while the Android devices tend to use more of the ON/OFF strategy.

In [Xu+13] authors studied fast caching strategy and derived the probability of starvation using Markovian models. Next, we focus on the ON/OFF strategy. We track the performance of a tagged user downloading a video whose duration, measured in seconds, is exponentially distributed with mean $\mu = \frac{1}{\theta}$. We model the arrival and departure of users as a birth-death Markov chain shown in Figure 3.4 with the following parameters:

- **Arrival of users:** arrivals follow a Poisson process with parameter λ.
- **Departure of users:** at time t, the time remaining for a fixed user, different from the tagged user, to leave the network follows an exponential distribution with parameter ν_i where i is the number of connected users. This is justified by the fact that the video duration is assumed to be exponentially distributed.
- **Absorbing state:** It corresponds to the case where the tagged user finishes downloading the video. When the system is at state i, the time remaining to the tagged user to enter the absorbing state follows an exponential distribution with parameter ϕ_i. This, also, is justified by the fact that the video duration is exponentially distributed. The dependence on i will be justified in the next section.

The tagged user starts with a prefetching phase where the player downloads an amount of data q_a during a time called start-up delay (more details are in section 3.2.3). When the playback starts, she switches between the ON and OFF mode. We model this ON/OFF system as a Markov chain shown in Figure 3.4. Let us consider K the maximum number of admitted users in the cell. There are $2K + 1$ possible states. The $2K$ first states are the product of the number of users connected combined with the two downloading states for the tagged user (ON and OFF). We denote each of these states in the form (i, ON) or (i, OFF) where i is the number of connected users other than the tagged user. The user enters an OFF state when she accumulates an amount of data

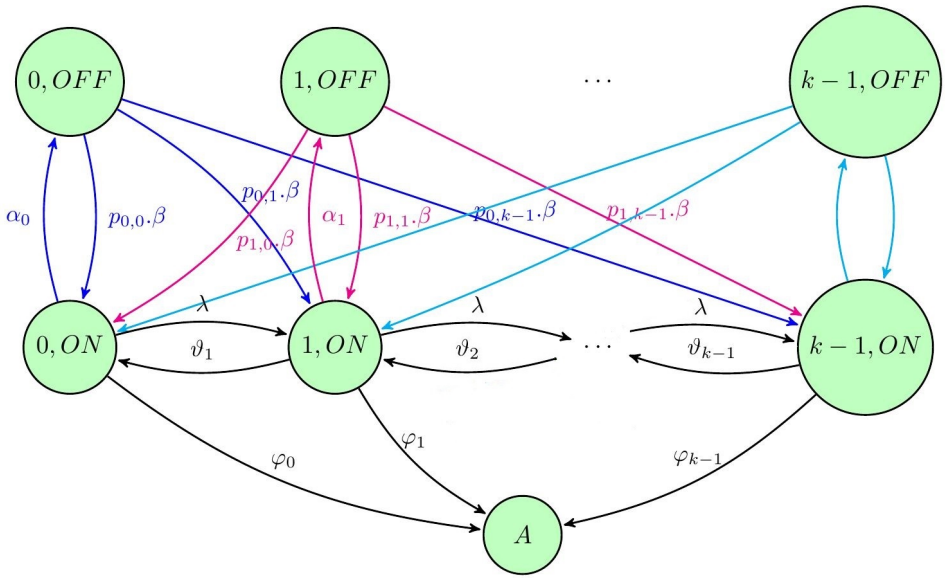


Fig. 3.4: Markov chain for user dynamics when the tagged user start playback.

$Q = HighWaterThreshold - LOWWaterThreshold$ in the buffer. Hence, The duration of state (i,ON), measured in seconds, is a random variable following an exponential distribution with parameter $\frac{1}{\alpha_i}$ which we define as follows. Let C be the aggregated capacity of the cell in bps and $Bitrate$ be the encoding rate of the video, also in bps. We assume that each user receives a throughput of $\frac{C}{i}$ where i is the number of connected users. In state (i, ON), the buffer of the tagged user fills in at a rate of $\frac{C}{i+1} - bitrate$, resulting in $\frac{1}{\alpha_i} = (\frac{C}{bitrate.(i+1)} - 1)\frac{1}{\theta_{ON}}$ where θ_{ON} is the average amount of time necessary to reach the high water threshold. The user switches from OFF state to an ON state when the playback reaches the low water threshold (consuming Q worth of data). The duration of (i, OFF) states do not depend on the system. We model the OFF duration as a random variable following an exponential distribution with parameter $\frac{1}{\beta}$. The last possible state corresponds to the full download of the video by the tagged user. This absorbing state is denoted by **A**. Note that when $\frac{C}{i+1} < bitrate$, the ON duration is higher than the video duration resulting in no transitions to OFF periods: the tagged user finishes downloading the entire video without entering an OFF state.

The ON and OFF period duration's are related as follows. The amount of video accumulated during the ON period (on top of the viewing process) is equal to what is consumed during the OFF period. Hence:

$$\theta_{on} \sum_{\substack{i \\ \frac{C}{i+1} > bitrate}} \pi_i \left(\frac{C}{Bitrate.(i+1)} - 1 \right) = \beta$$

We further introduce the following notations. $C_r = \frac{C}{bitrate}$ is the reduced capacity in seconds of video contents. We consider that all users have the same SNR and that the channel is static. If the system is in state i, each user receives an identical reduced throughput denoted $d_i := \frac{C_r}{i+1}$. During the ON mode, the reduced throughput observed at the playback buffer is $c_i = d_i - 1$, -1 accounting for the playback process. When the tagged user is on the OFF mode, $c_i = -1$ since there is no arrival of data. Next, our goal is to construct the complete model to compute the starvation probability as a function of startup delay q_a, and OFF and ON durations. Let $(\pi_i)_{0 \leq j \leq K-1}$ be the stationary

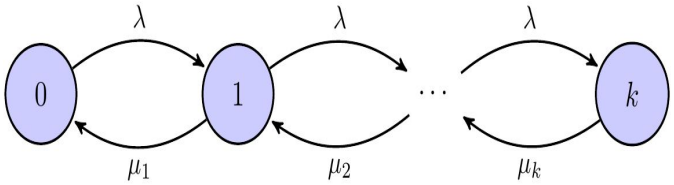


Fig. 3.5: Markov chain before the tagged user joins the network.

probability of having i users in the system when the tagged user joins the network, $(\vee_{ij})_{\substack{0\leq i\leq K-1\\0\leq j\leq K-1}}$ the transition probability from i users to j users during the prefetching delay q_a, and $(\Psi_j)_{0\leq j\leq K-1}$ the probability that a user experiences a starvation event given that she starts the playback at state j (j connected users other than the tagged user). By the law of total probability, the starvation probability $P_{starv}(q_a)$, can be expressed as:

$$P_{starv}(q_a) = \sum_{i=0}^{K-1}\sum_{j=0}^{K-1} \Pi_i.V_{ij}(q_a).\Psi_j(q_a) \tag{3.1}$$

In Sections 3.2.2, 3.2.3and 3.2.4, we compute, respectively, the stationary probabilities π_i, the prefetching delay matrix transition $(\vee_{ij})_{\substack{0\leq i\leq K-1\\0\leq j\leq K-1}}$ before computing $(\Psi_j)_{0\leq j\leq K-1}$, the probability that a user experiences a starvation event given that she starts the playback at state j.

3.2.2 Users and transitions distribution: $(\pi_i)_{0\leq j\leq K-1}$ and $(\nu_i)_{1\leq j\leq K-1}$

To compute the distribution π_i, let us consider the system before the tagged user join the network. The dynamic of users is shown in Figure3.5.

This users model appears in [Xu+13]. We have the same starting point except that our approach brings up the on/off mode later. More precisely, the ON/OFF mode has extreme benefits (see Section 3.1.3) compared to Fast Caching mode, but it has made very complicated calculations and resolution of our problematic, by doubling the dimension of the problem, which we were able to overcome in this chapter. The transition rate from a state i to $i-1$ is $\mu_i := C\theta_F$. The capacity is constant so, we let $\mu_i = \mu$ for $i = 1, 2, ..., K$ and $\mu_0 = 0$. Let $\rho := \frac{\lambda}{\mu}$ be the load of the network. Let ζ_i be the probability that there exist i users in the network. We have:

$$\zeta_0 = \tfrac{1-\rho}{1-\rho^{K+1}};\quad \zeta_i = \tfrac{\rho^i(1-\rho)}{1-\rho^{K+1}},\ \forall i = 1, ..., K.$$

When the tagged user joins the network successfully, she will observe i other users with probability π_i. π_i is the probability of being at state i knowing that we are at a state $j < K$. In other words, π_i is the probability that the network is at state i knowing that she is not at state K (Because otherwise the tagged user can not join the network). A simple computation leads to find:

$$\pi_i = \frac{\zeta_i}{1-\zeta_K} = \frac{\rho^i(1-\rho)}{1-\rho^K},\ \forall i = 0, ..., K-1. \tag{3.2}$$

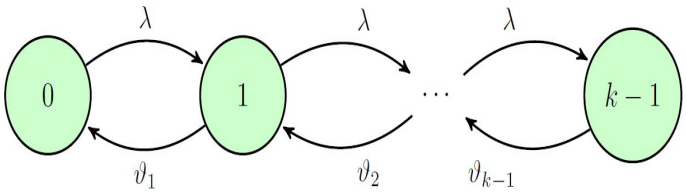


Fig. 3.6: Markov chain seen by the tagged user.

Note that $\pi_K = 0$ because the network can serve a maximum of K users simultaneously.

Our Markov chain will be altered with the presence of the tagged user. In essence, the state of the chain is the number of users seen by the tagged user. Denoted by ν_i the transition rate from state i to state $i-1$ conditioned on the presence of the tagged user. At state i, the throughput of each user is $\frac{C}{i+1}$, so $\nu_i := \frac{iC\theta_F}{(i+1)} = \frac{i}{i+1}\mu$ for $i = 0, 1, ..., K-1$[Xu+13]. As defined before, λ_i denotes the transition rate from state i to state $i+1$. For all $i \neq K-1$, we let $\lambda_i = \lambda$, and $\lambda_{K-1} = 0$. Figure 3.6 represents the users' dynamics seen by the tagged user, with these different transitions.

Before the playback starts, data prefetches in the buffer of the tagged user to reach the start-up threshold q_a to start the video playback. So in the section 3.2.3, we compute the prefetching delay distribution $(\vartheta_{ij})_{\substack{0\leq i\leq K-1 \\ 0\leq j\leq K-1}}$ which appears in equation 3.1 .

3.2.3 Prefetching delay distribution

During prefetching period, the state of the system evolves. Thus, it is necessary to define the transition probabilities matrix from a state i to a state j, $(\vee_{ij})_{0\leq i,j\leq K-1}$, during this period. We now determine these probabilities by studying the dynamics of the system represented by the Markov chain described in Figure 3.6.

Let $\vee_{ij}(q, q_a)_{0\leq i,j\leq K-1}$ be the conditional probability of being at state j when the video download level reaches q_a, such that the buffer was at q at state i. q is the queue length of the tagged user during the prefetching period, and q_a is the start-up delay threshold (time the tagged user needs to wait before playback starts). Hence, $\vee_{ij}(0, q_a)_{0\leq i,j\leq K-1}$ is the transition probability from state i to state j during the prefetching period, noted V_{ij} in formula 3.1.

The reduced throughput at the prefetching period is d_i. Let t be an instant during the prefetching period. In the infinitesimal time interval [t,t+h], the events that may occur are :

- no change on the network ,
- arrival of one user,
- departure of on user (different from the tagged user),
- occurrence of more than one events (the probability of this event is negligible comparing to other events when h go to 0.

Between t and $t+h$ and knowing that $q(t+h) = q + d_i h$, we deduce:

$$\begin{aligned} \vee_{i,j}(q, q_a) = (1 - (\lambda_i + \vartheta_i)h) \vee_{i,j} (q + d_i h, q_a) \\ +\lambda_i h \vee_{i+1,j} (q + d_i h, q_a) \\ +\vartheta_i h \vee_{i-1,j} (q + d_i h, q_a) + o(h) \end{aligned} \tag{3.3}$$

when h $\to$ 0, we obtain:

$$\begin{aligned} d_i \dot{\vee}_{i,j}(q, q_a) = (\lambda_i + \vartheta_i) \vee_{i,j} (q, q_a) - \lambda_i \vee_{i+1,j} (q, q_a) \\ -\vartheta_i \vee_{i-1,j} (q, q_a), \forall i, j \in 0, 1, \cdots, K-1 \end{aligned} \tag{3.4}$$

Posing $\vee = (\vee_{ij})_{0 \leq i,j \leq K-1}$, Equation 3.4 can be rewritten as:

$$\dot{\vee}(q, q_a) = M. \vee (q, q_a) \tag{3.5}$$

such that:

$$M = \begin{pmatrix} \frac{\lambda_0}{d_0} & -\frac{\lambda_0}{d_0} & 0 & \dots & 0 & 0 \\ -\frac{\vartheta_1}{d_1} & \frac{\lambda_1+\vartheta_1}{d_1} & -\frac{\lambda_1}{d_1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & -\frac{\vartheta_{K-1}}{d_{k-1}} & \frac{\vartheta_{K-1}}{d_{K-1}} \end{pmatrix} \tag{3.6}$$

We can easily show that $\vee(q_a, q_a) = I_d$, with I_d is the identity matrix, because we can not have transition if the prefetching period is finished [Xu+13]. Consequently, the solution of Equation 3.5 is:

$$\vee(q, q_a) = exp((q - q_a).M).I_d \quad for \;\; q \in [0, q_a] \tag{3.7}$$

3.2.4 Starvation probability

When the playback starts, the tagged user switches between the ON and OFF mode, and she leaves the network when she downloads the entire video. To take into account her departure, we modify our Markov chain (see figure 3.6) by adding an absorbing state defined in Section 3.2.1. The absorbing state **A** denotes the event the tagged user completes its downloading. By definition, its can not happen if the tagged user is in OFF mode. Let φ_i be the transition rate from state (i, ON) to state **A**. Due to the exponentially distributed video duration, this transition is exponentially distributed. The capacity of users is $\frac{C}{i+1}$, so $\varphi_i := \frac{\mu}{i+1}$. Our system can be modeled as a Markov chain shown in Figure 3.4.

When the user is in off mode, she does not download data. However, the system changes state. In order to keep the information on the system dynamics during the period OFF, we introduce $p_{i,j}$, the

probability that the tagged user enters the OFF period at state i and ends at state j. Let $F_{i,j}(t, t_{OFF})$ be the probability to end the OFF period at state j at time t_{OFF}, knowing that the OFF began at state i at time t. It is easy to see that $p_{i,j} = F_{i,j}(0, T_{OFF})$ with T_{OFF} is the mean of the OFF duration exponentially distributed. Let h be an infinitesimal time. Knowing that the tagged user can not end downloading the video at state OFF and using the Markov chain defined in Figure 3.6, we have:

$$\begin{aligned} F_{i,j}(t, t_{OFF}) = (1 - (\lambda_i + \vartheta_i)h)F_{i,j}(t + h, t_{OFF}) \\ +\lambda_i h F_{i+1,j}(t + h, t_{OFF}) \\ +\vartheta_i h F_{i-1,j}(t + h, t_{OFF}) + o(h) \end{aligned} \tag{3.8}$$

when h $\to$ 0, we obtain:

$$\begin{aligned} \dot{F}_{i,j}(t, t_{OFF}) = (\lambda_i + \vartheta_i)F_{i,j}(t, t_{OFF}) - \lambda_i F_{i+1,j}(t, t_{OFF}) \\ -\vartheta_i F_{i-1,j}(t, t_{OFF}), \quad \forall i, j \in \{0, 1, \cdots, K-1\} \end{aligned} \tag{3.9}$$

Posing **F**= $(F_{ij})_{0 \le i,j \le K-1}$, equation 4.21 can be rewritten as:

$$\dot{\mathbf{F}}(t, t_{OFF}) = R.\mathbf{F}(t, t_{OFF}) \tag{3.10}$$

with:

$$R = \begin{pmatrix} \lambda_0 & -\lambda_0 & 0 & \dots & 0 & 0 \\ -\vartheta_1 & \lambda_1 + \vartheta_1 & -\lambda_1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & -\vartheta_{K-1} & \vartheta_{K-1} \end{pmatrix} \tag{3.11}$$

It is easy to see that $\mathbf{F}(t_{OFF}, t_{OFF}) = I_d$ and so:

$$\mathbf{F}(t, t_{OFF}) = exp((t - t_{OFF}).R) \quad for \;\; t \in [0, t_{OFF}] \tag{3.12}$$

Finally, $\mathbf{p} = (p_{ij})_{0 \le i,j \le K-1}$ are deduced as follows:

$\mathbf{p} = \mathbf{F}_{i,j}(0, T_{OFF}) = exp(-T_{OFF}.R)$

Now, we will compute the starvation probability.

Let $\Psi_{i,ON}(q)$ be the conditional starvation probability of the tagged user such that:

- the tagged user buffer contains q seconds,
- the system is at state (i, ON).

$\Psi_{i,ON}(q_a)$ is so the conditional starvation probability knowing that the system is at state i when the playback start, noted Ψ_i at formula 3.1.

Let $\Psi_{i,OFF}(q)$ be the conditional starvation probability of the tagged user such that:

- the tagged user buffer contains q seconds,
- the system is at state (i, OFF).

Remember that the the reduced throughput during playback is $c_i = d_i - 1$ at state (i, ON), and -1 at (i, OFF).Let t an instant during the prefetching period. In the infinitesimal time interval [t,t+h], events that may occur are :

- no change on the network ,
- arrival of one user (if the system is at ON) ,
- departure of on user (different from the tagged user and the system is at ON),
- entry at the OFF state (if the system is at ON state at t^-),
- entry at the ON state(if the system is at OFF state at t^-),
- entry at the absorbing state **A** (if the system is at ON state at t^-),
- occurrence of more than one events (the probability of this event is negligible comparing to other events when h go to 0.

We deduce:

$$\begin{cases} \Psi_{i,OFF}(q) = (1-\beta h)\Psi_{i,OFF}(q-h) + p_{i0}\beta h\Psi_{0,ON}(q-c_i h) + \\ \qquad \cdots + p_{ik-1}\beta h\Psi_{k-1,ON}(q-c_i h) + o(h) \\ \Psi_{i,ON}(q) = (1-(\lambda_i+\mu_i+\alpha_i)h)\Psi_{i,ON}(q+c_i h) \\ \qquad +\lambda_i h\Psi_{i+1,ON}(q+c_i h) \\ \qquad +\vartheta_i h\Psi_{i-1,ON}(q+c_i h) \\ \qquad +\alpha_i h\Psi_{i,OFF}(q-h) + o(h) \end{cases} \tag{3.13}$$

When h $\to$ 0, we obtain for all i in $\{0, 1, \cdots, K-1\}$:

$$\begin{cases} \dot{\Psi}_{i,OFF}(q) = -\beta\Psi_{i,OFF}(q) + \sum_{j=0}^{k-1} p_{ij}\beta\Psi_{j,ON}(q) \\ \dot{\Psi}_{i,ON}(q) = \frac{\lambda_i+\mu_i+\alpha_i}{c_i}\Psi_{i,ON}(q) - \frac{\lambda_i}{c_i}\Psi_{i+1,ON}(q) \\ \qquad -\frac{\vartheta_i}{c_i}\Psi_{i-1,ON}(q) - \frac{\alpha_i}{c_i}\Psi_{i,OFF}(q) \end{cases} \tag{3.14}$$

To find the starvations probabilities $(\Psi_{i,OFF}(q))_{1\leq i\leq K-1}$ and $(\Psi_{i,ON}(q))_{1\leq i\leq K-1}$, let Ψ be the probability vector defined as follows:

$$\Psi := [\Psi_{0,OFF}(q), \cdots, \Psi_{K-1,OFF}(q), \Psi_{0,ON}(q), \cdots, \Psi_{K-1,ON}(q)]^T$$

The system of equations 3.14, is equivalent to the following formula:

$$\dot{\Psi}(q) = N.\Psi(q) \tag{3.15}$$

such that:

$$\dot{\Psi} := [\dot{\Psi}_{0,OFF}(q), \cdots, \dot{\Psi}_{K-1,OFF}(q), \dot{\Psi}_{0,ON}(q), \cdots, \dot{\Psi}_{K-1,ON}(q)]^T,$$

and

$$N = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{3.16}$$

with:

B:=Diag($(\frac{-\alpha_i}{c_i})_{0\leq i\leq K-1}$), D:=Diag($(-\beta)_{0\leq i\leq K-1}$),
C:=$(P_{i,j}\beta)_{\substack{0\leq i\leq K-1 \\ 0\leq j\leq K-1}}$ and

A=: $\begin{pmatrix} \frac{\varpi_0}{c_0} & -\frac{\lambda_0}{c_0} & 0 & \dots & 0 & 0 \\ -\frac{\vartheta_1}{c_1} & \frac{\varpi_1}{c_1} & -\frac{\lambda_1}{c_1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & -\frac{\vartheta_{K-1}}{c_{K-1}} & \frac{\varpi_{K-1}}{c_{K-1}} \end{pmatrix}$

with $\varpi_i =: \lambda_i + \mu_i + \alpha_i$ for all i ∈ S and $\vartheta_0 = \lambda_{K-1} = 0$.

Finally, we find:

$$\Psi(q) = exp(q.N).\Psi(0) \tag{3.17}$$

where $\Psi(0)$ denotes the starvation probability with no initial prefetching.The boundary conditions are:

- when the system is at the OFF state and the buffer is empty, there is necessarily starvation. So $\Psi_{i,OFF}(0) = 1 \ \forall\, i \ \in \{0, 1, \dots, K-1\}$, which sets K components of Ψ_0.

- when the system is at ON state, if the arrival reduced throughput d_i is less than the reduced bitrate 1, and the buffer is empty, the starvation happens. So, $\Psi_{i,ON}(0) = 1 \ \forall\, i \ such \ that \ d_i < 1$.

- If for some i, $d_i = 1$, then $\Psi_i(0)$ depends linearly on $\Psi_{i-1}(0)$ and $\Psi_{i+1}(0)$.

Let m be the minimum value of i such that $d_i > 1$. So, we just fixed $2K - m$ equations on $\Psi_i(0)$. we still have m equations to define, to fix the remaining components of $\Psi_i(0)$. When q is infinitely large, the starvation vector $\Psi(q)$ equal to 0. Alternatively, $\lim_{q\to\infty} \Psi_{i,ON}(q) = \lim_{q\to\infty} \Psi_{i,OFF}(q) = 0$ for all i. So, N is a strongly diagonally dominant and irreducible matrix and then, similar to a diagonal Matrix. Hence, there exists an invertible matrix P, and a diagonal matrix D such that $N := PDP^{-1}$. So, from equation 3.17 we obtain:

$$\lim_{q\to\infty} exp(q.PDP^{-1})\Psi(0) = \lim_{q\to\infty} P.exp(q.D).P^{-1}\Psi(0) = 0 \tag{3.18}$$

Knowing that P is an invertible matrix, we get:

$$\lim_{q\to\infty} exp(q\ D).P^{-1}\Psi(0) = 0 \tag{3.19}$$

and then,

$$\left[P^{-1}\Psi(0)\right]_i = 0 \tag{3.20}$$

for all i corresponding to positives eigenvalues.

Applying the Gershgorin theorem [Wik14a] on our Matrix N, there must have $\left[P^{-1}\Psi(0)\right]_i = 0$ for all i corresponding to positives eigenvalues, which is exactly the number of i such that $d_i > 1$. Consequently, the unknowns $\Psi_i(0)$ can be derived.

Now, we interconnect the prefetching threshold q_a and the starvation probability $\Psi(q)$. The starvation event will take place when the video duration is longer than the prefetching threshold. A user with video duration longer than prefetching threshold can be regarded as a tagged user. After the prefetching event, the playback starts. The probability that the playback event starts is $\pi.\vee(0, q_a)$. Let $\Psi_{ON}(q) := [\Psi_{0,ON}(q), \Psi_{1,ON}(q) \cdots, \Psi_{K-1,ON}(q)]^T$. Knowing that the tagged user is the ON mode after the prefetching process, The starvation probability in 3.1 with a prefetching threshold q_a is expressed as follows:

$$\begin{aligned} P_{starv} &= \mathbb{P}\{T_{video} > q_a\}.\pi.\vee(0, q_a).\Psi(q_a) \\ &= exp(-\theta q_a).\pi.\vee(0, q_a).\Psi_{ON}(q_a). \end{aligned} \tag{3.21}$$

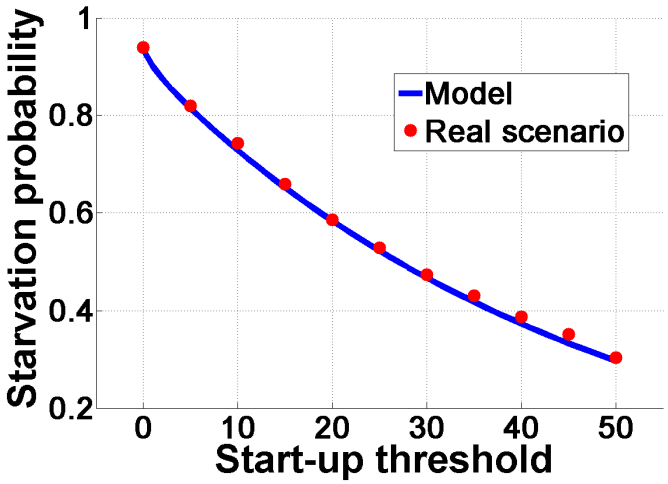


Fig. 3.7: Starvation Probability as a function of startup delay

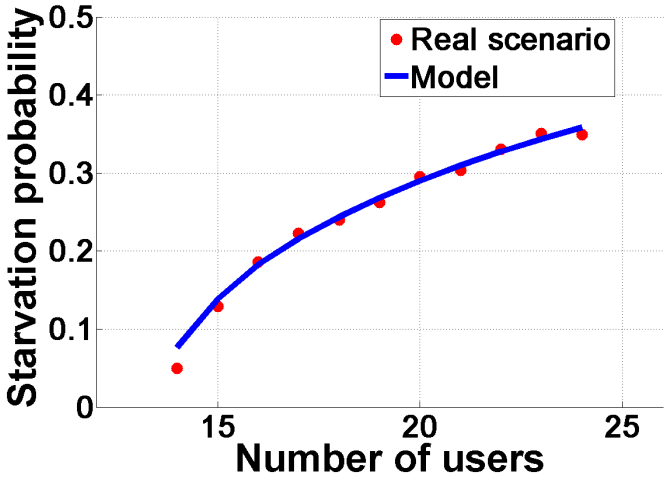


Fig. 3.8: Starvation Probability as a function of maximal number of users

3.3 Model validation

In this section, we present a validation of our Markovian model by comparing the probability of starvation obtained by the model with the ones obtained by simulating an actual client where the high and low water thresholds are fixed to a constant value. In order for both the Markov model and the actual client simulator, to perceive similar rate variations in the cells, we use the same model for clients arrivals and departures.

We use the following network set-up: the capacity of the cell is set to 6 Mbps, the video bitrate is 480 kbps, the mean video duration is 5 minutes and a maximum number of 16 simultaneously streaming users. These values are based on the operational deployment of a typical European mobile operator. We set the low water threshold equal to the start up threshold and the high water threshold equal to 25 Mbits (about 52 seconds of video).

Figure 3.7 shows the starvation probability with different values for the startup threshold. The OFF period is set to 50 seconds.

When the startup threshold increases from 0 to 50s of video content, the starvation probability decreases as expected. Our Markovian model matches the real client very well.

We consider a scenario in which we investigate the impact of the number of users. We let $\lambda = \frac{1}{25}s^{-1}$. Figure 3.8 pictures the variation of the starvation probability when the maximal number of users changes. The larger the maximal number of users, the larger the starvation probability. Again, the Markovian model matches very well with the actual client behavior.

3.4 Starvation probability: On-Off vs. Fast caching

In this section we compare the impact of the ON/OFF strategy on starvation probability to the fast-caching scenario that constitutes the best case scenario. For this purpose, we used simulation and our model (fast caching corresponds to ON duration $\geq$ the video size). Using both methods, we explored a variety of scenarios in which we vary: (i) the initial startup delay, (ii) the intensity of arrivals or (iii) the maximum admissible number of users in the cell. These three scenarios correspond to Figures 3.9, 3.10 and 3.11 respectively obtained from the theoretical models. The results from simulations are similar and not presented here. We observe that in all scenarios, the starvation probability of the Fast caching and On-Off strategies are very similar. A slight discrepancy is noticeable at low load in Figure 3.10. This is due to starvation events in off periods that would not be observable in practice. However, this discrepancy appears only for small values of probability of starvation.

These results show that using ON/OFF is a good strategy. Indeed, while ON/OFF can reduce the loss of energy and bandwidth due to abandoning behaviour, it does not impact the probability of starvation.

3.5 QoE optimization: starvation probability vs start-up threshold

The impatience of user watching a video (how long she is willing to wait before the playback starts) is a key metric for QoE. Let t be the impatience time limit. One would want to choose a small prefetching threshold q_a in order to maximize the number of videos starting before the impatience limit t. However, a small prefetching threshold leads to high starvation probability as shown in Section 4.3.3. In this section, we explore the optimization of the trade-off between customers impatience and probability of starvation.

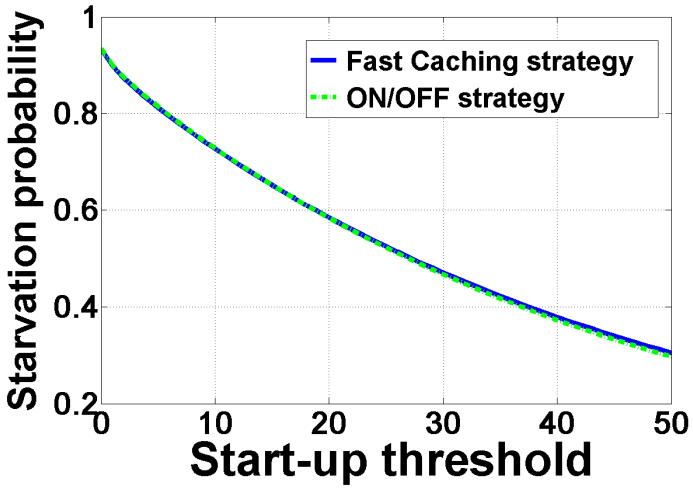


Fig. 3.9: Starvation Probability as a function of startup delay for ON/OFF and Fast Caching streaming strategies

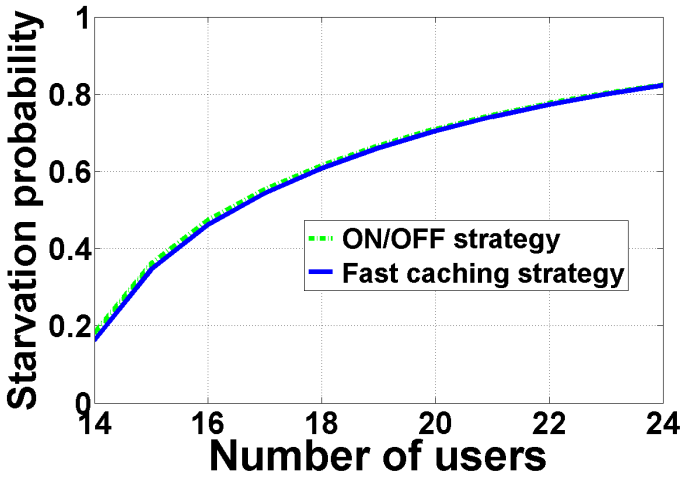


Fig. 3.10: Starvation Probability as a function of maximal number of users

3.5.1 QoE indicators

To satisfy user's impatience, it is necessary that video's playback begins before t. Let $U_m(q,t)$ be the probability of downloading a quantity q of video during the startup phase, given that the user is ready to wait at most t seconds before starting to watch the video. The user is satisfied when $U_m(q,t)$ is equal to 1, which means that she ends downloading q before time t and before video's playback. Thereafter, we use $U_m(q,t)$ and the probability of starvation P_{starv} as user utilities functions. Note that these two utilities are in direct competition: a high startup threshold q_a reduces the probability of starvation P_{starv} but increases the dissatisfaction of user's impatience $U_m(q,t)$, and vice-versa.

Figure 3.12 shows the variation of both the probability of non starvation $1 - P_{starv}$ and U_m with respect to q, the start-up threshold for a start-up delay equal to 10s, an arrival rate of $\frac{1}{20}s^{-1}$, a capacity equal to 6 Mbps, playback speed of 480 Kbs, a video of 5 minutes and a maximum of 18

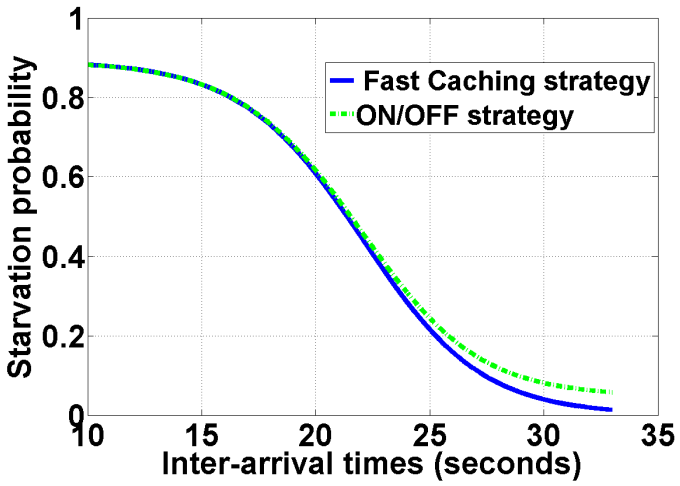


Fig. 3.11: Starvation Probability as a function of inter-aarival timefor ON/OFF and Fast Caching streaming strategies($q_a = 10s$, $\beta = \frac{1}{50}s^{-1}$ and 16 users)

users. This figure confirms what we said: there is a conflict between our two metrics, namely the probability of non starvation and the user's impatience.

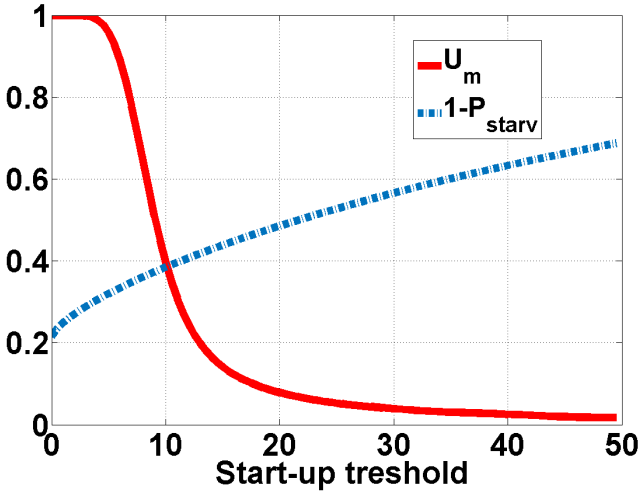


Fig. 3.12: the variation of the probability of non starvation and U_m as a function of start-up threshold($\lambda = \frac{1}{20}$, number of users=18 and startup delay=10s)

3.5.2 Goal function

To balance these two conflicting utilities, we define a cost function that describes a subjective tradeoff between P_{starv} and $U_m(q,t)$. The cost function can take many forms, see [MAri]. We used as a cost function the weighted product that enables to couple the two quantities.

Let consider the cost function as follows:

$$G_\gamma(t) := \max_q U_m(q,t)^\gamma (1 - P_{starv}(q))^{1-\gamma} \tag{3.22}$$

where γ is a positive constant in the interval $[0, 1]$ that specifies the relative weight of the impatience and probability of starvation from the user perspective.

Using the product in Equation 3.22 coerces both utilities to avoid extreme cases in the optimal trade-off choice, i.e. when q becomes large, $U_m(q,t) \to 0$ as there is no chance that q be downloaded in t seconds. However, the probability of starvation ramps up to one as most of the video (all the video in the extreme case) has been downloaded during the startup phase. When q tends to 0, $U_m(q,t) \to 1$ as it is more likely that q be downloaded in t seconds and the probability of starvation tends to 0.

Similarly to [Xu+13], we define $U_i(q,t)$ as the probability to read q seconds of video content before time t knowing that at the beginning of the lecture there are $i+1$ users (i users plus the tagged one) in the system, for all $i \in S$. $U_i(q,t)$ satisfies the partial differential equation of first order (Equation 3.23), which we can solve numerically.

$$\begin{aligned}\frac{\partial U_i}{\partial t} = -b_i \frac{\partial U_i}{\partial q} - (\lambda_i + \vartheta_i) U_i(q,t) \\ +\lambda_i U_{i+1}(q,t) + \vartheta_i U_{i-1}(q,t), \ \ \forall i \ \in \ S\end{aligned} \tag{3.23}$$

with the initial condition

$$U_i(q,0) = 0, \ \ \forall\, q > 0; \tag{3.24}$$

and the boundary conditions

$$U_i(0,t) = 1, \ \ \forall\, t \geq 0, \tag{3.25}$$

$$\lim_{q \to +\infty} U_i(q,t) = 0, \ \ \forall\, t \geq 0. \tag{3.26}$$

We then define $U_m(q,t)$ as $U_m(q,t) := U_m := \pi U(q,t)$ where π is the row vector of probabilities π_i defined in Equation (4.3) and $U(q,t)$ the column vector of $U_i(q,t)$ for all i in S.

At this step of work, we will not able to solve Equation 3.23. In fact, we attempt to find the expressions of $U_i(q,t)$ but we did not get the solution. We will see how we get that in next chapter. However, we propose a numerical solution.

Numerical solution of U$_i(q,t)$ for all i in 0,1,...,K-1

To solve the partial Equation 3.23 and find $U_i(q,t)$, we use the finite difference method [Wik17] on the space formed by q and t (start-up threshold and start-up delay respectively).

Let suppose that q and t vary in [0,q_{max}] and [0,t_max] respectively and let p_q and p_t the quantization steps of the two variables q and t.

Our goal is to solve numerically the following system of equation:

$$\begin{cases} \dfrac{\partial U_i}{\partial t} = -b_i \dfrac{\partial U_i}{\partial q} - (\lambda_i + \vartheta_i) U_i(q,t) + \lambda_i U_{i+1}(q,t) + \vartheta_i U_{i-1}(q,t), \ \ \forall i \in S \\ U_i(q,0) = 0, \ \ \forall\, q > 0, \ U_i(0,t) = 1, \ \ \forall\, t \geq 0, \ \lim\limits_{q \to +\infty} U_i(q,t) = 0, \ \ \forall\, t \geq 0. \end{cases} \tag{3.27}$$

Let $(t)_{0 \leq i \leq \frac{t_{max}}{p_t}}$ and $(q)_{0 \leq i \leq \frac{q_{max}}{p_q}}$ the sequence of discrete points of t and q respectively.

We have:

$$\frac{\partial U_i(q_j,t_n)}{\partial t} \approx \frac{U_i(q_j,t_{n+1}) - U_i(q_j,t_n)}{p_t} \ and \ \frac{\partial U_i(q_j,t_n)}{\partial q} \approx \frac{U_i(q_j,t_n) - U_i(q_{j-1},t_n)}{p_q} \tag{3.28}$$

Then, the fisrt equation in the system 3.27 becomes:

$$\frac{U_i(q_j,t_{n+1}) - U_i(q_j,t_n)}{p_t} = -b_i(\frac{U_i(q_j,t_n) - U_i(q_{j-1},t_n)}{p_q}) - (\lambda_i + \vartheta_i) U_i(q_j,t_n) + \lambda_i U_{i+1}(q_j,t_n) + \vartheta_i U_{i-1}(q_j,t_n) \tag{3.29}$$

which implies:

$$\begin{aligned} U_i(q_j,t_{n+1}) = U_i(q_j,t_n) - b_i \frac{p_t}{p_q}(\frac{U_i(q_j,t_n) - U_i(q_{j-1},t_n)}{p_q}) - p_t(\lambda_i + \vartheta_i) U_i(q_j,t_n) + p_t \lambda_i U_{i+1}(q_j,t_n) \\ + p_t \vartheta_i U_{i-1}(q_j,t_n) \end{aligned} \tag{3.30}$$

Then:

$$U_i(q_j,t_{n+1}) = (1 - b_i \frac{p_t}{p_q} - p_t(\lambda_i + \vartheta_i)) U_i(q_j,t_n) + b_i \frac{p_t}{p_q} U_i(q_{j-1},t_n) + p_t \lambda_i U_{i+1}(q_j,t_n) + p_t \vartheta_i U_{i-1}(q_j,t_n) \tag{3.31}$$

Let $\sigma_i = 1 - b_i \frac{p_t}{p_q} - p_t(\lambda_i + \vartheta_i)$, $\omega_i = b_i \frac{p_t}{p_q}$ and $\mathbf{U}_{i,t_{n+1}} = (U_i(0,t_{n+1}), U_i(p_q,t_{n+1}), ..., U_i(p_q[\frac{q_{max}}{p_q}], t_{n+1}))$.
We have then:

$$\mathbf{U}_{i,t_{n+1}} = A_i \mathbf{U}_{i,t_n} + B_i \mathbf{U}_{i+1,t_n} + C_i \mathbf{U}_{i-1,t_n} \tag{3.32}$$

With :

$$A_i = \begin{pmatrix} 1 & 0 & 0 & \dots & \dots & \dots & 0 \\ \omega_i & \sigma_i & 0 & \dots & \dots & \dots & 0 \\ 0 & \omega_i & \sigma_i & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & \omega_i & \sigma_i & 0 \\ 0 & \dots & \dots & \dots & 0 & \omega_i & \sigma_i \end{pmatrix} \tag{3.33}$$

and $B_i = diag((\lambda p_t)_i)$ and $c_i = diag((\vartheta_i p_t)_i)$.

Our problem will be resolved in recurrence following the hyperplans $(t, q, i)/t = t_i$ and here is the algorithm solving solving problem 3.27:

Require: $U_i(q, 0)$, A_i, B_i and C_i for all i
  for t from 0 to t_{max} **do** $\mathbf{U}_{i,t+1} = A_i\mathbf{U}_{i,t} + B_i\mathbf{U}_{i+1,t} + C_i\mathbf{U}_{i-1,t}$
  end for

By solving $U_i(q, t)$, we are able to link directly the start-up delay and start-up threshold.

We let λ be $\frac{1}{20}$, the capacity be 6 Mbs, the playback speed be 480 kbs, the video duration be 5 minutes and 18 users. figure 3.13 shows the variation of U_m as a function of start-up Threshold for different values of start-up delay. For fixed t, U_m is a decreasing function from 1 to 0 with respect to start-up threshold. In fact, when $q = \epsilon$ and for $t > 0$, $U_m = 1$. This is due to the fact that we are sure to reach reading q with the downloading speed before t. Also and for a fixed q, U_m increases with respect to t because, more user can wait before starting playback, more we have chance to read q.

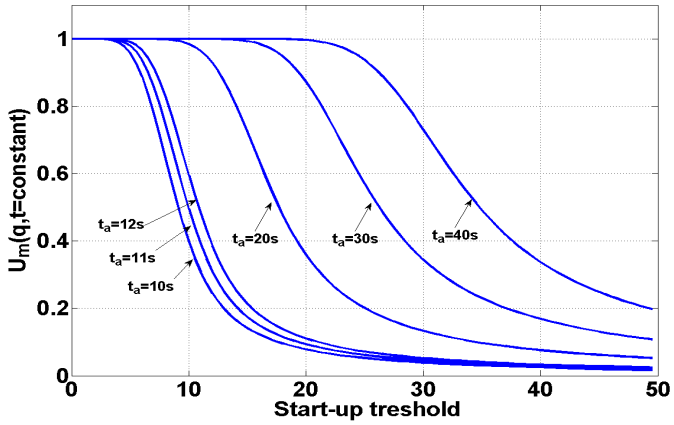


Fig. 3.13: U_m as a function of start-up threshold for different values of start-up delay t_s

For a value of star-up delay equal to 10s, figure 3.13 shows that more than 99 % of users will be satisfied if the start-up threshold is lower than 5 seconds of video content. To satisfy users, we should maximize U_m.

Next, we will study the trade-off between the start-up delay and starvation probability by analysing the simulation of our goal function 3.22.

3.5.3 Optimization of the goal function: results

We set the maximum number user to 18 and explore the cost function $G_\gamma(t)$ for several loads λ using two values for impatience : 10 seconds and 20 seconds.

Figures 3.14 and 3.15 (resp. 3.16 and 3.17), we represent $G_\gamma(t)$ for an impatience of 10 s (resp. 20 s) for $\lambda = 1/20$ and $\lambda = 1/30$ respectively. We use several values of γ to explore the importance of the impatience or starvation for the end users. We remark that for small values, impatience is not important compared to starvation, the maximum is of q_a is near the size of video. However, when γ is larger than a certain threshold, the maximum of the cost function is around a small value q_a. For instance, the optimal q_a is equal to 5 seconds worth of video for impatience t of 10 seconds.

If we consider that the users are equally sensitive to impatience and starvation probability i.e. $\gamma = 0.5$, on could find an optimal value for startup threshold that satisfy the majority of users.

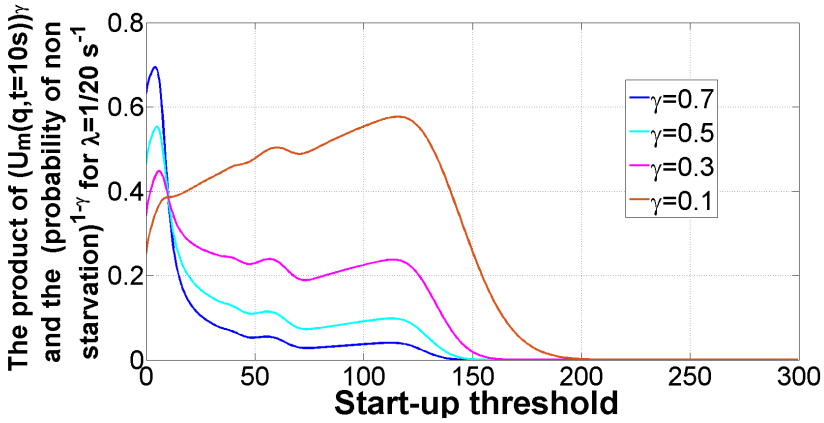


Fig. 3.14: The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{20}$, number of users=18 and startup delay=10s)

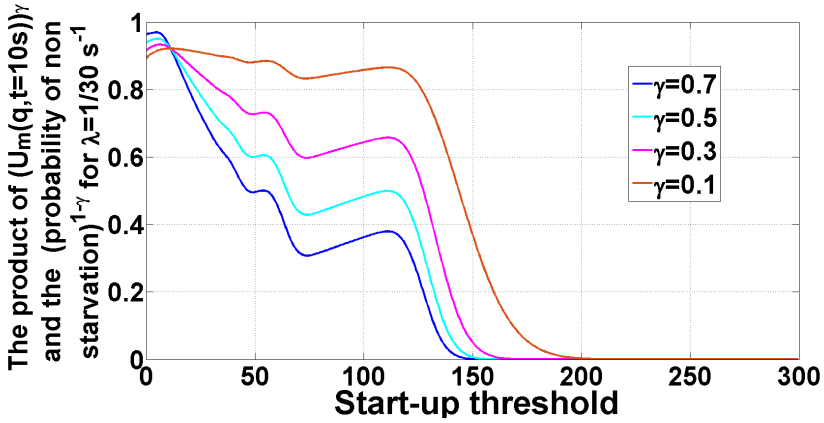


Fig. 3.15: The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{30}$, number of users=18 and startup delay=10s)

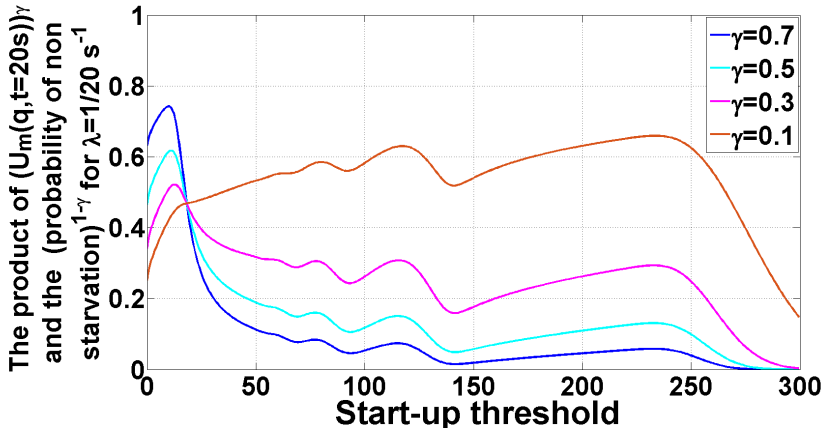


Fig. 3.16: The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{20}$, number of users=18 and startup delay=20s)

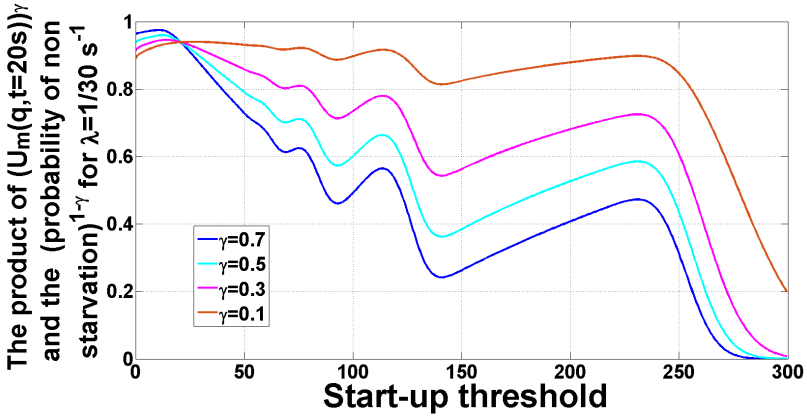


Fig. 3.17: The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{30}$, number of users=18 and startup delay=20s)

Finally, the weighted product's optimization has allowed us to seek and find the best startup threshold in order to satisfy user's impatience and video's freezes. In a network where the majority of customers are impatient to start playback of a video , it is preferable to set a value around 5 s of data accumulated in the buffer before beginning playback. If the customers experience is badly affected by video's freezes, it is more preferable to fix a high value of startup threshold to decrease the chance of freezing during the video playback.

3.6 Conclusion

dation In this chapter, we tackled the problem of modelling and optimizing the QoE metrics (i.e. the starvation probability and impatience) of media streaming service in mobile networks in the case of On-Off downloading strategy. We demonstrate that our Markovian-based model captures faithfully the behaviour of an actual client. We also showed that the use of ON/OFF strategy that was introduced to handle the premature departure of users does not increase the probability of starvation. Finally, we explored the trade-off between the impatience of customers and probability of starvation, and demonstrated the existence of an optimal startup threshold that can be used to maximize client

satisfaction. In the next chapter, we will quantify and compute loss (in term of energy) resulting from abandoning behaviours. We will also explore the optimization of this loss while considering the starvation phenomenon and a certain threshold level for start-up threshold.

4 Evaluation and optimization of QoE of On-Off strategy: Loss due to abondment

> *If you would be a real seeker after truth, it is necessary that at least once in your life you doubt, as far as possible, all things.*
>
> — **Rene Descartes**
> (Mathematician.)

Video data constitutes the majority of traffic in bytes that mobile and fixed line operators deliver to their customer. This type of traffic is both resource consuming and QoE sensitive. However, either because of content quality or QoE, a large fraction of users often abandon viewing prematurely. These abandonment phenomena lead to a huge waste of network resources and device batteries. Several strategies have been devised to account for all those dimensions. Dominant approaches are fast-caching where the server pushes traffic as fast as possible to the client in order to limit starvation, and On-Off strategy where the client forces the server to pause the transfer regularly in order to mitigate the wasted bytes and energy due to users abandonment. In this work, we focus on Fast-Caching and on On-Off strategies. In this chapter, we focus on the optimization of starvation probability and the wasted resource due to user behavior. We propose an analytic model to quantify the global bandwidth loss due to user abandonment for both On-Off and Fast Caching strategies. To the best of our knowledge, this is the first attempt to quantify the unused bytes. We formulate a multi-objective optimization problem to find the ON and OFF periods that strike a good trade-off between starvation probability and the wasted bytes.

4.1 From starvation phenomenon to loss due to abandonment

In chapter 3, we worked on starvation probability as indicator of the quality of experience. We developed a model for the On-Off strategy and derive an analytic expression of starvation probability as a function of two groups of parameters:

- Decision parameters: start-up delay, ON period and OFF period, and,

- Parameters related to the demand: throughput, bitrate, user's arrival rate, video duration.

We saw that when we increase the start-up delay, the probability that our video freezes later during playback decreases. However, a large value of start-up delay has two major disadvantages:(i) the user must wait a lot before video's playback, and (ii) loss of energy if the customer abandons reading. Concerning the first point (i), we developed what we called user impatience in Section 3.5.1 to study and take into account this phenomenon. For the second point (ii), we should find a way to estimate the amount of data lost if the customer decides to abandon viewing a video. In fact, we should be able to quantify the amount of buffered data at every time during a streaming session. Therefore, we need to estimate the average amount of buffered data, at a time during playback, while streaming a video. The goal of this estimation is twice: show the impact and benefits of the On-Off strategy comparing to Fast Caching and also, study the variation of buffered data as a function of On and Off duration. Consequently, our first objective is to estimate (compute) the amount of data in the buffer of a user who streams. Our work is motivated by the fact that telecommunication operators try to quantify and then limit the amount of streaming traffic sent and not watched by their customer. To do that, we place ourselves on a customer point of view and analyze the QoE indicator due to his abandonment. We keep the same user's dynamic defined in chapter 3.

The rest of the chapter is organized as follows. In Section 4.2 we define the network model. Section 4.3 presents the analytical formulation of the loss due to user abandonment. In Section 4.3.1, we compute the distribution of buffered data for the On-Off strategy. We use the distribution of buffered data to compute the loss due to abandonment for the On-Off scenario. We explore then the case of Fast Caching streaming strategy in Section 4.3.2 and compute the loss in this case. In Section 4.4 we present the optimization problem and application results. Finally, in Section 4.5 we conclude our chapter.

4.2 Network model

In this section we present our model. We keep the same user's dynamic defined in the chapter 3. Thus, we consider a cell using admission control. If the network is saturated, any request for additional user access will be denied. A user joins the network requests a video and leaves the network after completing the download of the entire video. At the client side, downloaded video packets are stored in the play-back buffer and played at a deterministic rate depending on the video encoding rate. As mentioned before, viewing of the video starts after an initial startup delay[1]. In the Fast Caching mode, the video download continues during the viewing phase until the video has been fully downloaded. In contrast, in the On-Off mode, the download stops when the playback buffer has reached the so-called high buffer threshold (see Section 1.3.6) and resumes when the playback process has consumed a fixed amount of video (up to the so-called low buffer threshold).

In chapter 3, we studied the On-Off strategy and derived the starvation probability using a Markovian model. We have shown how to adjust the start-up delay to avoid starvation during playback.

Guaranteeing download of a certain amount of data in order to avoid starvations is good but, we must also avoid downloading too much data. If a streaming user downloads a lot of data, she will have enough data to avoid video starvation but, she will have also an energy loss if she abandons

[1] A waiting time before start reading a video

the playback of the video: she consumed energy of her device to download data that has not been read. Thus, we introduce an indicator representing the average loss if the user abandons the video playback called:

- **loss due to abandonment:** it represents the average number of seconds, downloaded and unread, if the user decides to leave the playback of the video, before the end of the video.

We place ourselves on a customer point of view. We track the performance of a tagged user downloading a video lasting T_{video}.

Let C be the aggregated capacity of the cell in bps and Bitrate be the encoding rate of the video, also in bps. We assume that each user receives a throughput of $\frac{C}{i}$ where i is the number of connected users.

We model the arrival and departure of users as a birth-death Markov chain:

- **Arrival of users:** arrivals follow a Poisson process with parameter λ.

- **Departure of users:** at time t, the time remaining for a fixed user, different from the tagged user, to leave the network follows an exponential distribution with parameter ν_i where i is the number of connected users. This is justified by the fact that the video duration, T_{video}, is assumed to be exponentially distributed. The transitions ν_i are expressed as follow: $\nu_i = \frac{i.C}{(i+1).Bitrate.T_{video}}$ because there are i users, other than the tagged one, connected to the base station and each one has a throughput of $\frac{C}{i+1}$ at state i.

After a prefetching phase during which the tagged user buffers q_a seconds of data, the playback starts and she switches between the ON and OFF mode. We model this ON-OFF system as a Markov chain shown in Figure 4.1. Here, our vision is different from the one in chapter 3. In fact, The On-Off strategy defined in this chapter is server oriented: On-Off-S. Different from the On-Off defined in the previous chapter in which we switch from On period to Off period when the buffered data reaches a high threshold, the On-Off-S is a On-Off strategy in which the server (i.e service provider) pushes Q_{On} bytes to terminal and stops sending video data during T_{Off}. In the rest of this chapter, On-Off refers to On-Off-S Let K be the maximum number of admitted users in the cell. There are $2K + 1$ possible states. The $2K$ first states are the product of the number of users connected combined with the two downloading states for the tagged user (ON and OFF). We denote each of these states in the form (i, ON) or (i, OFF) where i is the number of connected users other than the tagged user. The user enters an OFF state when she downloads an amount of data Q_{on} in byte. The duration of state (i,ON), measured in seconds, is a random variable following an exponential distribution with parameter $\frac{1}{\alpha_i} = \frac{(i+1).Q_{on}}{C}$. This is justified by the fact that at state i, the throughput of the tagged user is $\frac{C}{i+1}$.

The user switches from ON state to OFF state when she downloads Q_{on} bytes. We model the OFF duration as a random variable following an exponential distribution with parameter $\frac{1}{\beta}$. We introduce the transitions $p_{i,j}.\beta$ in order to keep the information on the system dynamics during OFF periods. $p_{i,j}$ is the probability that the tagged user enters the OFF period at state i and ends at state j.

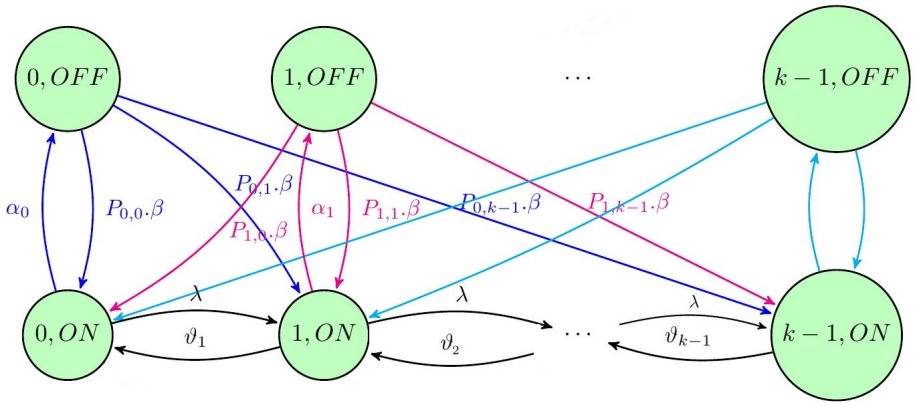


Fig. 4.1: User's dynamics form the tagged user point of view

$\mathbf{p} = (p_{ij})_{0 \leq i,j \leq K-1}$ is expressed as follows [Bou+16]:

$\mathbf{p}p = \mathbf{F}_{i,j}(0, T_{OFF}) = exp(-T_{OFF}.R)$ with:

$$R = \begin{pmatrix} \lambda_0 & -\lambda_0 & 0 & \dots & 0 & 0 \\ -\vartheta_1 & \lambda_1 + \vartheta_1 & -\lambda_1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & -\vartheta_{K-1} & \vartheta_{K-1} \end{pmatrix} \tag{4.1}$$

We use the following notations in the remainder of this chapter. $C_r = \frac{C}{bitrate}$ is the reduced capacity in seconds of video contents. We consider that all users have the same SNR and the channel is static. If the system is in state i, each user receives an identical reduced throughput denoted $b_i := \frac{C_r}{i+1}$.

In the next section, we introduce and formulate an analytic formula to our QoE Loss due to abandonment indicator.

4.3 Loss due to the abandonment

Our purpose here is to construct the complete model to compute the loss due to the abandonment of the tagged user as a function of startup delay q_a, and OFF and ON duration. Let $(\pi_i)_{0 \leq j \leq K-1}$ be the stationary probability of having i users in the system when the tagged user joins the network, and $Q_i(t)$ the downloaded data in seconds given that she starts the playback at state i (i connected users other than the tagged user) and the playback time is at second t. The loss due to abandon, $Loss_{abn}$, can be expressed as:

$$Loss_{abn} = \int_0^{T_{video}} f_{abn}(t).[\sum_{i=0}^{K-1} \pi_i.Q_i(t) - t + q_a].\mathrm{d}t \tag{4.2}$$

with $f_{abn}(t)$ is the distribution function of abandon at time t, and q_a is the start-up delay. Note that we seek the values of f_{abn} numerically by analyzing a streaming data traces. The cumulative distribution function (C.D.F) of abandon is shown in Figure **??**.

The stationary probability π_i is defined as follows [Ber+92]:

$$\pi_i = \frac{\rho^i(1-\rho)}{1-\rho^K}, \ \forall i = 0, ..., K-1. \tag{4.3}$$

where $\rho := \frac{\lambda}{\mu}$. Note that $\pi_K = 0$ because the network can serve a maximum of K users simultaneously.

$Loss_{abn}$ depends on several parameters namely the OFF duration, the quantity Q_{on} and the start-up threshold q_a.

Note that formula 4.2 can be applied for all streaming strategies. The case of Fast Caching will be studied later.

From this previous formula, it is obvious that to compute the loss due to abandonment, we should find a way to compute or even estimate the quantity of downloaded data Q_i: it is the only solution to completely calculate $Loss_{abn}$. The next section provides an answer to this problematic by proposing a technical solution for computing $Loss_{abn}$ in the case of the On-Off streaming strategy.

4.3.1 Probability distribution of buffered data $Z(q,t)$ for the On-Off streaming strategy

Our purpose here is to compute the quantity of downloaded data $Q_i(t)$ (see Eq. 4.2) while using ON-OFF strategy as a function of the start-up delay, noted q_a, OFF and ON durations. Typically, we want to know how much data Q(t), the tagged user downloads when she is at second t of playback. To solve this problem, it is difficult to use a direct approach whereby we consider that the evolution of the user's buffer starts from the value of start-up delay and calculate the buffer level at time t, Q(t). So, we will work and solve the following dual problem: Let's assume that a tagged user has Q(t) seconds of buffered data at the playback time t. The goal is then to compute the probability distribution to read this Q seconds with a speed of $\frac{capacity}{Bitrate(i+1)}$, before time t. In this dual problem, the buffer dynamics evolves in an infinitesimal interval [0,h] (h$\geq$0) according to:

$$Q(t+h) = Q(t) - \frac{capacity}{Bitrate(i+1)} * h \tag{4.4}$$

Let's $Z_i(q,t)$ be the probability that the buffer starts with q seconds and reaches 0 seconds before t, following Equation 4.4 and using ON-OFF strategy. Figure 4.1 represents the user's dynamics.

Let t be an instant of playback. In the infinitesimal time interval [t,t+h], the events that may occur are:

- no change on the network ,
- arrival of one user (if the system is at ON) ,
- departure of one user (different from the tagged user and the system is at ON),
- entry in the OFF state (if the system is at ON state at t^-),
- entry in the ON state (if the system is at OFF state at t^-),
- occurrence of more than one events (the probability of this event is negligible comparing to other events when h go to 0.

We deduce:

$$\begin{cases} Z_{i,OFF}(q,t) = (1-\beta h)Z_{i,OFF}(q,t-h) \\ \qquad\qquad +p_{i0}\beta h Z_{0,ON}(q,t-h) + \cdots \\ \qquad\qquad +p_{ik-1}\beta h Z_{k-1,ON}(q,t-h) + o(h) \\ Z_{i,ON}(q,t) = (1-(\lambda_i+\vartheta_i+\alpha_i)h)Z_{i,ON}(q-b_i h,t-h) \\ \qquad\qquad +\lambda_i h Z_{i+1,ON}(q-b_i h,t-h) \\ \qquad\qquad +\vartheta_i h Z_{i-1,ON}(q-b_i h,t-h) \\ \qquad\qquad +\alpha_i h Z_{i,OFF}(q,t-h) + o(h) \end{cases} \tag{4.5}$$

When h $\to$ 0, we obtain for all i in $\{0,1,\cdots,K-1\}$:

$$\begin{cases} \dfrac{\partial Z_{i,OFF}(q,t)}{\partial t} = -\beta Z_{i,OFF}(q,t) + \displaystyle\sum_{j=0}^{K-1} P_{i,j} Z_{j,ON}(q,t) \\ b_i \dfrac{\partial Z_{i,ON}(q,t)}{\partial q} + \dfrac{\partial Z_{i,ON}(q,t)}{\partial t} = -\omega_i Z_{i,ON}(q,t) \\ \qquad\qquad +\lambda_i Z_{i+1,ON}(q,t) \\ \qquad\qquad +\vartheta_i Z_{i-1,ON}(q,t) \\ \qquad\qquad +\alpha_i Z_{i,OFF}(q,t) \end{cases} \tag{4.6}$$

With $b_i = \frac{capacity}{Bitrate(i+1)}$ and $\omega_i =: \lambda_i + \vartheta_i + \alpha_i$ for all i $\in$ S.

The initial conditions are :

$$Z_{i,ON}(q,0) = 0\ ,\ Z_{i,OFF}(q,0) = 0\ \forall\ q > 0 \tag{4.7}$$

and the boundary conditions are:

$$Z_{i,ON}(0,t) = 1,\ Z_{i,OFF}(0,t) = 1\ \forall\ t > 0, \tag{4.8}$$

$$\lim_{q\to\infty} Z_{i,ON}(q) = \lim_{q\to\infty} Z_{i,OFF}(q) = 0\ \forall\ t > 0. \tag{4.9}$$

The initial condition in Equation 4.7 means that we cannot read q (> 0) seconds of data with speed b_i if we don't have time to read. The first boundary condition says that if there is no data to read (i.e: $q = 0s$), the probability $Z_{i,\{ON,OFF\}}(q,t)$ is 1. The second boundary condition says that if we have a lot of data, it is not possible to all read before t.

Let M be the matrix defined as follow:

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{4.10}$$

with:

B:=Diag($(-\alpha_i)_{0\leq i\leq K-1}$), D:=Diag($(\beta)_{0\leq i\leq K-1}$),
C:=$-(P_{i,j}\beta)_{\substack{0\leq i\leq K-1 \\ 0\leq j\leq K-1}}$ and

A=: $\begin{pmatrix} \omega_0 & -\lambda_0 & 0 & \dots & 0 & 0 \\ -\vartheta_1 & \omega_1 & \lambda_1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & -\vartheta_{K-1} & \omega_{K-1} \end{pmatrix}$

with $\vartheta_0 = \lambda_{K-1} = 0$.

M is a strongly diagonally and irreducible matrix and then, similar to a diagonal matrix. Hence, there exists a diagonal matrix D and a transition matrix P such that $M = P.D.P^{-1}$. The solution of Equation 4.6 is given by [Xu+13] by:

$$Z(q,t) = P.exp(-t.D).P^{-1}.S(q,t) \tag{4.11}$$

with

$$\begin{cases} Z(q,t) = [Z_{0,ON}, ..., Z_{K-1,ON}, Z_{0,OFF}, ..., Z_{K-1,OFF}]^T \\ S(q,t) = [S_{0,ON}, ..., S_{K-1,ON}, S_{0,OFF}, ..., S_{K-1,OFF}]^T \\ S_{i,ON}(q,t) = \begin{cases} 0 \ if \ q - b_i t \ > \ 0 \\ 1 \ if \ q - b_i t \ \leq \ 0 \end{cases} \\ S_{i,OFF}(q,t) = \ 0 \end{cases} \tag{4.12}$$

In this next section, we study the case of the Fast Caching strategy and show how to derive the equations to compute the distribution of buffered data.

4.3.2 Probability distribution of buffered data $Z(q,t)$ for Fast Caching streaming strategy

Let's $Z_{i,fc}(q,t)$ be the probability that the buffer starts with q seconds and reach 0 seconds before t, following Equation 4.4 and using the Fast Caching strategy.

In the case of Fast Caching, the system of equations 4.6 is reduced to:

$\forall\, i \in [0, K-1]$.

$$\begin{aligned} b_i \frac{\partial Z_{i,fc}(q,t)}{\partial q} + \frac{\partial Z_{i,fc}(q,t)}{\partial t} = &-(\lambda_i + \vartheta_i) Z_{i,fc}(q,t) \\ &+\lambda_i Z_{i+1,fc}(q,t) \\ &+\vartheta_i Z_{i-1,fc}(q,t) \end{aligned} \tag{4.13}$$

with the initial condition

$$Z_{i,fc}(q,0) = 0, \;\; \forall\, q > 0; \tag{4.14}$$

and the boundary conditions

$$Z_{i,fc}(0,t) = 1, \;\; \forall\, t \geq 0, \tag{4.15}$$

$$\lim_{q \to +\infty} Z_{i,fc}(q,t) = 0, \;\; \forall\, t \geq 0. \tag{4.16}$$

and the solution is:

$$Z_{fc}(q,t) = P.exp(-t.D).P^{-1}.S(q,t) \tag{4.17}$$

with:

$$\begin{cases} Z_{fc}(q,t) = [Z_{0,fc}, ..., Z_{K-1,fc}]^T \\ S(q,t) = [S_0, ..., S_{K-1}]^T \\ S_{i,fc}(q,t) = \begin{cases} 0 \;\; if \;\; q - b_i t \;\; > \;\; 0 \\ 1 \;\; if \;\; q - b_i t \;\; \leq \;\; 0 \end{cases} \\ R = P.D.P^{-1} \end{cases} \tag{4.18}$$

and R is the matrix defined at section 4.2.

In the next section, we show our method to compute the QoE indicator $Loss_{abn}$. We present then simulations and results pf our model.

4.3.3 Model result and simulation

In this section, we present the results of our Markovian model. To compute the buffered data $Q_{i,ON-OFF}(t)$ defined at Eq. 4.2, we seek the set of values q satisfying Eq. 4.6 following the distribution $Z(q,t)$, and then take the mean over i to compute $Q_{average,ON-OFF}(t) = \sum_{i=0}^{K-1} \pi_i . Q_{i,ON-OFF}(t)$ the average buffered data for all t $\in$ [0, T_{video}].

We use the following network set-up: the capacity of the cell is set to 6 Mbps, the video bitrate is 480 kbps, the mean video duration is 5 minutes, the start-up threshold is at 10 seconds, the inter arrival time at 35 seconds, and a maximum number of 18 simultaneously streaming users. These values are based on the operational deployment of a typical European mobile operator. We set the OFF period duration to 50 seconds and a quantity Q_{on} equal to 40 Mbits.

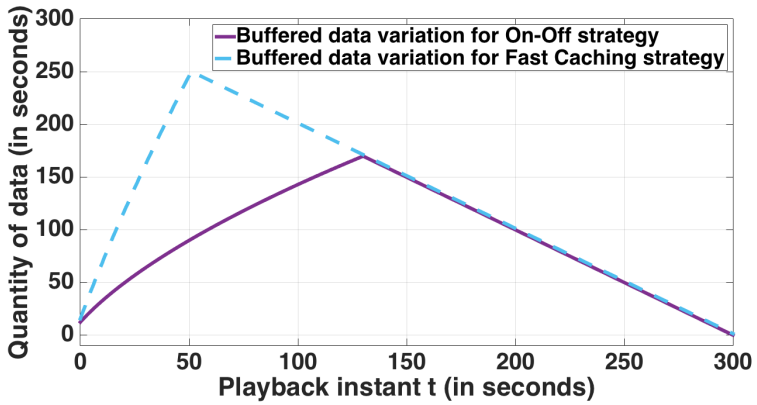


Fig. 4.2: downloaded and buffered data as a function of the advancement of playback time for ON-OFF and Fast Caching streaming strategies($q_a = 5s$, $Q_{on} = 40\ Mbites$, $\beta = \frac{1}{50}s^{-1}$, $\frac{1}{35}s^{-1}$ and 18 users)

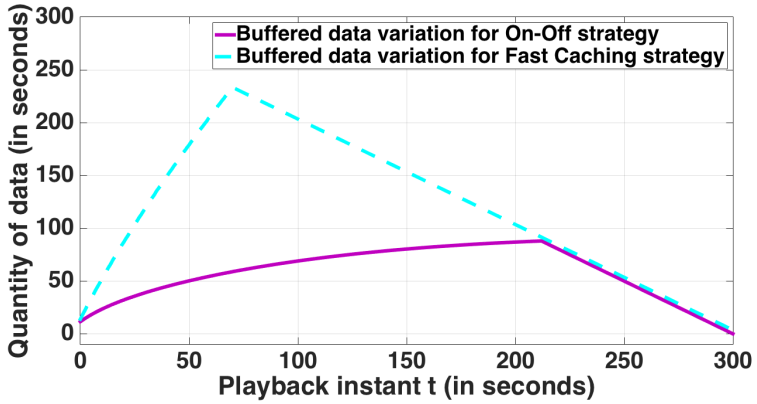


Fig. 4.3: downloaded and buffered data as a funtion of the advancement of playback time for ON-OFF and Fast Caching streaming strategies($q_a = 5s$, $Q_{on} = 40\ Mbites$, $\beta = \frac{1}{50}s^{-1}$, $\frac{1}{30}s^{-1}$ and 18 users)

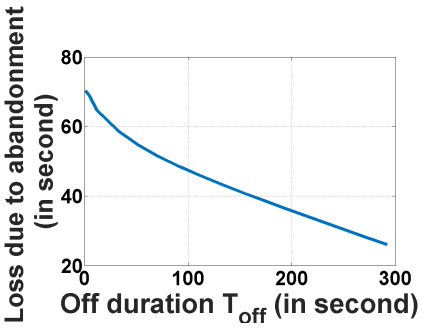


Fig. 4.4: Loss due to the abandonment as a function of OFF duration($Q_{on} = 50Mbits$)

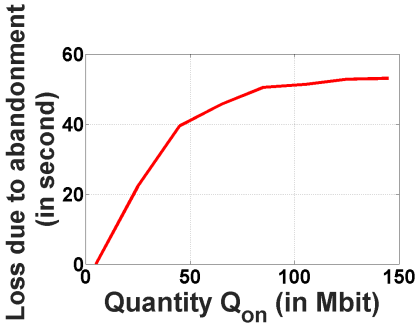


Fig. 4.5: Loss due to the abandonment as a function of quantity Q_{on} ($\beta = \frac{1}{50}s^{-1}$)

Figure 4.2 shows the downloaded and buffered data for both streaming strategies, ON-OFF and Fast Caching, as a function of the advancement of the playback.

First, we remark that, as expected, the Fast Caching streaming strategy quickly downloads and buffer streaming traffic relative to the ON-OFF strategy. Also in our case, there is no starvation for the tagged user during playback which means that the base station serves well the tagged user. For the rest of users, we cannot decide if they are well served by the base station as it depends on their $Bitrates$. What is certain is that any user who has a $Bitrate$ lower or equal to the $Bitrate$ of the tagged user will also experience no starvation. Also, if the tagged user decides to stop the video after, for example, 90 seconds and leaves the network, with fast caching strategy, there will be 125 seconds of unread data in the buffer while with the On-Off strategy, it remains just 75 seconds. It is clear that the ON-OFF strategy is better in that case since it has a gain of 50 seconds compared by the fast caching. This is the key reason behind that work. Specially, We can say that this tagged user represents all users that have a $Bitrate$ lower or equal than the $Bitrate$ of the tagged user in terms of starvation performance. For other users who have a higher $Bitrate$ than that of the tagged user, we can ensure better performance quality using the following strategy (that we have not implemented): offer more throughput (send more data) by depriving users that are in excess in terms of data. Figure 4.3 shows the same plots as before but with a greater load on the base station by taking an inter-arrival time of $30s$ ($\lambda = \frac{1}{30}$). As expected, the tagged user buffers less data and it is because there is more demand for access to the base station and hence less throughput than in the prior case for the tagged user.

Figure 4.4 and 4.5 show, respectively, the variation of $Loss_{abn}$ defined in Equation 4.2 as a function of OFF duration and quantity Q_{on} respectively for ON-OFF streaming strategy for a cell capacity of 6 Mbps, video bitrate of 480 kbps, mean video duration of 5 minutes, a start-up threshold of 10 seconds, an inter arrival time of 35 seconds, and a maximum number of 18 simultaneously streaming users. As expected, $Loss_{abn}$ decreases with respect to OFF duration and increases with respect to the quantity Q_{on}. In the next session, we present a case where all customers can abandon before proceeding to the optimization of QoE indicators: starvation probability and loss due to abandonment.

4.3.4 Case where every user can abandon the video playback

In this section, we introduce the abandon phenomenon for all users different from the tagged user and see the impact on the tagged user performance. We introduce the abandonment phenomenon in our Markovian model as follows. The transitions ν_i defined at equation 4.1 become:

$$\nu_i = \frac{i.C}{(i+1).Bitrate.\sigma_i.T_{video}} \tag{4.19}$$

σ_i is a coefficient in the interval $[0, 1]$ that specifies that users download only $\sigma_i.T_{video}$ of the video. This coefficient depends on users and therefore on i. σ_i is randomly drawn from the distribution of abandonment shown in Figure **??**.

To simulate the impact of this phenomenon on the tagged user, we calculate $Q_{average,On-Off}(t)$ from 100 randomly picked values of $(\sigma_i)_{0\leq i\leq K-1}$ for different users, take the average over this 100 cases with a confidence interval and compare it with the case of section 4.3.1 without abandonment.

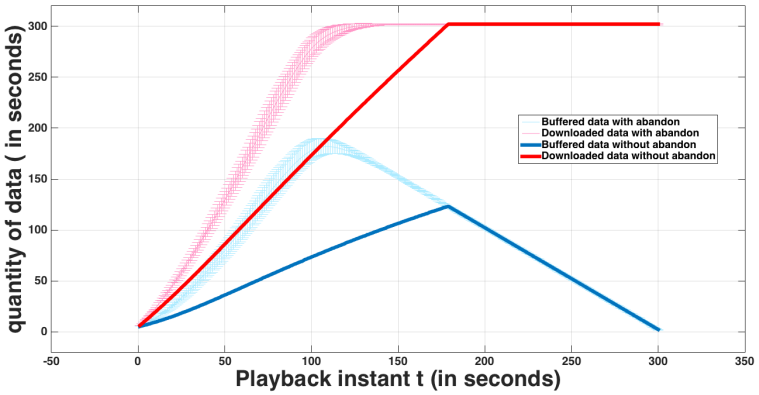


Fig. 4.6: downloaded and buffered data as a funtion of the advancement of playback time for On-Off streaming strategy($q_a = 5s$, $\beta = \frac{1}{50}s^{-1}$ and 18 users)

Figure 4.6 (resp. 4.7) represents and compares the variation of the average downloaded and buffered data as a function of playback time t for On-Off (resp. Fast Caching) strategy taking into account the abandonment phenomenon or not. We note that in the case when users can abandon the playback of the video, the tagged user downloads and buffers rapidly data, for both strategies. The abandonment of other users affects positively the performance of the tagged user: it releases more bandwidth to quickly download the video.

In the remaining of the chapter, we will focus on the case where the other users (different from the tagged user) download the entire video: it constitutes worst case for the tagged user.

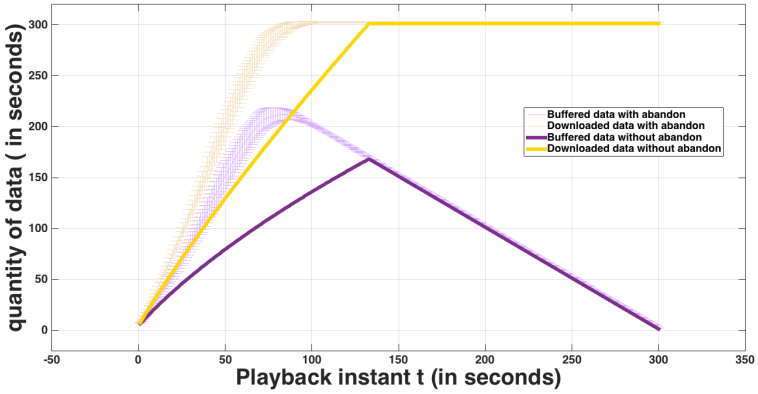


Fig. 4.7: downloaded and buffered data as a funtion of the advancement of playback time for Fast Caching streaming strategy($q_a = 5s$, $\beta = \frac{1}{50}s^{-1}$ and 18 users)

4.4 Optimization: starvation vs wasted bytes

Starvation during playback of a video is a key QoE metric. One would like to ensure downloading enough data in order to avoid video starvations. However, downloading a lot of data leads to a significant loss in terms of energy if the user decides to abandon watching the video. In this section, we explore the optimization of the trade-off between probability of starvation P_{starv} [Bou+16] and loss due to the abandoned $Loss_{abn}$ defined in Section 4.3.

In Chapter 3, we used the same users' dynamic to compute the probability to have a starvation during video playback P_{starv}, expressed as follows:

$$\begin{aligned} P_{starv} &= \mathbb{P}\{T_{video} > q_a\}.\pi.\vee(0,q_a).\Psi(q_a) \\ &= exp(-\theta q_a).\pi.\vee(0,q_a).\Psi(q_a). \end{aligned} \tag{4.20}$$

Such that:

- $\Psi(q_a) = exp(q_a.N).\Psi(0)$ is the starvation probability knowing that the playback starts with q_a seconds of prefetched video content. N is a square matrix depending on network parameters and ON and OFF durations,

- q_a is the prefetching time before starting to read a video,

- $\mathbb{P}\{T_{video} > q_a\}$ is the probability that the video duration is greater than q_a,

- π is the vector of π_i defined at Equation 4.3,

- $\vee(0, q_a)$ is the vector of transition probabilities during the prefetching period before playback start.

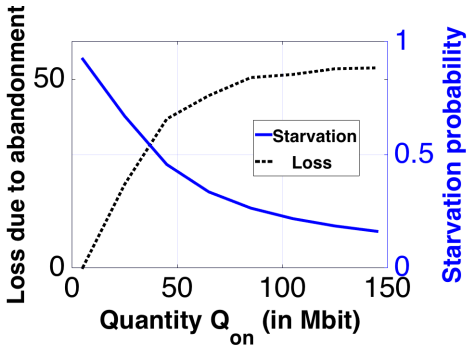


Fig. 4.8: Loss due to the abandonment and starvation probability as a function of quantity Q_{on}($\beta = \frac{1}{50}s^{-1}$)

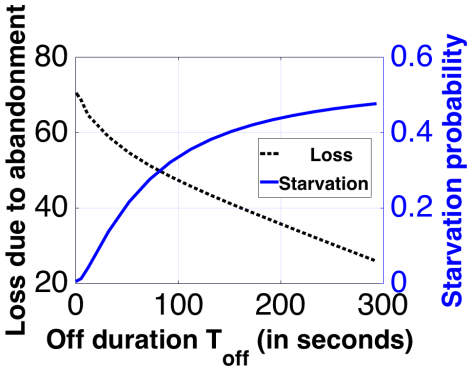


Fig. 4.9: Loss due to the abandonment as a function of OFF duration ($Q_{on} = 50Mbits$)

There are two major requirements or objectives to keep in mind when optimizing QoE of streaming video over wireless networks which are avoiding starvations and minimizing the loss due to abandonment. Unfortunately, these two objectives cannot be explored separately because they are coupled in conflicting ways such that, improvements in one objective lead to deterioration of the other. Figures 4.8 and 4.9 show the variation of both: loss due to abandonment and starvation probability with respect to the quantity Q_{on} and OFF duration respectively for a start-up threshold equal to 10s, an arrival rate of $\frac{1}{35}s^{-1}$, a capacity equal to 6 Mbs, playback speed of 480 Kbs, a video of 5 minutes and a maximal of 18 users. We remark that our two utilities are in direct competition. As a simple example, a high OFF duration reduces the loss due to abandonment $Loss_{abn}$ but increases the probability to have a starvation during playback P_{starv}, and vice versa. These figures confirm what we said: there is a conflict between our two metrics in order to satisfy and maximize both, the probability of starvation and the loss due to abandonment.

Thereafter, we use $Loss_{abn}$ and the probability of starvation P_{starv} as objectives functions.

A key assumption is that the two objectives are not ordered or preferential and then studied without preconceptions. We assume that there is a network optimizer that seeks to optimize the two objectives simultaneously:

$$\underset{T_{off},Q_{on}}{minimize}\ \ g(T_{off},Q_{on}) \tag{4.21}$$

with $g(T_{off},Q_{on} = [P_{starv}(T_{off},Q_{on}), Loss_{abn}(T_{off},Q_{on})]^T$

Equation 4.21 is the minimization of both objectives functions simultaneously which is known as a multi-objective optimization problem (MOOP). For more informations and details concerning the MOOP, please refer to the survey [MAri].

We have now defined our MOOP. The two objectives are: $P_{starv}(T_{off}Q_{on})$ and $Loss_{abn}(T_{off},Q_{on})$. Note that T_{off} varies within the interval $[0,T_{video}]$ and Q_{on} in $[0,T_{video}.Bitrate]$.

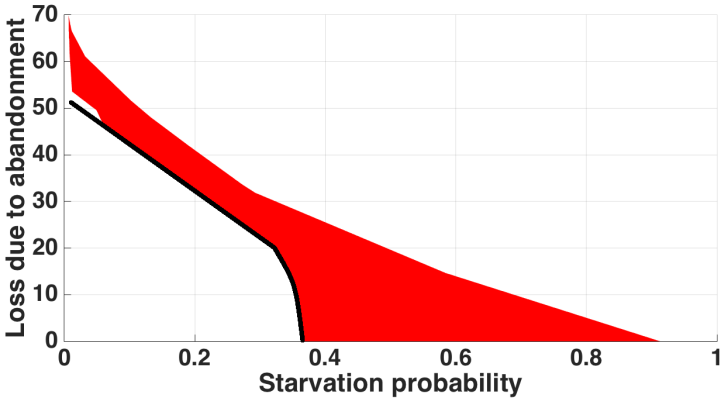


Fig. 4.10: Visualization of the tradeoff between loss due to abandonment and starvation probability($q_a = 10s$, $\lambda = \frac{1}{35}s^{-1}$ and 18 users)

In Figure 4.10, we plot the area defined by:

$$A = \{(P_{starv}(T,Q), Loss_{abn}(T,Q))/T \in [0, T_{video}] \\ and\ Q \in [0, T_{video}.Bitrate]\} \tag{4.22}$$

We are interested by the optimal operating points which belong to the black curve: the Pareto boundary. Each point belonging to Pareto optimal points describes a particular tradeoff between our 2 objectives functions. Consequently, our Pareto boundary describes the set of efficient potential operating points from which we should select the one that is subjectively preferable to us as a user quality optimizer. First, we confirm that our two objectives are conflicting also on the curve of Pareto boundary. Indeed, we can decrease the loss due to abandonment only by making sacrifices in starvation probability. A very important point to note also, is that we can significantly decrease the loss due to abandonment with only minor losses in starvation probability at the right part of Pareto boundary. Instead of operating in a point where the loss is 20 seconds and the starvation probability is around 0.4, there is a better operational point where the loss is zero and the probability is almost constant around 0.4. Finally, we remark that to have low values of starvation probability (≤ 0.2), we should necessarily take a small OFF duration ($T_{off} \leq 30$ seconds) and a relatively large Q_{on} values ($Q_{on} \geq 10Mbit$).

Figure 4.11 shows also the trade-off between our two objectives functions but with more load for the base station with an inter-arrival time of $25s$. We remark that the shape of our area does not change much. The difference from the previous case is that, for example, there is no possible T_{off} and Q_{on} to have a starvation probability lower than 0.2 even if we use a small OFF duration: it is due to the load of the base station.

The optimal values of our problem depend on the criteria of users and the weight of each objective functions: there is no global optimum. Each optimal Pareto operating point represents a user or operator preference. Authors, in [Bjo+14], propose different methods for linearising the objective functions in order to find the optimum points. To address this issue, we can, for example, specify a

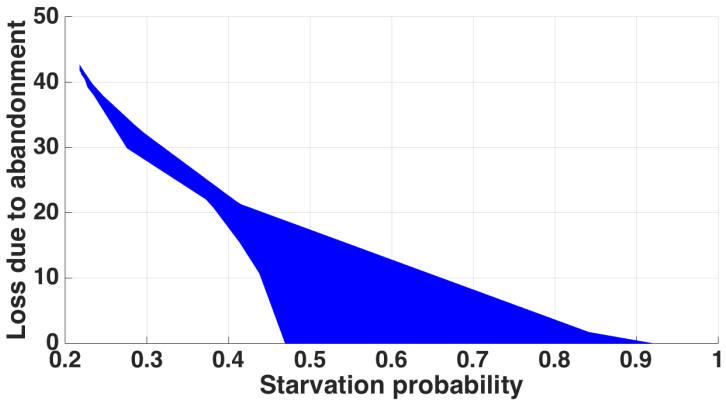


Fig. 4.11: Visualization of the tradeoff between loss due to abandonment and starvation probability($q_a = 10s$, $\lambda = \frac{1}{25}s^{-1}$ and 18 users)

goal function $h : \mathbb{R}^2 \longrightarrow \mathbb{R}$ (weighted sum, geometric mean) that for any optimal Pareto operating point produces a scalar describing how preferable that point is. The goal function provides a certain subjective tradeoff between our two objectives functions and then imposes preferences articulated and decided before any computations take place. We left this approach to future work due to time constraints.

4.5 Conclusion

In this chapter, we tackled the problem of modeling and optimizing the loss due to abandonment and the starvation probability of media streaming service in mobile networks in the case of ON-OFF and Fast Caching downloading strategies. We define and show how to compute the loss due to abandonment based on our Markovian users dynamic model. We also showed that the use of ON-OFF strategy can provide a good performance and anticipate abandon by minimizing the loss criteria. Finally, we explored the trade-off between the loss due to abandonment and probability of starvation, and show how the trade-off of these two QoE indicators behaves while using ON-OFF strategy, with respect to ON and OFF phases. In the next chapter, we will relax our Markovian model and study the performance of a dynamic model controlling the management of data sent to each user, in order to satisfy a group of clients, while taking into account user behavior, video starvations, and network resources. In fact, we adopt a linear model and look for satisfying a group of client by optimizing QoE indicator of streaming service.

5 Enforcing QoE at the base station

> *If I were to awaken after having slept for a thousand years, my first question would be: Has the Riemann hypothesis been proven?.*
>
> — **David Hilbert**
> (Mathematician..)

The increasing video streaming traffic and dense deployment of base stations are adding more and more pressure to telecommunication operators in order to serve their end users efficiently. In the last years, operators and content providers have made the clients QoE imperative. Serving well a client during a streaming session means to ensure no video freeze during a video session and without a terminal energy drain. This energy drain problem comes mainly from the data downloaded and not read by the end users: user's abandonment (see Section 4.1). This phenomenon is not only consuming device energy, but also expands core network operations and energy drains in base stations. Consequently, research efforts are focusing on finding the best streaming strategy to route streaming data from service providers to end users. In the previous chapters, we studied the two most popular streaming strategies which are On-Off, used by Android devices, and Fast Caching, used by iOS devices. We modeled the On-Off strategy and studied its performance by measuring the two QoE indicators: video freeze and loss due to user's abandonment. We opted for a model where user's dynamics follows a birth and death process. In this chapter, the idea is to find an efficient BS rate allocation algorithm in order to improve the QoE of streaming video end users. As an illustration, let's suppose we have two users. The first denoted A has a single time unit of buffered data (data equivalent to a second of video playback) and a second denoted B which has 5 data units at instant t+1. Let's assume that the base station can distribute 4 units of data at instant t. The question is then, what is the share of the units at time t+1 that the station will send to each of the two users A and B? The answer is not simple. In fact, the sharing algorithm can decide to offers user A more units than user B in order to ensure and anticipate a smooth playback. If user A abandons viewing the video at time t+1, it is not a good strategy.

In the streaming strategies currently implemented, abandonment is not taken into account, even if we avoid overfilling the buffer (i.e On-Off streaming strategy). In the best case, a premature abandonment will result in a loss of a few seconds of video content (i.e it can vary from a few seconds to the size of the video), sent to the user but never viewed. Nowadays, we have sufficient information on user abandonment behavior to take into account in order to reduce this loss, thus saving the resources used unsuccessfully.

The goal of this chapter is to determine the quantity of streaming data sent to each streaming demand every step time in order to optimize QoE indicators. In the next section, we explain some key

differences between Chapters 3, 4 and this chapter and also, present some related works to our linear modeling of the QoE.

5.1 QoE for streaming video: from the study of existing models to proposing new strategy

This section is a bridge between what we have studied and analyzed in the previous chapters and what we will do in this chapter. In fact, in Chapter 3 and 4, our goal was to study and evaluate the most popular streaming strategies, which are the On-Off and Fast Caching strategies. We proposed a Markovian model to analyze the trade-off between different QoE indicators for streaming video services. The question is: are these streaming strategies optimal in terms of QoE? Can we find another strategy that will be better in terms of quality of experience?

Starting from this, our idea is to find an algorithm or even a strategy whose performances in QoE are better. To this end and in order to control the sending of all the streaming data to different users, and have more freedom to find a better strategy, we decided to use a model at the packet level (see Section 2.3.1). In this family of model, we assume that packets arrive in the user terminal buffer following a predefined distribution, and the service rate of packets is also assumed to follow a predefined distribution. In the literature, there is a lot of works that align with this modeling technique with different objectives. In [SV13], the authors adopt a packet level modeling to minimize the fraction of data sent from the BS to end users while ensuring a smooth playback and allocating channel resources by predicting the wireless link. They propose a linear algorithm to trade off the allocation of the bandwidth and user buffer size. Their model, for one user, is as follows:

$$\begin{cases} r_1 - z_2 &= V \\ r_t + z_t - z_{t+1} &= V \\ r_T - z_T &= V \end{cases} \tag{5.1}$$

With:

- r_i is the allocated bits per time slot,
- z_i is the bits in the buffer of the user,
- V is the playback speed or rate,
- T is the look-ahead window which is the next time slots where the average channel gain is predictable by the video user,
- t is the time slot (i.e equal to 2,3,...T-1).

In addition the prediction of wireless channel was expressed as follows:

$$r_t = w_t.T_d.B.log_2(1 + SINR) \tag{5.2}$$

With T_d is the duration of slot t, w_t the number of sub-channels affected at slot t, B is the bandwidth and the SINR is the signal to interference ratio.

Using those linear Equations (Eq 5.1) and the channel prediction (Eq. 5.2), the authors minimize the quantity of data sent to users in the few next slots. The problem with this approach is that, apart from the fact that it is based on an average prediction of the channel, it is a short-term technique. In fact, at the macro level, it may turn out that there is a strategy that could allow very good results independently of the prediction of the wireless channel. Finally, this approach will surely be expensive in terms of energy and price, since it should necessarily calculate and predict the channels for all the users.

In [Xu+12], the authors assume that packets sent from the BS to users follow a Poisson process with intensity λ, and the packet service rate follow an exponential distribution with parameter γ. Thus, the buffer is an M/M/1/K with K the buffer size. Based on this model, they compute a probability to have a starvation during playback using The Ballot theorem [**wikibal**]. The expression of this starvation probability P_s is the following:

$$P_s = \sum_{k=x_1}^{N-1} \frac{x_1}{2k - x_1} \binom{2k - x_1}{k - x_1} p^{k-x_1}(1-p)^k \tag{5.3}$$

Knowing that:

- x_1 is packets accumulated during the start-up delay,
- N is the file size,
- p is the probability to receive a new packet, expressed as follows: $p=\frac{\lambda}{\lambda+\gamma}$.

The authors use after this Formula 5.3 to compute the distribution of starvations and analyze this distribution regarding to the start-up and rebuffering delay (i.e delay to download x_1 packets after a starvation happens).

In this work, our goal is to model the exchange between a BS and users at packet level, in order to find a strategy that optimizes the QoE indicators, namely starvation events and loss due to abandonment (see Section4.3). In the next section, we introduce our linear model and present the problem to optimize and our way to find, analyze or even propose a better streaming technique.

5.2 Linear modeling of the QoE

In this section, we present our linear model of the QoE. We place ourselves in a favorable framework in which all the video contents are hosted by a server near to the base station. We assume that there is a smart mechanism at the base station side that has access to all the user useful informations (i.e buffered data during a session, abandonment distribution, playback speed...) It is up to this mechanism to carry out the optimization of the quality from this information and by means of the predictions on the behavior of the users. The problem is therefore to determine and optimize the quantity of data to be sent to each request or user, and at each time step. In the next Section, we start our modeling by the base station.

5.2.1 Modeling of the base station and main client variables

We consider a cell using admission control. If the network is saturated, any request for additional user access will be denied.

The base station (BS) is characterized by:

- Capacity C in bit per second,
- The maximum number K of clients which can simultaneously be connected to the base station.

We assume that the base station can decide on the different amounts of data to send to users regardless of the radio conditions. A user joins the network, requests a video and leaves the network after downloading the whole video. Note that a streaming client cannot watch more than one video from his terminal device.

Let t be an instant between 0 and T. T is a time horizon in which we carry out our model and the optimization. Note that T is extendable if we seek to perform an optimization over a wider time horizon.

At the users side, the downloaded video packets are stored in the play-back buffer and played at a deterministic rate depending on the video encoding rate. We are interested in the family of variables $z_{i,t}$, defined as follows:

$z_{i,t}$ the amount of data in seconds available in the buffer of user i at time t ($i \in \{1, 2, .., K\}$). Noting that $z_{i,0} = 0$ for all i in $\{1, 2, .., K\}$. We have the following equation:

$$z_{i,t} \geq 0, \ \forall i \in \{1, ..., K\}, \ and \ t \in \{1, ..., T\} \tag{5.4}$$

We also introduce the parameter b_i such that:

Notation	Definition
C	Capacity of the base station in bit per second
K	Maximal number of users which can simultaneously be connected to the base station
t	Simulation instant
$t_0(i)$	Instant where the user i connects to the base station
$t_e(i)$	Instant where the user i leaves the base station
T_i	Video duration of user i
T	A time horizon in which we carry out our model
δt	The time step
t_f	Constant duration in second
q_a	Start-up threshold in seconds (i.e pre-loaded time required to start the video playback)
$r_{i,t}$	The amount of data in bytes to be sent from the BS to the user i at the instant t
$z_{i,t}$	the amount of data in seconds available in the buffer of user i at time t
b_i	Bitrate of user i
λ	users arrival rate
i	User
$p_{i,t}$	Boolean = 1 if user i is connected to the base station at time t and 0 otherwise
$n_{i,t}$	Boolean = 1 if user i is in prefetching period at time t and 0 otherwise
$a_{i,t}$	Boolean which is equal to 1 if user i reads data at time t and 0 otherwise
$s_{i,t}$	Boolean which is equal to 1 if the user i experiences a video starvation at time t and 0 otherwise

Tab. 5.1: Notations of the problem

b_i is the bitrate of the user i considered constant during all the streaming session. b_i is a positive real number.

The connection between the BS and the clients is ensured by the family of variables $r_{i,t}$ which represent the amounts of data in bytes to be sent from the BS to the user i at the instant t. We have the following equation:

$$r_{i,t} \geq 0, \ \forall i \in \{1, ..., K\}, \ and \ t \in \{1, ..., T\} \tag{5.5}$$

We assume that there is an intelligent module between the BS and the clients, enabling values of the $z_{i,t}$ variables of the users to be traced back to the BS. Next, we will introduce variables that control user status.

5.2.2 variables controlling client status

In this section, we present variables controlling the users state. In fact, the downloaded video data by users are stored in the play-back buffer and played at rate b_i. As we mentioned at Section 3.2.1, the download and playback process define two critical phases in a streaming session: prefetching and playback period.

User's dynamic

Let $t_0(i)$ be the instant when user i connects to the base station and requests a video; and $t_e(i)$ the instant she finishes downloading her entire video.

First, the arrival of the users is modeled by a Poisson process of density λ. Consequently, $t_0(i)$ is a known parameter of the problem.

On the other hand, we assume that the departure of customers is totally determined by the size of the video. Let T_i be the duration of the video in seconds, requested by user i. User i leaves the base station after downloading the quantity T_i.

In order to control the end of the download of the whole video, we introduce a variable $v_{i,t}$ defined by:

$$v_{i,t+1} = v_{i,t} + \frac{r_{i,t+1}}{b_i},\ \forall i \in \{1, ..., K\},\ t\ \geq\ t_0(i) \tag{5.6}$$

$$v_{i,t_0(i)} = \frac{r_{i,t_0(i)}}{b_i},\ \forall i \in \{1, ..., K\} \tag{5.7}$$

$$v_{i,t} = 0,\ \forall i \in \{1, ..., K\},\ for\ t < t_0(i) \tag{5.8}$$

$v_{i,t}$ represents the total amount of seconds of video downloaded from the initial instant, $t_0(i)$, until an instant t.

Remark. Unlike $t_0(i)$, $t_e(i)$ is not a known parameter of the problem. The departure time of a user i, $t_e(i)$ is the time t in which $v_{i,t}$ catches T_i. User i leaves then the system when $v_{i,t}$ exceeds T_i. To ensure that $v_{i,t}$ does not exceed too much T_i, we introduce the following inequality:

$$v_{i,t} \leq T_i + \epsilon,\ \forall i \in \{1, ..., K\},\ t\ \geq\ t_0(i) \tag{5.9}$$

With ϵ is a negligible quantity in seconds compared to $\frac{r_{i,t}}{b_i}$ for all i in , that ensures the downloading of the whole video. Consequently, it is the optimization process (see Section 5.3) that decides on the departure times of the clients $t_e(i)$ for all i in $\{1, ..., K\}$.

In order to control the arrival and departure of users, we introduce the variable $p_{i,t}$ such that:

$p_{i,t}$ = 1 if the user i is connected to the base station at time i and 0 otherwise.

Note that $p_{i,t}$ takes the value 1 between $t_0(i)$ and $t_e(i)$ and the value 0 otherwise.

After downloading the entire video T_i, user i logs out. We then have the following equation:

$$-p_{i,t+1}.T_i \leq v_{i,t} - T_i \leq (1 - p_{i,t+1}).T_i,\ \forall i \in \{1, ..., K\},\ \forall t\ \geq\ t_0(i) \tag{5.10}$$

T_i intervenes in equation 5.10 just to have boundaries of the expression $v_{i,t} - T_i$. When $v_{i,t} - T_i$ is negative, user i continues the download of the video and decides 1 for $p_{i,t+1}$. Once $v_{i,t} - T_i$ becomes positive, the user disconnects just at the next instant and $p_{i,t+1}$ takes the value 0. Once disconnected, the variable $p_{i,t}$ remains at 0 for the remaining time to reach T. The case where the same user reconnects may well be considered as a new video request.

Once connected, user i is in one of the following states:

- Prefetching state: before starting reading and after each video starvation, user i starts with a prefetching period in which he accumulates a quantity q_a seconds in the buffer. We let then $n_{i,t}$ be a boolean which is equal to 1 if the user i is in the prefetching period at time t and 0 otherwise,

- Playback state: after accumulating q_a seconds in the buffer, the user starts the video playback while continuing to download the video. Let $a_{i,t}$ be a Boolean equal to 1 if the user i reads the video at the time t and 0 otherwise,

- Starvation state: a state where the video freezes. We let $s_{i,t}$ be a boolean which is equal to 1 if the user i experiences a famine at time t and 0 otherwise.

In order to impose that a user, when connected, is either reading, prefetching or experiencing a video starvation, we introduce the following equation:

$$a_{i,t} + n_{i,t} + s_{i,t} = p_{i,t},\ \ \forall i \in \{1, ..., K\},\ \forall t \in \{1, ..., T\} \tag{5.11}$$

Note that the equation above takes action for all i $\in \{1, ..., K\}$ and t $\in \{1, ..., T\}$.

Prefetching period

Once $p_{i,t} = 1$, user i begins with a prefetching period.

To trigger this period of prefetching at the initial instant when a clients connects to the network, we impose the following equations:

$$n_{i,t_0(i)} = 1,\ p_{i,t_0(i)} = 1,\ a_{i,t_0(i)} = 0,\ s_{i,t_0(i)} = 0,\ z_{i,t_0(i)} = 0, \forall i \in \{1, ..., K\}. \tag{5.12}$$

Note that when a user logs in and requests a video, she starts directly by the prefetching state.

Later in this section, we will define the remainder of the equations linking these two variables $n_{i,t}$ and $p_{i,t}$ as well as the filling of the buffers during this prefetching period.

Let q_a be the start-up threshold. Once q_a is buffered, user i ends this prefetching period and starts video playback. The variable $n_{i,t}$ must then reset to 0. We have so the following inequality:

$$-(1-n_{i,t}).T_i - n_{i,t+1}.T_i \leq \frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - q_a \leq (1-n_{i,t+1}).T_i + (1-n_{i,t}).T_i,\ \forall i \in \{1, ..., K\},\ \forall t \geq t_0(i) \tag{5.13}$$

Knowing that the $r_{i,t}$ are data sent from BS to user i at time t, $z_{i,t}$ are user buffer values with $z_{i,t_0(i)}$ = 0 for all i, T_i and b_i are constants (i.e the size of the video and the bitrate of user i, respectively). Note that T_i intervenes in this equation just to have boundaries of the expression $\frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - q_a$. In fact, the inequality 5.13 only becomes active when user i is in prefetching phase at the instant t (i.e $n_{i,t} = 1$ and so $(1 - n_{i,t}).T_i = 0$. If user i was in prefetching phase at t, the above inequality decides if user i continues in prefetching at instant $t + 1$ or not. When $\frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - q_a$ is negative, user i continues in prefetching and $n_{i,t+1}$ is equal to 1. Once $\frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - q_a$ becomes positive, the prefetching ends at the next instant and therefore the $n_{i,t+1}$ takes the value 0. We remind our reader that, in this case, $n_{i,t} = 1$ and $a_{i,t} = s_{i,t} = 0$.

The start-up threshold should also verifies the following equation:

$$q_a \geq \delta t \tag{5.14}$$

With δt is the data processing and decision time step. The interpretation of this inequality 5.14 is that the start-up threshold should at least be greater than δt to satisfy the minimum buffered data.

Next, we will present equations of data playback and video starvation

Playback and starvation

After the prefetching phase, playback starts. Equation 5.13 triggers $n_{i,t+1}$ to 0 end ensures the end of the prefetching period. Using Equation 5.11, user i could be either in starvation or reading the video. To fix that, we introduce the following equation:

$$s_{i,t} \leq n_{i,t+1} \leq 1 - a_{i,t},\ \forall i \in \{1, ..., K\},\ \forall t \geq t_0(i) \tag{5.15}$$

The interpretation of this equation 5.15 is the following: when the prefectching phase ends for user i at time t, $n_{i,t+1}$ is equals to 0 and the video playback is forced (i.e $a_{i,t}$=1 and $s_{i,t}$=0). In addition, once a video starvation has occurred at time t (i.e $s_{i,t}$=1), user restarts the prefetching at t+1 (i.e $n_{i,t+1}$ to take the value 1).

Next, we will define the equations and the constraints of the evolution of the buffers of the users and the shares of the resources of the BS .

5.2.3 Filling and updating clients' buffer

In this section, we will define the equations controlling the update as well as the constraints of filling of the users' buffer. Indeed, the BS must try to ensure a smooth playback for all users, while avoiding sending a lot of data to each user (i.e avoid data loss if a user abandonment occurs).

Buffers update

Let δt be the data processing and decision time step. The update of the buffers is done according to the following equation:

$$\frac{z_{i,t+1}}{b_i} = \frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - a_{i,t}.\delta t,\ \forall i \in \{1, ..., K\},\ \forall t \geq t_0(i). \tag{5.16}$$

Filling buffers

User buffers are filled according to the decision of the BS. The goal is to optimize the amount sent to each user to ensure smooth playback of the video. It is then necessary, at least, that the amount of

data sent by the base station to user i allows to survive to the starvation phenomenon during δt when user i is not in prefetching phase. We have the following inequality:

$$-n_{i,t+1}.Ti-(1-a_{i,t+1}).T_i \leq \frac{z_{i,t+1}}{b_i}+\frac{r_{i,t+1}}{b_i}-\delta t \leq a_{i,t+1}.T_i+n_{i,t+1}.Ti,\ \forall i \in \{1,...,K\},\ \forall t \geq t_0(i). \tag{5.17}$$

First, the terms of the inequality 5.17 are all expressed at time $t+1$ and not t: it is due to the domain of validity which is $t \geq t_0(i)$. Second, the term $n_{i,t+1}.Ti$ appears in the inequality to ensure that user i is not in prefetching phase (i.e if $n_{i,t+1} = 1$, the inequality 5.17 is not active. Consequently, if user i is not in prefetching (i.e $n_{i,t+1} = 0$), she is either reading the video or in starvation according to equation 5.11 (i.e $a_{i,t+1} = 0$ or $s_{i,t+1} = 0$). However, when $\frac{z_{i,t+1}}{b_i} + \frac{r_{i,t+1}}{b_i} - \delta t$ is negative, user i cannot be in playback because $a_{i,t+1}$ takes the value 0 (i.e $a_{i,t+1} = 0$). Consequently, user i is in starvation according to Equation 5.11 and $a_{i,t+1} = 1$. On the other hand, if $\frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - \delta t$ becomes positive, the playback is active and there is no starvation (i.e $a_{i,t+1} = 1$ and i.e $s_{i,t+1} = n_{i,t+1} = 0$).

Newt, we will discuss equations and inequalities related to capacity.

5.2.4 Capacity constraint

The data sent to all connected users must not exceed the capacity C at any time t. Consequently, the main constraint for BS is that the sum of all the data sent to the different clients must not exceed the capacity. We have the following equation:

$$\sum_{i=0}^{K} r_{i,t} \leq C, \forall i \in \{1,...,K\},\ \forall t \in \{1,...,T\}. \tag{5.18}$$

It is obvious that $r_{i,t}$ should be 0 if user i is not connected to the BS (i.e $p_{i,t} = 0$). We have then:

$$r_{i,t} \leq C.p_{i,t}, \forall i \in \{1,...,K\},\ \forall t \in \{1,...,T\}. \tag{5.19}$$

With C the capacity of the BS.

Remark. The capacity C represents the coupling between the user.

5.3 Objective function

The overall objective can be defined according to the need and the objective of the operator. In fact, the objective function may be a function of the different the sub-objectives. One of sub-objectives may be to ensure a smooth playback for all connected users by minimizing the overall number of starvations. It can be expressed as follows $\frac{1}{K.T}\sum_{t=0}^{T}\sum_{i=0}^{K} s_{i,t}$. Another sub-objectives may be to ensure a small duration for the prefetching period and can be directly expressed as a function of

the start-up threshold q_a. In Section 3.5.2, we used the weighted product function by considering the two sub-objectives cited before. Also, forcing the buffer to not exceed a high threshold (i.e to not have too much loss if the user abandons the video playback) may be a good sub-objective too. For the moment, we do not consider the case of user abandonment of video playback. In order to keep the generality of our work, we present the objective function in its general form which is: $f(z_{i,t}, r_{i,t}, n_{i,t}, a_{i,t}, p_{i,t}, s_{i,t}, q_a)$ with $z_{i,t}$, $r_{i,t}$, $n_{i,t}$, $a_{i,t}$, $p_{i,t}$, $s_{i,t}$ and q_a are defined above.

5.3.1 Formulation of the optimization problem

In order to recap all the equations and constraints, we write below the global problem to optimize:

$$\text{Optimize} \quad f(z_{i,t}, r_{i,t}, v_{i,t}, n_{i,t}, a_{i,t}, p_{i,t}, s_{i,t}, q_a)$$

subject to

$$r_{i,t} \leq C.p_{i,t}, \forall i \in \{1, ..., K\}, \ \forall t \in \{1, ..., T\}, \tag{5.20}$$

$$\sum_{i=0}^{K} r_{i,t} \leq C, \forall i \in \{1, ..., K\}, \ \forall t \in \{1, ..., T\}, \tag{5.21}$$

$$a_{i,t} + n_{i,t} + s_{i,t} = p_{i,t}, \ \ \forall i \in \{1, ..., K\}, \ \forall t \in \{1, ..., T\}, \tag{5.22}$$

$$s_{i,t} \leq n_{i,t+1} \leq 1 - a_{i,t}, \ \forall i \in \{1, ..., K\}, \ \forall t \geq t_0(i), \tag{5.23}$$

$$\frac{z_{i,t+1}}{b_i} = \frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - a_{i,t}.\delta t, \ \forall i \in \{1, ..., K\}, \ \forall t \geq t_0(i), \tag{5.24}$$

$$-n_{i,t+1}.Ti - (1 - a_{i,t+1}).T_i \leq \frac{z_{i,t+1}}{b_i} + \frac{r_{i,t+1}}{b_i} - \delta t \leq a_{i,t+1}.T_i + n_{i,t+1}.Ti, \ \forall i \in \{1, ..., K\}, \ \forall t \geq t_0(i), \tag{5.25}$$

$$-(1 - n_{i,t}).T_i - n_{i,t+1}.T_i \leq \frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} - q_a \leq (1 - n_{i,t+1}).T_i + (1 - n_{i,t}).T_i, \ \forall i \in \{1, ..., K\}, \ \forall t \geq t_0(i), \tag{5.26}$$

$$v_{i,t+1} = v_{i,t} + \frac{r_{i,t+1}}{b_i}, \ \forall i \in \{1, ..., K\}, \ t \ \geq \ t_0(i), \tag{5.27}$$

$$v_{i,t_0(i)} = \frac{r_{i,t_0(i)}}{b_i}, \ \forall i \in \{1, ..., K\}, \tag{5.28}$$

$$v_{i,t} = 0, \ \forall i \in \{1, ..., K\}, \ for \ t < t_0(i), \tag{5.29}$$

$$v_{i,t} \leq T_i + \epsilon, \ \forall i \in \{1, ..., K\}, \ t \ \geq \ t_0(i), \tag{5.30}$$

$$-p_{i,t+1}.T_i \leq v_{i,t} - T_i \leq (1 - p_{i,t+1}).T_i, \ \forall i \in \{1, ..., K\}, \ \forall t \ \geq \ t_0(i), \tag{5.31}$$

$$q_a \geq \delta t, \tag{5.32}$$

$$n_{i,t_0(i)} = 1, \ p_{i,t_0(i)} = 1, \ a_{i,t_0(i)} = 0, \ s_{i,t_0(i)} = 0, \ z_{i,t_0(i)} = 0, \forall i \in \{1, ..., K\}, \tag{5.33}$$

$$z_{i,0} = 0, \ \forall i \in \{1, ..., K\}, \tag{5.34}$$

$$n_{i,t}, a_{i,t}, p_{i,t}, s_{i,t} \in \{0,1\},$$

$$r_{i,t}, z_{i,t}, v_{i,t}, t_e(i), q_a \in \mathbb{R}+.$$

To solve this optimization problem, we used GUSEK under GLPK [1]. However, the complexity of our model (i.e algorithm) is high. In fact, the number of variables, $Cardinal_{variable}$, is:

$$\begin{aligned} Cardinal_{variable} \simeq & \, card\{r_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{z_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{v_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{n_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{a_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{p_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{s_{i,t}/\ i \in \{1,..,K\}, t \in \{1,..,T\}\} \\ & + card\{q_a\} \\ Cardinal_{variable} \simeq & \, 7.K.T + 1 \end{aligned} \tag{5.35}$$

And the number of constraints, computed in the same way as in Eq 5.35 is:

$$Cardinal_{constraint} \simeq 17.K.T + 1 \tag{5.36}$$

Consequently, the complexity and the computation time explode (i.e once T takes values greater than 10 with 3 users in our simulations). We note that q_a may also be a vector depending on i (i.e $q_a(i)$) which will also increase the number of variables and constraints. In addition, we assumed that there is no difference between different users who are connected at a given time t, which adds more decision complexity in the combinatorial algorithm that we have implemented.

In the next section, we simulate the results of our model while a base station manages 1 or and two users respectively.

5.4 Simulations

In this section, we will show the results of our model by analyzing the behavior of the base station (i.e data end to users $r_{i,t}$), the buffered and accumulated data in the user's buffer.

For the first simulation, we place ourselves under the same hypothesis as in the section 5.2. We suppose that there is a base station that manages one user. So, the number of users, K, is equal to 1. The time horizon, T, is set at 150 seconds. The initial time of connection is $t_0(i) = 2$, the capacity is set to 400 kbps, the bitrate at 480 kbps, the start-up threshold at 5s and the video duration at 60s. For the objective function f, we put the start-up delay[2] as a function to minimize which is $f = \sum_{t=1}^{T} n_{1,t}$. In fact, f takes large values means that the user spends more time in prefectching period and therefore a greater start-up delay.

[1] The GLPK (GNU Linear Programming Kit) package is intended for solving linear programming models, mixed integer programming (MIP), and other related problems

[2] the duration between the time that a user initiates a session and the time that the playback start, in seconds

Notation	Value
K	1
T	150 seconds
$t_0(i)$	2
C	400 Kbits
b_i	480 Kbits
T_i	60 seconds

Tab. 5.2: The parameter values of the first simulation.

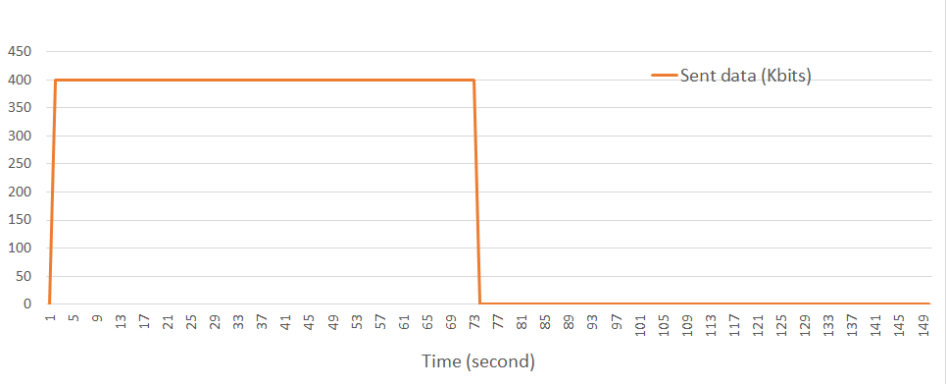


Fig. 5.1: Sent data $r_{1,t}$ in Kbits as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2)

Figures 5.1, 5.2 and 5.3 show, respectively, the sent data r_1, t, the buffered data z_1, t and the accumulated data v_1, t of our optimization algorithm.

We remark that our optimization model has decided that the base station assigns the capacity all the time to the user. This result is logical because our objective is to minimize the start-up delay. To do so, the maximum data must be sent to the user so that she can finish her first prefetching period as soon as possible, and thereafter continue to send the maximum data to avoid a starvation and thus go back to prefetching again. We note that in this example, the user experiences one starvation during all the connection time as the capacity of the channel is smaller than the bit rate of the video.

In another example that we tested for one user and in which we put $C = 10Kbps$, $T = 10s$, $t_s = 5$, $t_o(i) = 2$, $b_i = 480kbps$, $T_i = 60s$ and for the same objective function previously defined, the optimization model decides to not send data to user because even if the base station sends all the capacity, the user will not be able to finish his prefetching period before T. To force the base station to send data, we can modify the objective function by adding $\sum_{t=1}^{T} r_{1,t}$ as objective with the start-up delay.

In the second simulation, we suppose that the base station manages two users. So, the number of users, K, is equal to 2. The time horizon, T, is set at 75 seconds. The starting time of the connections is $t_0(1) = 2,\ t_0(2) = 12$, the capacity is set to 600 Kbps, the bitrate at 480 Kbps for the two users, the start-up threshold at 5s and the video duration at 60s and 30s for user 1 and user 2 respectively. The table below summarizes the values of the parameters of the simulation.

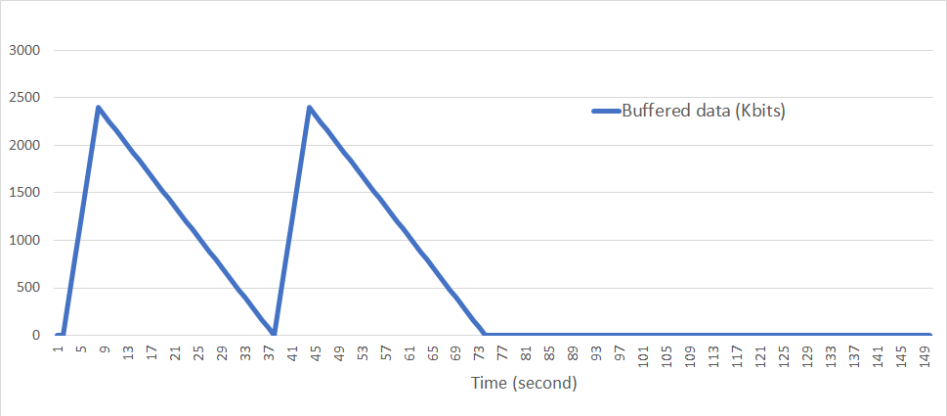


Fig. 5.2: Buffered data $z_{1,t}$ in Kbits as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2)

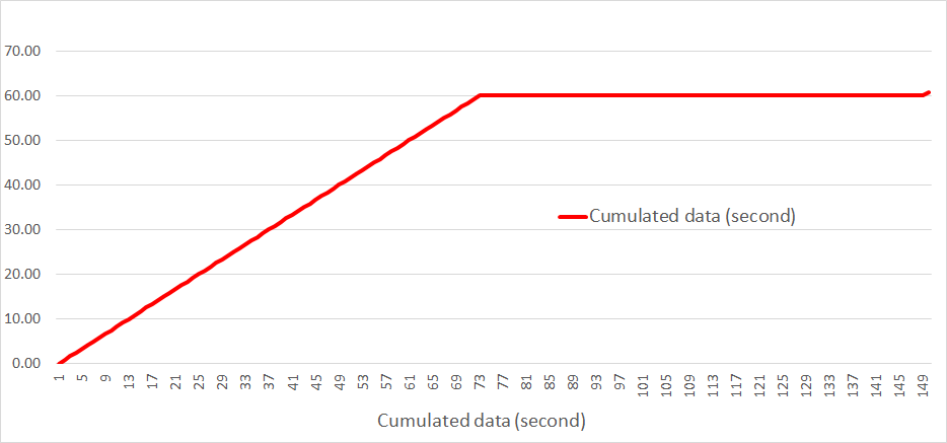


Fig. 5.3: Accumulated data $v_{1,t}$ in seconds as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2)

For the objective function f, we put the start-up delay, the number of starvations and the sent data in the objective function to minimize as follows:

$$f = \sum_{i=1}^{K}\sum_{t=1}^{T} n_i, t + \sum_{i=1}^{K}\sum_{t=1}^{T} s_i, t - \sum_{i=1}^{K}\sum_{t=1}^{T} r_i, t$$

(5.37)

In fact, we would like to minimize the start-up delay and the number of starvation while forcing the base station to send all the possible data. The idea is to see how our optimization algorithm will behave while managing two users.

Figures 5.4 and 5.5 show, respectively, the accumulated data $v_{i,t}$, for user 1 and 2 of our optimization algorithm.

Notation	Value
K	2
T	75 seconds
$t_0(1)$	2
$t_0(2)$	12
C	600 Kbits
b_1	480 Kbits
b_2	480 Kbits
T_1	60 seconds
T_2	30 seconds

Tab. 5.3: The parameter values of the second simulation.

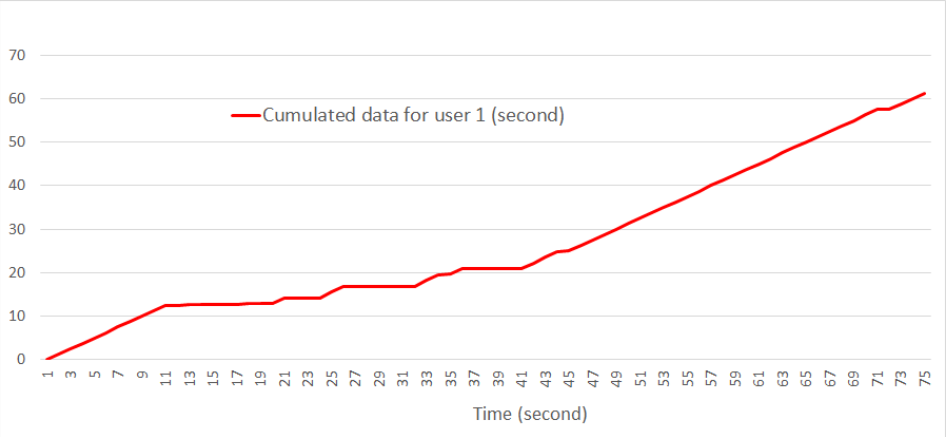


Fig. 5.4: Accumulated data $v_{1,t}$ in seconds as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_1=60s, T_2=30s, t_s=5s, T=150s, K=2, $t_o(1)$=2 and $t_o(12)$=12)

We notice that the sending of the data to user 1 has been disturbed by the connection and video request of user 2 from the instant t = 12.

We note that in this example, user 1 experiences two starvations during all the connection time while user 2 experiences 0 starvation.

We have tried to put more users and a large time horizon in order to simulate behavior that is close to reality. The computing time becomes too large to obtain results in a reasonable time. The next step will be to find a way to minimize the computation time of the algorithm. This could be done in different ways:

- Add valid constraints: in Inequality 5.17, the combinatory (i.e the time to relax this inequality) is high and the inequality still active all time while the algorithm looks for an integer solution of our model. Consequently, it takes time to decide on the reading binary variable $a_{i,t}$ at time t+1. A solution is to add a valid constraint that helps our algorithm to quickly decide on $a_{i,t}$. It can be expressed as follows: $a_{i,t-1} - a_{i,t} \leq s_{i,t},\ \forall i \in \{1, ..., K\},\ \forall t \geq 2$;

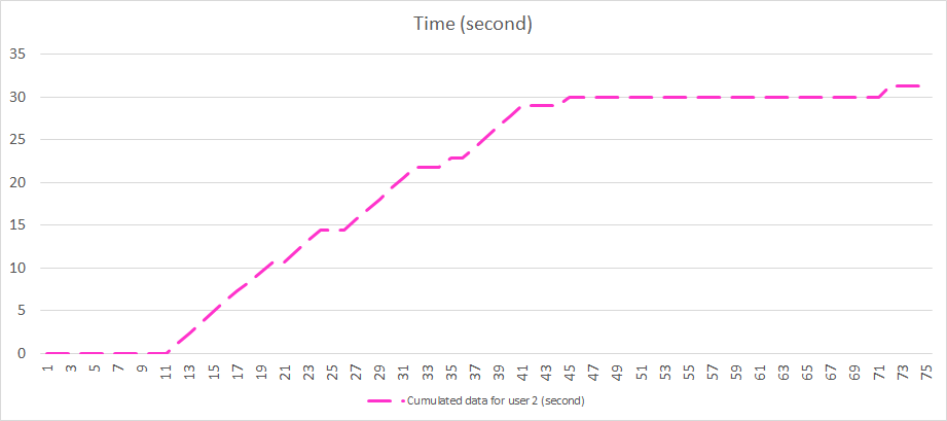


Fig. 5.5: Accumulated data $v_{2,t}$ in seconds as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2)

- Try to solve or even reduce the problem of dynamic coupled over time t and capacity for users(for all variables between t and t+1) or even propose a solution at x percent of the objective to optimize.

Due to time constraints at the time of writing this manuscript, we did not have time to instantiate the above ideas.

Before concluding this chapter, we propose two heuristics that could be compared against our optimization model (once we will have overcome our complexity issues, at least for a small set of users).

In the first scenario (referred to as scenario 1), we try first to satisfy all data requested from users instantaneously and share the rest fairly between user. If the capacity can not satisfy the demand, we exclude the user who has the largest demand in terms of data quantity minus buffered data.

We use the following scenario:

- BS: capacity C, max number of users k,
- users: arrival rate λ , video size T_{vid} between 600 and 1200 s, start-up threshold between 2 and 5s,
- $r_{i,0}$=0,
- User leaves after downloading all the video,
- the scenario is run for 1200 s,
- t=1,2,..,1200.

While time t < 1200 s

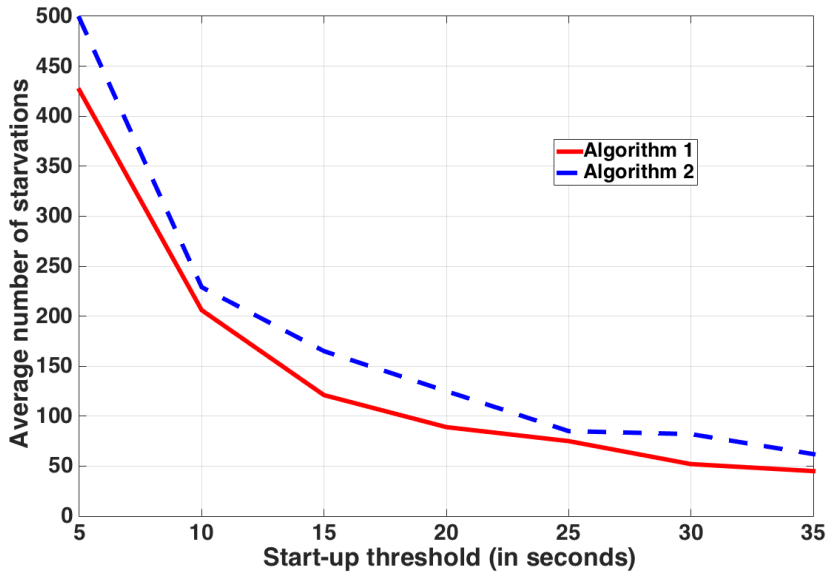


Fig. 5.6: Average number of starvations as a function of start-up threshold

- Users uploads the amount of buffered data and Bitrate ($z_{i,t}$, $b_{i,t}$) to base station at the second t.

- BS sends data to user ($r_{i,t}$) as follows: if sum of ($r_{i,t}$) < C :

  satisfy all the requested demand $b_{i,t} - z_{i,t}$ and share the rest fairly between all users;

  else While sum $r_{i,t}$ > C: exclude the user who has the largest $b_{i,t} - z_{i,t}$ compared to the average.

This first strategy even if it may be not optimal, is surely a good strategy while trying to reduce video starvations. For the second scenario, we consider the strategy that shares fairly the resources, regardless of the situation of user's buffer.

So, the second algorithm (refereed to as algorithm 2) corresponds to the case where the capacity is shared evenly between all users.

To compare the two algorithms, we decided to measure the average number of video starvations compared to the start-up threshold (q_a) and inter arrival time of users (λ), under very favorable conditions for video starvation. Figures 5.6 and 5.7 show the average number of starvations as a function of start-up threshold and inter arrival time of users, respectively, for a capacity equal to 6 Mbs, video duration between 10 and 20 minutes, 18 users with a bitrate between 360 Kbs and 480 Kbs and an inter arrival user time of 35 s. We note that Algorithm 1 has better performance in terms of number of starvations (i.e better QoE). We also notice that the number of starvations varies as a negative exponential with the start-up threshold. Algorithm 1 can thus be a good starting point in our goal of finding a strategy that improves video starvations and loss due to user abandonment.

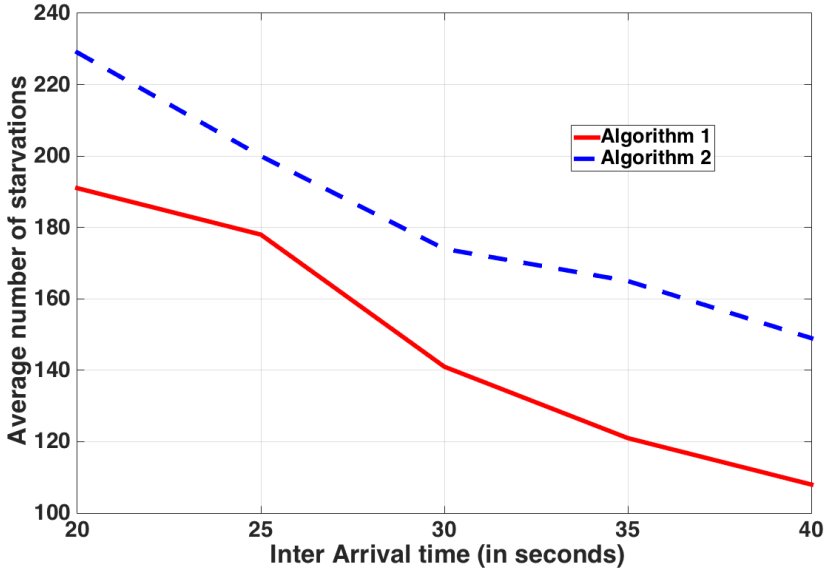


Fig. 5.7: Average number of starvations as a function of inter arrival time (λ)

I would like also to extend the model, especially to add an equation to control wasted data due to user abandonment behavior. The equation can be written as follows:

$$\frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} \leq max(E(i,t), t_f),\ \forall i \in \{1, ..., K\},\ \forall t \in \{1, ..., T\} \tag{5.38}$$

Equation 5.39 ensures that the quantity of data sent from the BS to a user i at time t does not exceed the expected reading time E(i,t)[3] of user i at time t. t_f is a constant time second (i.e can be fixed to 2 or 3 seconds). The t_f serves us in case the user exceeds our expected time of reading and / or the case where she finishes the playback of the whole video.

5.5 Conclusion

In this chapter, we were able to establish a global framework describing the behavior of all actors (BS and streaming users) by a linear program. Our approach was focused on the video buffers of the users and on how to manage these buffers in order to ensure a good performance for all users. We started the chapter by a study of existing works in literature and especially models of streaming video at packet level. We introduced then our linear model with a series of equations reflecting the nature of the exchange between the BS and users. We formulate after our global problem to optimize before presenting our simulations results and proposing two heuristics to be compared against our model. We present some perspectives.

Due to time constraints, we were not able to run the linear program with a large enough number of users and a large time horizon to enable a comparison with heuristics as the ones explored at the end

[3] the estimated time at at t for user i before abandons the video playback

of the chapter. I would like also to extend the model, especially to add an equation to control wasted data due to user abandonment behavior. The equation can be written as follows:

$$\frac{z_{i,t}}{b_i} + \frac{r_{i,t}}{b_i} \leq max(E(i,t), t_f),\ \forall i \in \{1, ..., K\},\ \forall t \in \{1, ..., T\} \tag{5.39}$$

Equation 5.39 ensures that the quantity of data sent from the BS to a user i at time t does not exceed the expected reading time E(i,t)[4] of user i at time t. t_f is a constant time second (i.e can be fixed to 2 or 3 seconds). The t_f serves us in case the user exceeds our expected time of reading and / or the case where she finishes the playback of the whole video.

[4]the estimated time at time t for user i before abandons the video playback

6 Conclusion and perspectives

Our work throughout these three years has consisted of modeling, analyzing and optimizing indicators of QoE for streaming data. Its main originality lies in the proposal for the first time of a model at flow level for the On-Off streaming. We proposed a study and analysis of the QoE indicators according to different parameters, linked to the operator's network or to the demand. In fact, we introduce our thesis subject by explaining the concept of the quality of experience especially for streaming video services. We then analyzed some different QoE indicators and streaming strategies through experimentations.

In Chapter 3, we have developed an exact model and analysis of the starvation probability for On-Off streaming strategy with a Markovian model of users and an exponential duration for ON, OFF and video duration. Our approach is carried out using ordinary differential equations. We found an analytical expression of starvation probability of a video in function of network and mobile station parameters. We finally apply the theoretical results to calibrate the start-up threshold in order to optimize the QoE for streaming service using an On-Off strategy.

In Chapter 4, we tackled the problem of modelling and optimizing the loss due to abandonment of media streaming service in mobile networks in the case of On-Off and Fast Caching downloading strategies. We define and show how to compute the loss due to abandonment based on our Markovian users dynamic model. We also showed that the use of On-Off strategy can provide a good performance and anticipate abandon by minimizing the loss criteria. Finally, we explored the trade-off between the loss due to abandonment and probability of starvation, and show how the trade-off of these two QoE indicators behaves while using On-Off strategy, with respect to ON and OFF phases. The study of the distribution of On-Off streaming strategy starvations and a network in which users can choose to stream with Fast Caching or On-Off strategy may be a good perspective while modeling at flow level. Also, even if the birth and death process is the best simple model found in literature to model the arrival and departure of users, we can imagine a more smooth model which takes into account user's mobility under user's dynamic (i.e user's dynamic refers to the dynamic of users connected to the same BS while the user's mobility refers to the mobility of users between BS).

In Chapter 5 we study the performance of a dynamic model controlling the management of data sent to each user, in order to satisfy a group of clients, while taking into account user behavior, video starvations, and network resources. We propose a dynamic algorithm that minimizes instantaneously the QoE indicators of data sent to users. An interesting perspective would be to integrate into the model the wireless channel prediction and per user video behavior (i.e. YouTube collects statistics about the viewing behavior in for each video clip [You]).

Finally, while most actors working on streaming video service data tackle the issue of studying and optimizing the QoE for streaming services, adaptive bitrate streaming and caching can be a good perspective and the opportunity to expand across multiple approaches. As the client is now exposed to increasingly higher quality of video streaming data, it is incumbent on the industry and especially on R&D departments to pursue every possible option for delivering the highest video quality possible while ensuring good overall QoE.

6.1 Conclusion and perspectives (French)

Notre travail au cours de ces trois années a consisté à modéliser, analyser et optimiser des indicateurs de la qualité d'expérience (QdE) pour les données de streaming. Son originalité principale réside dans la proposition pour la première fois d'un modèle Markovian au niveau flux pour la stratégie du streaming On-Off. Nous avons proposé une étude et une analyse des indicateurs de la QdE selon différents paramètres, liés au réseau de l'opérateur ou à la demande. En fait, nous introduisons notre sujet de thèse en expliquant le concept de la qualité de l'expérience, en particulier pour les services de vidéos en streaming. Nous avons ensuite analysé certains indicateurs de QdE et quelques stratégies de streaming via des expérimentations.

Dans le chapitre 3, nous avons développé un modèle ainsi qu'une analyse de la probabilité de famine pour la stratégie de streaming On-Off avec un modèle markovien d'utilisateurs et une durée exponentielle pour les périodes ON, OFF et la durée de la vidéo. Notre approche se base essentiellement sur des équations différentielles ordinaires. Nous avons trouvé une expression analytique de la probabilité de famine d'une vidéo en fonction des paramètres du réseau et de la station mobile. Nous appliquons finalement les résultats théoriques trouvés pour calibrer le seuil de démarrage d'une vidéo afin d'optimiser la QdE pour le service de streaming adoptant une stratégie On-Off.

Dans le chapitre 4, nous avons abordé le problème de la modélisation et de l'optimisation de la perte due à l'abandon du service de streaming dans les réseaux mobiles dans le cas des stratégies de téléchargement en mode On-Off et Fast Caching. Nous définissons et montrons la méthode pour calculer cet indicateur de la QdE, la perte due à l'abandon, en utilisant notre modèle Markovian de dynamique des utilisateurs. Nous avons également montré que l'utilisation de la stratégie On-Off peut fournir une bonne performance et anticiper l'abandon en minimisant les critères de perte. Enfin, nous avons exploré le compromis entre la perte due à l'abandon et la probabilité de famine, et montrons comment le compromis de ces deux indicateurs de la QdE se comporte lors de l'utilisation de la stratégie On-Off, en fonction de la durée des phases ON et OFF. L'étude de la distribution de la famine pour la stratégie de streaming On-Off et d'un réseau dans lequel les utilisateurs peuvent choisir de diffuser avec la stratégie Fast Caching ou On-Off peut être une bonne perspective pour ce travail. En outre, même si le processus de naissance et de mort est le meilleur modèle simple trouvé dans la littérature pour modéliser l'arrivée et le départ des utilisateurs, nous pouvons imaginer un modèle plus lisse qui prend en compte la mobilité de l'utilisateur sous une dynamique de l'utilisateur (la dynamique de l'utilisateurs réfère à la dynamique des utilisateurs connectés à la même station de base (SB) alors que la mobilité de l'utilisateur fait référence à la mobilité des utilisateurs entre différentes SBs.

Dans le chapitre 5, nous étudions les performances d'un modèle dynamique contrôlant la gestion des données envoyées à chaque utilisateur afin de satisfaire un groupe de clients tout en tenant compte du comportement de l'utilisateur, du phénomène de la famine de la vidéo et des ressources du réseau. Nous proposons un algorithme dynamique qui minimise instantanément les indicateurs de la QdE des données envoyées aux utilisateurs. Une perspective intéressante serait d'intégrer dans le modèle la prédiction du canal sans fil et le comportement des utilisateurs par vidéo par (YouTube recueille des statistiques sur le comportement de visualisation dans chaque clip vidéo [You]).

Enfin, alors que la plupart des acteurs travaillant sur le streaming des données de la vidéo sur les services vidéo abordent la question de l'étude et de l'optimisation de la QdE pour les services de streaming, le streaming adaptative et la mise en cache peuvent constituer de bonnes perspectives et une opportunité d'étendre la travail sur des approches multiples. Dans nos jours, le client est exposé à une qualité croissante des données de streaming de la vidéo, il incombe à l'industrie et surtout aux départements R&D de poursuivre toutes les options et opportunités possibles pour offrir la meilleure qualité vidéo possible tout en assurant une bonne QdE globale.

Bibliography

[AL08] Florence Agboma and Antonio Liotta. „QoE-aware QoS management“. In: *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*. ACM. 2008, pp. 111–116 (cit. on pp. 10, 29, 47).

[Ash+11] Rao Ashwin, Lim Yeon-sup, Barakat Chadi, et al. „Network Characteristics of Video Streaming Traffic“. In: ACM CoNEXT. 2011.

[Az+14] Hatem Abou-zeid, Hossam S Hassanein, and Stefan Valentin. „Energy-efficient adaptive video transmission: Exploiting rate predictions in wireless networks“. In: *IEEE Transactions on Vehicular Technology* 63.5 (2014), pp. 2013–2026 (cit. on p. 30).

[Ber+08] Francisco Bernardo, Nemanja Vucevic, Anna Umbert, and Miguel Lopez-Benitez. „Quality of experience evaluation under QoS-aware mobility mechanisms“. In: (2008), pp. 1–7 (cit. on p. 28).

[Ber+92] Dimitri P Bertsekas, Robert G Gallager, and Pierre Humblet. *Data networks*. Vol. 2. Prentice-Hall International New Jersey, 1992 (cit. on p. 71).

[Bjo+14] Emil Bjornson, Eduard Jorswieck, Mérouane Debbah, Bjorn Ottersten, et al. „Multiobjective Signal Processing Optimization: The way to balance conflicting metrics in 5G systems“. In: *Signal Processing Magazine, IEEE* 31.6 (2014), pp. 14–23 (cit. on p. 80).

[Boo] „NGN integrated subsystem architecture.“ In: *ETSI TS 182 028, v3.5.1.* (Cit. on pp. 3, 4).

[Bou+16] Mohamed Bouzian, Mustapha Bouhtou, Taoufik En-Najjary, Lucile Sassatelli, and Guillaume Urvoy-Keller. „QoE optimization of ON/OFF video streaming strategy in wireless networks“. In: *IEEE Wireless Days (WD) 2016*. 2016, pp. 1–8 (cit. on pp. 70, 78).

[Che+15a] Chang Wen Chen, Periklis Chatzimisios, Tasos Dagiuklas, and Luigi Atzori. *Multimedia Quality of Experience (QoE): Current Status and Future Requirements*. John Wiley & Sons, 2015 (cit. on p. 11).

[Che+15b] Liang Chen, Yipeng Zhou, and Dah Ming Chiu. „Smart streaming for online video services“. In: *Multimedia, IEEE Transactions on* 17.4 (2015), pp. 485–497.

[CIS] CISCO. *Cisco Visual Networking Index: Forecast and Methodology, 2015–2020* (cit. on p. 37).

[CM09] Bogdan Ciubotaru and Gabriel-Miro Muntean. „SASHA—a quality-oriented handover algorithm for multimedia content delivery to mobile users“. In: *IEEE Transactions on Broadcasting* 55.2 (2009), pp. 437–450 (cit. on p. 29).

[ETI12] EU FP7 Project ETICS. *Economics and technologies for inter-carrier services*. 2012-2013 (cit. on p. 3).

[Fie+10] Markus Fiedler, Tobias Hossfeld, and Phuoc Tran-Gia. „A generic quantitative relationship between quality of experience and quality of service“. In: *IEEE Network* 24.2 (2010) (cit. on p. 5).

[Fie+99] Roy Fielding, Jim Gettys, Jeffrey Mogul, et al. *Hypertext transfer protocol–HTTP/1.1*. 1999.

[Gon+09] Yan Gong, Fangchun Yang, Lin Huang, and Sen Su. „Model-based approach to measuring quality of experience“. In: (2009), pp. 29–32 (cit. on p. 26).

[Ham+08] Abdelwahab Hamam, Mohamad Eid, Abdulmotaleb El Saddik, and Nicolas D Georganas. „A quality of experience model for haptic user interfaces“. In: (2008), p. 1 (cit. on p. 27).

[Hoq+] M.A. Hoque, M. Siekkinen, J.K. Nurminen, and M. Aalto. „Dissecting mobile video services: An energy consumption perspective“. In: (), pp. 1–11 (cit. on pp. 19, 38, 43).

[Hoq+12] Mohammad Ashraful Hoque, Matti Siekkinen, Jukka K Nurminen, and Mika Aalto. „Investigating streaming techniques and energy efficiency of mobile video services“. In: *arXiv preprint arXiv:1209.2855* (2012) (cit. on p. 25).

[Hoq+13] Mohammad Ashraful Hoque, Matti Siekkinen, and Jukka K Nurminen. „TCP receive buffer aware wireless multimedia streaming: an energy efficient approach“. In: *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM. 2013, pp. 13–18 (cit. on p. 13).

[Hoq+15] Mohammad Ashraful Hoque, Matti Siekkinen, Jukka K Nurminen, Mika Aalto, and Sasu Tarkoma. „Mobile multimedia streaming techniques: QoE and energy saving perspective“. In: *Pervasive and Mobile Computing* 16 (2015), pp. 96–114.

[Hoß+12] Tobias Hoßfeld, Florian Liers, Raimund Schatz, et al. „Quality of Experience Management for YouTube: Clouds, FoG and the AquareYoum“. In: *PIK-Praxis der Informationsverarbeitung und Kommunikation* 35.3 (2012), p. 133 (cit. on pp. 12, 47).

[Hua+14] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. „A buffer-based approach to rate adaptation: evidence from a large video streaming service“. In: *ACM SIGCOMM 2014 Conference, SIGCOMM'14, Chicago, IL, USA, August 17-22, 2014*. 2014, pp. 187–198.

[Jul+11] Parikshit Juluri, Louis Plissonneau, and Deep Medhi. „Pytomo: A tool for analyzing playback quality of YouTube videos“. In: *23rd International Teletraffic Congress, ITC 2011, San Francisco, CA, USA, September 6-9, 2011*. 2011, pp. 304–305 (cit. on p. 47).

[Ket+10a] Istvan Ketyko, Katrien De Moor, Wout Joseph, Luc Martens, and Lieven De Marez. „Performing QoE-measurements in an actual 3G network“. In: *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*. IEEE. 2010, pp. 1–6 (cit. on p. 14).

[Ket+10b] István Ketykó, Katrien De Moor, Toon De Pessemier, et al. „QoE measurement of mobile YouTube video streaming“. In: *Proceedings of the 3rd workshop on Mobile video delivery*. ACM. 2010, pp. 27–32 (cit. on p. 12).

[Kim+08] Hyun Jong Kim, Dong Hyeon Lee, Jong Min Lee, et al. „The QoE evaluation method through the QoS-QoE correlation model“. In: 2 (2008), pp. 719–725 (cit. on p. 26).

[Liu+] Yao Liu, Fei Li, Lei Guo, Bo Shen, and Songqing Chen. „A comparative study of android and iOS for accessing internet streaming services“. In: *in Passive and Active Measurement (PAM) 2013* (), pp. 104–114 (cit. on pp. 11, 20, 34, 45).

[Liu+13] Yao Liu, Fei Li, Lei Guo, Bo Shen, and Songqing Chen. „A comparative study of Android and IOS for accessing Internet streaming services“. In: Springer. 2013, pp. 104–114 (cit. on p. 48).

[Lua+10] Tom H Luan, Lin X Cai, and Xuemin Shen. „Impact of network dynamics on user's video quality: analytical framework and QoS provision“. In: *Multimedia, IEEE Transactions on* 12.1 (2010), pp. 64–78.

[Mar+07] Ourania Markaki, Dimitris Charilas, and Dimitris Nikitopoulos. „Enhancing quality of experience in next generation networks through network selection mechanisms“. In: *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. IEEE. 2007, pp. 1–5 (cit. on p. 5).

[Mar15] Carla Marshall. *By 2019, 80 % of the World's Internet Traffic Will Be Video*. `http://tubularinsights.com/2019-internet-video-traffic/`. June 2015 (cit. on pp. 37, 38).

[MAri] R Timothy Marler and Jasbir S Arora. „Survey of multi-objective optimization methods for engineering“. In: *Structural and multidisciplinary optimization* 26.6 (Springer, 2004), pp. 369–395 (cit. on pp. 60, 79).

[Mat+15] Siekkinen Matti, Hoque Mohammad Ashraful, and Nurminen Jukka K. „using viewing statistics to control energy and traffic overhead in mobile video streaming“. In: *ACM Transactions on networking* (2015), pp. 1036–6692 (cit. on pp. 34, 36).

[MR14] Sebastien Moller and Alexander Raake. „Quality Of Experience, Advanced Concepts, Applications and Methods“. In: *Library of Congress, Springer Cham Heidelberg New York Dordrecht London* (2014) (cit. on pp. 15–18).

[Nam+13] Hyunwoo Nam, Bong Ho Kim, Doru Calin, and Henning G Schulzrinne. „Mobile Video Is Inefficient: A Traffic Analysis“. In: *Technical report, Department of Computer Science, Columbia University* (2013) (cit. on p. 11).

[OS12] Ozgur Oyman and Sarabjot Singh. „Quality of experience for HTTP adaptive streaming services“. In: *Communications Magazine, IEEE* 50.4 (2012), pp. 20–27 (cit. on p. 11).

[P.112] ITU-R Recommendation P.1202.1. „Parametric non-intrusive bitstream assessment of video media streaming quality–lower resolution application area“. In: *International Telecommunication Union, Geneva* (2012) (cit. on p. 18).

[Par+11] Ali ParandehGheibi, Muriel Médard, Asuman Ozdaglar, and Srinivas Shakkottai. „Avoiding interruptions – a qoe reliability function for streaming media applications“. In: *IEEE Journal on Selected Areas in Communications* 29.5 (2011), pp. 1064–1074 (cit. on p. 47).

[Rao+11] Ashwin Rao, Arnaud Legout, Yeon-sup Lim, et al. „Network characteristics of video streaming traffic“. In: *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*. ACM. 2011, p. 25 (cit. on pp. 11–13).

[REC01] RECOMMENDATION. „ITU-T G.1010 end user multimedia QoS categories“. In: *ITU Telecom* (2001) (cit. on p. 5).

[REC11] RECOMMENDATION. „ITU-T G.1000 qualité de service des communictions: cadre et définitions“. In: *ITU Telecom* (2011) (cit. on p. 5).

[RM+14] Juan J Ramos-Muñoz, Jonathan Prados-Garzon, Pablo Ameigeiras, Jorge Navarro-Ortiz, and Juan M López-Soler. „Characteristics of mobile youtube traffic“. In: *IEEE Wireless Communications* 21.1 (2014), pp. 18–25 (cit. on pp. 19, 44).

[Sie+15] M. Siekkinen, M.A. Hoque, and J.K. Nurminen. „Using Viewing Statistics to Control Energy and Traffic Overhead in Mobile Video Streaming“. In: *IEEE/ACM Transactions on Networking* PP.99 (2015), pp. 1–15 (cit. on p. 44).

[SK13] Lea Skorin-Kapov. „Survey and Challenges of QoE Management Issues in Wireless Networks“. In: *Journal of Computer Networks and Communications* 2013 (2013) (cit. on pp. 10–12, 14, 15).

[Sol+07] David Soldani, Man Li, and Renaud Cuny. *QoS and QoE management in UMTS cellular systems*. John Wiley & Sons, 2007 (cit. on pp. 4, 5).

[Son+12] Wei Song, Dian W Tjondronegoro, and Michael Docherty. „Understanding user experience of mobile video: framework, measurement, and optimization“. In: *Mobile Multimedia: User and Technology Perspectives* (2012), pp. 3–30 (cit. on p. 10).

[Sta+11] Barbara Staehle, Matthias Hirth, Rastin Pries, Florian Wamser, and Dirk Staehle. „Aquarema in action: Improving the YouTube QoE in wireless mesh networks“. In: *Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future*. IEEE. 2011, pp. 33–40 (cit. on p. 14).

[Sto11] Thomas Stockhammer. „Dynamic adaptive streaming over HTTP–: standards and design principles“. In: *Proceedings of the second annual ACM conference on Multimedia systems*. ACM. 2011, pp. 133–144 (cit. on p. 11).

[SV13] Sanam Sadr and Stefan Valentin. „Anticipatory buffer control and resource allocation for wireless video streaming“. In: *arXiv preprint arXiv:1304.3056* (2013) (cit. on p. 84).

[Tak+08] Akira Takahashi, David Hands, and Vincent Barriac. „Standardization activities in the ITU for a QoE assessment of IPTV“. In: *IEEE Communications Magazine* 46.2 (2008) (cit. on p. 30).

[Ti] (Cit. on p. 9).

[Tst14] Tstat. *TCP STatistic and Analysis Tool*. `http://http://tstat.polito.it/index.shtml`. 2014 (cit. on pp. 37, 45).

[Vs] .

[Wik13] Wikipedia. *Osi model*. `http://en.wikpedia.org/wiki/OSImodel`. Apr. 2013 (cit. on p. 9).

[Wik14a] Wikipedia. *Gershgorin circle theorem*. `http://en.wikipedia.org/wiki/Gershgorin_circle_theorem`. June 2014 (cit. on p. 56).

[Wik14b] Wikipedia. *Law of total probability*. `http://en.wikipedia.org/wiki/Law_of_total_probability`. Dec. 2014.

[Wik17] Wikipedia. *Finite difference method*. `https://en.wikipedia.org/wiki/Finite_difference_method`. Feb. 2017 (cit. on p. 61).

[Xu+11] Yuedong Xu, Eitan Altman, Rachid El-Azouzi, et al. „Analysis of Buffer Starvation with Application to Objective QoE Optimization of Streaming Services“. In: (2011) (cit. on p. 13).

[Xu+12] Yuedong Xu, Eitan Altman, Rachid El-Azouzi, et al. „Probabilistic analysis of buffer starvation in Markovian queues“. In: *INFOCOM, 2012 Proceedings IEEE*. IEEE. 2012, pp. 1826–1834 (cit. on pp. 32, 85).

[Xu+13] Yuedong Xu, Salah Eddine Elayoubi, Eitan Altman, and Rachid El-Azouzi. „Impact of flow-level dynamics on QoE of video streaming in wireless networks“. In: *INFOCOM, 2013 Proceedings IEEE*. IEEE. 2013, pp. 2715–2723 (cit. on pp. 13, 33, 39, 47, 48, 50–52, 61, 73).

[Xu+14a] Yuedong Xu, Eitan Altman, Rachid El-Azouzi, et al. „Analysis of Buffer Starvation with Application to Objective QoE Optimization of Streaming Services“. In: *Multimedia, IEEE Transactions on* 16.3 (2014), pp. 813–827.

[Xu+14b] Yuedong Xu, Yipeng Zhou, and Dah-Ming Chiu. „Analytical QoE models for bit-rate switching in dynamic adaptive streaming systems“. In: *Mobile Computing, IEEE Transactions on* 13.12 (2014), pp. 2734–2748.

[Xu+14c] Yuedong Xu, Salah-Eddine Elayoubi, Eitan Altman, Rachid El Azouzi, and Yinghao Yu. „Flow-level QoE of Video Streaming in Wireless Networks“. In: (2014).

[Yam+07] Hideaki Yamada, Norihiro Fukumoto, Manabu Isomura, Satoshi Uemura, and Michiaki Hayashi. „A QoE based service control scheme for RACF in IP-based FMC networks“. In: (2007), pp. 611–618 (cit. on p. 30).

[You] Youtube. *Audience Retention* (cit. on pp. 103, 105).

[ZA11] Jingjing Zhang and Nirwan Ansari. „On assuring end-to-end QoE in next generation networks: Challenges and a possible solution“. In: *Communications Magazine, IEEE* 49.7 (2011), pp. 185–191 (cit. on p. 10).

Webseiten

[Teo11] Luanne Teoh. *85400,000 activations daily*. 2011. URL: `http://www.biztechday.com/85-increase-in-smartphone-users-androidwith-50-market-share-and-4000-activations-daily/`.

List of Figures

1.1 diagram linking QoE and QoS. . . . 6
1.2 General mapping curve between QoS and QoE. . . . 6
1.3 Interplay of TCP and applications via buffers. . . . 9
1.4 Example of possible PCC architecture performing end-to-end QoS/QoE delivery for DASH services [SK13]. . . . 12
1.5 Categorization of video QoE evaluatin algorithms [MR14]. . . . 15
1.6 Impact of packet loss when one slice per frame is used. a Loss occurred in the current frame. b Loss propagated from previous frames [MR14]p.281. . . . 16
1.7 Progressive Download Model. . . . 17
1.8 HTTP Based Adaptative Streaming. . . . 18
1.9 Downloading strategies by the YouTube player for different types of terminals [RM+14]. 19

2.1 Different perspectives of network quality. . . . 26
2.2 Markov chain of user's dynamic. . . . 33
2.3 Downloading strategy for an Android device. . . . 35
2.4 Downloading strategy for an iOS device. . . . 35
2.5 The percentage of each service in the overall volume of traffic for different types of network. . . . 38
2.6 Consumer Internet streaming video services predictions for 2020.[Mar15] . . . 38
2.7 Streaming strategies technical match between service provider and terminal device. . 39
2.8 Fraction of downloaded data while watching only 25 % of the video for On-Off and Fast Cching strategies. . . . 40
2.9 histogram of the number of buffer starvations for ON/OFF strategy. . . . 40
2.10 histogram of the number of buffer starvations for Fast Caching strategy. . . . 40

3.1 Downloading strategies for different types of terminals [RM+14]. . . . 44
3.2 Fraction of abandon during a streaming session. . . . 46
3.3 Fraction of downloaded data while watching only 25 % of the video for ON/OFF and Fast Cching strategies. . . . 46
3.4 Markov chain for user dynamics when the tagged user start playback. . . . 49
3.5 Markov chain before the tagged user joins the network. . . . 50
3.6 Markov chain seen by the tagged user. . . . 51
3.7 Starvation Probability as a function of startup delay . . . 57
3.8 Starvation Probability as a function of maximal number of users . . . 57
3.9 Starvation Probability as a function of startup delay for ON/OFF and Fast Caching streaming strategies . . . 59
3.10 Starvation Probability as a function of maximal number of users . . . 59
3.11 Starvation Probability as a function of inter-aarival timefor ON/OFF and Fast Caching streaming strategies($q_a = 10s$, $\beta = \frac{1}{50}s^{-1}$ and 16 users) . . . 60

3.12 the variation of the probability of non starvation and U_m as a function of start-up threshold($\lambda = \frac{1}{20}$, number of users=18 and startup delay=10s) 60
3.13 U_m as a function of start-up threshold for different values of start-up delay t_s 63
3.14 The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{20}$, number of users=18 and startup delay=10s) 64
3.15 The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{30}$, number of users=18 and startup delay=10s) 64
3.16 The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{20}$, number of users=18 and startup delay=20s) 65
3.17 The variation of the goal function as a function of startup threshold ($\lambda = \frac{1}{30}$, number of users=18 and startup delay=20s) 65

4.1 User's dynamics form the tagged user point of view 70
4.2 downloaded and buffered data as a function of the advancement of playback time for ON-OFF and Fast Caching streaming strategies($q_a = 5s$, $Q_{on} = 40\ Mbites$, $\beta = \frac{1}{50}s^{-1}$, $\frac{1}{35}s^{-1}$ and 18 users) 75
4.3 downloaded and buffered data as a funtion of the advancement of playback time for ON-OFF and Fast Caching streaming strategies($q_a = 5s$, $Q_{on} = 40\ Mbites$, $\beta = \frac{1}{50}s^{-1}$, $\frac{1}{30}s^{-1}$ and 18 users) 75
4.4 Loss due to the abandonment as a function of OFF duration($Q_{on} = 50Mbits$) . . . 76
4.5 Loss due to the abandonment as a function of quantity Q_{on} ($\beta = \frac{1}{50}s^{-1}$) 76
4.6 downloaded and buffered data as a funtion of the advancement of playback time for On-Off streaming strategy($q_a = 5s$, $\beta = \frac{1}{50}s^{-1}$ and 18 users) 77
4.7 downloaded and buffered data as a funtion of the advancement of playback time for Fast Caching streaming strategy($q_a = 5s$, $\beta = \frac{1}{50}s^{-1}$ and 18 users) 78
4.8 Loss due to the abandonment and starvation probability as a function of quantity Q_{on}($\beta = \frac{1}{50}s^{-1}$) 79
4.9 Loss due to the abandonment as a function of OFF duration ($Q_{on} = 50Mbits$) . . 79
4.10 Visualization of the tradeoff between loss due to abandonment and starvation probability($q_a = 10s$, $\lambda = \frac{1}{35}s^{-1}$ and 18 users) 80
4.11 Visualization of the tradeoff between loss due to abandonment and starvation probability($q_a = 10s$, $\lambda = \frac{1}{25}s^{-1}$ and 18 users) 81

5.1 Sent data $r_{1,t}$ in Kbits as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2) 95
5.2 Buffered data $z_{1,t}$ in Kbits as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2) 96
5.3 Accumulated data $v_{1,t}$ in seconds as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2) 96
5.4 Accumulated data $v_{1,t}$ in seconds as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_1=60s, T_2=30s, t_s=5s, T=150s, K=2, $t_o(1)$=2 and $t_o(12)$=12) . . 97
5.5 Accumulated data $v_{2,t}$ in seconds as a function of time in seconds (Capacity=400Kbits, bitrate=480 Kbits, T_i=60s, t_s=5s, T=150s, K=1 and $t_o(1)$=2) 98
5.6 Average number of starvations as a function of start-up threshold 99
5.7 Average number of starvations as a function of inter arrival time (λ) 100

List of Tables

1.1 List of abbreviations 2

5.1 Notations of the problem 87
5.2 The parameter values of the first simulation. 95
5.3 The parameter values of the second simulation. 97