# THÈSE

**pour obtenir le grade de**

## Docteur de l'Université Sorbonne Paris Nord

**Discipline : "Doctorat en Informatique"**

*présentée et soutenue publiquement par*

**Mustafa OTHMAN**

le 22 juin 2021

# Objective video quality metric aware Adaptation mechanisms for video streaming based on DASH

Directeur de thèse : **Prof. Ken CHEN**

Co-directeur de thèse : **Prof. Anissa MOKRAOUI**

## JURY

| | | |
|---|---|---|
| Hossam AFIFI | Professeur, Telecom SudParis | Rapporteur |
| Nadjib AIT SAADI | Professeur, Université de Versailles Saint-Quentin-en-Yvelines | Examinateur |
| François-Xavier COUDOUX | Professeur, Université Polytechnique, Hauts de France | Rapporteur |
| Amine NAIT-ALI | Professeur, Université Paris-Est Créteil | Président |
| Ken CHEN | Professeur, Université Sorbonne Paris Nord | Directeur |
| Anissa MOKRAOUI | Professeur, Université Sorbonne Paris Nord | Co-Directrice |

It is with my deepest gratitude and appreciation that I dedicate this thesis

To my parents;

# Abstract

The DASH (Dynamic Adaptive Streaming over HTTP) standard is widely adopted for video streaming. The Adaptive BitRate (ABR) style adaptation mechanism, which is a key component of DASH, is not standardized, since it must take various elements into account, in particular the context of the communication and the system, but also the quality perceived by the users, to maximize the QoE (Quality of Experience). Many ABR algorithms have been proposed. Few of them attach importance to perceived, and objectively calculated, quality as an adaptation parameter. This thesis proposes a generic framework, called VQBA (Video Quality Metric Based Adaptation algorithm), allowing to integrate an objective metric of the video quality of one's choice as an adaptation parameter. The idea is to maximize the efficient use of the available bandwidth by deciding to switch to a higher speed not only because it is feasible, but also because it provides a significant visual improvement. We carried out numerous tests with video sequences of various kinds and by placing them in real network situations with traces from operational mobile networks. These tests, through three usual video quality metrics, namely SSIM (Structural Similarity Index Measurement), PSNR (Peak Signal to Noise Ratio) and VMAF (Video Multimethod Assessment Fusion), and in comparison with a selection of ABR algorithms, show that the path we explored, that is to say, giving importance to video quality as an adaptation parameter, is an effective path for better QoE.

***Keywords***— *Video Streaming; QoE; ABR; DASH; SSIM; PSNR; VMAF; Mobile Networks.*

# Résumé

La norme DASH (Dynamic Adaptive Streaming over HTTP) est largement adoptée pour la diffusion de vidéo. Le mécanisme d'adaptation du style ABR (Adaptive BitRate), qui est un des composants clé de DASH, n'est pas normalisée, car il doit prendre divers éléments en compte, notamment le contexte de la communication et du système, mais également la qualité perçue par les usagers, pour maximiser la QoE (Quality of Experience). De nombreux algorithmes ABR ont été proposés. Peu d'entre eux accordent une importance à la qualité perçue, et objectivement calculée, comme paramètre d'adaptation. Cette thèse propose un cadre générique, nommé VQBA (Video Quality Metric Based Adaptation algorithm), permettant d'intégrer une métrique objective de la qualité vidéo de son choix comme paramètre d'adaptation. Le principe consiste à maximiser l'utilisation efficace de la bande passante disponible en décidant d'adopter un débit plus élevé non seulement parce qu'il est faisable, mais aussi parce que cela apporte une amélioration visuelle significative. Nous avons mené de nombreux tests avec des séquences vidéo de diverses natures et en les plaçant dans de vraies situations de réseaux avec des traces issues des réseaux mobiles opérationnels. Ces tests, à travers trois métriques usuelles de la qualité vidéo, nommément SSIM (Structural Similarity Index Measurement), PSNR (Peak Signal to Noise Ratio) et VMAF (Video Multimethod Assessment Fusion), et en comparaison avec une sélection d'algorithmes ABR, montrent que la voie que nous avons explorée, c'est-à-dire, accorder une importance à la qualité vidéo comme paramètre d'adaptation, est une voie efficace pour une meilleure QoE.

*Mots-clés Streaming Video; QoE; ABR; DASH; SSIM; PSNR; VMAF; Réseaux mobiles.*

# Contents

# List of Tables

# List of Figures

# Abbreviations

| | |
|---|---|
| ABR | Adaptive BitRate algorithm |
| AVC | Advanced Video Coding |
| BBA | Buffer Based Adaptation |
| C-DMRC | Compressive Distortion Minimizing Rate Control |
| DASH | Dynamic Adaptive Streaming over HTTP |
| dB | decibels |
| DLM | Detail Loss Metric |
| FESTIVE | Fair, Efficient, and Stable adapTIVE algorithm |
| FPS | Frames Per Second |
| FR | Full-Reference |
| HCI | Human-Computer Interaction |
| HDS | HTTP Dynamic Streaming |
| HLS | HTTP Live Streaming |
| HTTP | Hyper Text Transfer Protocol |
| HVS | Human Visual System |
| IQA | Image Quality Assessment |
| ITU | International Telecommunication Union |
| LCV | Laboratory for Computational Vision |
| LIVE | Laboratory for Image and Video Engineering |
| MOS | Mean Opinion Score |
| MOVIE | MOtion-based Video Integrity Evaluation |
| MPD | Media Presentation Description |
| MPEG | Moving Picture Experts Group |
| MSE | Mean Squared Error |
| MSS | Microsoft Smooth Streaming |
| MS-SSIM | MultiScale-Structural SIMilarity index |

| | |
|---|---|
| NR | No-Reference |
| OSMF | Open Source Media Framework |
| PBA | PSNR Based Adaptation |
| PSNR | Peak Signal to Noise Ratio |
| QA | Quality Assessment |
| QoE | Quality-of-Experience |
| 2QAV | QoS/QoE system video control solution |
| QoS | Quality-of-Service |
| RMSE | Root Mean Square Error |
| RR | Reduced-Reference |
| RTSP | Real-Time Streaming Protocol |
| RTP | Real-Time Protocol |
| SBA | SSIM Based Adaptation |
| SSIM | Structural Similarity Index Measurement |
| SVC | Scalable Video Coding |
| SVM | Support Vector Machine |
| TCP | Transmission Control Protocol |
| TI | Temporal Information |
| VBA | VMAF Based Adaptation |
| VCEG | Video Coding Experts Group |
| VIF | Visual Information Fidelity |
| VMAF | Video Multimethod Assessment Fusion |
| VNI | Visual Networking Index |
| VoD | Video on Demand |
| VQA | Video Quality Assessment |
| VQ | Visual Quality |
| VQM | Video Quality Metric |
| VQBA | Video-Quality metrics Based-Adaptation algorithm |

Chapter 1

# Introduction

> “*Great things are done by a series of small things brought together.*”
>
> Vincent van Gogh

## 1.1 Overview

Video communication is taking more and more importance both in our professional activities and our everyday life. This trend is regularly monitored and predicted by major actors, for instance Cisco Systems's 2016's white paper, *Cisco Global Cloud Index: Forecast and Methodology, 2015–2020* (C11-738085-00, 11/16). As a collateral effect, it is being dramatically accelerated by the ongoing global COVID-19 crisis.

The popularity of video streaming, as well as applications using video-streaming, is increasing steadily these last years. This includes the major video-based entertainment activities and social networking, such as those provided by NETFLIX or YouTube. Video streaming takes a prominent place in video communication and occupies a huge and increasing part of Internet traffic. More and more streamed video contents are "consumed" on mobile devices (such as smart phone). Thus, it is not surprising to learn that, according to a recent report of CISCO, the video content now dominates cellular traffic, with a weight of about 60% of all mobile data traffic and and is expected to reach 82% by 2022 [1].

Today's dominant video steaming technology is the Dynamic Adaptive Streaming over HTTP

(DASH) [2,3] system which is an international standard (MPEG-DASH). DASH is an adaptive bitrate streaming technique that enables high-quality streaming of media content over the Internet through conventional HTTP-based web services.

Video content delivered over the networks suffers from different kinds of distortions on their path from the source to the user, such as the fluctuations/limitation of bandwidth, etc. Consequently, the quality of the video eventually received by the user may be not as good as the original one.

This leads to an need of accurate and efficient assessment of perceptual image and video quality at the user side, under different kinds of distortions [4,5]. Actually, the perceptual quality is the single most important criterion for a video service. By nature, the perceptual quality is *subjective*. For obvious operational reasons, this kind of assessment should be *objectively* and *automatic*.

From networking's viewpoint, the multimedia services such as video streaming depends on a set of QoS parameters (e.g., bandwidth, delay, etc.) which have direct impact on the perceptual quality. For instance, packets loss may cause *mosaic* in images. However, QoS is not the sole factor for image quality, which is, it worth to be recalled, basically *subjective*. Different strategies of transmissions and/or playback policies may lead to totally different perceptual quality, even under the same networking environment.

The term *Quality of Experience* (QoE) has been coined to take into account all the parameters to be considered in order to achieve the best possible perceptual quality. Actually, the ultimate criterion for a video streaming service is the *feeling* that users get through their eyes.

The term *Quality of Service* (QoS) is conventionally used to assess transmission quality through network, i.e. quality of the transmission service provide by the underlying network. A satisfactory QoS depends on the availability of resources.

QoE is a combined measure of performance expectations of the end-user; it depends on QoS which ensure the necessary quantitative resources for video transmission, It also take into account parameters which do impact on the perception, such as the quality of the video, but also other parameters, such as the pattern (duration, frequency) of sequence freezing. The latter will be one of our major concerns, under the vocabulary of *rebuffering*.

The correlation between the technology-focused QoS and the user-aware QoE has captured huge interest in both industry and academic community during the last years. Efforts have been done on investigation of reliable and objective (and so computable) metrics for linking together technical system parameters (e.g. delay, jitter, loss rates, connection setup time, and further typical QoS parameters) with the perceptual quality of the user [6]. nowadays, QoE is usually assessed through a set of objective measures, such as mean bitrate, rebuffering frequency/duration, instability frequency, objective video quality measurements, etc.

Adaptive BitRate Streaming was introduced with the main goal of adapting video quality to network bandwidth variations in order to decrease video buffering (also known as stalling/freezing) and maximize the overall video quality. This leafs to the DASH (Dynamic Adaptive Streaming over HTTP) approach, which is the current standardized framework for video streaming.

Hereafter, we give a short history of investigation on video adaptation mechanisms which eventually leads to DASH. Since 2008, several adaptive streaming mechanisms have been proposed, among which we can mention (cf Fig. 1.1): DASH, HTTP Live Streaming (HLS) of Appel, IIS Smooth Streaming (SS) of Microsoft, and HTTP Dynamic Streaming (HDS) of Adobe. All of these mechanism take the Adaptive BitRate (ABR) approach to dynamically adjust the video quality in function of the currently available bandwidth. The common principle of these mechanisms is the following (cf Fig. 1.1):

- Video service provider cut the original video content into segments (referred as *chunk*).
- Each chunk is pre-compressed with several versions, at various coding schema, resolution and/or bitrate levels. Thus, needs for bandwidth can be finely tuned with the granularity of the duration of a chunk. This provides freedom for adaptation mechanism.
- Actual adaptation decision is made by Client: For each chunk, client issues a *fetch* order for a particular version (and so at a certain bit level), according to some adaptation algorithm.

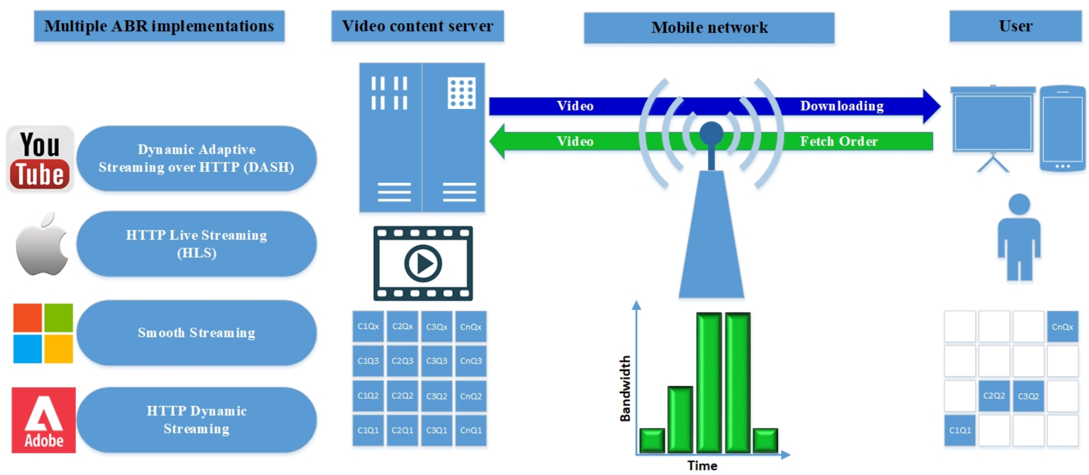



**Figure 1.1:** *Overview of various adaptive bitrate (ABR) streaming proposals*

DASH became a Draft International Standard in January 2011, and an International Standard in November 2011. The standard DASH has been published in April 2012. It has been revised in 2019 as MPEG-DASH ISO/IEC 23009-1:2019.

DASH aims to deliver video with high Quality of Experience [7–9]. The principle of DASH consists in sending video chunks through the HTTP protocol. For DASH, versions of a video chunk are ranged through their bitrate, from best (highest bitrate, best quality) to worst (lowest birate, 1st grade quality). Content description (in particular, information about the media, such as bitrate (qualities), segment lengths, format, URL, etc., is provided by Media Presentation Description (MPD). MPD is encoded in XML format and is made available to end users by DASH server. In this way, the receiver (end-user) can decide which bitrate level is most suited and then fetch it from the DASH server. Figure 1.2 depicts the DASH streaming process.

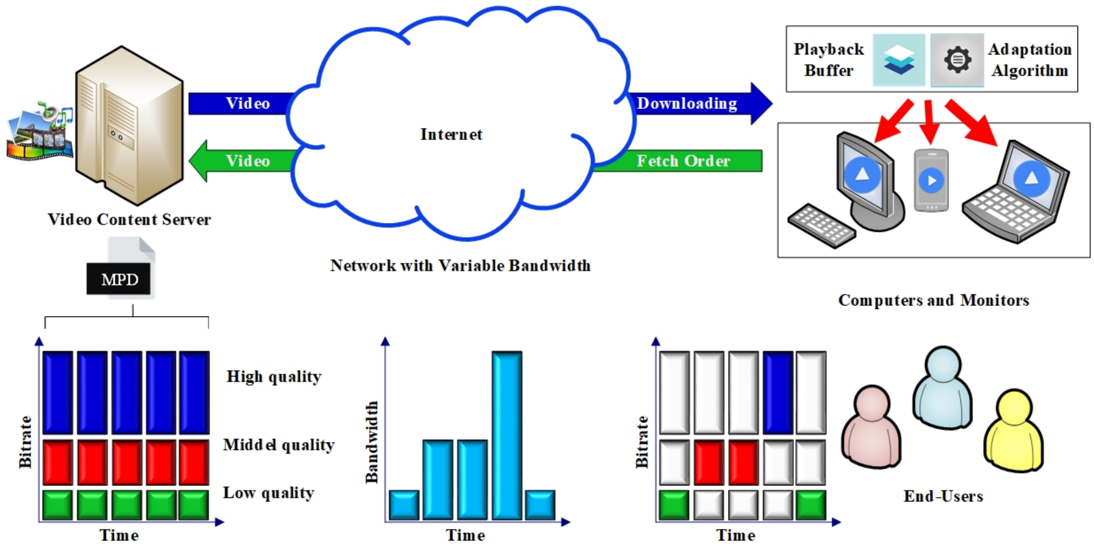



**Figure 1.2:** *DASH streaming flow process.*

DASH marks several points:

- It can be built above the omnipresent HTTP;
- It privileges user context by allowing users to choose the best suited solution which is **adapted** to his/her networking context in order to provide an optimal watching experience;
- it is open to various existing video encoding technologies and easily evolving to future technologies.

Today, the majority of the video content providers such as (NETFLIX & YouTube) use DASH.

Among the major challenges related to DASH, the video playback freezing, termed as **rebuffering**, is a main issue. This phenomenon occurs when the network is incapable of transmit subsequent video chunks before the starvation of the already downloaded video content. This is typically an adaptation problem: the required video chunks need better networking conditions to be downloaded *on time*. In other words, there is a lack of adequation between the version of video chunks being downloaded and the networking context.

A corollary phenomenon is the **instability** of the visual quality of the video being played back. This phenomenon occurs when there are frequent fluctuations of video quality during the video playback. Indeed, different versions of chunk encoding offers different video quality (due to different bitrates and/or resolutions). When two adjacent video chunks are displayed with different versions, it is possible that the difference in video quality produces an unpleasant watching experience.

DASH offers a framework for adaptation, whereas it is up to each end-user to make the adaptation decision. The adaptations strategy, termed subsequently as *adaptive bitrate mechanism* (**ABR**), is thus a very critical point. The ABR mechanism is located on the user's side. It tries to select the optimal version of the next chunk to be fetched (downloaded) based on various parameters, such as estimated bandwidth, buffer occupancy etc.

It is not easy to design an ABR streaming algorithm to achieve a balance between conflicting metrics, including minimizing rebuffering events, minimizing instability, or maximizing mean bitrate. For instance, minimizing jointly the risk of rebuffering and also the instability is really a challenging issue. Actually the natural approach for avoiding rebuffering and the one for avoid instability are not quite the same:

- the former consists in *predicting* the most suitable version, with possible switching to a new one, i.e. frequent version changes if necessary; whereas

- the latter advocated for conservation of the existing version as long as possible.

Furthermore, the predicting itself is not a easy job, since it is based on the prediction of the evolving networking conditions. A "*safe*" way to minimize both rebuffering and instability would be keeping claiming the lowest bitrate version. Of course, it is at the price of the lowest bitrate by wasting available bandwidth. It offers certainly the worst video quality and so watching experience.

Most of the proposed adaptation mechanisms work mainly with parameters at network and/or system level, such as the bandwidth forecast, buffer occupation, etc. We believe that it is important to give importance to objective video quality measurements in the design of ABR algorithm, and consider it as a **key** parameters, in addition to the more widely technical and resources-oriented parameters such as buffer occupancy, bandwidth. Indeed, the user's watching experience has to be taken into account. Objective video quality measurement offers an efficient way to combine the feasibility (offered by the automatic computation) and the user experience (quality metric). This is the general guideline and motivation of this thesis.

## 1.2 Motivations and Contributions

The main objective of this thesis aims to propose a novel adaptation framework for DASH-based ABR which takes into account the user QoE in general, and, more particularly the objective video quality. The main idea of this framework is the following:

- we make use of objective video quality metric (which is rarely used in the existing works) for the simple reason that, after all, users are fundamentally and mainly aware of video quality.
- based on this objective indicator, we develop our adaptation mechanism in a way that a possible upgrade to a higher bitrate version, which is possible according networking context forecast, takes actually place only when it does carry a noticeable upgrade in video quality also. In other words, we decide to:
  - increase the bitrate level only when the objective Video-Quality-Metric (VQM) indicates a significant improvement in the video quality (in this way, we reduce the no-necessary bandwidth consumption and minimize the risk of rebuffering), and
  - decrease the bitrate level only when there is a real risk of rebuffering (thus minimize the instability).
- The combined effect of this approach contributes to maximizing the video quality and minimizing the rebuffering and instability.

From operational point of view, our approach is compatible with the DASH scheme, since the objective video quality metric can be pre-computed and made available, through MPD, to end-user for adaptation purpose.

This idea can be applied to all objective video metric. In this sense, we consider it as a generic framework. We have firstly tested it with the Structural Similarity Index Measurement

(SSIM) [10]. We then carried experiments [11] with Peak Signal-to-Noise Ratio (PSNR) and Video Multimethod Assessment Fusion (VMAF) as well.

We have carried experiments under real traffic situations, by using traces captured in real mobile network. Our studies were conducted with comparison to some non video-quality-aware ABR (BBA, FESTIVE, OSMF, see § 3.5 for a detailed presentation of these algorithms). These studies show that our framework does achieve our design objectives.

Our works have been published on two international conferences [10, 11]. Also, there is an ongoing submission process to the journal *Signal, Image and Video Processing* (Springer).

## 1.3 Outline

The rest of this thesis is organized as follows:

i. Chapter 2 presents and analyses fundamental concepts and related works about quality-of-service (QoS), quality-of-experience (QoE). We then make focus on visual quality metric (VQM) assessment, which is a key component of this thesis. We presents and analyses fundamental concepts and works related to visual quality assessment. We then deal with DASH-related adaptive bitrate (ABR) streaming algorithm.

ii. Chapter 4 presents firstly our generic framework termed as *Video-Quality metric Based-Adaptation algorithm* (VQBA). We then present our experimentations with the SSIM visual quality metric (VQM). The results prove that our design goals are justified.

iii. Chapter 5 extends the studies to other two common VQM, i.e. PSNR and VMAF. We provide also a focused study on the rebuffering phenomena with our algorithm.

iv. Chapter 6 presents the conclusions and future working directions.

Chapter 2

# State of the Art on QoS & QoE for Video Streaming

> "*To know what you know and what you do not know, that is true knowledge.*"
>
> Confucius

This chapter first introduces the fundamental concepts on quality-of-service (QoS) and quality-of-experience (QoE). A state-of-the-art on visual quality assessment for video streaming is then presented where the different QoE assessment techniques are discussed. The concept of objective visual quality assessment (VQM) metrics is presented where the most common and well known objective metrics to assess/enhance the QoE for video streaming such as SSIM, PSNR and VMAF are presented.

## 2.1 Quality of Service and Quality of Experience

### 2.1.1 Quality of Service (QoS)

The concept of Quality-of-Service (QoS) was proposed by the International Telecommunication Union (ITU) in 1994 [12]. QoS refers to the measurement of the performance of a *service*, such as a computer network or a cloud service, seen by the users of the service.

Typical QoS metrics of the network service include packet loss, jitter, transmission delay, throughput, bitrate, etc. QoS characterises services provided by the underlying netwoking infrastructure. Adequat resource dimensioning and/or provisioning allow to ensure a specific level of performance to a data flow, or data flows. QoS metric related to video streaming services include usually packet loss, transmission delay, throughput, bitrate.

### 2.1.2 Quality of Experience (QoE)

The term *Quality of Experience* (QoE) refers to the overall perception by a user of the quality of a video under his/her visioning. According to ITU-T [13], QoE is the overall acceptability of an application or service, as perceived subjectively by the end-user. For video QoE, it is a perceptual assessment that reflects viewers' satisfaction with their video streaming experience.

It is obviously a paramount concept for all video communications systems, It is also by definition a very *subjective* concept. Moreover, it is also application and context dependent.

The perceptual system of human-being is specific to each person, the perceived quality threshold varies from user to user. A user may not perceive a service in the same way as his/her peers. Thus, the perceptual quality is by definition subjective and application dependant.

One of the major challenge for video content provider is how to delivery the video content to end-user with the best achievable QoE under various networking and system constraints. One of the major challenges for networking operators is how to achieve a fair-share of network resources so that the user QoE is maximized for all users in a network.

Roughly speaking, the concept of QoE describes the subjective perceived quality of end-users versus a service, which is typically a video service. [14–17]. In some extends, it offers a user-oriented complementary indicator to the (very) objective QoS metrics.

## 2.2 Relation between QoS and QoE

The term "quality of experience" itself is an extension of the "quality of service" concept from the networking community. The relationship between QoS and QoE can be illustrated as given by Fig 2.1 [18, 19].

Since the 1990s, as the video communication was taking an increasing importance, it becomes obvious that the sole networking-oriented QoS parameters can no longer provide satisfactory criteria for video communication.

The purpose of QoE is to understand the end user's experience and expectations (the end user's experience and their level of satisfaction with the offered service). The QoE metrics help improving existing technologies and developing better future services.

User experience and the concept of QoE was originally promoted by Human-Computer Interaction (HCI) researchers to stress concern with the outcomes of people's experience with-

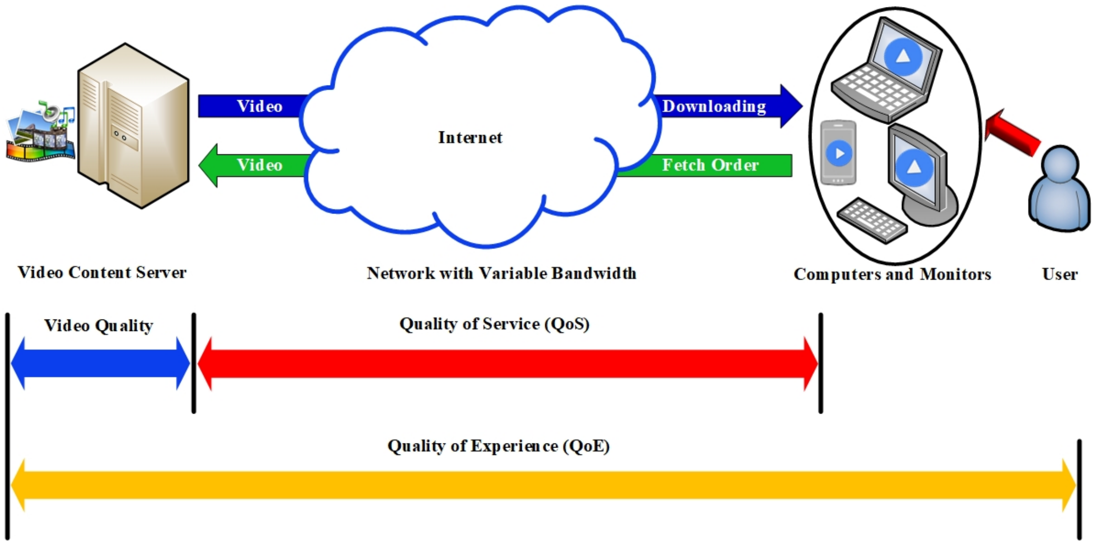



**Figure 2.1:** *The relationship between QoS and QoE.*

/through technology [20]. Some researchers give the QoE even a wider definition by stating: quality of experience deals with all relevant aspects that define how satisfied a person is with a service [21, 22]. The perceptual system of human-being is specific to each person, the perceived quality threshold varies from user to user. A user may not perceive a service in the same way as his/her peers. Thus, the perceptual quality is by definition subjective and application dependant.

While monitoring and managing QoS parameters are important for high quality delivering of video content, it is also important to evaluate the received video's *perceptual* quality from the users' perspective. Indded, the latter has a major impact on QoE. QoE-based video quality evaluation is challenging because the user experience is subjective, hard to quantify, and measure. The QoE is then a set of performance metrics that concentrates on the viewpoint of user satisfaction. The current measurement studies typically classify QoE into two categories: *subjective* QoE versus *objective* QoE. Fig. 2.2 gives [23] some of the existing metrics of QoE and QoS, respectively.

Subjective QoE is based on users' opinions. An example of the most generally used subjective QoE metric is the *Mean Opinion Score* (MOS). In this thesis, we concentrate on the objective metrics, i.e., metrics that can be automatically computed.

## 2.3 QoE Impact Factors

The existing QoE prediction models take into consideration the so called *Influence Factors* (IFs) (cf. Figure 2.3 [24]).

A QoE IFs is "*any characteristic of a user, system, service, application, or context whose actual state or setting may have an influence on the QoE for the user*" [24, 25]. The impact

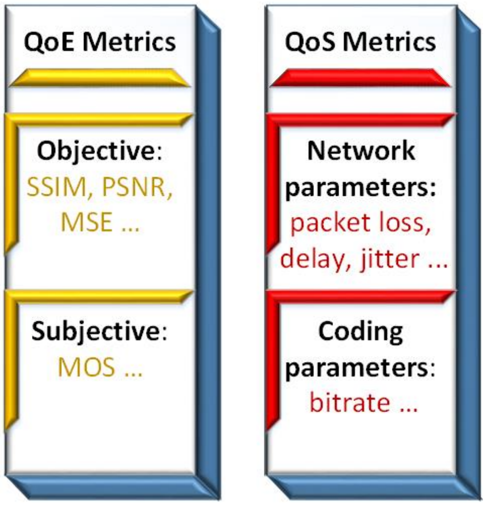



**Figure 2.2:** *Some metrics of QoE and QoS .*

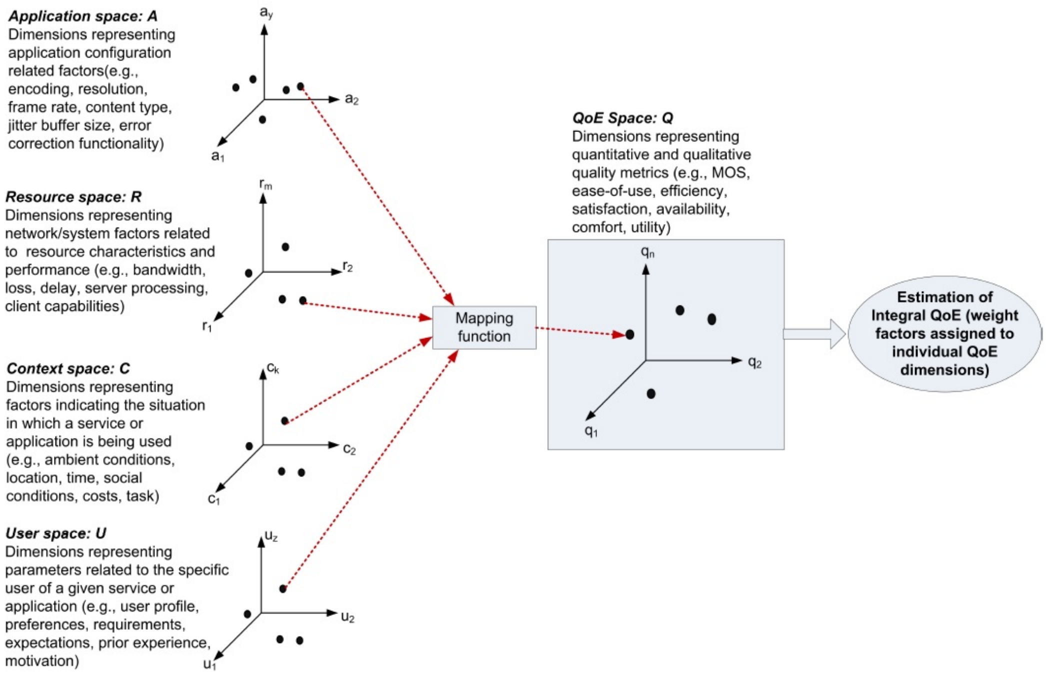



**Figure 2.3:** *the model QoE influence factors.*

factors on any QoE model can be categorized into four multi-dimensional spaces [24, 26, 27]:

i. System IFs: includes many aspects that are related to media such as characteristics that determine the technically produced quality of an application or service or network related such as (wired/wireless/mobile, bandwidth, delay, jitter, packet loss, etc.).

ii. Human IFs: include individual characteristics of a user such as memory and recency effect, his usage history of the application (e.g., browsing history), his expectations from the service and the characteristic can describe the demographic and socio-economic background, the physical and mental constitution, or the user's emotional state.

iii. Context IFs: include location, viewing environment, time of the day, type of usage, and time of service consumption; "are factors that embrace any situational property to describe the user's environment in terms of physical, temporal, social, economic, task, and technical characteristics" (peak time, etc.).

iv. Content IFs: explains the characteristics of the contents such as type of video, its duration and content aspects related to complexity including (temporal and spatial complexity).

## 2.4 Measurement of QoE

Service Providers (SPs) use often QoS parameters, such as bandwidth, jitter or delay as indicator of QoE. Actually, these ones are valuable operational parameters. However, we know that they are not enough.

The ultimate performance indicator for video streaming services is clearly the end-user QoE [28–31].

The challenging point of QoE is two folders:

- How to find pertinent networking and system parameters?
- How to choose and take into account other parameters with impact on QoE?

In [29], the issue of fair QoE measurements in networking is addressed, in order to enhance the user experience (to get user satisfaction). Their work provides a brief survey on how to measure the quality of the experience and how different layers in the network environment affect the quality of the perceived experience. To control their resources while maintaining user satisfaction, the video content providers need to take into account not only the QoS, but also the QoE.

In [31], they offer three approaches namely subjective approach, objective approach, and hybrid approach in order to assess as accurately as possible this perceptual quality in video streaming applications over wireless networks in different network conditions. They focus on a hybrid approach called Pseudo Subjective Quality Assessment (PSQA), which can be run in real-time.

In [28] they conducted their experiments based on the subjective video quality assessment (MOS) to determine user satisfaction of video streaming services. They investigated video-user interaction (engagement) through various video QoE metrics such as i) buffering ratio, ii) average bitrate and iii) buffering events, etc.

In [30] they proposed two dynamic server selection systems based on users' QoE feedback for the Video on Demand (VoD) system, where they use users' an online QoE model (that evaluates bitrate and rebuffering events for each video chunk during the video streaming session) as an assessment of server performance.

## 2.5 Basic Concepts on Video Quality Assessment

Most of the visual quality assessment metrics were first designed to evaluate the visual quality of the distorted images and were later extended to videos [32]. The VQA of a distorted video can be performed in two ways: subjectively or objectively. The subjective VQA follows the protocol described in the standard [33]. Under the same experimental conditions, a large panel of observers evaluate the quality of a given video. Each observer ranks the observed video according to his/her visual perception on a scale ranging from 1 (i.e., worst quality) to 5 (i.e., best quality). The final score, known as the MOS (Mean Opinion Score), is obtained by averaging the individual assessments [34]. Although the subjective VQA is in accordance with the human perception, its implementation remains complex (e.g. the cost of conducting this type of experiment is high and requires a large number of participants, etc). Table 2.1 compares the video quality assessments methods (subjective and objective).

However, objective VQA is derived from an analytical expression validated by subjective quality analysis during its development [35]. Indeed, several objective metrics with reference (i.e. Full-Reference (FR) such as SSIM, MS-SSIM VQM VFD, MOVIE, ST-MAD, VMAF and FLOSIM), without reference (i.e. No-Reference (NR) such as MREBN) or Reduced-Reference ((RR) such as STRRED) have been proposed, for more details one can refer to [36]. Most of them aim to get as close as possible to the Human Visual System (HVS) to apprehend the visible distortions. Next section proposes the study to a wider overview of video streaming QoE

**Table 2.1:** *Comparative of quality assessments methods.*

| **Characteristic** | **Subjective methods** | **Objective methods** |
|---|---|---|
| A direct measure of QoE | Yes | No |
| Real-time | No | Yes \ No |
| Cost | High | Low |
| Wide Application | Limited | Limited |

assessment methods.

## 2.6 QoE Video Quality Assessments Methods

The scientific literature shows that video quality assessment results naturally from the techniques originally proposed to assess the visual quality of distorted still images. To measure the QoE, one can classify the proposed VQA methods into subjective and objective methods.

The subjective method refers to the evaluation of the quality of the services, where human subjects measure or quantify performance and quality (many participants observe a sample video to understand their personal perception of quality service). The main drawbacks of the subjective method is that they cannot be implemented in real-time due to the lack of repeatability. These limitations and challenges have encouraged and motivated the creation and development of objective methods that predict subjective quality from the network/media parameters. However, it is difficult to correlate the objective methods to human perception and some may require high computational.

There are different strategies for measuring subjective and objective visual quality. Table 2.1 compares the subjective and objective VQA methods. The following subsections discuss the common QoE measurement methods [37].

### 2.6.1 Subjective Quality Assessment

The International Telecommunication Union (ITU) describes QoE as the total admissibility of service, as perceived subjectively by the end-user. For the quality of service DASH video streaming, it can be concluded that the users' satisfaction level of video content delivered by the video content server which is considered. This definition means that the evaluation of the quality of experience of video streaming is done through subjective experiments. Subjective experiments represent the most accurate method for obtaining visual quality ratings. In subjective tests, commonly a number of users are required to observe a set of video clips for later on to rate what they have seen and experienced. The evaluation is usually done in accordance with the MOS. The quality level of a video sequence based on a MOS model is evaluated on a scale ranging from 1 to 5. Observers are required to rate quality parameters utilizing a standardized five-point scale with labels such as Excellent, Good, Fair, Poor and Bad, where 5 is the best quality while 1 represents the worst one [34]. Since this is in the subjective domain, one has to expect some variability of the users' ratings as people have different interests and expectations for the video. One can minimize such factors through specific guidance and training. The ITU proposes standard viewing conditions, criteria for evaluation procedures for selecting test users and techniques for analyzing data and materials.

### 2.6.2 Objective Quality Assessment

Objective measurements are typically technology-centric where data is automatically collected by monitoring tools. User tests are then essential to identify and verify the relationship between technical parameters and the perceived quality (which is part of QoE). Objective QoE measuring techniques are based on network-related parameters that need to be gathered to predict the users' satisfaction. Objective measurement methods follow either an intrusive approach, requiring image/video reference content, or a non-intrusive approach that does not require reference information to predict the quality of the experience. Many algorithms referred to as objective quality metrics have been proposed in the existing work for the in-service objective quality evaluation of video sequences. The objective quality assessment methods can be based on full reference quality metric, reduced-reference quality metric and no-reference quality metric.

The studies carried out on video streaming work show that VQA metrics are used for a variety of purposes:

- to assess the quality of the received video using VQA metrics;
- to improve the quality of the transmitted video; and
- to predict the QoE that should be correlated to the subjective VQA.

In what follows, full-reference video objective metrics as well as for improving or predicting the QoE are first described.

#### 2.6.2.1 Full-Reference Video Quality Metrics

Full-reference video quality metrics are the most common, normally use as an evaluation of the degree of distortions in the received content [38]. The simplest way to assess the video quality metrics and most widely technique is the Mean Squared Error (MSE), which is computed by averaging the squared intensity differences of distorted values from the original values. These are attractive because they are simple to compute meanings. But these metrics are not very well since they do not correspond to the perceived visual video quality. In the latest three-decade, a major effort contributed to the development of video quality assessment methods that take into account the features of the HVS. The majority of the suggested perceptual quality evaluates models that followed a strategy of modifying the MSE measure so that errors are penalized in accordance with their accuracy. Evaluation of image quality algorithms based on a full reference video quality assessment measure like the MSE, SSIM and PSNR.

Full-reference video quality metrics that are usually used in video codecs comparisons are expected to reflect any variations and changes in videos. The only weakness of the FR approach compared to the others is the need to have the original video for comparison with the encoded video (observed), which is often not handy (unavailable). For example, SSIM, as it is a full-reference metric (i.e., its computation requires full information of the original video chunk), it cannot be calculated at the client video streaming side, but its values when varying the video representation can be conveniently pre-computed, stored on the video streaming server-side, and included as a matrix in the MPD. The computational procedure for calculating the SSIM values is done on the video streaming server-side. [39–41].

#### 2.6.2.2 No-Reference Video Quality Metrics

Contrary Non-Reference metrics are metrics assessing the visual quality of the distorted information without using the original information as a reference. These kinds of QoE metrics are more related to online services, where the delivery network is shared by other services. In video streaming services, it is hard to determine if the variation in the quality is due to the quality of the reference or due to the intermediate elements. Therefore NR metrics are the most suitable to measure the online streaming services, due to the distance between the video users and the video streaming server and also due to a lack of separate feedback channels. This is a hindrance to deriving QoE-QoS relationships aiming at capturing the impact of the network [42]. In many practical applications, where the reference image/video is not available, and a no-reference or “blind” quality assessment approach is desirable. This is because no-reference metrics assessment are designed to imitative visual perception (i.e., learned from large-scale perceptual scores) without requesting original images as a reference to be compared. Objective metrics such as the bitrate, no-reference metrics, which gauge the video distortion solely from the received frames (i.e., no external quality reference is provided). No reference image quality assessment algorithm that corresponds to visual perception [39]. Existing NR image quality evaluation methods are all learning-based, but the training images are low quality by compression, noise or fast fading rather than high-resolution. As a result, state-of-the-art on NR image quality evaluation methods are less effective for accounting for the artefacts such as incorrect high-frequency details introduced.

#### 2.6.2.3 Reduced Reference Video Quality Metrics

The RR video quality assessment, where only incomplete information from the reference can be made available in addition to the distorted video to conduct quality evaluation RR video quality assessment methods estimate the amount of distortion in the distorted video using particular information from the original video. RR methods only use a limited number of features of the original video. The same set of parameters are derived and compared from both reference and encoded copy. Certain parameters can exist at the network layer such as packet loss or at the application layer such as bitrate [43–45].

## 2.7 SSIM, PSNR and VMAF metrics

Among the different metrics available in the scientific literature, the section below presents the relevant metrics on which we have based the proposed Adaptive BitRate algorithms that will be presented in the next two chapters.

### 2.7.1 Structural Similarity Index Measurement (SSIM) QoE Metric's

The SSIM is a metric that predicts the perceived goodness of images and videos. The basic model was developed in the Laboratory for Image and Video Engineering (LIVE) at the university of Texas at Austin and further developed jointly with the Laboratory for Computational Vision (LCV) at New York university [39]. SSIM metric intended to take the HVS aspects into consideration throughout the evaluation process. The SSIM index is a full reference metric, the measurement or prediction of image/video quality is related to the original image/video as a reference. The SSIM is a based design taking into account the image/video degradation as the perceived changes in the structural information including important perceptual aspects such as luminance and contrast aspects and combining them into a single value, called index. The SSIM index is a decimal value ranging between 0 and 1, where zero represents the worst visual [39].

Figures 2.4, 2.5 and 2.6 provide the SSIM values of different video quality levels using "animation", "documentary" and "sport" video test used during our experiments (cf. § 4.4.3 for a more detailed explanation of these sequences).

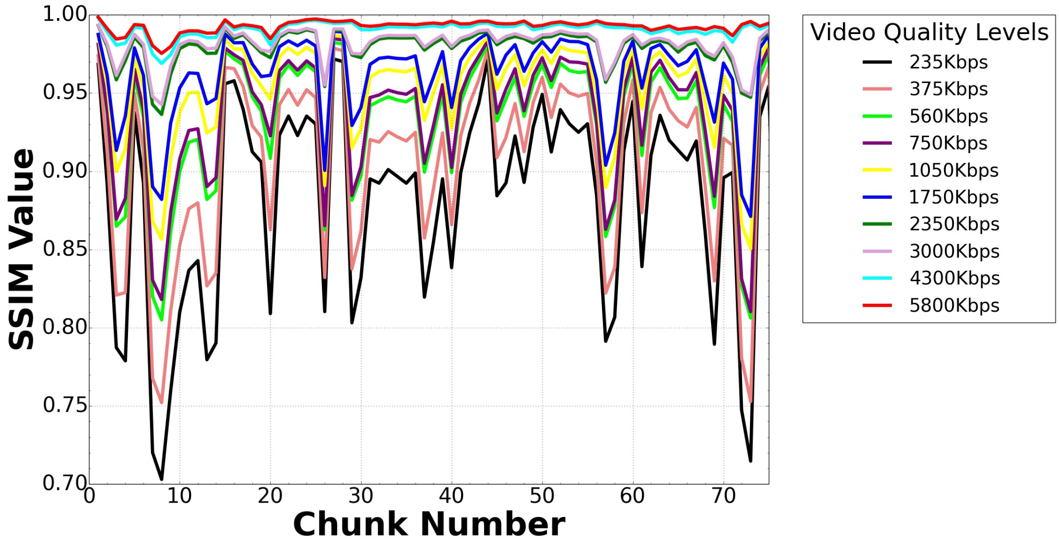



**Figure 2.4:** *SSIM values of different video quality levels using "Animation" video test.*

Figure 2.7 shows the SSIM index map of the movie Big Buck Bunny.

The SSIM metric computed between the original samples $x$ and its distorded version denoted $y$ is given as follows [39, 46]:

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma}, \tag{2.1}$$

where *l(x,y)* is given by:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{2.2}$$

and *c(x,y)* is the constrat comparison given by:

$$c(x,y) = \frac{2\varphi_x\varphi_y + C_2}{\varphi_x^2 + \varphi_y^2 + C_2}, \tag{2.3}$$

and *s(x,y)* is the structure comparison defined as follows:

$$s(x,y) = \frac{\varphi_x y + C_3}{\varphi_x\varphi_y + C_3}. \tag{2.4}$$

where $\mu_x$, $\mu_y$, $\varphi_x$, $\varphi_y$ and $\varphi_x y$ are the local means, standard deviations and cross-covariance for samples $x$, $y$.

Setting the weights $\alpha$, $\beta$, $\gamma$ to 1 and $C_3 = \frac{C_2}{2}$ the formula can be reduced to the form shown at the top of this section.

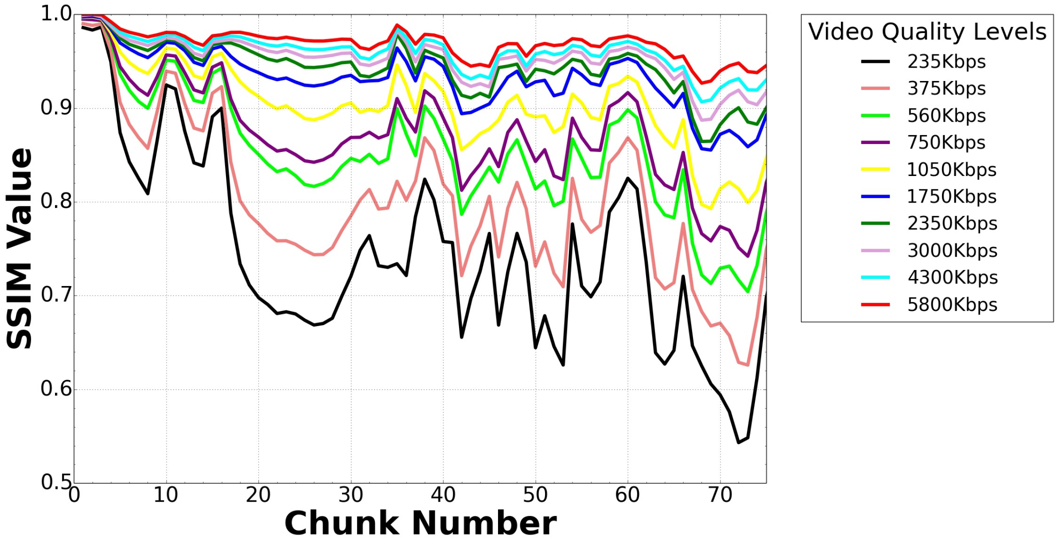



**Figure 2.5:** *SSIM values of different video quality levels using "Documentary" video test.*

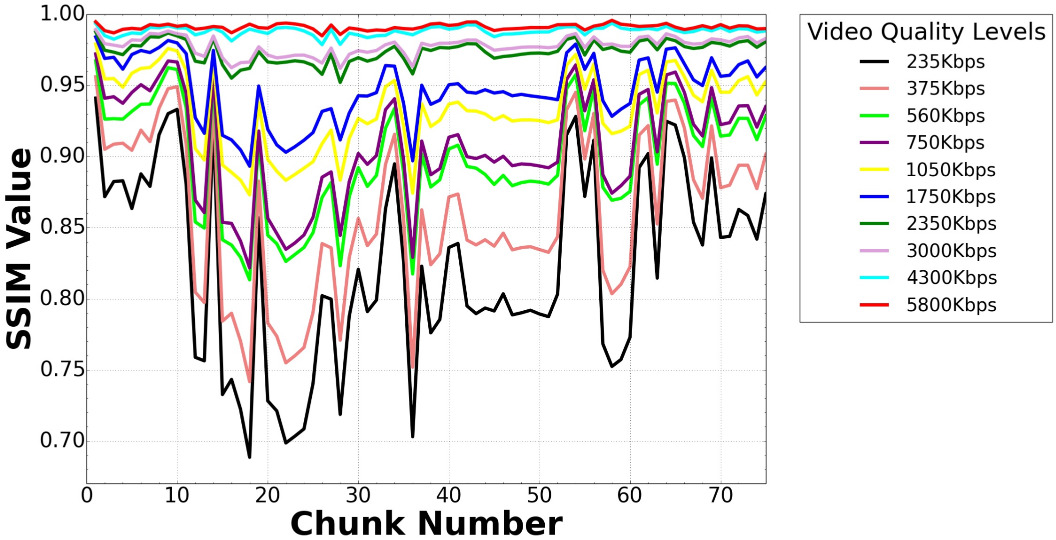



**Figure 2.6:** *SSIM values of different video quality levels using "Sport" video test.*

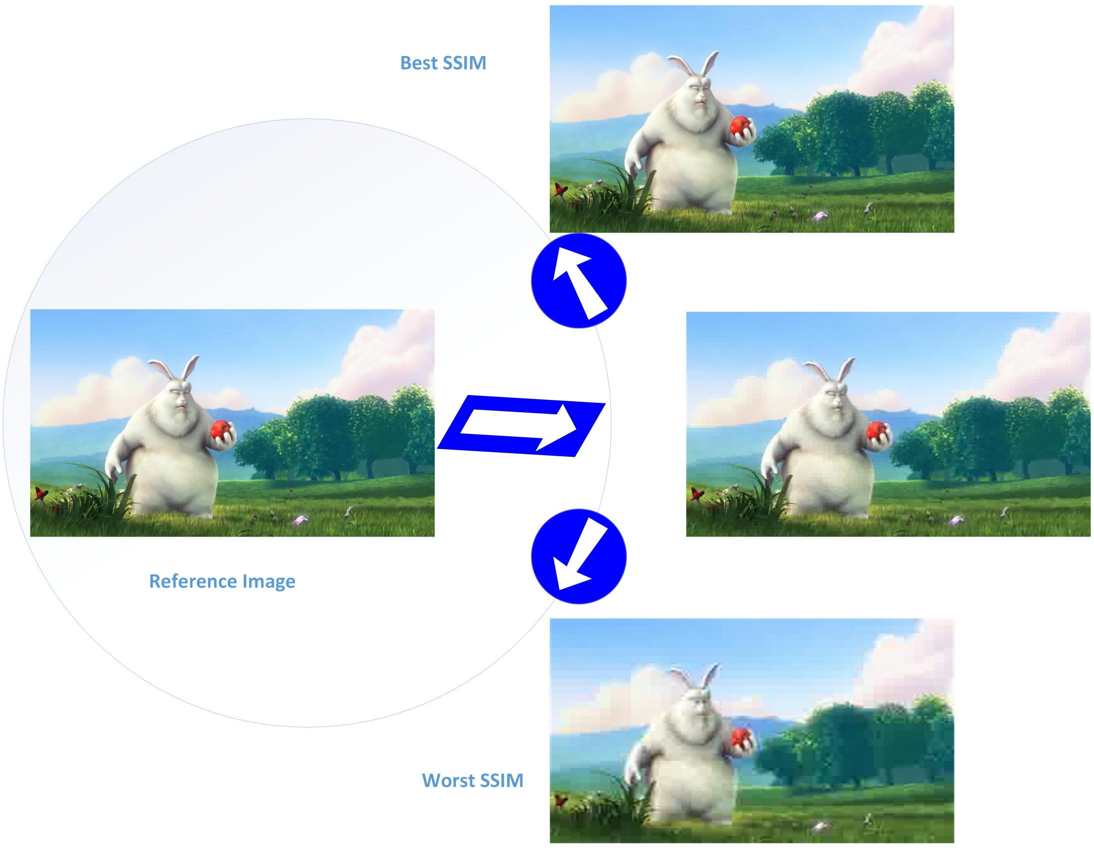



**Figure 2.7:** *Comparison of SSIM index - the movie (Big Buck Bunny).*

**Table 2.2:** *Mapping of objective QoE (SSIM) to subjective QoE (MOS).*

| MOS | Quality | SSIM |
|---|---|---|
| 5 | excellent | $> 0.99$ |
| 4 | good | $\geq$0.95 and: $<$0.99 |
| 3 | fair | $\geq$0.88 and: $< 0.95$ |
| 2 | poor | $\geq$0.5 and: $< 0.88$ |
| 1 | bad | $< 0.5$ |

This results in a specific form of the SSIM index:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\varphi_x y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\varphi_x^2 + \varphi_y^2 + C_2)}, \tag{2.5}$$

where $C_1$, $C_2$, $C_3$ are constants for the luminance, contrast, and structure, specified as a non-negative numbers.

In our work, we calculate the video quality metric (SSIM) for each video chunk as an average of the SSIM of each image of the video as:

$$SSIM(k,r) = \frac{1}{T \times FPS} \sum_{x=0} SSIM(K,r)_x, \tag{2.6}$$

where $k$ is the chunk at the $r$-th bitrate level and $T$ is the video chunk duration. Each video chunk is encoded in a given frames per seconds ($FPS$).

Among the different metrics, the similarity index is considered as a good objective metric due to its proven performance [32]. The state of the art shows that this metric has also been selected as a metric to assess the QoE (see e.g., [47–50]).

The metric outputs a value from 0 to 5 (5 is the best possible score) to present the image/video quality level based on the human visual system and subjectivity aspects, including blurring, block distortion and color distortion. Table 2.2 represents a mapping of objective QoE (SSIM) to a nominal 5-point MOS scale subjective quality assessment QoE based on [39, 51].

### 2.7.2 Peak Signal-to-Noise Ratio (PSNR) QoE Metric's

The objective quality assessment metric PSNR is just a logarithmic representation of the MSE, computed between the original (i.e. reference) visual information and its degraded version [38, 52]. The PSNR mathematically is as simple to understand and implement as it is easy and fast to compute. Over the years, video researchers have developed a familiarity with PSNR that allows them to interpret the values immediately [52,53]. The higher the PSNR, the better the quality of the compressed, or reconstructed image. At the other end of the scale, a small value of the PSNR implies high numerical differences between the original and a decoded image. PSNR is based on a pixel by pixel comparison of the data without considering what the pixels actually represent, and thereby only represents an approximate relationship with the video quality perceived by human observers.

Consider an original image denoted $I$ of size $M \times N$ and $K$ its distorted version. Denoted $I(i,j)$ the pixel of the image $I$ located at position $(i,j)$. Denoted $m$ the number of the bits used to represent a pixel. The PSNR, then comparing the distorted image to its original version, is given by the following expression:

$$PSNR = 10\log_{10}\left(\frac{b^2}{MSE}\right), \tag{2.7}$$

where

$$b = 2^m - 1, \tag{2.8}$$

and

$$MSE = \frac{1}{N.M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[I(i,j) - K(i,j)]^2. \tag{2.9}$$

The full reference metric (PSNR) is one of the most popular image quality metrics. It is a well-known measurement method that aims to also measure the video fidelity. In our thesis, we calculate the video quality metric (PSNR) for each video chunk as of the PSNR values to all images as:

$$PSNR = 10\log_{10}\left(\frac{b^2}{\frac{1}{T\times FPS}\sum_{x=0} MSE(K,r)_x}\right), \tag{2.10}$$

Table 2.3 presents a mapping of the objective QoE (PSNR) to a nominal 5-point MOS scale subjective quality assessment [54,55].

### 2.7.3 Multimethod Assessment Fusion (VMAF) QoE Metric's

Video multimethod assessment fusion has been developed by Netflix [56]. VMAF is a metric that needs the availability of a full reference information. VMAF is a fusion-based metric that is gaining popularity in the area of video quality assessment. Its essential concept is to combine multiple elementary video quality features, to balance between high performance and computational efficiency, such as (i) Visual Information Fidelity (VIF) [57] (ii) Detail Loss Metric (DLM) [58] and (iii) Temporal Information (TI). All these features/metrics are integrated into a

**Table 2.3:** *Mapping of objective QoE (PSNR) to subjective QoE (MOS).*

| MOS | Quality | PSNR |
|---|---|---|
| 5 | excellent | $\geq 45$ |
| 4 | good | $\geq 33$ and: $< 45$ |
| 3 | fair | $\geq 27.4$ and: $< 33$ |
| 2 | poor | $\geq 18.7$ and: $< 27.4$ |
| 1 | bad | $< 18.7$ |

final metric using a machine learning algorithm Support Vector Machine (SVM). A large sample of MOS scores were used as ground truth to train a quality estimation model. The resulting regressor is used for estimating per-frame quality scores on new videos. VMAF which combines scores from three different metrics which were mentioned above to obtain a single score between 0 and 100, with a higher score denoting a higher quality, where observed that VMAF predictions have a very high correlation with subjective video quality rating.

## 2.8 Conclusion

This chapter introduces the background information and related work of this thesis. We firstly provide an introduction to the concepts fundamental concepts of quality-of-service (QoS), quality-of-experience (QoE) as well as related work on QoS and QoE. We then make focus on objective visual quality metric (VQM) assessment, which is a key component of this thesis. In the next chapter, we will present also background information and related to Dynamic Adaptive Streaming over HTTP (DASH) standards and DASH problems, in addition to reviewing the specimen of Adaptive Bitrate (ABR) algorithms that we used in our work.

Chapter 3

# State of the Art on DASH Video Streaming

> *"Though no one can go back and make a brand new start, anyone can start from now and make a brand new ending."*
>
> Carl Bard

In this chapter, we present the Dynamic Adaptive Streaming over HTTP (DASH) standards and its chalenges. Following, this chapter reviews the significant Adaptive Biterate (ABR) algorithms to improve the Quality of Experience (QoE) for video streaming application.

## 3.1 MPEG-DASH Technology for Multimedia Streaming

The dominant video streaming technology is Dynamic Adaptive Streaming over HTTP (DASH) through the Internet based on TCP/HTTP [2,3,7,59]. Where the live and Video on Demand (VoD) services increasingly using this technology, which is an international standard MPEG-DASH. Figure 3.1 depicts the DASH flow streaming process.

DASH uses Hypertext Transfer Protocol (HTTP) as application layer protocol and Transmission Control Protocol (TCP) as the transport protocol. Where, DASH consists of the Adaptive Bitrate (ABR) algorithm as a key element, wherever ABR still remains an open issue.

Today, many video content providers (including NETFLIX and YouTube) have switched to ABR streaming to maximize Quality of Experience (QoE) for video users [7,60]. Adaptive bitrate streaming was introduced with the primary purpose of adapting video quality to network

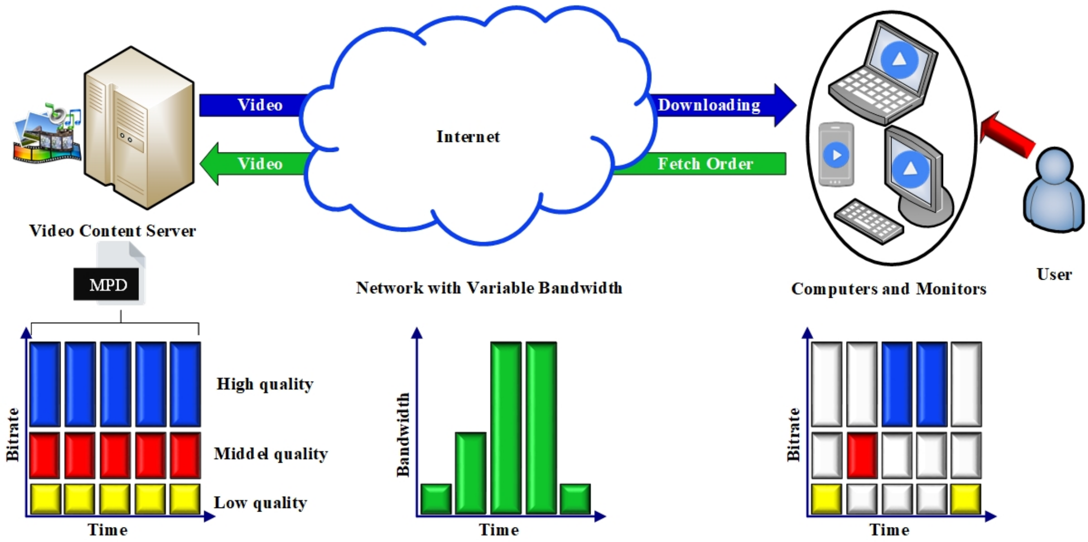



**Figure 3.1:** *DASH Streaming Flow Process.*

bandwidth variations to decrease video rebuffering, rate switching, and maximize the overall video quality in order to achieve the satisfaction of the end user of the video.

Well-known standards for ABR algorithms were introduced by Move Networks and are now being developed and used by Adobe Systems, Apple, and Microsoft, etc. Streaming includes the following:

- Adobe: HTTP Dynamic Streaming (HDS)
- Apple: HTTP Live Streaming (HLS)
- Microsoft: Smooth Streaming (MSS)
- HTTP Adaptive Streaming (HAS)

In addition, the DASH industry forum has been formed in order to enable smooth implementation of DASH, one of their achievement is DASH-AVC/H264 a recommendation of profiles and settings serving as guidelines for the DASH implementation with H.264/AVC video. Despite the many ABR implementations that have been proposed related to the DAHS approach. Nevertheless, there are still differences not only between commercial products but also inside the same product, even within MPEG-DASH, such as:

(a) the buffer size of playback,

(b) duration of the video chunk, and

(c) the conditions of the ABR algorithm that selects the bitrate level of the next video chunk.

The impact of QoE metrics such as (Rebuffering, Rate switching, etc.) on user satisfaction (DASH users) is significant [41, 61]. This is due to the fluctuations of the mobile network over time.

Additionally, the use of HTTP allows efficient traffic of NATs and Firewalls [62], This is not the case with Real-Time Transport Protocol (RTP) over User Datagram Protocol (UDP) streaming solutions.

In the DASH server, the media file is segmented into chunks of fixed duration, e.g., (1 10) seconds, which can be pre-compressed with several versions at different bitrates and/or qualities. For instance, when using a single (multi) layer codec such as AVC (SVC), each segment has different versions (layers). The segments are provided on a web server and can be downloaded through HTTP standard-compliant GET requests. The adaptation to the bitrate is done on the client-side for each segment, e.g., the client can switch to a higher bitrate - if bandwidth permits - on a per-segment basis. The temporal and structural relationships between segments are described in the Media Presentation Description (MPD) file.

### 3.1.1 DASH Server Side

The common principle of server-side DASH is as follows [63–67]:

i. The original video segmented into chunks.

ii. Each video chunk is pre-compressed with several versions at a different resolution, and/or bitrate levels; where the information about each video chunk used like an elementary unit for adaptation. Our contribution is to add additional information about each video chunk based on video quality metrics into MPD in order to optimize the adaptation mechanism video QoE.

iii. Actual adaptation decision is made by client: we propose an ABR algoritme based on video quality metrics.

### 3.1.2 DASH Adaptation Interaction: Server/Client

The DASH streaming flow process is the interaction between the server and the client. Where on the client-side, the DASH player requests chunks decode and displays them successively on the client screen. To load a video [63, 66, 67]:

- A video client periodically requests an individual chunk from a video server.
- As the video chunks are downloaded, they are stored in a playback buffer.

- Video players use ABR algorithms to selects bitrate for the next video chunk.

The specificity of DASH streaming technology is that it offers an adaptation to the network/system conditions; in fact, it enables the DASH client to switch from one bitrate level to another within the same DASH server video content stream; in our contribution: we add additional information related to visual quality metrics for each video chunk for adaptation. In common ABR implementations, the selection of the bitrate level of each requested chunk depends essentially on the estimation of the available bandwidth on the client-side.

This can become problematic if the chunks we request have higher bitrates than the network can support, this ultimately leads to an empty buffer and thus a rebuffering period. To prevent issues like this, video players use ABR algorithms.

## 3.2 Progressive download of video chunks via HTTP

The process of transmission video streaming refers to the delivery of the video content where the video chunk is displayed at the video client-side while being delivered by a video content provider simultaneously. The video client-side starts to download the video chunks progressively and stored them in the application buffer (playback buffer). Definitely, the first video chunk starts playing just when it's completely downloaded on the video client-side. The video chunks are downloaded at the same time as the video is playback, i.e., the buffer is filled with newly downloaded chunks, while it replaced as the downloaded chunks are played out [68,69].

With fluctuating mobile network conditions, the buffer slowly fills compared with video playback time, resulting in re-buffering events. Conventional, the video content server was based on protocols such as Real-Time Protocol (RTP) and Real-time Streaming Protocol (RTSP) for real-time delivery of video content to the video client side. However, with UDP, video playback may experience degradation, that is, some video frames might get distorted or dropped due to packet losses. While modern video streaming systems today use the HTTP protocol to stream video content. Transmission of existing models via Internet standards established protocol in the TCP transport layer, HTTP in the application layer. Nevertheless, major developments like the current HTTP / 2 Template Challenge, for example, attempt to predict instances of buffering loading times of videos and web pages (websites) to adapt to new transmission technologies. The general assumption is that a standard HTTP infrastructure (omnipresent interface to applications) is used which is deployed on top of TCP for the delivery of both MPD and video segments (which provide Reliable data transfer).

## 3.3 Transmission Control Protocol (TCP) & DASH

As we mentioned earlier, there has been a move to the ABR algorithm (video streaming technologies) as it has many advantages over conventional (non-adaptive) vide. Additionally, HTTP over TCP has gained popularity as a criterion transmission protocol due to various deployment advantagesas shown in3.2.


**Figure 3.2:** *Diagram of HTTP messages.*

TCP is used to transfer data in network communications such as the Internet. It is connection-oriented, a connection between the client and the server is established before data can be sent "Three-way handshake, retransmission, and error detection process; this is the reason why TCP is a reliable protocol". TCP used by DASH was intended for general purposes and not for video streaming in special. Indeed, the TCP congestion control protocol target to take the whole end-to-end available bandwidth and its behavior may have many differences with DASH streaming technology behavior. Nonetheless, these variants have various behaviors that may induce forked impacts on the QoE of DASH and the QoS of the access network. Moreover, the competition between DASH video clients for bandwidth is a special situation of competition between TCP flows; the only difference is the download of numerous video chunks, instead of a unique video file, within the same TCP session [70].

Video streaming service running over HTTP due to its ability to traverse through network address translations and firewall, reliability of video packet transfer, flexibility to respond to unstable network conditions, which uses TCP, devices of users receive all the information sent from the video server via IP networks for these matters the video player is incorporated into a web browser [62, 71, 72]

However, if the TCP control mechanisms discover losses on the IP network, the TCP transfer rate drops. Therefore, the quality of experience of users during a video streaming session under network congestion is mostly affected by rate switching or rebuffering event [73–75].

In recent years, it has been shown that TCP does not necessarily increase the performance of video streaming applications, particularly if the video player is able to adapt to large variations in throughput. A streaming protocol based on TCP, progressive downloading over HTTP, was used by the first generations of HTTP / TCP-based video streaming applications [76].

Nowadays, the majority of video content providers have chosen to use DASH technology, to adopt the video bitrate based on the capacity of the network on the user side; in order one goal, to improve the users' QoE during video streaming session. Nevertheless, if the quality of the video changes frequently, the user's attention may be negatively affected and other events may occur such as video playback freezing during the video session, all of this led to leaving the user the video session.

## 3.4 QoE Issues with DASH

### 3.4.1 General consideration

Traditional objective Video Quality Assessment (VQA) metrics were designed for quality estimation of video at various (destructive) compression levels. These metrics also take into account the impact of *packet loss* during the transmission process.

With Dynamic Adaptive Streaming over HTTP (DASH), the transmission mode has changed. There is no more packet loss, due to the use of TCP. Instead, the phenomena of content stravation, termed as **rebuffering**, occur and become one of the most critical issues. Indeed, with the TCP-based per chunk transmission scheme, it is possible that the playback buffer is empty before the reception of the next chunk.

Another phenomenon, always due to the chunk granularity, and also the possibility of multi-rates contents per chunk, is the video quality instability. Indeed, it is possible to fetch two adjaent chunks at totally different bitrate levels, and so with totally different video qualities.

There are quite a number of technical indicators, each of them provides some insight about the overall QoE. For instance, a high mean bitrate suggests a playback which is done with a good visual quality in general. However, the watching could be boring if the rebuffering phenomena are quite frequent. In a similar way, an overall good playback with a few moments of low quality would also provide a high mean bitrate. However, it is solely up to the user to prefer a fluid and stable yet mean quality playback, or a high quality playback with some rebuffering and/or quality switching.

DASH provides an architecture which allows end-users to make adaptation to the current networking and systems contraints, in order to get a best possible overall QoE. Adaptation mechanism is the central point of a DASH-based video streaming system.

QoE is a combination of many factors, including, on the physical conditions: chunk resolution, per chunk available video bitrates, available bandwidth, buffer size. Among these factors, the first two are provider dependant but is known to end-users through MPD. The solution is to hold a balance between these factors [77, 78].

As we already stated, QoE is a complex issue. In fact, the importance and effect of each QoE metric depends on the opinion of end-users [7, 79, 80]. In general, each QoE metric offers only a *partial* viewpoint on QoE. For this matter, to characterize the overall QoE, there is a need to consider multiples QoE metrics. In the case of QoE for DASH-based video streaming, we usually use these metrics:

- ***Average BitRate-Video Quality***: the average bitrate of the video chunks that have been displayed on the screen. Low video bitrate leads generally to bad quality (and so bad user experience). However, a high video bitrate, while providing better quality, claims also more bandwidth and so is more sensitive to bandwidth fluctuation.

- ***Rebuffering Events***: Several indicators can be driven to this phenomenon: frequency, maximal single duration, total duration. Long and/or frequent rebuffering are irritating and can lead to the abandon of a playback session [7, 81]. We consider here the total duration of the freezing events during the video session. Actually, a well designed ABR mechanism will not allow frequent rebuffering phenoma, but the adaption strategy does have impact on rebuffering duration.

- ***Video Quality Switching***: This concerns video quality changing when the adjacent chunks belongs to two different encoding versions. This objective video quality switch can be *perceptible* by users and impact on their experience.

### 3.4.2 About Rebuffering

The **Rebuffering** concept is defined: as the freezing of video playback during the video streaming session these events happen when the playout buffer gets empty. If the video bitrate is higher than the throughput of the video streaming application, the playout buffer will consume. Ultimately, insufficient data is available in the buffer and the playback of the video cannot continue. The playback is interrupted until the buffer contains a certain amount of video data [59]. The impact of rebuffering events on video streaming QoE has been widely studied in literature and is shown to depend on the number, the duration, and the position of the rebuffering (freezing playback) events in the playout, where the rebuffering events considered among the most important metrics that affect the user's QoE. These rebuffering events during the playback (video streaming session) lead to a bad user experience [82]. In [83], show that an increased duration of rebuffering decreases the quality of experience. They also find that one long rebuffering event is preferred to frequent short ones. In [84] they show interesting results that there is an impact of the position of freezing events to video streaming users' responses, where the video user reacts differently with regard to the rebuffering location where the problem happens. This indicates that the rebuffering and its location have a combined impact on human perception. While the users who experienced more rebuffering in the video tends to watch the video for shorter durations [61] and are expected to be not satisfied in the case of four or more rebuffering events for videos [85].

Rebuffering is a complex issue. As a matter of fact, it is not easy to choose the ideal bitrate for the next video chunk, since it depends basically on the network bandwidth availability forecast. An aggressive (higher bitrate) choice leads more probably to depletion of the buffer but offers better overall bitrate so video quality; a conservative (lower bitrate) choice prevents buffer depletion but offers lower overall bitrate (bad so worse video quality). There is clearly a trade-off between video quality and rebuffering risk.

Recent studies on video QoE (cf. for instance, [79]) converge to the conclusion that the rebufferings in DASH should be avoided in order to enhance the QoE. However, users' QoE can vary depending on the manner of rebuffering : for instance how long or how often rebufferings arise during video playback.

Concerning a single rebuffering event, a duration of *up to* approximately 360 ms has been showed to be acceptable by the end-users [81]. S. van Kester et al. provided a detailed study [81] on the impact of a single rebuffering as well as multiple rebufferings on the perceived quality of the users. A subjective test was performed to study how these rebuffering influence the quality perceived by users through MOS records. This study reveals that an acceptable rebuffering time (MOS>3.5) is 360 ms. This study also showed that the perceived quality depends not only on the duration of the rebuffering but also on the number of rebuffering. It depends also on the pattern of rebuffering: actually, we may have a single long rebuffering or multiple short rebuffering periods.

A large scale study has been presented in [7]. This study is based data collected, by using the *YouSlow* tool, from more than 400,000 YouTube views situated at over than 100 countries. The authors found that the viewers stay 5 minutes and 1 second on average per video session, including rebuffering and start-up latency. More than 40% of viewers closed YouTube videos in the middle of the playback, due to unexpected playback events such as rebuffering and bitrate changes. They compared in particular rebuffering against two other factors which are *start-up delay* and *bitrate switching*:

- *Rebuffering vs start-up delay*: In terms of abandonment, the rebuffering events cause video users abandonment rates six times compared with start-up delay during the video session.
- *Rebuffering events vs switching bitrate*: This study is focused on the comparison between two groups: in the first group, there are 9,577 video sessions where the viewers experienced a single rebuffering event without any bitrate changes and any ads. The second group is constituted with 4,991 video sessions where the viewers experienced a single bitrate change with no rebufferings and no ads. The first group (rebuffering) has an abandonment rate which is three times higher than that of the 2nd group (bitrate switching).

### 3.4.3 About Bitrate Switching

Quality instability due to bitrate switching is particularly present in wireless mobile networks. In [86], a subjective video QoE study has been conducted on multiple mobile platforms and encompassed a wide variety of distortions, including dynamically-varying distortions as well as uniform compression and wireless packet. They observed that time-varying quality has a definite impact on video end-user subjective judgement of quality. Their study revealed also that humans appear to be far more forgiving of lost segments than they are of switching quality. Also, humans seemingly prefer longer rebuffering event over more frequently but shorter rebuffering events.

In [87], they proposed QoE aware quality-level switching algorithm, in order to adaptive video streaming with fluctuating network , where this systme switching adaptively based on the network-throughput estimations. Their proposal works on the controller of DASH system. This system has two features: insertion of average quality level and determination of quality-level switching. Generally this method have shown effectiveness to enhance QoE through fluctuating network.

In [88], they suggest analytical framework in wireless networks to compute starvation probability of the buffer, continuous playback time and average video quality, given the switching bitrate logics. where this framework can be used to predict the QoE metrics of dynamic adaptive streaming with a set of features: a) buffer-aware bit-rate switching b) receiver-side stream control. In this work they proposed two frame work analytical the first analytical framework to predict the QoE of adaptive streaming is based only on channel variation (wireless channel is modelled as a continuous time Markov process), while the second framework is the bit-rate switching algorithms is based on both channel variation and buffer length (playout buffer is modelled as a fluid queue with Markov modulated fluid arrival). Where their study in this work shows good practical value in guiding the design of the bitrate switching algorithm.

In [89], they proposed control algorithm called BOLA that uses Lyapunov optimization to minimize rebuffering and maximize video quality. Where they show how BOLA can be adapted to avoid switching bitrate events during video playback; despite the switching bitrate are less annoying than rebuffering events, but in their work they see the switches bitrate events when it occur too frequently, it will effected negatively in user involvement.

In [90] they propose an in-network resource management framework, AVIS, that schedules HTTP-based adaptive video flows on cellular networks, is to control the frequency bit rate switching per user via scheduling, AVIS is effective in allocating the resources of a base station across multiple adaptive video flows and effectively balances between three important goals: 1) fair resource allocation 2) Stability of a user's bit-rate (average bit rate switches between different users) and 3) enables a balance of optimal bit rate for individual users. Therefore a good QoE are obtained.

## 3.5 ABR Algorithms for DASH

### 3.5.1 ABR Algorithms

As stated before, Adaptive BitRate (ABR) algorithm is a key element in DASH. There are many factors to be taken into account when making adaptation on the one hand, and, on the other hand, various ways to take these factors individually into account, as well as to make balance among them. in addition, the target QoE may also have various factors. Thus, there are quite a number of ABR algorithms which have been developed and this topic still remains an open issue.

In this section, we will review related work on ABR algorithms. ABR algorithms are usually categorized into three classes:

i. **Buffer-based Adaptation**: these adaptation algorithms mainly utilize only the client's current buffer level to select a bitrate quality of next video chunks for playback. For example, a low buffer level means the adaptation algorithm selects the lowest quality for download and enters a conservative mode of behavior. Similarly, higher buffer levels means an increasingly aggressive quality selection process as the buffer fills up. (e.g., BOLA and BBA by NETFLIX) which avoids rebuffering events by constantly monitoring the buffer levels before selecting a quality for download [60, 91].

ii. **Rate-based Adaptation**: these adaptation algorithms simply selects a quality for next video chunk based on the measured download rate use the throughput achieved in recent prior downloads for decision-making (e.g., throughput rule in dash.js and Fair, Efficient, and Stable adapTIVE algorithm (FESTIVE)) [92, 93].

iii. **Mixture**: a mix of the two previous categories (algorithms combine both types of information) [47].

### 3.5.2 The Buffer Based Adaptation (BBA) Algorithm

Huang et al. proposed the Buffer Based Adaptation (BBA) [60, 94] method. Briefly, the algorithm, which was part of a wide-scale Netflix experiment, defines a class of functions that map current buffer occupancy to a quality bitrate (denoted rate map) which aims to: a) Avoid unnecessary rebuffering and b) Maximize the overall video quality. They used the buffer occupancy as a control signal to select the level (bitrate) of the next video chunk instead of the estimated bandwidth. They calculate dynamically a couple of <maximum, minimum> bitrate levels. When possible, the actual video rate keeps growing to the maximum, until there is not enough (with respect to a threshold value) room in buffer, then the video bitrate drops to the minimum level.

To guarantee that the algorithm never unnecessarily buffers is to simply fetch rate Rmin when the buffer approaches empty (When the buffer indicator falls in critical zone), permitting the buffer to grow as long as C(t) > Rmin. Inspired by this observation, Huang et al. design their algorithms BBA as follows. First step , they focus on a buffer-based design: BBA select the video rate directly as a function of the current buffer level. They call this design the buffer-based approach. Their suggestion might be thought of as an “likeness” of Figure 3.4 [60]: namely, they start by using only the playback buffer, and then “regulate” this algorithm using capacity estimation if necessary.

An ABR algorithm is buffer-based if it selects the level of next video chunk as a function of the current buffer occupancy, B(t). The model space for this category of algorithms is explained as a buffer-rate plane where the buffer-axis is buffer occupancy and the rate-axis is video rate. The region between [0, Bmax] on the buffer-axis and [Rmin, Rmax] on the rate-axis defines the feasible region. Any curve f(B) on the craft within the funciolnal region defines a rate map, a function that produces a video rate between Rmin and Rmax given the current buffer occupancy see Figure 3.3 [60].

To avoid rebuffering, it is important to get content when the buffer occupancy approaches empty (when the buffer indicator falls in critical zone). The best way to get content is to use the minimal rate. Then, when the buffer is being refilled, it is possible to get a higher download rate (see Figure 3.3 [60]).

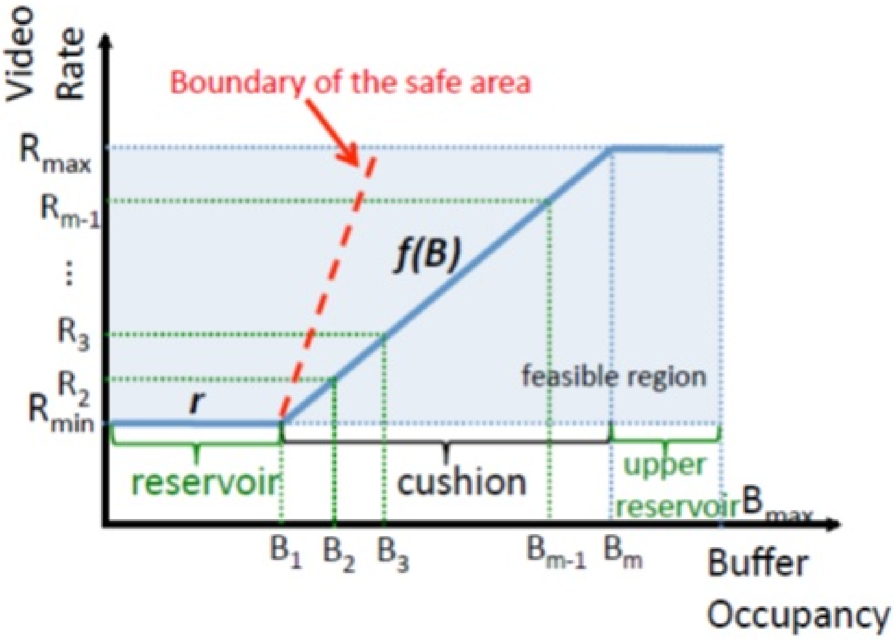



**Figure 3.3:** *The rate map used in the buffer based algorithm.*

Inspired by this observation, Huang et al. design their algorithms BBA which is mainly uses buffer-indicator to estimate the level of the next chunk to be fetched. Nevertheless, they also uses (networking) capacity estimation as a regulator parameter. This leads to the following closed-loop adaptation schema (see Fig. Figure 3.4 [60]).

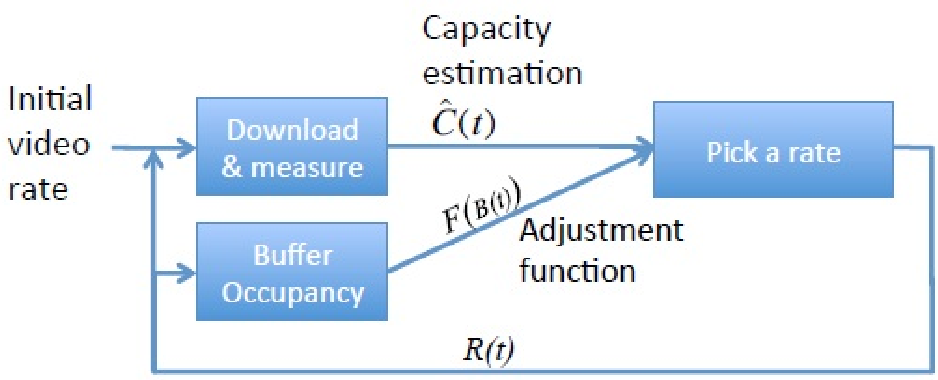



**Figure 3.4:** *Current practice adjusts the estimation based on the buffer occupancy.*

The BBA algorithm permits to reduce the rebuffer rate by 10˘20% compared to the ABR algorithm used by Netflix at that time, while delivering a similar average video rate, and a higher video rate in steady state.

### 3.5.3 The FESTIVE algorithm

Within the context of multiple bitrate-adaptive players share a bottleneck link, and for achieving three key metrics: efficiency, fairness, and stability, Jiang et al. proposed a general bitrate adaptation algorithm framework called (FESTIVE) [92].

This adaptation algorithm aims to improve fairness, stability and efficiency of the DASH player by predicting throughput to be the harmonic mean of the experienced throughput for the past downloaded chunks, as well as a stability score as a function of the bitrate switches in the past chunks. As the prediction does not take into account the buffer occupancy, FESTIVE may have rebuffering which affects the QoE performance.

Through this framework, they identify the underlying causes of several unwanted interactions that have severe negative impact on video bitrate adaptation over HTTP. From that, they develop a generic framework to achieve trade-offs between stability, fairness and efficiency with a solid video adaptation shema. Figure 3.5 [95] gives an overview of FESTIVE.

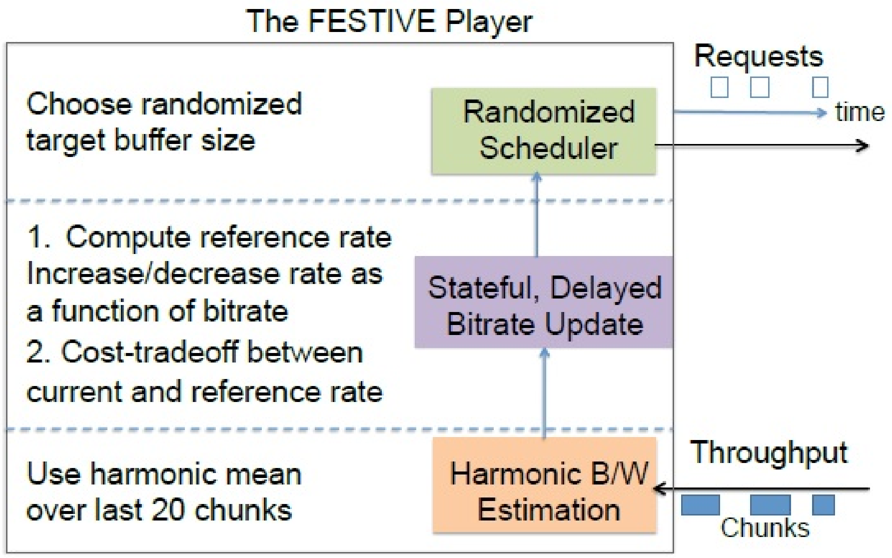



**Figure 3.5:** *Overview of the FESTIVE adaptive video player.*

FESTIVE has Three Key Components:

i. **The harmonic bandwidth estimator** computes the harmonic mean of the last throughput estimates. In the premier stage (before having a enough number of samples), FESTIVE does not use any rate switches because its bandwidth estimate will be unreliable.

ii. **The stateful and delayed bitrate** update module receives throughput estimates from the bandwidth estimator and computes a reference bitrate. As a specific implementation, they use a gradual switching strategy; i.e., switchs are made only to the next higher/lower level. This ensures that the bitrates eventually converge to a fair allocation despite the biased bitrate-to-bandwidth relationship.

iii. **The randomized scheduler** The next chunk is downloaded immediately if its playback buffer is less than the target buffer size. Otherwise, the next chunk is scheduled with a random delay by selecting a randomized target buffer size. This ensures there are no start time biases.

### 3.5.4 The Open Source Media Framework (OSMF)

The Open Source Media Framework (OSMF) [96] is an HTTP video streaming platform developed by Adobe Systems, for building solid, feature-rich video players. It is designed as a flexible architecture permitting developers to easily adjust their player for the browser. OSMF combines plug-ins for video content delivery along with standard player features such as play/pause, download progress, buffering, and bitrate switching. Figure 3.6 [97] shows basic structure of an OSMF player. The OSMF decreases the complexity of player development, permitting the developer more time to focus on the overall video user experience.

The bitrate adaptation algorithm in OSMF [78, 98, 99], mainly uses these inputs to select the quality of the next video chunk:

i. video chunk duration;

ii. the time needed to download the last video chunk;

iii. the current bitrate;

iv. the potential bitrate candidate for the next video chunk;

v. lowest quality level;

vi. highest quality level; and

vii. available bitrate.

OSMF's adaptation algorithm uses an indicator (a kind of quality switching ratio) $\alpha$ which is computed with the last last downloaded video chunk. More precisely, $\alpha$ is computed by dividing

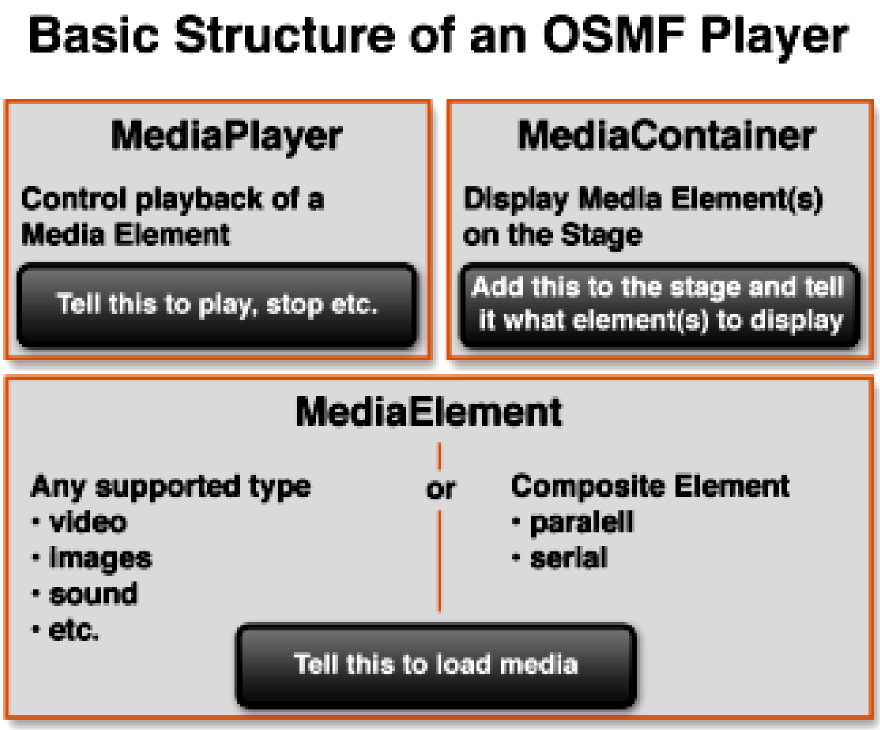



**Figure 3.6:** *Basic structure of an OSMF player.*

the time duration of the last downloaded video chunk by the actual duration of its download. Thus $\alpha < 1$ means that the download time is larger than the playback time, if this situation continues, a rebuffering would happen. On the contrary, $\alpha > 1$ suggests that the networking conditions allow a higher volume (higher level) to be downloaded.

The OSMF works as follows

(a) If $\alpha < 1$, then a lower level (lower bitrate) is selected for the next chunk. This process continue, if necessary, until the the lowest bitrate is selected.

(b) For $\alpha > 1$, a higher bitrate is selected as a potential candidate until the highest one.

With the bitrate adaptation algorithm in OSMF, the performance quality of video-rate switching is very frequently. This is due to the fact that the quality adaptation does not take into account the available buffer occupancy.

## 3.6 Video-quality metrics as adaptation factor

Video-quality metrics are valuable factors for video adaptation. We present hereafter works using respectively SSIM, PSNR and VMAF metrics

### 3.6.1 With SSIM

In [50], the authors used SSIM as objective QoE metrics in their framework for video streaming systems based on the H.264/SVC codec, the scalable extension of H.264/AVC. They took into account the nature of the video (e.g., interview, soccer match, movie). Their results showed that they can use SSIM to determine the QoE behaviour of different contents. They also showed that video sequences with lower resolution perform better than video sequences with a lower frame-rate.

In [48], the authors also incorporated the SSIM into their *Utility Function* as a quality metric. The rationale can be summarized as follows:

- The relationship between bitrate and perceptual quality is **not linear**; the bitrate increases, the gain in video quality is gradually saturated.
- The equal division of network bandwidth for video streams of different resolutions results in unfair video quality levels as perceived by end-users.

In [49], the authors used the SSIM to measure the quality of video transmission through their system (Compressive Distortion Minimizing Rate Control, C-DMRC). The latter uses a distributed cross-layer control algorithm that aims to maximize the received video quality over a multi-hop wireless network with lossy links.

In [100], they proposed a SSIM-based adaptation algorithm for DASH with SVC in mobile networks to improve overall QoE through delivering high average quality with low switching frequency in fluctuating mobile network conditions. This paper considered the difficulty that faces the ABR video streaming algorithms for appropriate evaluation of the video quality. The majority of the ABR streaming algorithm takes bitrate as the input to evaluate the quality of the video. However, bitrate is not strongly correlated with the visual quality. In this work, to make the adaptation process more reliable, the authors proposed to exploit SSIM of individual video chunks as an adaptation algorithm input.

In [36], the authors introduced a classification scheme for FR and RR media-layer objective video quality assessment. They shown that the natural visual statistics based MultiScale-Structural SIMilarity index (MS-SSIM), the natural visual feature based Video Quality Metric (VQM), and the perceptual spatio-temporal frequency-domain based MOtion-based Video Integrity Evaluation (MOVIE) index give the best performance for the LIVE Video Quality Database.

In [101], the authors considered three metrics influencing the QoE for DASH: (i) instantaneous visual quality using SSIM; (ii) quality fluctuation and (iii) rebuffering events.

### 3.6.2 With PSNR

In [51], the authors rely on the video quality QoE metrics like PSNR which allows to carry large measurement studies and to derive simple relationships applicable in QoE control. In special, to take a closer look at the impact of (i) the video resolution; (ii) the scaling method; (iii) the impact network conditions, respectively packet loss; and (iv) the video content types on the QoE by means of PSNR full-reference metrics.

In [102], in order to evaluate the level of satisfaction for the users in the connected emotion-aware intelligent system network, the authors suggested in their work three metrics as the QoE measurement: PSNR, buffering ratio that captures the stalling periods of video playing observed by users and modified the MOS which is determined by the average bitrate and packet loss.

### 3.6.3 With VMAF

In [103], the authors proposed a learning-based approach for QoE prediction model taking into account the memory features, video quality models, and rebuffering-aware.

In [104], a strong correlation between subjective MOS and the computed objective VMAF score with a high correlation has been shown.

In [105], the authors suggested a QoE prediction model based on three QoE-aware inputs: an objective measure of perceptual video quality; a QoE memory of prior events; and rebuffering-aware information. They selected various VQA models, including VMAF for quantitative analysis of the outcomes of the experiment.

## 3.7 Conclusion

This chapter introduces the background information and related work of this thesis. We present and analyze the fundamental concept of Dynamic Adaptive Streaming over HTTP (DASH) and its challenges. We then deal with the DASH-related adaptive bitrate (ABR) streaming algorithm. As a conclusion of this presentation of the state of the art related to various topics concerned by the DASH based video streaming, we do think that there is a need to investigating some new adaptation algorithms which make a combination of Qos-related factors and video-quality-related factors. This will be our working direction. Our proposal and results will be presented through the two next chapters.

CHAPTER 4

# A Generic framework for Video-Quality metric-Based Adaptation Algorithm (VQBA)

> *Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time.*
> THOMAS A. EDISON.

## 4.1 Introduction

In this chapter, we present our proposal of a generic framework, named *Video-Quality Metric Based-Adaptation Algorithm* (VQBA for short) for the DASH-based video mechanism. We consider it as a *framework* since it is designed for various objective video quality metrics, among them we have tested SSIM, PSNR and VMAF.

This generic framework is based on a joint consideration of a) the objective Video-Quality-Metric (VQM) such as (SSIM, PSNR, and VMAF) and b) the physical resources such as buffer occupancy and estimation bandwidth, to minimize both rebuffering and visual quality instability, as well as to maximize the overall video quality given by the objective video-quality metric.

The main idea of this framework can be stated as follows.

- We give importance to objective video quality metric by making use of it in our adaptation scheme (which is rarely used in the existing work) for the simple reason that, after all, users are aware of video quality.

- Based on this objective indicator, we develop our adaptation mechanism in a way that a possible upgrade to a higher (bitrate) level, as predicted by networking context forecast, takes actually place only when it carries a noticeable upgrade in video quality also. In this way, we try to maximize the use of bandwidth for effective quality contribution and avoid rebuffering.

Most of the proposed work is mainly driven by networking (bandwidth) and/or system (buffer occupation) conditions. Few of the existing ABR algorithms really take into account the video quality as a main parameter for adaptation.

From the view point of image (and video) processing, the quality improvement is not directly proportional to the video bitrate increase, but follows a more complex relation. Figure 4.1 helps to better illustrate this point. It gives the contrast between two chunks of a video sequence. It can be observed that for chunck number 27, all the bitrate levels (x-axis) offer nearly the same (*SSIM*) value [39]; whereas for chunck number 140, the lifting in SSIM value for higher bitrate levels are rather noticeable.

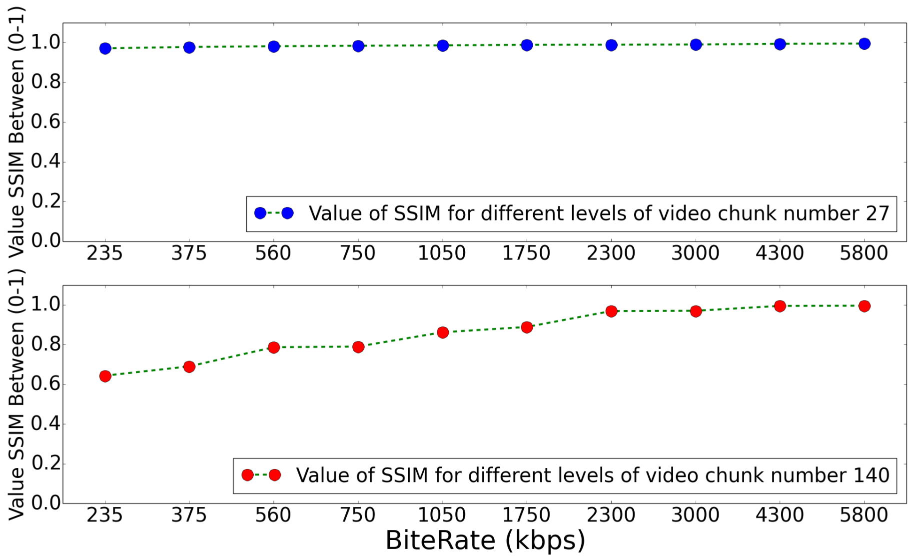



**Figure 4.1:** *SSIM for different resolution for the chunks number 27 and 140.*

We believe that the visual quality, that we can assess through some objective metric, is a key parameter for adaptation. This has been the start point of our working direction, which led to an algorithm [10] using the SSIM as a main adaptation metric, which is the first version of VQBA.

## 4.2 Rational and Illustration

The rationale of our algorithm (cf. section 4.3 for more details) consists in using the video quality metric as a criterion for an actual bitrate increase when the latter is allowed by networking conditions. Our algorithm aims to prevent bitrate increases which do not really bring a *significant* visual improvement. Indeed, each bitrate increase comes with a higher rebuffering risk, we use SSIM to know if the increase is really worth the risk.

In order to illustrate the basic idea of our proposal, we present these two examples (cf. Figures 4.2, 4.3). We consider the SSIM metric. Our algorithm keeps trace of SSIM of the video chunks being displayed (i.e. the SSIM of video *actually viewed* by end-user). In particular, we compute the SSIM variation between adjacent video chunks (the blue curve), from this, we compute also a SSIM threshold curve (the red curve).

In addition to the system/networking conditions, which are the usual criteria for bitrate adaptation, in particular for an increase of the bitrate for the next video chunk, our proposal checks also the impact of a bitrate increase: we try to figure out if this increase worth really to be done, which is if it really gives a *real gain* in visual quality to improve overall QoE.

In the first example (Fig. 4.2), the SSIM variation is smaller than the SSIM threshold; the bitrate will be kept at the same level for the next video chunk. In the second example (Fig. 4.3), the SSIM variation is bigger than the SSIM threshold; an bitrate increase can be taken place for the next video chunk.

From an operational point of view, our approach is compatible with the DASH scheme. Actually, the objective video quality metrics (such as SSIM, PSNR and VMAF) can be pre-computed for each level of each video chunks. These metrics can also be made available to end-user (the adaptation maker) through the MPD.

## 4.3 A Video-Quality Metric Based-Adaptation Algorithm (VQBA)

In this section, we provide a detailed presentation of our framework.

### 4.3.1 Conditions and Notations

The video stream is encoded at $R$ bitrate levels, denoted as $\mathcal{R} = \{r_j\}_{j=1...R}$ with $r_1 < r_2 < \cdots < r_R$, and it is divided into $K$ chunks (video segments) of equal duration (denoted as $T$). We have thus a total of $K \times R$ elementary video contents which are qualified for being fetched then displayed. Let $v(k,r)$ denotes such a content of $k$-th chunk at $r$-th bitrate level.

Each elementary video content $v(k,r)$ is associated with its corresponding VQM, noted as

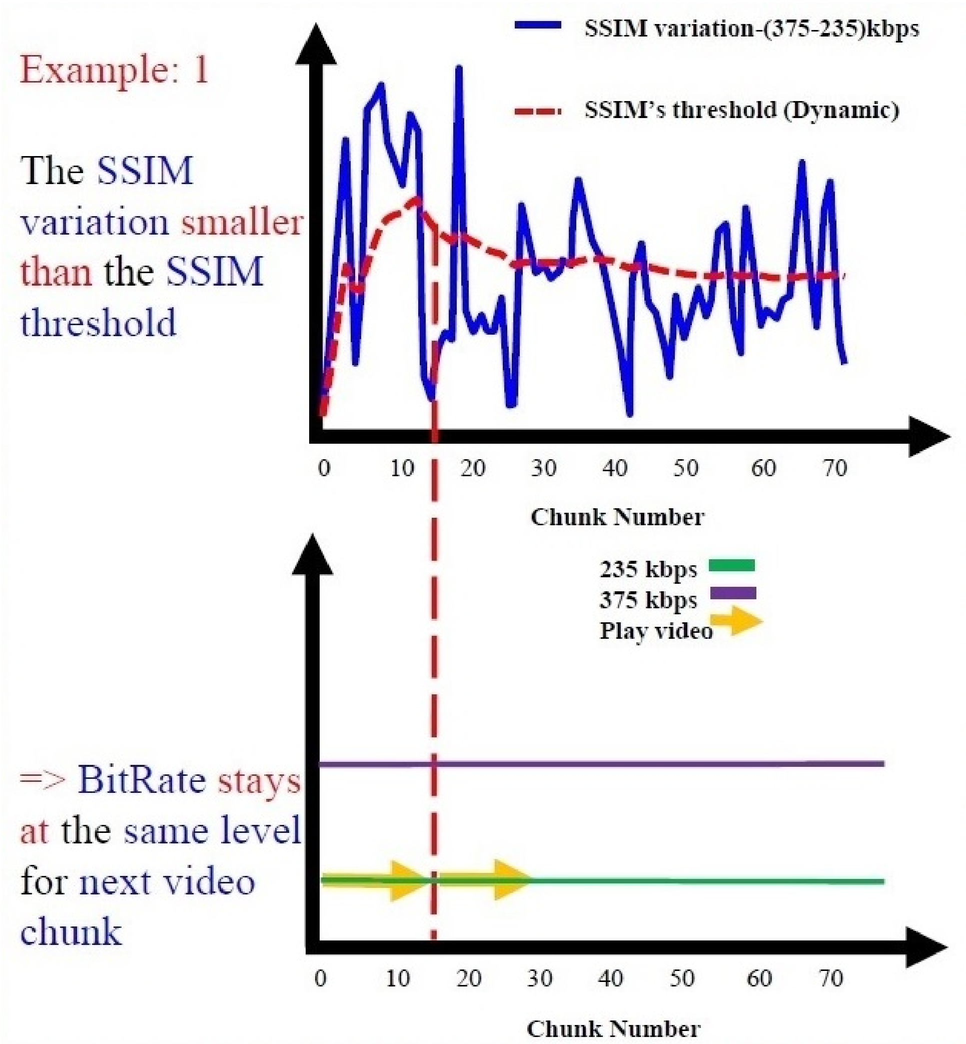



**Figure 4.2:** *1st example: SSIM indicator to determine the bitrate level for the next video chunk.*

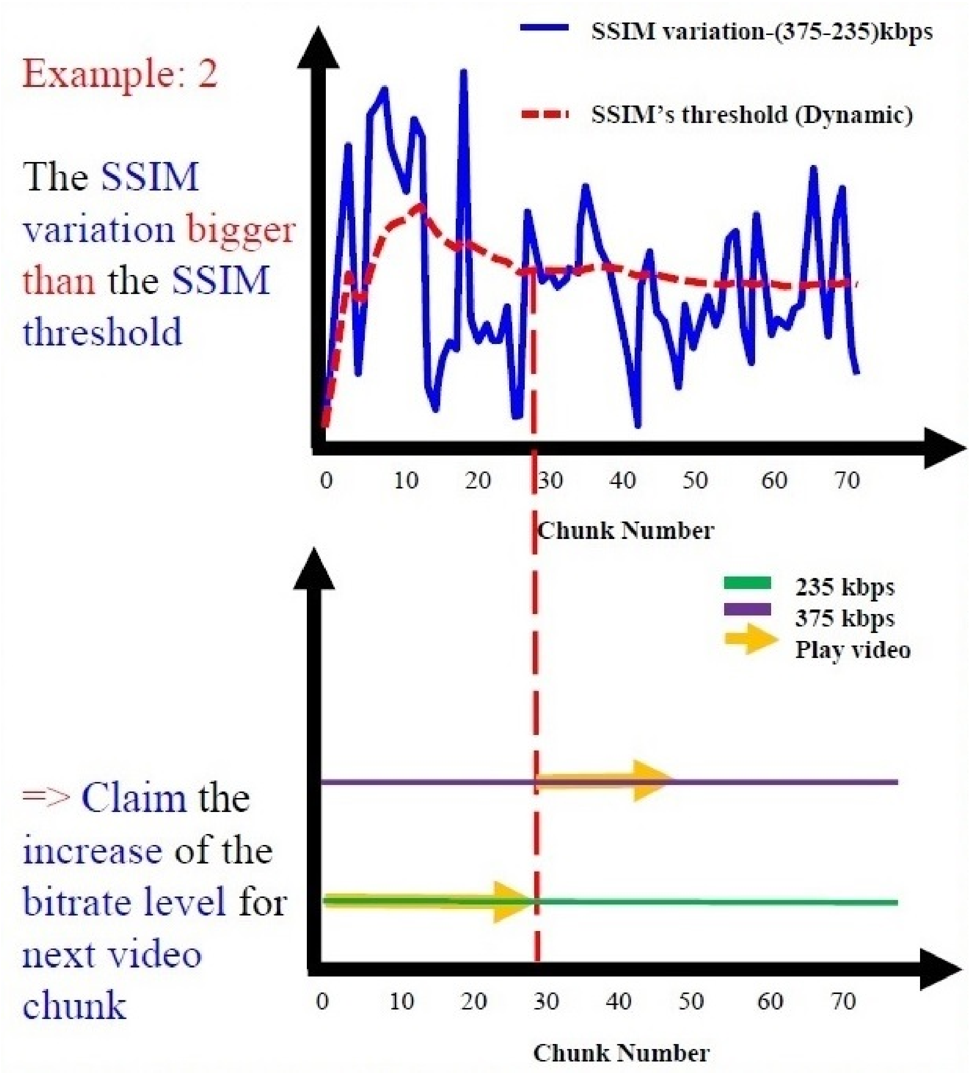



**Figure 4.3:** *2nd example: SSIM indicator to determine the bitrate level for the next video chunk.*

$\mathbf{Q}(k,r)$. This metric depends of course of the target quality assessment (SSIM, PSNR, VMAF in our study). It is pre-computed and stored in the server. During a DASH session, the client can get it through the MPD mechanism.

### 4.3.2 Rationale of the Adaptation Logic

Recall that the main point of this framework is a joint consideration of the networking level control (buffer-based and rate-based) and the target video quality metric. Of course, we still have to take into account the networking and buffer conditions are Indeed, these ones are hard constraints that we have to respect. The key point here consists in using the **video quality metrics indicator** as an additional criterion to determine the level (bitrate and so video quality) of the next chunk to be fetched.

Hereafter is the guideline behind the design of our algorithm, for the choice of the bitrate of the next chunk (assumed to be the $l$-th one).

- Our primary concern is to prevent rebuffering. Thus, we aims to insure a minimum amount of available video content by establishing a *critical zone*. When the buffer occupancy is below a critical value ($L_c$, cf. § 4.3.3), we adopt a TCP-like approach by imposing the lowest bitrate. This is in particular true when a rebuffering actually happens, as well as at the initial phase. Indeed, it is the most secure way to (re)fill the buffer.

- When the buffer occupancy is out of the critical zone, we then first estimate the available bandwidth (denoted by $EBW(l)$).
  - If $EBW(l) < r_1$, the networking condition is *critical*, we have to take $r_1$ which is the lowest available bitrate.
  - Otherwise, possibility of a bitrate switching exists among $\{r_2, \ldots, r_R\}$.
    * If the network conditions go worse, we decide to keep the current bitrate, since the degradation could be temporary and we still have some margin (buffer occupancy is not critical). In this way, we try to avoid unnecessary quality switching.
    * If the network conditions go better, there is possibility of bitrate increase. At this time, we make use of our criterion on video quality metric. We actually increase the bitrate to the best allowed level, say $u$, only if this one ($Q(l,u)$) *does* bring a *significant* improvement (versus some *threshold*) of quality. Otherwise, we keep the current bitrate level (so no bitrate switching).

Thus, our algorithm clearly gives priority to prevent rebuffering by minimizing *visually non-effective* bitrate increases. It tries also to minimize the visual effect oscillation by minimizing the bitrate switching in case of network condition degradation.

### 4.3.3 Presentation of the VQBA algorithm

Hereafter, we give the formal description of the Video-Quality Metrics Based-Adaptation algorithm framework (cf. pseudo-code in Figure 4.4) and present the parameters and metrics used by it.

- $l$ is the number of the next chunk to be fetched, $f$ is the bitrate level at which the next chunk will be fetched.
- $b$ is the current buffer occupancy and $L_c$ is the *critical value*.
- $\alpha(l)$ is the *current* VQM threshold, $d_{l-1}$ is the bitrate level at the past ($(l-1)$-th) chunk.
- The available chunks and their levels ($\mathcal{R}$), as well as the associated Video Quaity Metric ($Q$) are provided by the server and can be got by the client through MPD at initialization.
- $EBW(l)$ is the estimated bandwidth which would be available during the fetching of the next ($l$-th) chunk.

**Input:** $\mathcal{R}$, $Q_{VQM}$, $b$, $L_c$, $l$, $\alpha_{VQM}(l)$, $EBW(l)$, $d_{l-1}$,
**Output:** $f$

**if** $(b \leq L_c)$ **OR** $(r_1 \geq EBW(l))$ **then**
  $f = 1$
**else**
  $f = \max\{j \,/\, r_j \in \mathcal{R}, r_j < EBW(l)\}$.
  **if** $Q(l,f) - Q(l-1, d_{l-1}) \leq \alpha(l)$ **then**
    $f = d_{l-1}$ (*No significant improved, keep the current*)
  **end if**
**end if**
**return** $f$ (*the level at which the next chunk will be fetched*)

**Figure 4.4:** *The Video-Quality metrics Based-Adaotation algorithm framework (VQBA).*

The VQBA algorithm is run each time a fetch order can be issued, i.e. either when a chunk is totally downloaded or, upon the first availability of a free space (this is the case when a chunk is totally whereas the buffer was previously full).

The parameters ($\alpha(l)$ and $EBW(l)$) are assumed to be estimated through parallel processes. Indeed, quite a number of options can be made about these two metrics.

### 4.3.4 Discussions

This framework is rather generic, since it offers room for various declinations:

- First of the declinations is the video metric. This point, which is main focus of this thesis, will be deeply investigated in the rest of this manuscript. In § 4.4, there is a general presentation of these tests and experimental conditions.

- The success of adaptation depends also on the accurary of the bandwidth forecast. This challenging topic is still an open research problem. We give subsequently (cf. § 4.3.4.1) a short description of the default algorithm we used

- Our adaptation algorithm relies also on the *delta* in terms of gain about video quality. Again, there are quite a lot possibilities to determine the *threshold* values. In § 4.3.4.2, we give discussions on this issue.

#### 4.3.4.1 Available Bandwidth Forecast

The forecast of available bandwidth, noted here as $EBW$ (estimated bandwidth), is a challenging issue. In the current version, we adopt a very classical approach. The $EBW$ is estimated with a simple smoothing function which computes the average bitrate at which the past chunks were actually downloaded. $EBW(l)$, the estimated bandwidth for the coming $l$-th chunk, is estimated as follows.

$$EBW(l) = \frac{\sum_{i=1}^{l-1} UBW(i)}{l-1}, \tag{4.1}$$

where $UBW(i)$ is the actual bitrate at which the $i$-th chunk has been downloaded (UBW stands for used bandwidth). It is obtained by.

$$UBW(i) = \frac{V_i}{t_i - s_i}, \tag{4.2}$$

where $t_i$ (resp. $s_i$) is the instant at which the fetch of the $i$-th chunk ended (resp. started), and $V_i$ is the volume of the $i$-th chunk.

#### 4.3.4.2 Quality Improvement Threshold ($\alpha(l)$)

The setting and computation of the quality improvement threshold ($\alpha(l)$) depends on the nature of the metric. Indeed, different metrics have different value spaces. SSIM (resp. VMAF) provides a normalized ((0..1) (resp.(0..100)) objective visual quality metric, whereas PSNR is a metric in dB. Besides, different strategies of threshold setting have obviously impact on the behaviour and performance of the algorithm.

Without loss of generality, we present here a very simple threshold setting scheme, which consists in getting the mean value of the quality change, in order to smoothing the quality change. A more detailed study focusing the threshold is given in § .

- For each chunk (say $l$ for $l > 1$), we compute the the variation $\Delta(l)$ in terms of VQM related to the previous chunk $(l-1)$, i.e.

$$\Delta(l) = Q(l, d_l) - (l-1, d_{l-1})$$

  where $d_l$ and $d_{l-1}$ are the respective bitrate levels of chunk $l$ and chunk $l-1$.

- The cumulative variation till $l$ is then computed with:

$$S(l) = \sum_{k=2}^{l} \Delta(k).$$

- The adaptive threshold $\alpha(l)$ is then computed with:

$$\alpha(l) = \frac{S(l)}{l-1} = \frac{\sum_{k=2}^{l} \Delta(k)}{l-1}. \tag{4.3}$$

Notice that, in practice, the video playback begins with the pre-fetch of several chunks, so $l > 1$ is assured.

## 4.4 Experimental analysis of VQBA Framework

### 4.4.1 Presentation

We have carried extensive experimental studies of our algorithms in order to assess the performance of our algorithm and its behaviour with three of the most used visual quality metrics, namely SSIM, PSNR and VMAF, with real network traces.

In order to get better understanding about the behavior and effectiveness of the VQMs,

- we also carried runs, under the same networking conditions, of three of the non-quality-aware ABR, namely BBA [60, 94], FESTIVE [92] and OSMF [96].

- We have tested three kinds of video : *Animation, Documentary* and *Sport* (cf § 4.4.3)

All these experimental results allow us to carry various comparative studies : between VQMs, between VQBA and traditionnl ABR, for various types of video sequence.

Hereafter, we will provide description of components of this evaluation framework.

### 4.4.2 Realistic Networking conditions

In order to obtain a realistic networking case, we have used two sets of real traffic traces.

The first set (24 traces [10]) has been captured by ourselves from the 4G mobile network of a big network provider, at different locations and timeslots in Paris city. For this, we used the *iPerf* tool with the command `iperf3 -c iperf.scottlinux.com`. Details are given in § 7.1. This set of traffic will be referred as *USPN-set*.

The second set is constituted of traces available at [106]. These traces were collected from 4G networks within the city of Ghent, Belgium, in January and February 2016 [107], over various routes while downloading a large file over HTTP through various modes of mobility: foot, bicycle, bus, tram, train and car. This set of traffic will be referred as *BE-set*.

By default, and except explicitly mentioned otherwise, we use the USPN-set.

### 4.4.3 Test sequences

For the evaluation test, among the test video sequences used in DASH, we select the following three :

- **Animation** (*Big Buck Bunny*) a 9-min, 56s video [108],
- **Documentary** (*Of Forests and Men*) is created a 7-min, 33s short film on forests filled with aerial images [109]. and
- **Sport** (*The World's Best Bouldering in Rocklands, South Africa*) a 13-min, 18s video [110].

Figures 4.5, 4.6 and 4.7 provide an idea of each of these three sequences: a) Big Buck Bunny_**animation** b) Of Forests and Men_**documentary**and c) The World's Best Bouldering in Rocklands, South Africa_**sport** used in our experiments.

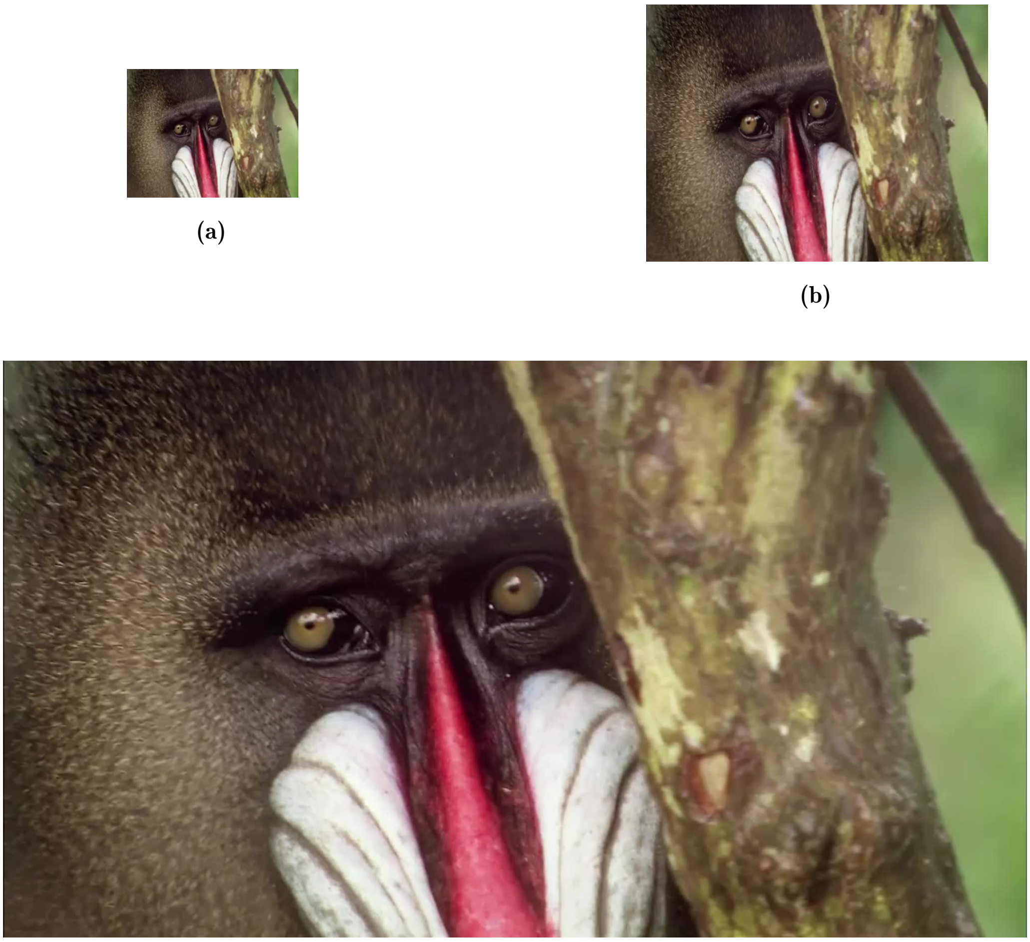
These videos are encoded at 24 frames/s (FPS) with the FFMPEG package by using its H.264 codec. We generate a video at 10 ($R = 10$) different resolutions (in kbps) (the same as those used by Netflix [111, 112]):

$$\mathcal{R} = \{235, 375, 560, 750, 1050, 1750, 2350, 3000, 4300, 5800\}$$

Videos are then split into chunks of 4 ($T = 4$) seconds by using MP4Box-GPAC framework [113].

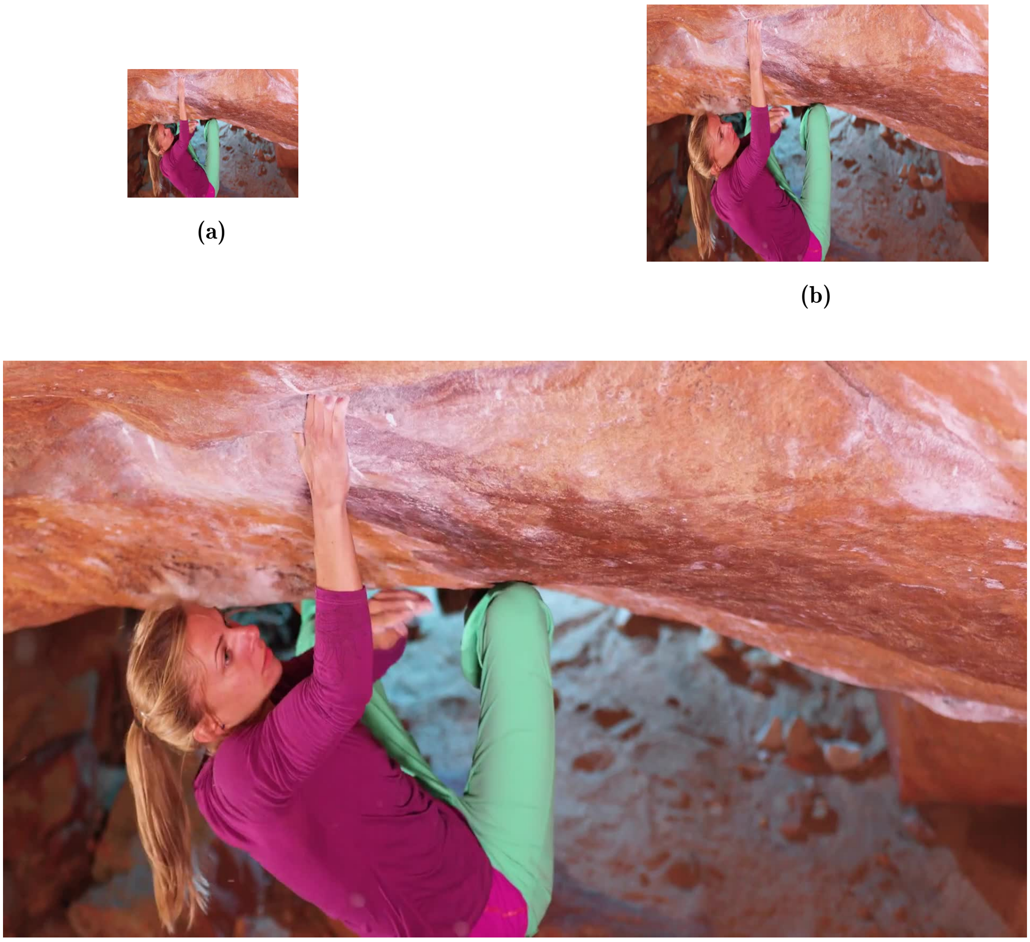
Each elementary video content unit $v(k, r)$, i.e. the content of $k$th chunk coded at $r$th bitrate level, has a corresponding metric value given by th VQM Matrix ($Q(k, r)$). This value is computed as the mean value of the metric for each image. For instance, if the target metric is SSIM, then the VQM metric value $Q(k, r)$ is computed as the mean value of the SSIM of each single image

Figure 4.5: *Quality images for video chunks of the movie (Big Buck Bunny) {(a) Low Quality=235 kbps, (b) Medium Quality =1050 kbps and (c) Reference}.*



(a) (b) (c)

**Figure 4.6:** *Quality images for video chunks of the movie (Of Forests and Men) {(a) Low Quality=235 kbps, (b) Medium Quality =1050 kbps and (c) Reference}.*



(a) (b) (c)

**Figure 4.7:** *Quality images for video chunks of the movie (The World's Best Bouldering in Rocklands, South Africa) {(a) Low Quality=235 kbps, (b) Medium Quality =1050 kbps and (c) Reference}.*

in $v(k,r)$. In this way, the VQM matrices can be pre-computed and stored in the server. We racall that for DASH operations, clients can get it through the MPD mechanism.

Details of the preparation of these video sequences can be found in § 7.2

### 4.4.4 DASH Emulation

We developed (in Python) a simulator in order to evaluate the performance of DASH-based adaptation algorithms. This simulator can work in trace-driven mode, i.e., the networking context is reconstituted with real networking traces. The simulator reproduces timely the instants of video chunk download completion (which depends on network condition) as well as the video chunk playback (which can be blocked by re-buffering).

The main objective of this simulator is to assess the performance of ABR algorithms for DASH video streaming using video chunks encoded and streaming it over the collected bandwidths traces.

The default critical value for buffer occupancy ($Lc$) is set to 12 seconds (3 chunks). The default different buffer sizes is either $BS = 120$ seconds or $BS = 240$ seconds.

### 4.4.5 Performance metrics

The performance of the algorithms is assessed through 4 metrics (i.e., Rebuffering, Instability, SSIM, birate). For each metric, the average value is computed on 24 tests:

i. **Average Rebuffering** is the average value of rebuffering (freezing) duration.

ii. **Average Instability** is the average value of bitrate changes.

iii. **Average of SSIM** is the average value of the SSIM of the video being displayed.

iv. **Average of bitrate** is the average bitrate of the video being displayed.

## 4.5 A First Experimental Study of VQBA Framework with SSIM Metric

In this section, we present a first study of our VQBA framework with SSIM as video quality metric. For short, we refer it as SBA (SSIM-Based Adaptation).

Recall that the main idea of our framework resides in the fact that we decide to:

- *Increase* the bitrate level only when the SSIM indicates a significant improvement in the video quality (thus getting more video content at almost the same *user perceived* video quality);

- *Decrease* the bitrate level only when there is a real risk of rebuffering (thus minimize the instability).

This study compares and discusses the performance of the SBA to the following three traditional ABR: BBA, FESTIVE and OSMF.

We have tested two scenarios with two different buffer sizes: a) $BS = 120$ seconds, b) $BS = 240$ seconds. Each scenario is tested with 24 different traces (our traces ). The threshold value ($Lc$) is set to 12 seconds (3 chunks) in both scenarios.

### 4.5.1 SBA with *Animation* Video

Here are the experimental results by using the **Animation** video stream. We first give presentation at a per metric basis. We then provide summarized results in Table 4.1.

#### 4.5.1.1 Rebuffering

Figure 4.8 shows that the SBA algorithm introduces zero rebuffering for both scenarios. Actually, we give priority to rebuffering avoidance by setting a critical zone with drastic bitrate drop-off. As BBA works in a similar way, it shows also the same zero rebuffering. We can actually notice that neither SBA nor BBA is *visible*. On the contrary, FESTIVE and OSMF have rebuffering during video playback for 21.208 and 46.25 seconds respectively. Here, our algorithm achieves its design goal and performs better than FESTIVE and OSMF for the given scenarios.

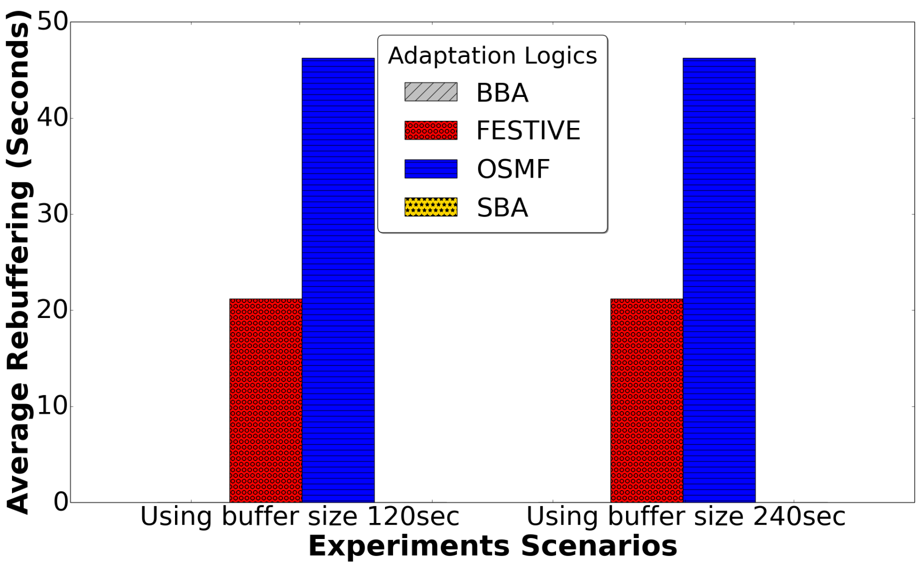



**Figure 4.8:** *Average Rebuffering duration for different algorithms with buffer sizes of* 120 *and* 240 *seconds and with animation (big buck bunny).*

#### 4.5.1.2 Instability

Figure 4.9 shows that SBA achieves good performance, since it is respectively at the first (for $BS = 120$ sec.) and second (for $BS = 240$ sec.) places. For the scenario with $BS = 240$ seconds, BBA algorithm is slightly better than SBA: this is due to a more conservative bitrate increase approach of BBA. But the price to pay is a much lower average bitrate of BBA compared to the others, where as our algorithm keeps the highest average bitrate (cf. Figure 4.11).

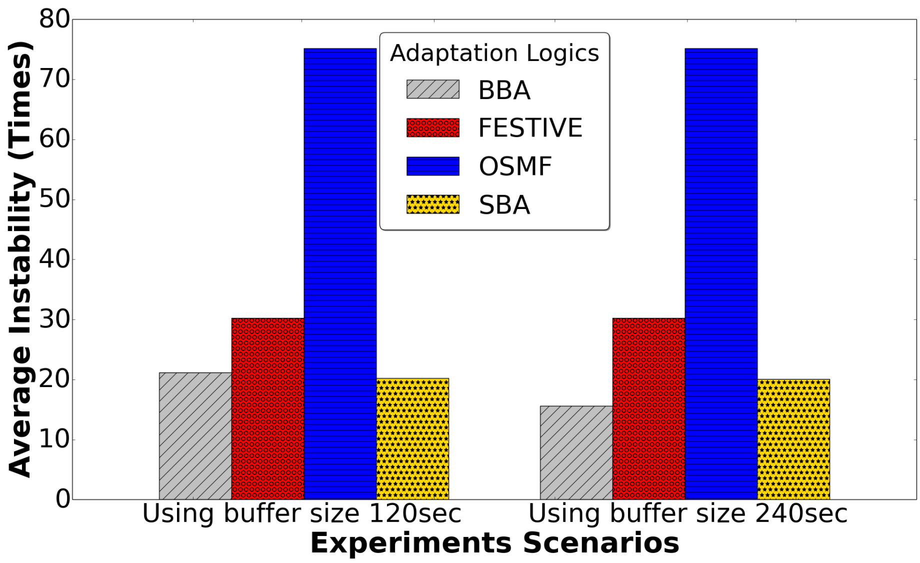



**Figure 4.9:** *Average Instability for different algorithms with buffer sizes of* 120 *and* 240 *seconds and with animation (big buck bunny).*

#### 4.5.1.3 Video quality (SSIM)

Concerning the video quality (here the SSIM metric), we can observe that SBA and BBA have similar performance (see Figure 4.10), which is much better than the two others. This means in particular that our choice of upgrading only if there is a real gain in SSIM is justified.

#### 4.5.1.4 Average bitrate

Figure 4.11 gives the average bitrate for different algorithms. As it is shown, our proposal SBA achieves the highest average bitrate for both scenarios. We notice also that BBA, which have similar performance as our algorithm for the first 3 metrics, gets here the lowest bitrate, probably because there is an excessively conservative consideration for rebuffering avoidance.

#### 4.5.1.5 Summary

Table 4.1 summarizes the results of the two scenarios. For the first 4 lines of the table $BS = 120$ s, whereas for the last 4 lines $BS = 240$ s. One can observe that the SBA algorithm achieves the

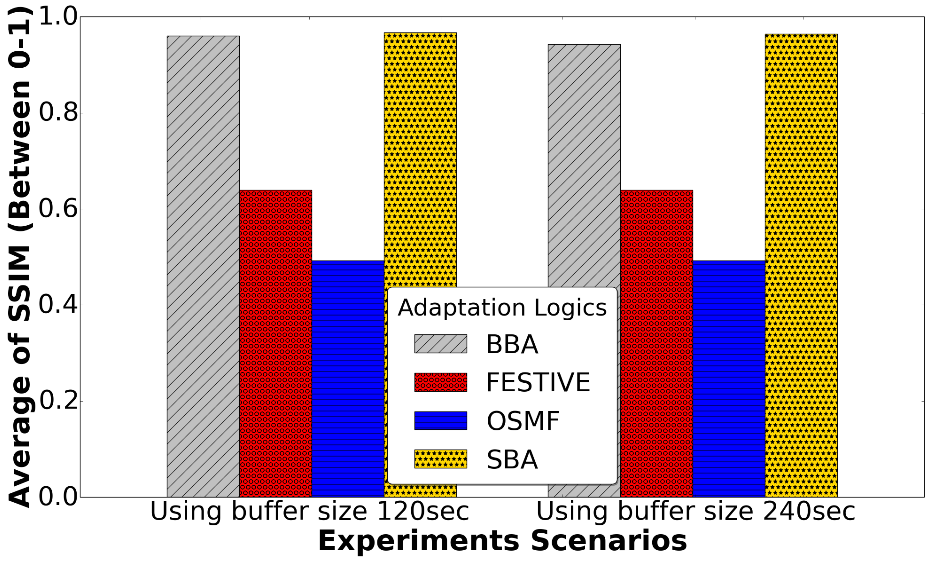



**Figure 4.10:** *Average SSIM for different algorithms with buffer sizes of* 120 *and* 240 *seconds and with animation (big buck bunny).*

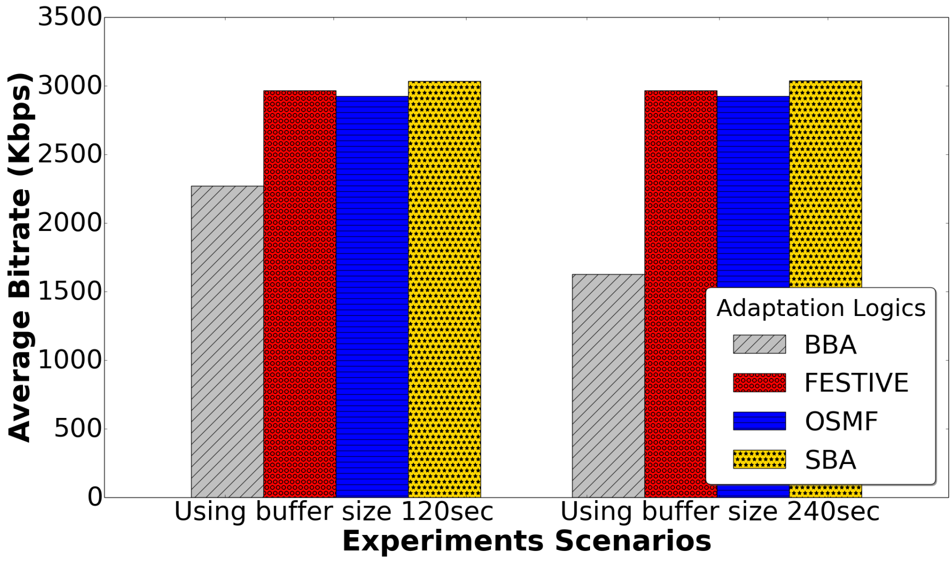



**Figure 4.11:** *Average Bitrate for different algorithms with buffer sizes of* 120 *and* 240 *seconds and with animation (big buck bunny).*

**Table 4.1:** *Summarized results of the two scenarios with* ***Animation****.*

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| SBA | **0.0** | **20.25** | **0.967** | **3035.165** |
| BBA [94] | **0.0** | 21.166 | 0.960 | 2270.699 |
| FESTIVE [92] | 21.208 | 30.208 | 0.638 | 2968.263 |
| OSMF [96] | 46.25 | 75.166 | 0.492 | 2926.625 |
| SBA | **0.0** | 20.083 | **0.964** | **3039.998** |
| BBA [94] | **0.0** | **15.583** | 0.942 | 1629.180 |
| FESTIVE [92] | 21.208 | 30.208 | 0.638 | 2968.263 |
| OSMF [96] | 46.25 | 75.166 | 0.492 | 2924.321 |

desired objective with shorter rebuffering, less instability at a good bitrate level.

### 4.5.2 SBA with *Documentary* and *Sport* Videos

We also conducted experiments with the **Documentary** and **Sport** videos streams. The results are summarized in Table 4.2 (for **Documentary**) and Table 4.3 (for **Sport**).

We observed results which are similar to those obtained with **Animation** video stream. One can notice that for both scenarios, our proposal SBA (SSIM-Mode) of VQBA framework, achieves better ranking for most of the metrics.

## 4.6 A Study on the Choice of Threshold Value for SSIM Metric

### 4.6.1 Motivation

As stated earlier, we use the video quality metrics as a key criterion for bitrate adaptation to optimize the use of available bandwidth. A higher bit rate is chosen not only because the network resources allow to do so but also due to the significant improvement in visual quality (here measured through SSIM metric). In this manner, an upgrade in bandwidth utilization is allowed only when there will be a significant visual quality observed.

Thus, a critical issue is how to determine what is a significant visual improvement. In the previous study, we used a dynamic function which provides the threshold. Here, we go further on the investigation of the impact of the threshold for the SSIM metric (which is normalized to

**Table 4.2:** *Summarized results of the two scenarios with* ***Documentary****.*

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| SBA | **0.0** | 23.5 | **0.941** | 2927.646 |
| BBA [94] | **0.0** | **19.916** | 0.921 | 2326.034 |
| FESTIVE [92] | 22.625 | 29.708 | 0.601 | **2963.134** |
| OSMF [96] | 44.625 | 76.958 | 0.459 | 2848.442 |
| SBA | **0.0** | 23.416 | **0.941** | 2929.382 |
| BBA [94] | **0.0** | **14.458** | **0.941** | 1635.275 |
| FESTIVE [92] | 22.625 | 29.708 | 0.601 | **2963.134** |
| OSMF [96] | 44.625 | 76.958 | 0.459 | 2841.568 |

**Table 4.3:** *Summarized results of the two scenarios with* ***Sport*** *video test.*

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| SBA | **0.0** | **17.791** | **0.957** | **3086.453** |
| BBA [94] | **0.0** | 17.875 | 0.952 | 2470.888 |
| FESTIVE [92] | 18.541 | 28.0 | 0.673 | 2992.655 |
| OSMF [96] | 46.625 | 77.083 | 0.478 | 2866.815 |
| SBA | **0.0** | 17.75 | **0.954** | **3078.769** |
| BBA [94] | **0.0** | **12.708** | 0.929 | 1937.89 |
| FESTIVE [92] | 18.541 | 28.0 | 0.673 | 2992.655 |
| OSMF [96] | 46.625 | 77.083 | 0.478 | 2864.495 |

0..1), through various options. Our comparative studies are always based on real networking context (mobile network) captures and real video sequences. Without loose of generality, there are two approaches: fixed threshold vs dynamic one.

### 4.6.2 Fixed Threshold

For fixed values, we tested the set $\{0.005, 0.02, 0.1\}$ which offers a certain idea both on range and granularity. These fixed values have been chosen after a preliminary study on the SSIM variation between adjacent chunks at adjacent levels[1] for the three sequences under test. The results are shown through 4.12, Fig. 4.13 and Fig. 4.14, respectively. For these figures, we computed the SSIM variation for three couples of bitrates: **low** (235 vs 375), **middle** (1050 vs 1750) and **high** (5800 vs 4300). These variations are compared against fixed values (horizontal lines) and the dynamic formula $\alpha(l)$. Variations are mainly of small extend. It is why we selected $\{0.005, 0.02, 0.1\}$ in a non-linear manner.

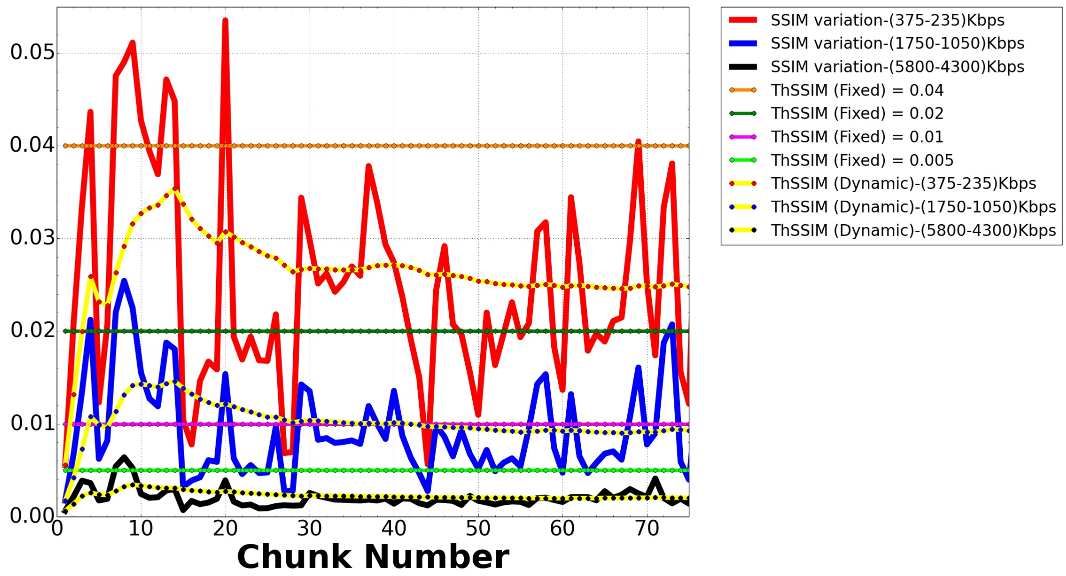



**Figure 4.12:** *Variation of SSIM for high, medium and low quality levels for* ***Animation****.*

### 4.6.3 Dynamic Threshold

For dynamic one, we conducted the following experiments. We first measure, for each chunk (say $l$ for $l > 1$), the variation ($\Delta(l)$ in terms of SSIM related to the previous chunk $(l-1)$, i.e. $\Delta(l) = Q(l, d_l) - Q(l-1, d_{l-1})$ where $d_l$ and $d_{l-1}$ are the respective bitrate levels of chunk $l$ and chunk $l-1$. We then memorize the sum till $l$ with $S(l) = \sum_{k=2}^{l} \Delta(k)$. The adaptive threshold $\alpha(l)$ is then computed with:

$$\alpha(l) = \frac{S(l)}{l-1} = \frac{\sum_{k=2}^{l} \Delta(k)}{l-1}. \tag{4.4}$$

[1]Indeed, a major part of the bitrate increases are gradual.

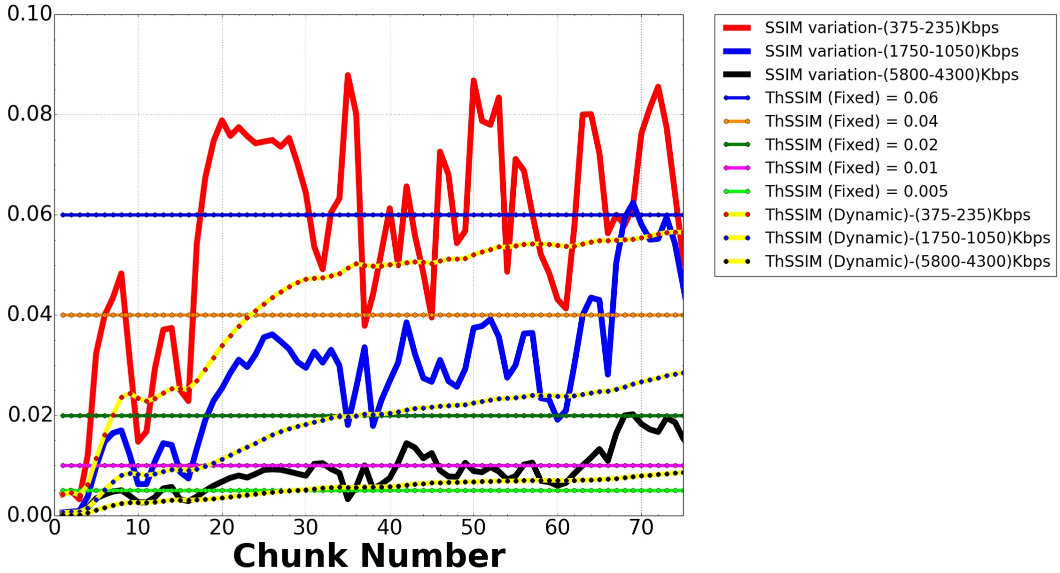



**Figure 4.13:** *Variation of SSIM for high, medium and low quality levels for* ***Documentary****.*

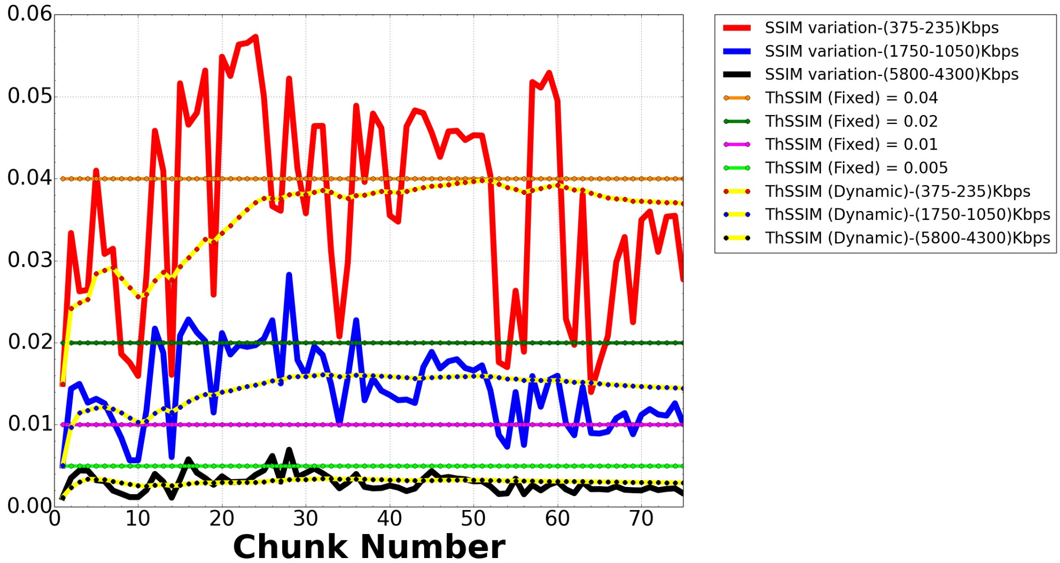



**Figure 4.14:** *Variation of SSIM for high, medium and low quality levels for* ***Sport****.*

**Table 4.4:** *Summarized results with* ***Animation***.

| Scenarios | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| $th_{SSIM}=\alpha(l)$ | 0.7 | 17.3 | 0.934 | 2202.3109 |
| $th_{SSIM}=0.1$ | 0.9 | 13.0 | 0.930 | 2163.437 |
| $th_{SSIM}=0.02$ | 0.7 | 21.8 | 0.936 | 2345.468 |
| $th_{SSIM}=0.005$ | 1.2 | 22.1 | 0.933 | 2437.971 |
| $BBA$ | 0.0 | 23.1 | 0.956 | 1852.852 |
| $FESTIVE$ | 28.0 | 38.6 | 0.504 | 2609.941 |
| $OSMF$ | 60.6 | 76.1 | 0.299 | 2737.966 |

Notice that, in practice, the video playback begins with the pre-fetch of several chunks, so $l > 1$ is assured.

### 4.6.4 Results and analysis

For the evaluation test, we still use the **Animation**, **Documentary** and **Sport** video streams. The performance is always assessed through 4 metrics (Rebuffering, Rate Switching, SSIM, bitrate). Each sequence is tested with 10 different traces. For each metric, we provide its average value over the 10 tests.

The results (in terms of mean values) are shown through the following three tables, Table 4.4, Table 4.5 and Table 4.6, for each of the three sequences through four QoE metrics: Rebuffering *(in seconds)*, Rate Switching *(times)*, SSIM *(value between 0-1)* and BitRate *(kbps)*.

Each table has actually two parts: the upper part gives results obtained with our SBA algorithm with various threshold values. The second part gives, as comparison, the result of the same sequence with three other adaptation algorithms (BBA, FESTIVE, OSMF) under the same experimental conditions.

We begin our discussion by consider only the first parts of the tables, i.e., we limit our discussion inside the SBA algorithm. From these tables, we get the following observation and analysis.

- We can see that the fixed threshold at 0.1 would offer the worst QoE based on BitRate. Indeed, among the 4 metrics, it offers only the lowest rate switching. But, when looking at

**Table 4.5:** *Summarized results with* ***Documentary****.*

| **Scenarios** | **Rebuffering** | **Rate Switching** | **SSIM** | **BitRate** |
|---|---|---|---|---|
| $th_{SSIM}$=$\alpha(l)$ | 0.3 | 13.0 | 0.899 | 2227.075 |
| $th_{SSIM}$=0.1 | 0.0 | 12.8 | 0.905 | 1867.652 |
| $th_{SSIM}$=0.02 | 1.1 | 13.0 | 0.885 | 2241.053 |
| $th_{SSIM}$=0.005 | 1.0 | 17.4 | 0.891 | 2414.369 |
| $BBA$ | 0.0 | 20.5 | 0.916 | 1876.896 |
| $FESTIVE$ | 31.0 | 39.5 | 0.448 | 2588.269 |
| $OSMF$ | 61.0 | 77.9 | 0.300 | 2620.008 |

**Table 4.6:** *Summarized results with* ***Sport****.*

| **Scenarios** | **Rebuffering** | **Rate Switching** | **SSIM** | **BitRate** |
|---|---|---|---|---|
| $th_{SSIM}$=$\alpha(l)$ | 0.5 | 10.6 | 0.937 | 2424.941 |
| $th_{SSIM}$=0.1 | 0.1 | 5.9 | 0.929 | 1820.094 |
| $th_{SSIM}$=0.02 | 0.6 | 17.4 | 0.932 | 2424.555 |
| $th_{SSIM}$=0.005 | 1.1 | 19.6 | 0.919 | 2487.851 |
| $BBA$ | 0.0 | 18.8 | 0.948 | 2078.227 |
| $FESTIVE$ | 19.7 | 28.2 | 0.635 | 2724.609 |
| $OSMF$ | 63.6 | 76.8 | 0.331 | 2691.957 |

the bitrate, which is the lowest, we can also conclude that this is because the bitrate is often kept at lower level since the threshold is too high. So, 0.1 seems to be a too conservative threshold.

- At the other end, the threshold value of 0.005 makes the adaptation too *agressive*. Actually, also it provides a slightly higher bitrate, it also leads to higher rebuffering and rate switching, both are rather irritant. After all, it does not offer a better SSIM.

- Between the fixed threshold at 0.02 and the dynamic $\alpha(l)$, the tie-break is not so easy:

(i) The fixed threshold at 0.02 offers a slightly better SSIM and higher bitrate, whereas ;

(ii) The dynamic threshold seems offers a lower rebuffering risk. As the rebuffering is our primary concern, we have a preference for the dynamic threshold.

We then take the whole table into account, i.e. we make a comparative study of SBA algorithm and the three others. SBA algorithm offers in general a more balanced result among the 4 metrics, even for fixed thresholds at 0.1 and 0.05. This confirms the phenomenon we already observed with only the dynamic threshold ($\alpha(l)$) in our previous studies.

### 4.6.5 Conclusion

In conclusion, the results support the following two points:

- The use of an objective visual quality metric as an adaptation parameter is really effective in termes of QoE improvement, versus some representative algorithms which are not visual quality aware, both with fixed and dynamic threshold values.

- In the case of SBA, an adaptive threshold is better suited since it provides a better trade-off between various factors of a video streaming session.

## 4.7 Experimental analysis of VQBA with a second set of traces (BE traces)

### 4.7.1 Presentation

This section provides another set of experiments under the SBA mode, by using the Belgium traces (BE-set) (cf. § 4.4.2 available at [106].

A comparative study has been made with BBA, FESTIVE and OSMF algorithms through the same performances metrics (§ 4.4.5)

### 4.7.2 Detailed results

Hereafter, we present results in a per metric basis, for different experiments based on types of videos (animation, documentary and sport).

#### 4.7.2.1 Rebuffering

Concerning the Rebuffering, Figure 4.15 shows that our proposal SBA and BBA continue to avoid rebuffering. On the contrary, FESTIVE and OSMF suffer from many rebuffering events during video chunks playback.

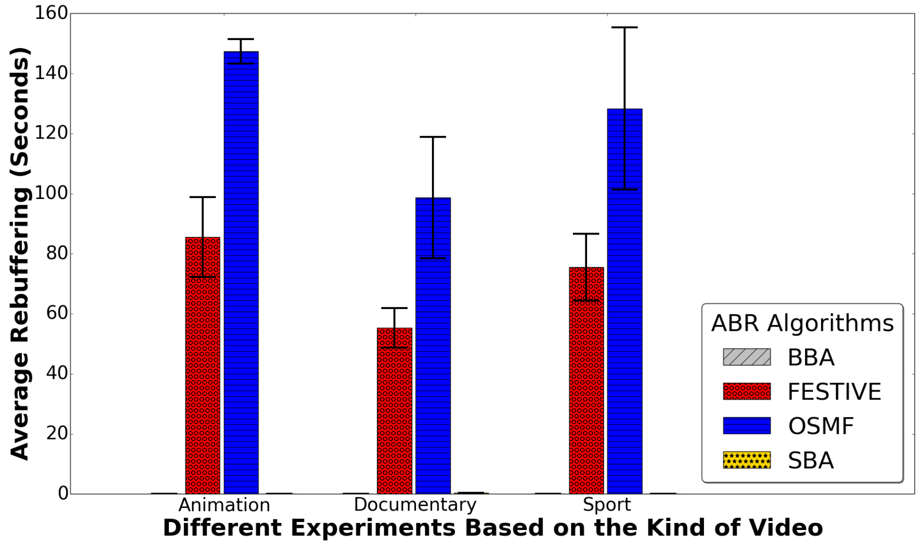



**Figure 4.15:** *Average Rebuffering for different algorithms based on types of videos (Animation, Documentary and Sport).*

#### 4.7.2.2 Instability

Figure 4.16 shows in particular that our proposal SBA achieves good performance. BBA algorithm is slightly better than our proposal SBA: this is probably due to a more cautious bitrate increase approach of BBA. But the price to pay is a much lower average bitrate of BBA compared to the others ABR algorithms, where as our algorithm keeps the good average bitrate (cf. Figure 4.18).

#### 4.7.2.3 SSIM

Figure 4.17 shows that SBA and BBA ostensibly have achieved a much better performance for SSIM than FESTIVE and OSMF. This means in particular that our choice of upgrade only if there is a real gain in SSIM is jonce again ustified.

#### 4.7.2.4 Average bitrate

Figure 4.18 represents the average bitrate, as shown, our proposal SBA (SSIM-Mode) achieve good average bitrate. We notices also BBA, which have good performance as our algorithm for the first 3 metrics, gets here the lowest bitrate, probably due to its excessively conservative rebuffering avoidance policy.

### 4.7.3 Summary and conclusion

The following tables summarizes the results, namely Table 4.7 (for **Animation**), Table 4.8 (for **Documentary**) and Table 4.9 (for **Sport**).

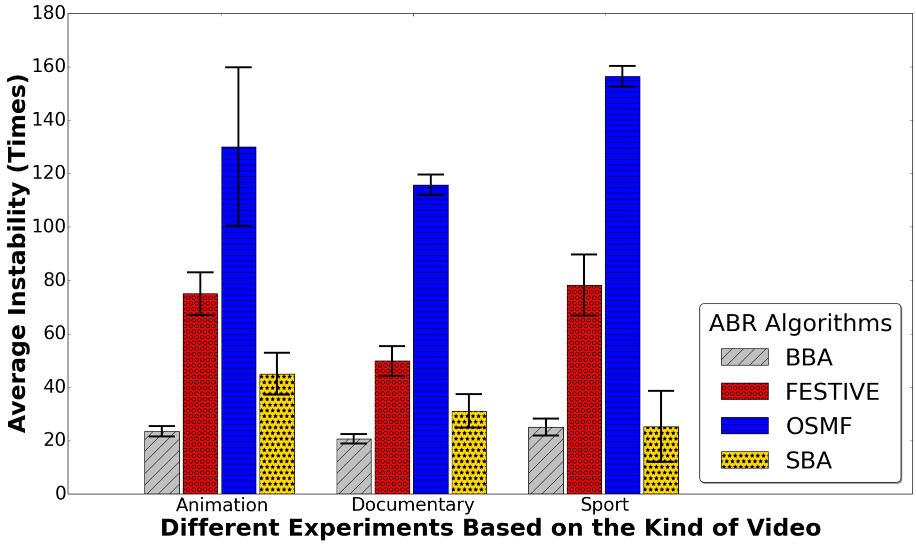



**Figure 4.16:** *Average Instability for different algorithms based on types of videos (Animation, Documentary and Sport).*

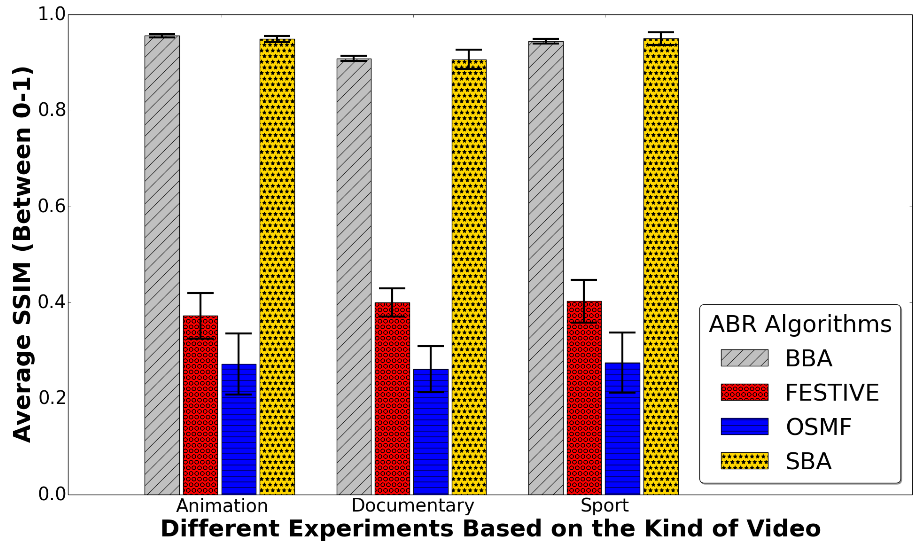



**Figure 4.17:** *Average SSIM for different algorithms based on types of videos (Animation, Documentary and Sport).*

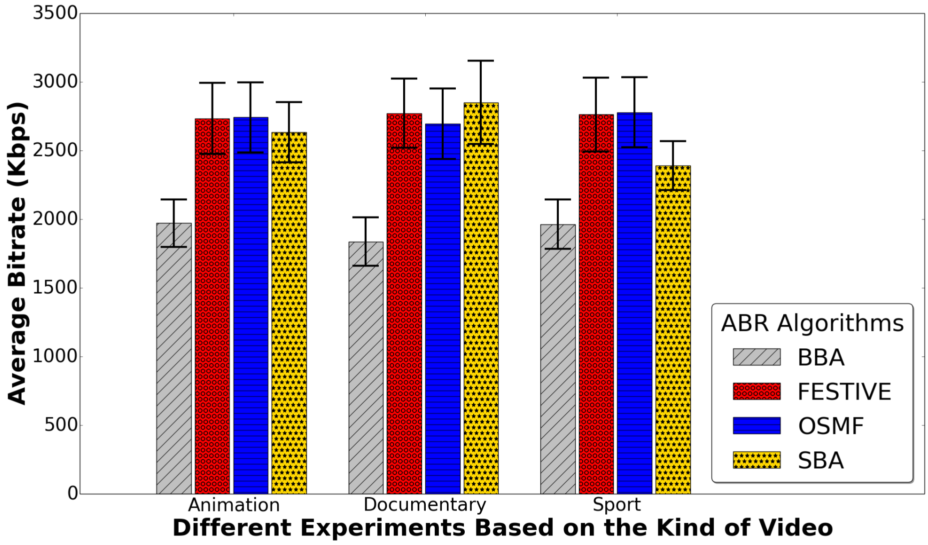



**Figure 4.18:** *Average BitRate for different algorithms based on types of videos (Animation, Documentary and Sport).*

**Table 4.7:** *Summarized results with **Animation**.*

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| SBA | 0,1 | 45,6 | 0,949 | 2635,094 |
| BBA [94] | 0 | 23,5 | 0,956 | 1972,148 |
| FESTIVE [92] | 85,5 | 75 | 0,372 | 2735,336 |
| OSMF [96] | 147,4 | 130,1 | 0,272 | 2742,497 |

We can conclude that through this serie of experimentation under different networking conditions (in Gant, Belgium), we got observation of the performance comparison between our algorithm (SBA mode) and non VQM-aware ones which are similar to those we got under our 1st serie of experimentation (in Paris, France).

## 4.8 Conclusion

In this chapter, we presented a generic framework, named *Video-Quality Metric Based-Adaptation algorithm* (VQBA), to be used under the DASH paradigm for video streaming. The main goals of this contribution are:

- It aims to combining information given by objective video quality metrics, along with those

**Table 4.8:** *Summarized results with* ***Documentary***.

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| SBA | 0,3 | 31,1 | 0,906 | 2851,748 |
| BBA [94] | 0 | 20,7 | 0,908 | 1836,882 |
| FESTIVE [92] | 55,3 | 49,8 | 0,4 | 2772,336 |
| OSMF [96] | 98,7 | 115,8 | 0,261 | 2696,007 |

**Table 4.9:** *Summarized results with* ***Sport***.

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | BitRate |
|---|---|---|---|---|
| SBA | 0,1 | 25,3 | 0,95 | 2390,165 |
| BBA [94] | 0 | 25,1 | 0,944 | 1964,502 |
| FESTIVE [92] | 75,5 | 78,2 | 0,402 | 2763,075 |
| OSMF [96] | 128,3 | 156,5 | 0,275 | 2779,043 |

provided by system/networking-aware information, to make adaptation decision;

- It aims to work with various Video Quality metrics.

To evaluate the performance of our proposal, we have conducted experimental tests, by means of trace-driven emulation, under various networking conditions provided by real traffic traces and with several types of video sequences.

As a first series of tests, we worked with the SSIM metric (the SBA declination of our VQBA framework). Comparisons with some representative non-vido-quality-aware algorithms (BBA, FESTIVE, OSMF) through an objective video quality metric (SSIM) show that our generic framework achieves an efficient adaptation by minimizing both the effect of rebuffering and instability, with good quality for video playback

We can conclude that these results validate our research direction, i.e., the VQM-ware adaptation approach does provide good QoE.

CHAPTER 5

# Further experimental Studies with VQBA

*In theory there is no difference between theory and practice. In practice there is.*

LAWRENCE "YOGUI" BERRA,1925

In this chapter, we present our further experimental investigation of the VQBA framework. We extend our studies to two other video quality metrics, namely PSNR and VMAF. For the sake of simplicity, they will be referred as **PBA** (for PSNR-Based Adaptation) and **VBA** (for VMAF-Based Adaptation), respectively. wth the Animation sequence.

For PSNR, we also carried experimentations on two other video sequences (documentation , sport).

Lastly, we also studied the rebuffering with SBA.

## 5.1 Performance Comparison of SBA, PBA and VBA algorithms

### 5.1.1 Presentation

We discuss and compare the performance between the different modes of VQBA framework (SBA, PBA, VBA) and the BBA, FESTIVE & OSMF algorithms.

The tests presented here were done with the (already presented) **Animation** (*Big Buck Bunny*) [108] sample video sequence (cf. § 4.4.3 for more details about preparation for the experiments). Each sequence has been tested with buffer sizes of 240 seconds (60 chunks). The

**Table 5.1:** *Summarized results objective QoE (SSIM, PSNR & VMAF) With* ***Animation****.*

| Adaptation Logics | SSIM | PSNR | VMAF |
|---|---|---|---|
| $SBA$ | 0,942 | 35,981 | 54,405 |
| $PBA$ | 0,954 | 35,443 | 53,150 |
| $VBA$ | 0,950 | 36,446 | 57,288 |
| $BBA$ | 0,958 | 35,387 | 29,830 |
| $FESTIVE$ | 0,636 | 25,251 | 43,806 |
| $OSMF$ | 0,323 | 30,291 | 47,483 |

critical value for buffer occupancy ($Lc$) is set to 12 seconds (3 chunks).

The performance of the algorithms is assessed through 6 metrics :

- Three video quality metrics: SSIM, PSNR, VMAF;
- Three more conventional QoE metrics: Rebuffering duration, Bitrate Switching, Mean BitRate.

Each sequence is tested with 20 different traces. For each metric, we provide its average value over the 20 tests.

### 5.1.2 Performance Comparison between SBA(SSIM), PBA(PSNR) and VBA(VMAF)

Table 5.1 summarizes the results of our experiments for different adaptation algorithms by using buffer size 240 seconds & with Animation through three objective QoE metrics (SSIM, PSNR & VMAF). One can observe that different modes (SBA, SBA & VBA) of VQBA framework achieved a good performance through QoE metrics (SSIM, PSNR & VMAF) compared with the (control) relevant algorithm.

The following three figures (Figure 5.1, Figure 5.2, Figure 5.3) provide a more visual comparison.

#### 5.1.2.1 With SSIM metric

Figure 5.1 shows that our modes (SBA, PBA & VBA), along with BBA algorithm, ostensibly have achieved a better average of SSIM compared with the two other algorithms (FESTIVE & OSMF). More precisely (cf. Table 5.1), BBA achieved the first ranking while the PBA, VBA, SBA, FESTIVE & OSMF achieved the (second, third, fourth, fifth & sixth) ranking respectively.

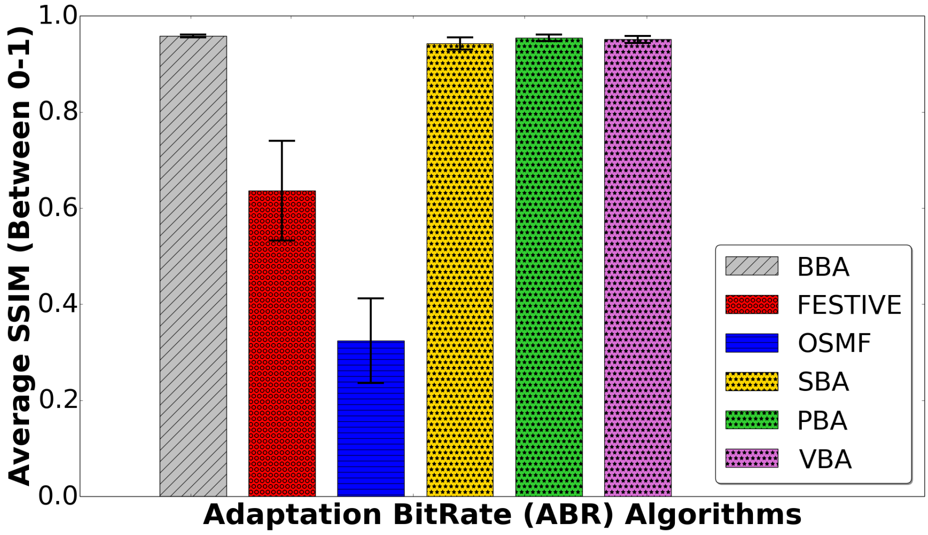



**Figure 5.1:** *Average SSIM for different algorithms with buffer sizes of* 240 *seconds and with animation (big buck bunny).*

#### 5.1.2.2 With PSNR metric

Figure 5.2 shows that our modes (SBA, PBA & VBA) have achieved a better average of PSNR compared to non VQM-aware algorithms (BBA, FESTIVE, OSMF). More precisely (cf. Table 5.1), VBA arrives at the first rank, SBA the second, while PBA (resp. BBA, FESTIVE & OSMF achieved the third (resp. fourth, fifth & sixth) ranks.

#### 5.1.2.3 With VMAF metric

Figure 5.3 shows that the VBA ostensibly has achieved the best average of VMAF, compared to SBA & PBA on the one hand, and, on the other hand, BBA, FESTIVE & OSMF algorithms. More precisely (cf. Table 5.1), SBA that achieved the second-ranking while PBA, OSMF, FESTIVE & BBA achieved the (third, fourth, fifth & sixth) ranking respectively.

### 5.1.3 Performance Comparison with other QoE metrics

Figure 5.4, Figure 5.5 & Figure 5.6 show that our modes (SBA, PBA & VBA) of VQBA framework ostensibly have achieved a good performance also through the classicl QoE metrics (Rebuffering, Rate Switching & BitRate) compared with BBA, FESTIVE & OSMF algorithms.

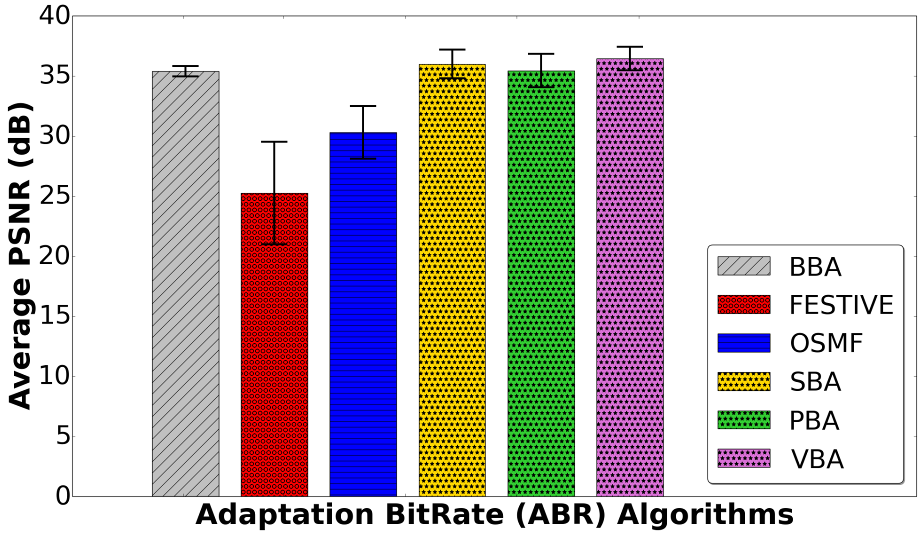



**Figure 5.2:** *Average PSNR for different algorithms with buffer sizes of* 240 *seconds and with animation (big buck bunny).*

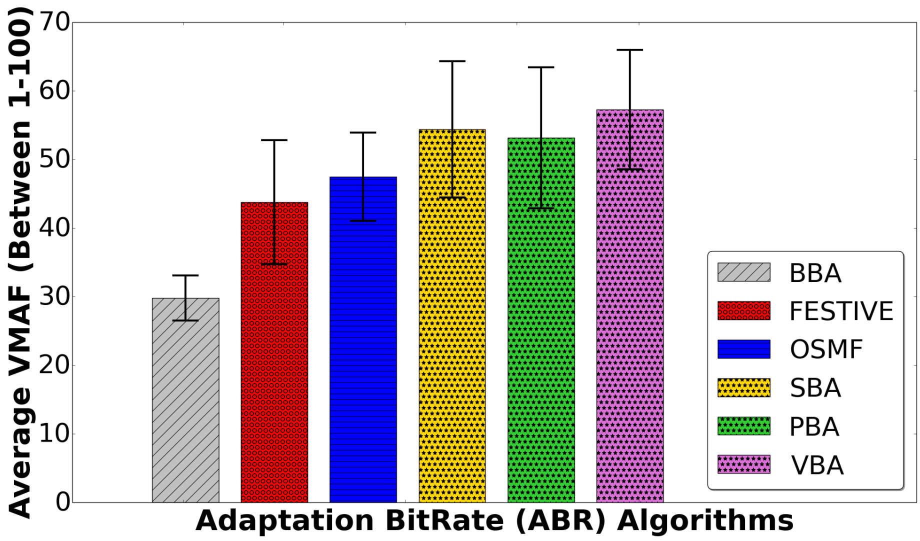



**Figure 5.3:** *Average VMAF for different algorithms with buffer sizes of* 240 *seconds and with animation (big buck bunny).*

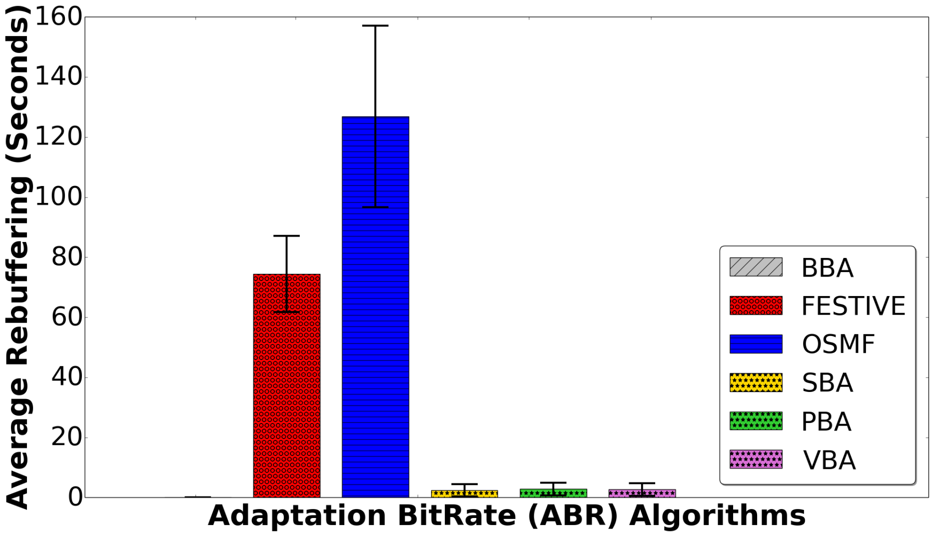



**Figure 5.4:** *Average Rebuffering for different algorithms with buffer sizes of* 240 *seconds and with animation (big buck bunny).*

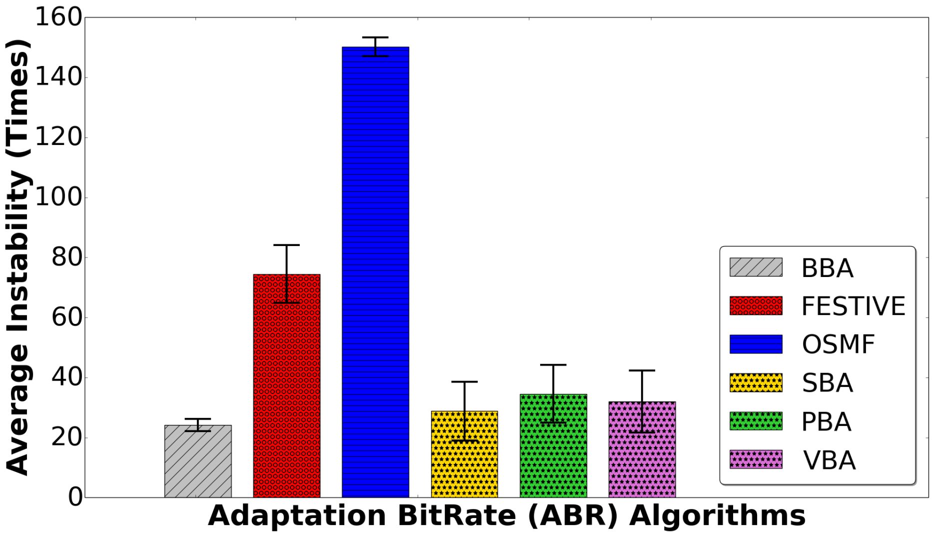



**Figure 5.5:** *Average Instabilty for different algorithms with buffer sizes of* 240 *seconds and with animation (big buck bunny).*

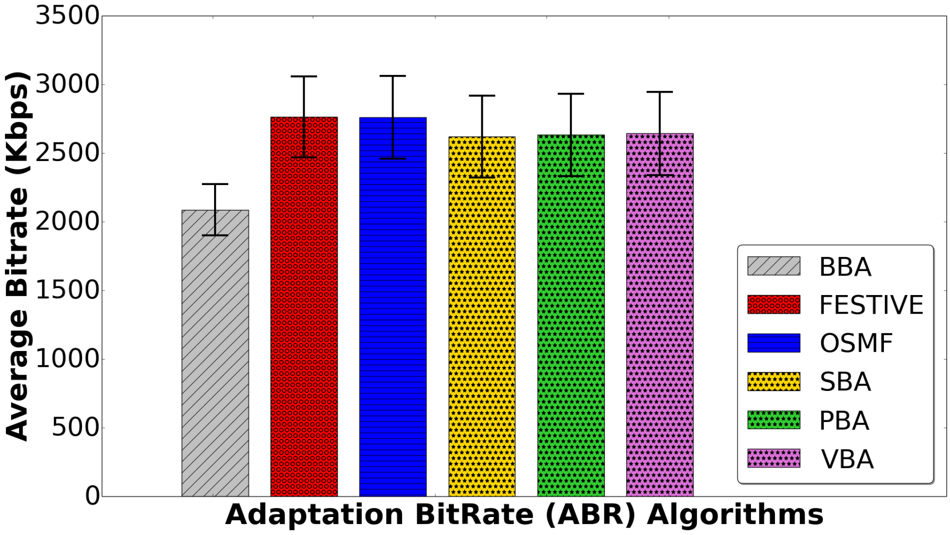



**Figure 5.6:** *Average BitRate for different algorithms with buffer sizes of* 240 *seconds and with animation (big buck bunny).*

### 5.1.4 Conclusion

This study, which is conducted mainly through the viewpoint of video quality accessement, confirm that our framework works not only for the SSIM metric, but also for other metrics. Here the PSNR and VMAF metrics, which ar among the usual QoE metrics. A more comprehensive conclusion is provided in § 5.3.

## 5.2 SBA and PBA with Documentary and Sport Videos

### 5.2.1 Presentation

To further validate our proposed VQBA framework, we conducted additional experiments on **Documentary** and **Sport** video streams [109, 110], in the SBA and PBA cases. Comparisons are made with BBA, FESTIVE & OSMF through five quality metrics (Rebuffering, Rate Switching, SSIM, PSNR & BitRate). The results are summarized in Table 5.2 (for Documentary) and Table 5.3 (for Sport), respectively.

### 5.2.2 SSIM and PSNR metrics with SBA experiences

Figure 5.7 and Figure 5.8 give respectively the SSIM and PSNR metrics under SBA (i.e. the indicator metric is SSIM) with three types of videos streams (Animation, Documentary and Sport) through 20 sets of bandwidth traces.

**Table 5.2:** *Summarized results with* ***Documentary****.*

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | PSNR | BitRate |
|---|---|---|---|---|---|
| SBA | 1,35 | 25,15 | 0,889 | **34,962** | 2760,669 |
| PBA | 0,7 | 25,85 | **0,917** | 34,803 | 2840,669 |
| BBA [94] | **0** | **17,45** | 0,909 | 33,986 | 1902,073 |
| FESTIVE [92] | 46,55 | 54,35 | 0,471 | 32,676 | 2884,559 |
| OSMF [96] | 66,4 | 120,7 | 0,405 | 32,219 | **2900,013** |

**Table 5.3:** *Summarized results with* ***Sport****.*

| Adaptation Algo. | Rebuffering | Rate Switching | SSIM | PSNR | BitRate |
|---|---|---|---|---|---|
| SBA | 3,9 | 40,9 | 0,895 | **36,6** | 2640,24 |
| PBA | 4,25 | 42,3 | **0,947** | 34,799 | 2670,962 |
| BBA [94] | **0** | **25,4** | 0,946 | 35,601 | 2048,604 |
| FESTIVE [92] | 73,7 | 83,35 | 0,432 | 33,666 | 2733,326 |
| OSMF [96] | 126,2 | 160,5 | 0,298 | 31,815 | **2775,614** |

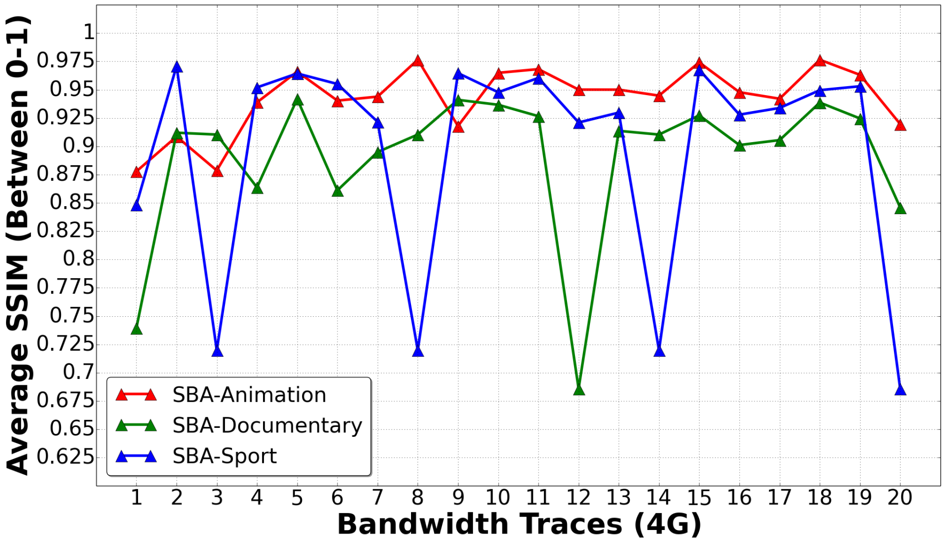



**Figure 5.7:** *SBA performance with movies (Animations, Documentary and Sport) based on SSIM.*

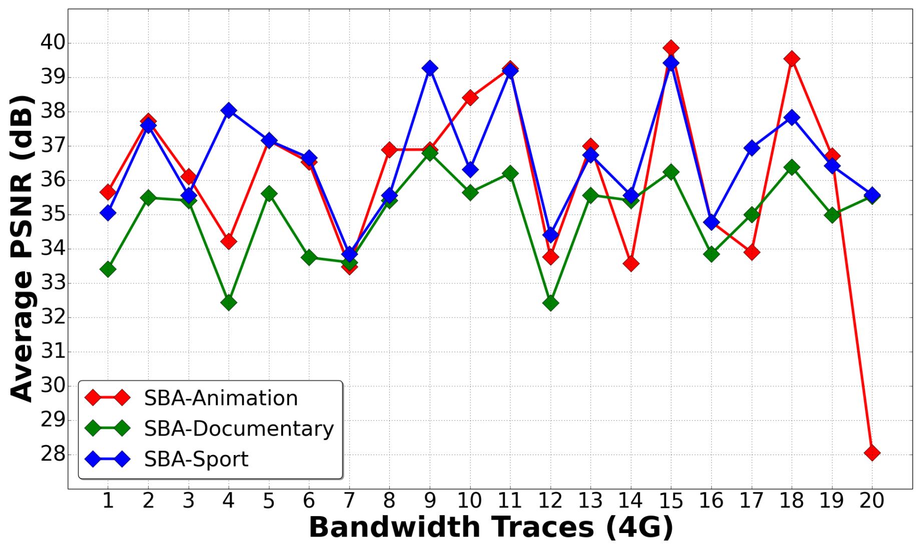



**Figure 5.8:** *SBA performance with movies (Animations, Documentary and Sport) based on PSNR.*

#### 5.2.2.1 SSIM and PSNR metrics with PBA

Figure 5.9 and Figure 5.10 give respectively the SSIM and PSNR metrics under PBA (i.e. the indicator metric is PSNR) with three types of videos streams (Animation, Documentary and Sport) through 20 sets of bandwidth traces.

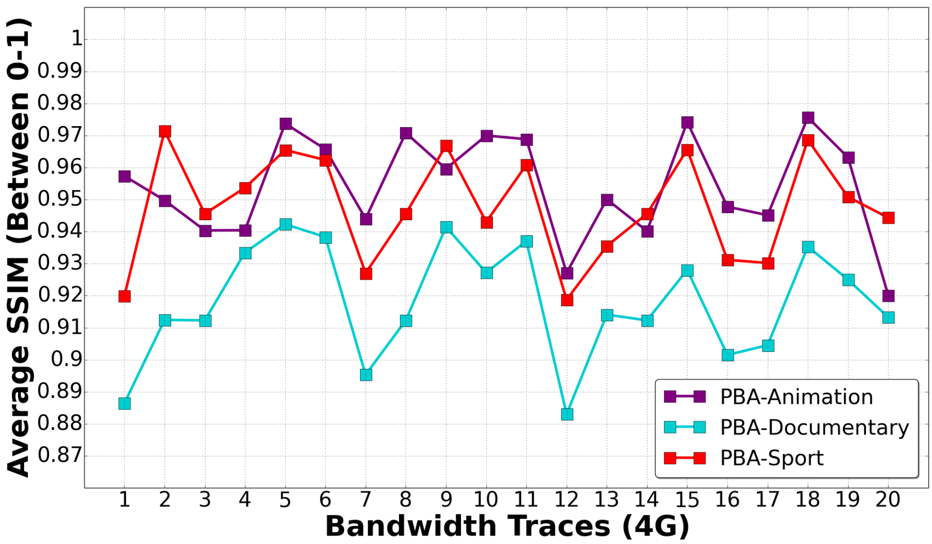



**Figure 5.9:** *PBA performance with movies (Animations, Documentary and Sport) based on SSIM.*

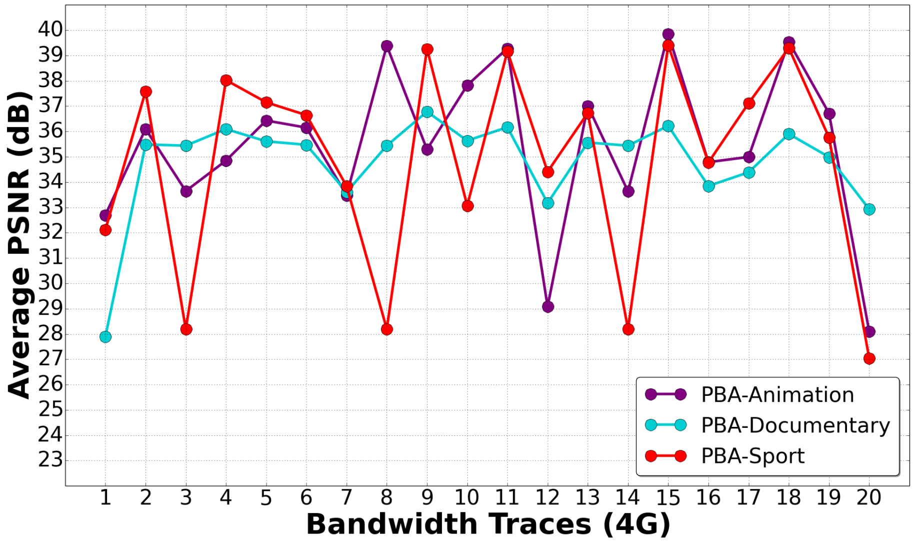



**Figure 5.10:** *PBA performance with movies (Animations, Documentary and Sport) based on PSNR.*

### 5.2.3 Average SSIM and PSNR metrics comparison

The previous results are brute data from our experiences, which are provided in order to get a more insight vision of the various situations.

Figure 5.11 and Figure 5.12 provide a comparison of the average values (through the brute data) of SSIM and PSNR, under respectively SBA and PBA, always in function of the type of the video (Animation, Documentary and Sport).

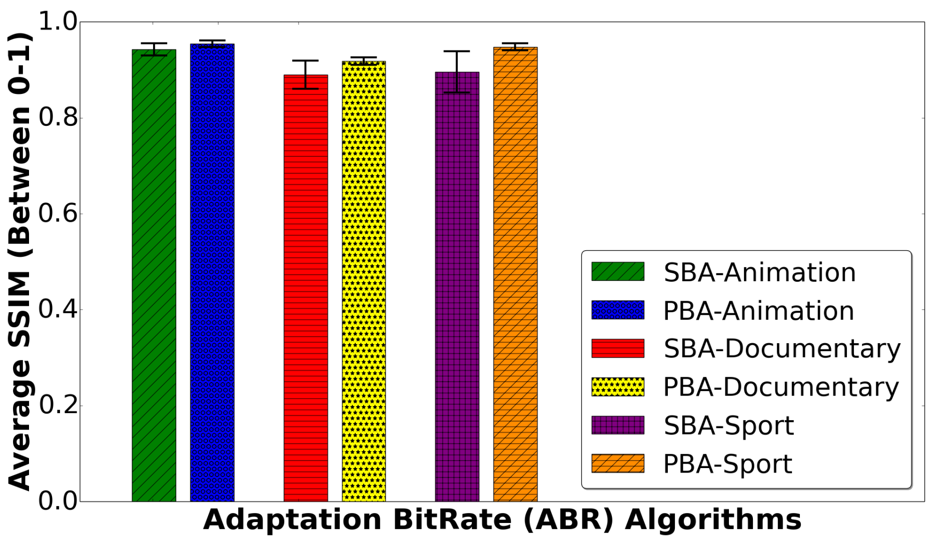



**Figure 5.11:** *SBA vs PBA with movies (Animations, Documentary and Sport) based on SSIM.*

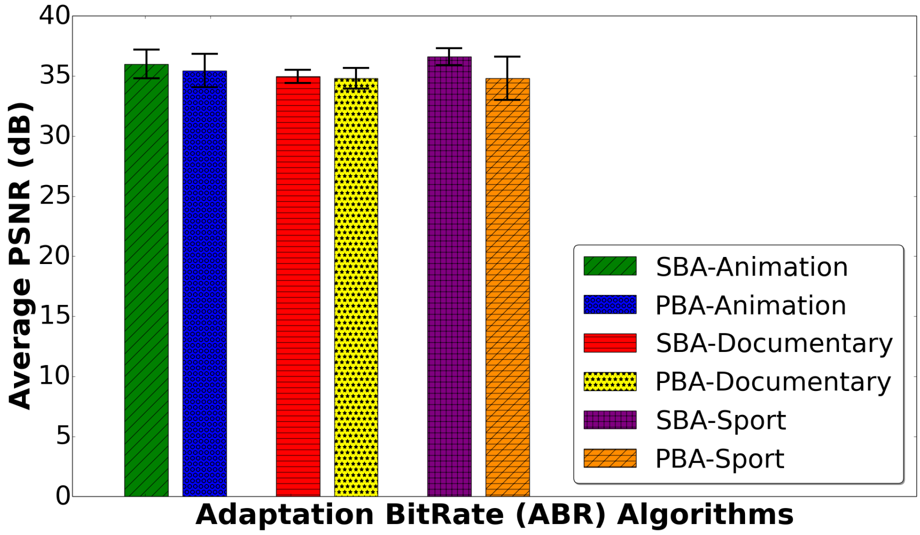



**Figure 5.12:** *SBA vs PBA with movies (Animations, Documentary and Sport) based on PSNR.*

## 5.3 Conclusion of extended studies to PSNR and VMAF

Through these extended studies to PSNR and VMAF, we obtained broader and consolidated results related to our *generic framework* VQBA. This framework is actually designed to be used by any objective video quality metric.

At this point, we recall the main features of our framework which are:

- Take advantage of the objective video quality metrics that are seldom used in the existing ABR streaming algorithm for the simple reason that, after all, users are aware of video quality,
- Based on the objective VQM indicator, we make adaptation decision for a better bitrate (and so expected better QoE) in a way that thkis leverage *does actually* improves the video quality (QoE) according to the VQM, not only because it is allowed by network conditions. This allows to get more video content at almost the same user-perceived video quality.
- We make use a more efficient and effective use of the available bandwidth, which contribute to minimise the rebuffering and the video quality fluctuation.

.

To evaluate the performance of this generic VQBA framework, we carried experimentations by emulating real networking circumstance with various traffic traces captured from real mobile networks and three types of video content (animation, documentary, Sport). We compared the performance of our algorithm (under SBA, PBA and VBA modes) with non VQM-aware adaptation algorithms (BBA, FESTIVE & OSMF) through six video quality metrics (SSIM, PSNR, VMAF, Rebuffering, Rate Switching & BitRate).

The good performance of our algorithms confirm that our proposal does make sense and make difference by taking into account the objective video-quality metric as an adaptation parameter.

## 5.4 Analysis of the Rebuffering phenomena under SBA mode

### 5.4.1 Introduction

Recall that, with DASH, a video stream is segmented into contiguous *chunks*, each chunk is downloaded via HTTP to the video end-user then stored in the buffer, before being played back when its display times comes.

A very boring phenomenon is the absence of the chunk to be displayed when its time comes. Actually, from the view point of application, the only possible action is to **freeze** (get a pause of) the video playback, till the availability of this very chunk. One can easily imagine that this kind of pause is not pleasant at all for end-users. This is the *rebuffering* phenomenon that we already introduced.

These phenomena are usually due to the lack of adequation between the amount of data of the chunk(s) that one claim and the actual available bandwidth.

As the bandwidth fluctuation is a reality, usually, there is always a prefetch of a first set of chunks before the starting of playback. In this way, there is a margin for minimising the rebuffering.

Many studies on DASH video streaming show that rebuffering events (freezing playback) are the key influence factors of QoE [25, 59, 85, 114]. Here, we will provide a detailed study of the rebuffering events under the SBA mode.

### 5.4.2 DASH and Rebuffering Events

DASH player uses adaptation algorithms to determine the most suitable representation for the next video chunk The representation is selected to play the best possible quality of the video with the least number of interruptions. Thus, the DASH player is able to adjust the quality of the video at the boundary of each segment in order to ensure uninterrupted playback [59, 115].

This functional behaviour makes DASK very closely linked to the rebuffering, actually, a bad adaptation could lead easily to rebuffering, which should be avoided [9].

The ABR algorithms aim to enhance the satisfaction of the video end-user by selecting the appropriate video bitrate for network conditions. However, selecting the suitable bitrate is challenging due to variations in mobile network throughput specifically, to achieve the balance between QoE metrics, which are conflicting in their nature (e.g. maximize video quality and minimizing rebuffering/instability) [116, 117].

It has been established for a while that QoE is influenced by both the duration and the frequency of rebuffering events [118]. Rebuffering events have remarkably negative effects on subjective QoE [119, 120]. It is interesting to notice that single rebuffering (long rebuffering duration) is preferred compared to multiple short rebuffering and regular (e.g. one rebuffering event every 3 s) over irregular video rebufferings.

### 5.4.3 Related Work on Objective QoE Metric-Rebuffering Events

With respect to rebuffering events: the majority of the recent studies on video QoE [79, 82], agree on the fact that it should be avoided rebufferings events. If possible to enhance the video QoE.

On the other hand, that satisfaction of the viewers can vary relying on a rebuffering manner, i.e., how many times or the duration of the rebuffering events arise during video playback.

In general, the selection of the ideal bitrate for the next video chunk is very difficult because of the coarse-grained nature of ABR decisions. Where the selection not good bitrate of the next video chunk leads to depletion of the buffer and thus to the occurrence of rebuffering events the cause of these events is due to the variability of network throughput and the conflicting video QoE metrics requirements (minimize rebuffering and switching events and the same time maximize average bitrate, etc.).

Many studies work in with the fact that video rebuffering in DASH should be avoided at all times to get better QoE of user's video streaming, while it is a common hypothesis that rebuffering events are more annoying than video quality switching during video session. In general, the conclusion that video rebuffering must be avoided at any time has been confirmed (cf. § 3.4.2 see more related work on rebuffering events).

In [120] despite using DASH is being increasingly deployed by video content providers, such as Netflix and YouTube. Where the client is able to fetch the appropriate bitrate for the next video chunk to be played next based on the estimation bandwidth (network condition). However, this can introduce impairment rebuffering events, which can severely impact an end user's quality of experience (QoE). They create a new video quality database, which simulates a typical video streaming application, using long video sequences and interesting Netflix content, the new database contains highly diverse video content and it includes the subjective opinions of a sizeable number of human subjects regarding the study on QoE of both rebuffering and compression distortions. In their study, they observe that rebuffering is always clear or noticeable for the video users and it is not preferred (unloved) to subjects, while switching bitrate may be less clear due to content-related dependencies. the video quality drops were preferable over rebuffering.

In [121] one of their main contribution (based on consensus on the requirements for a good ABR algorithm) designing and implement two novel ABR algorithms: BOLA-E and DYNAMIC. The Both minimize rebuffering, while maximizing the average video quality. Maximizing the QoE of the user includes metrics that are often in conflict. ABR algorithms must stream at a high bitrate with low rebuffering. Also they developed a FAST SWITCHING adaption that can replace video chunks that have already been downloaded with higher-quality video chunks. The new adaptation provide higher QoE to the user in terms of higher video quality, thus some oocurrence rebuffers eventes. In [122] their contribution aims to DASH video delivery rebuffering events while ensuring a low latency delay between the displayed and the original video flows. They suggested in their work to utilise a novel approach to deal with bandwidth lack occurring through a video chunk delivery : The implementation of video frame ignore the policies with HTTP/2 until an the end of video chunk they assess their adaptation by combining dynamic and static videos with cellular and WiFi network bandwidth traces. They tried to get rid from rebuffering events by developing optimal and practical video frame discarding algorithms to meet the 1s latency constraint. In their algorithm, they request the video frames individually through HTTP/2 multiple streams, and we selectively drop the least meaningful video frames thanks to HTTP/2 stream resetting feature.

### 5.4.4 Analysis of Rebuffering Events with SBA

We present here a comparative experimental study which is focused on the the rebuffering phenomenon, under SBA, BBA, FESTIVE and OSMF, respectively. The reference video stream is the *Big Buck Bunny* (animation). The buffer size is set to 240 seconds. The tests have been run for ten trafic traces.

The rebuffering events took place in three out of ten sets of bandwidth traces and most of these events did not occur at the beginning of the video streaming session. Hereafter, we will only focus on these three traces. Figure 5.13 (for the 1st dataset traces), Figure5.14 (for the 2nd dataset traces) and Figure 5.15 (for the 3rd dataset traces) give insight of bandwidth variation related to eache of these three traces.

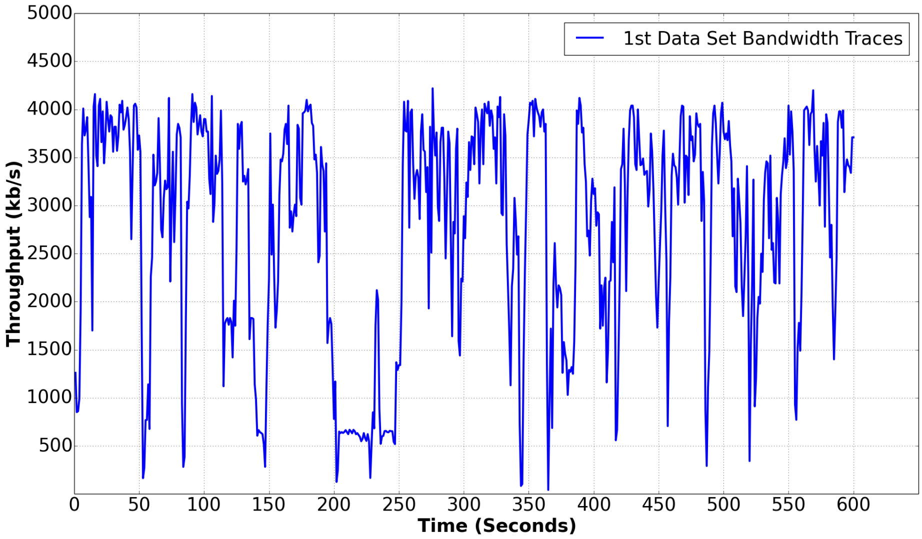



**Figure 5.13:** *The first dataset with bandwidth traces used in experiments.*

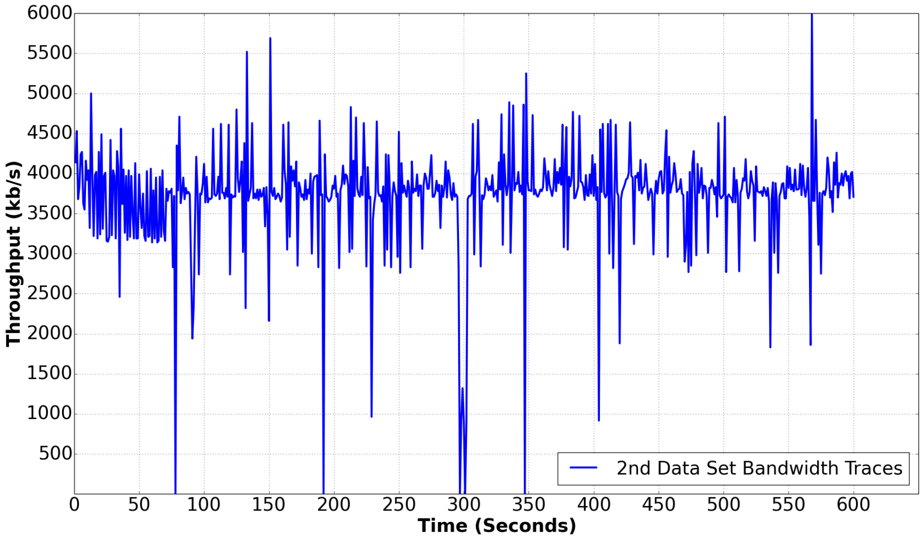



**Figure 5.14:** *The second dataset with bandwidth traces used in experiments.*

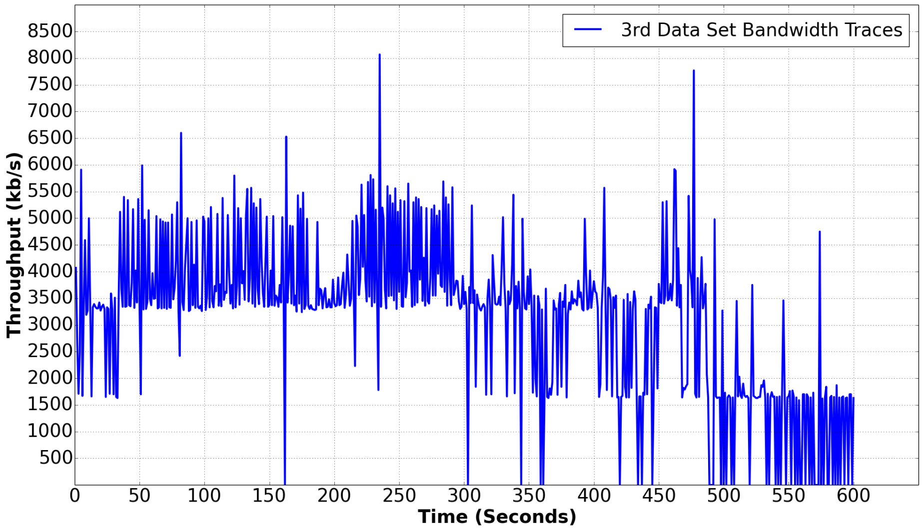



**Figure 5.15:** *The third dataset with bandwidth traces used in experiments.*

#### 5.4.4.1 Observation at an ealy stage of playback with SBA

Figure 5.16 through mini-figures (a), (b) and (c) illustrates the video playback rates actually chosen by our SBA algorithm during the time slice [100, 180] (20 chunks) of the video streaming session. We observe that when the rates are kept steadily at respectively 2350 kbps and 3000 kbps for the trace 1 and trace 3; there is an oscillation between 235 kbps and 4300 kbps the trace 2. There is no rebuffering, but a significant instability which does yield also a negative effect, even though it is generally perceived as less boring than the rebuffering.

#### 5.4.4.2 Rebuffering phenomena with SBA

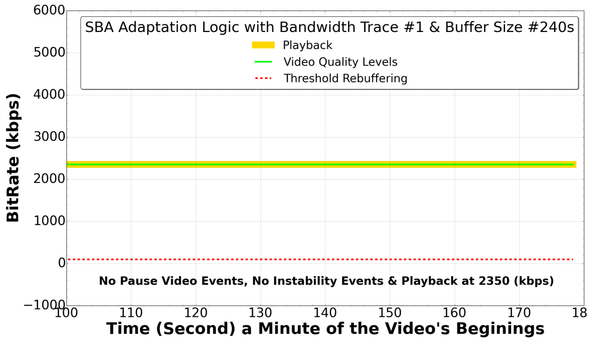
The observed rebuffering phenomena occur relatively far away from the beginning. Figure 5.17 through mini-figures (a), (b) and (c) illustrate some rebuffering events that occurred for three (out of ten) traces with our SBA algorithm. In our examples, the rebuffering last for 6, 3 and 1 seconds, respectively. The observation window is 60 seconds in all cases.

This kind if rebuffing has a limited negative impact on the overall QoE ( [83–85]), due to two main reasons:

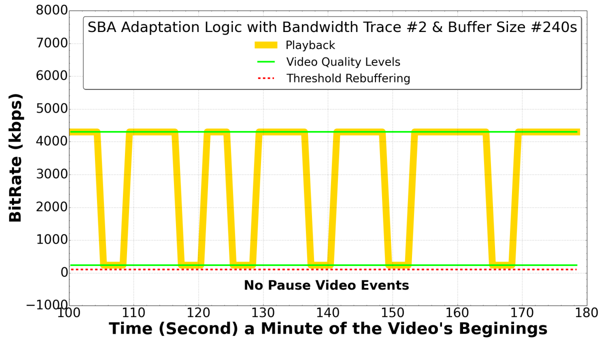
i. The rebuffering happen quite a few times (actually one time) and for a short period during the observation window (60 seconds)

ii. The ebuffering did not occur at the beginning of the video session.

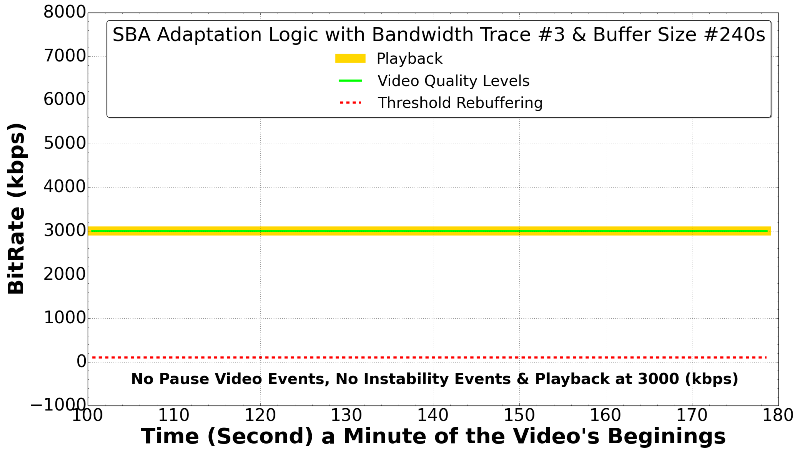
Figure 5.18 through mini-figures (a), (b) and (c) shows the estimation bandwidth by the SBA




(a)




(b)




(c)

**Figure 5.16:** *The situation of video playback at the beginning the video session during using our SBA mode.*

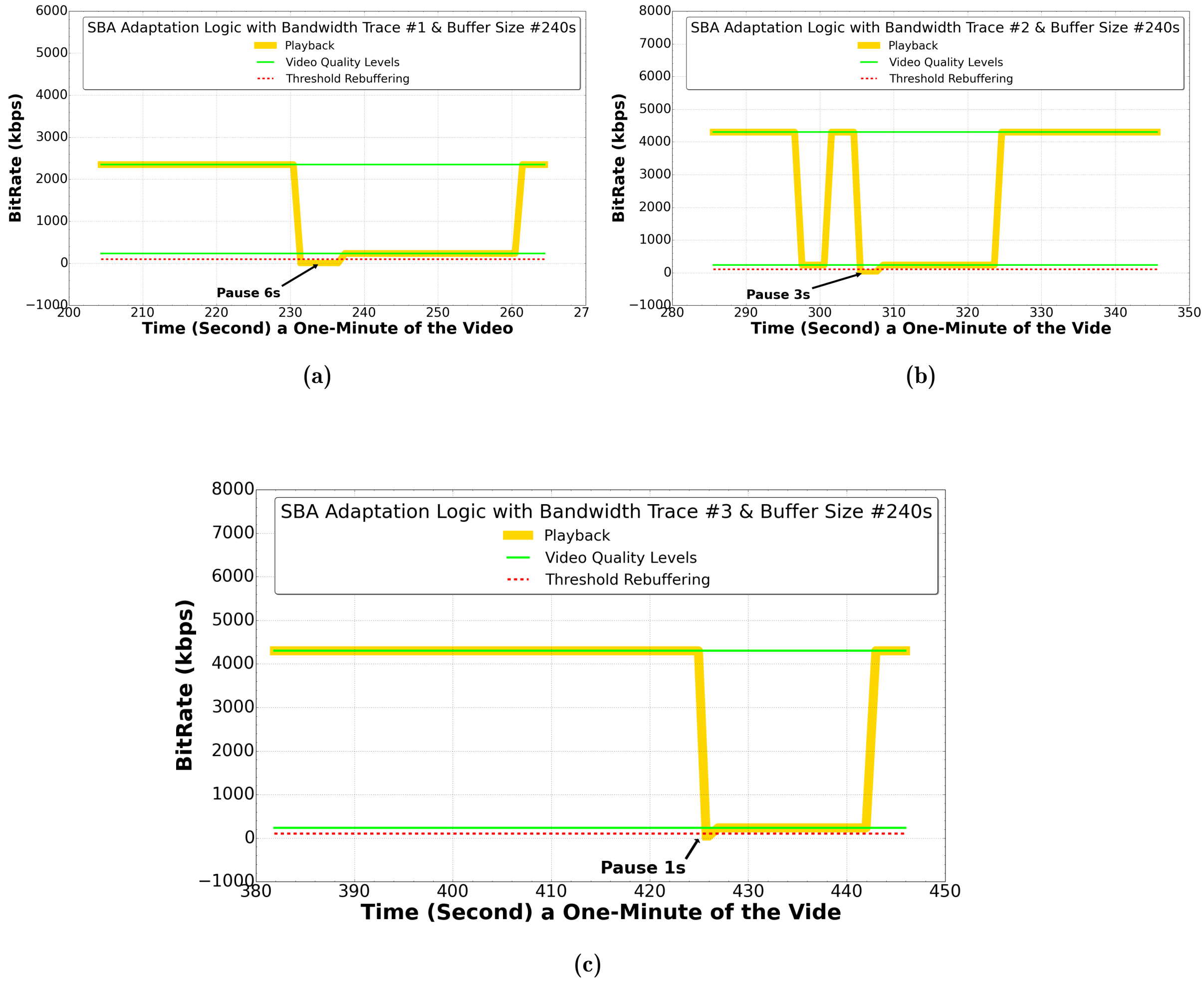



**Figure 5.17:** *The rebuffering events that happened during using our SBA mode with animation video.*

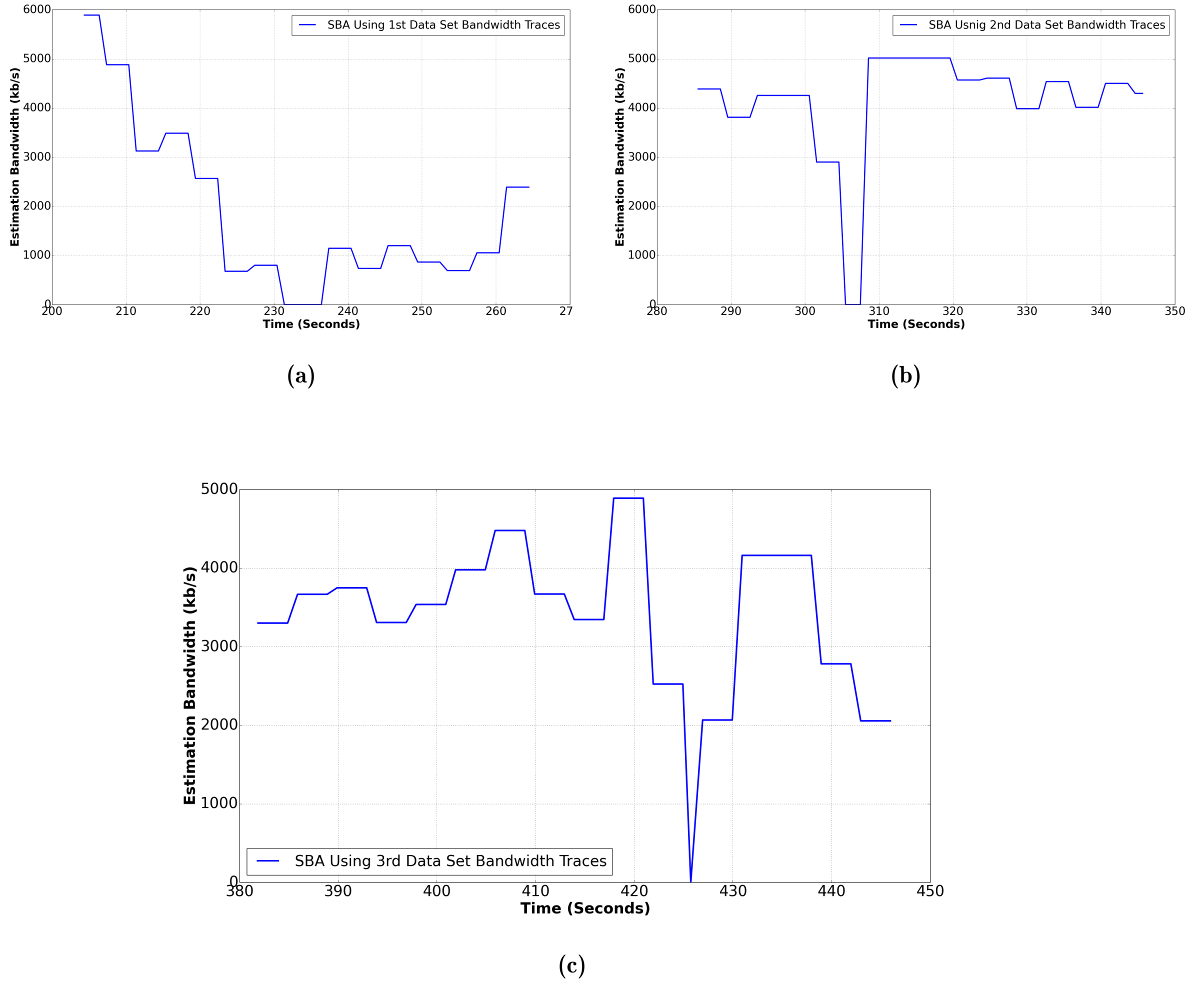



**Figure 5.18:** *The estimation bandwidth by the SBA (SSIM-Mode) of VQBA framework.*

(SSIM-Mode) of VQBA framework using the first, second and third bandwidth traces when the rebuffering events that occurred.

### 5.4.5 Phenonema under BBA, FESTIVE and OSMF

As a contrast, we provide the behavior of BBA, FESTIVE and OSMF for the same traces.

#### 5.4.5.1 Behavior with BBA

Figure 5.19 through mini-figures (a), (b) and (c) illustrate the behavior of the BBA adaptation algorithm with the same traces, at the same time range where rebuffering occurs with SBA. We observed at there is no rebuffering event, but there is some rates instability for the trace 1 (1050, 1750 and 2350 kbps) and trace 2 (2350 and 3000 kbps).

Figure 5.20 through mini-figures (a), (b) and (c) show the estimation bandwidth by the BBA adaptation algorithm using the first, second and third bandwidth traces related to Figure 5.19.

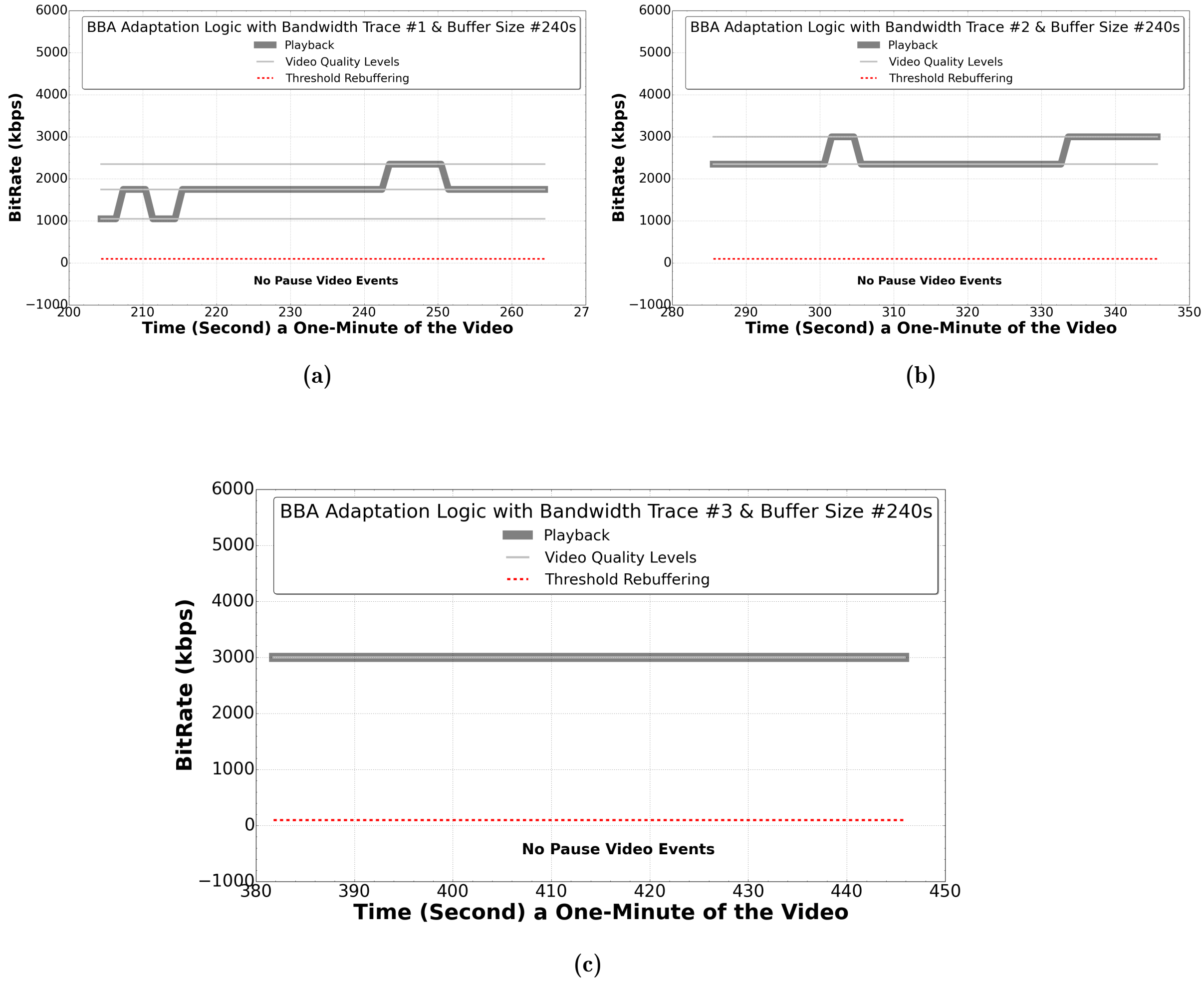
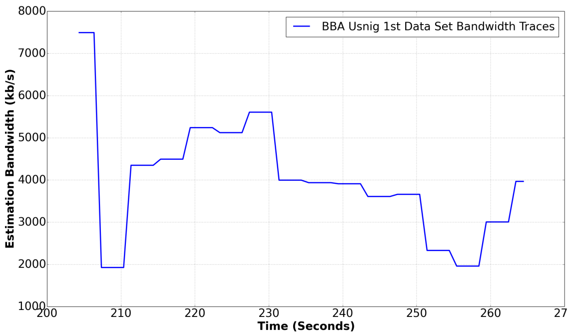
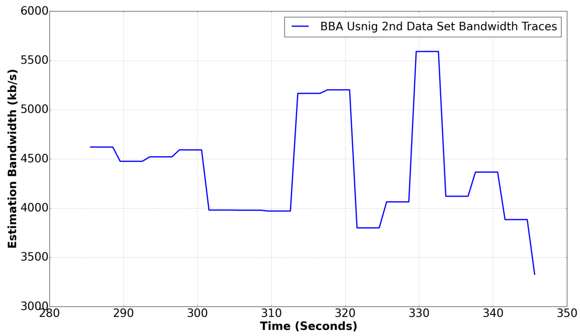


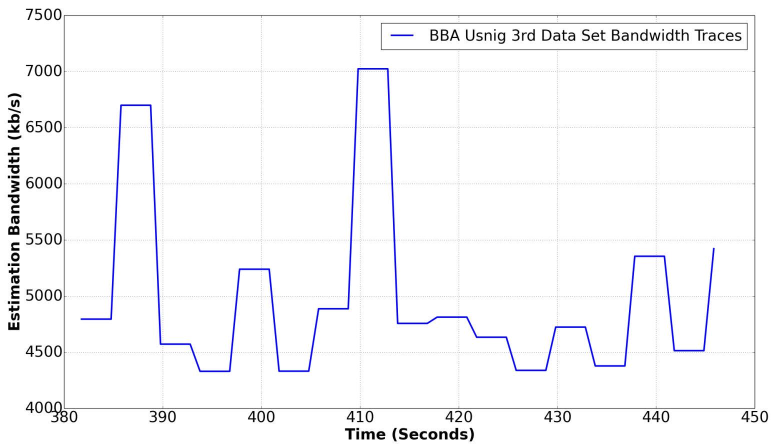

**Figure 5.19:** *The situation of video playback during using BBA adaptation algorithm at the same pauses places that happened during using our SBA mode.*




**(a)**

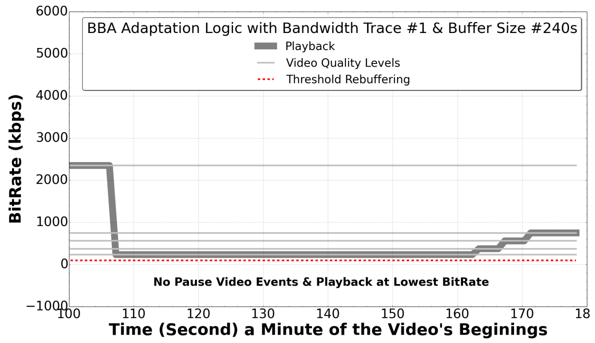



**(b)**

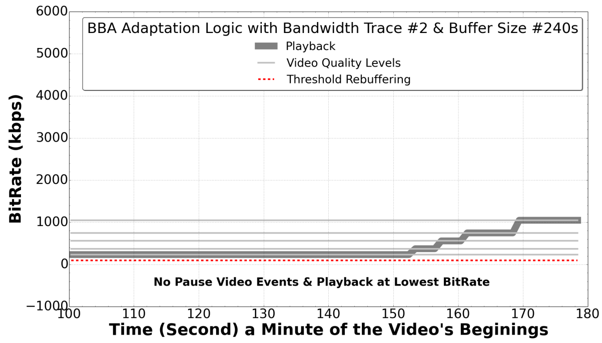



**(c)**

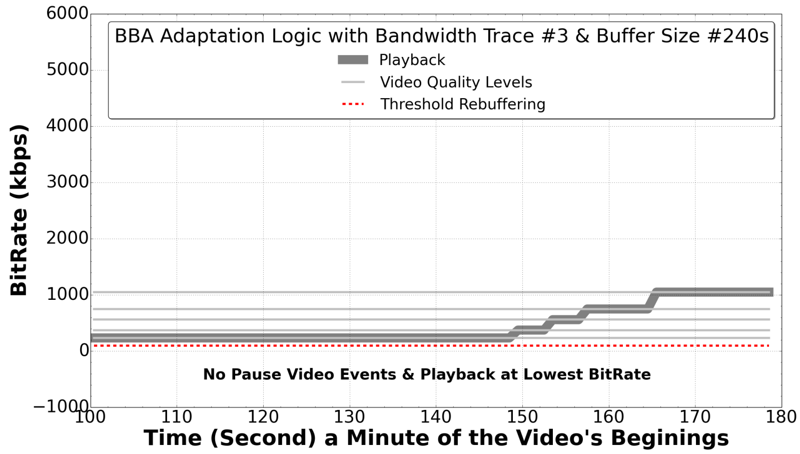
**Figure 5.20:** *The estimation bandwidth by the BBA adaptation algorithm.*




**(a)**




**(b)**




**(c)**

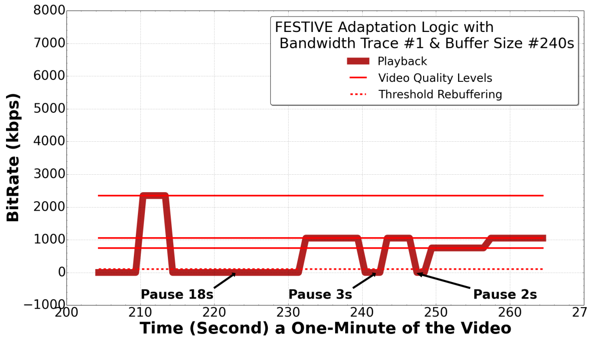
**Figure 5.21:** *The situation of video playback at the beginning the video session during using BBA adaptation algorithm.*

The Figure 5.21 through mini-figures (a), (b) and (c) show the situation at the begining of the video playback, where the bitrate is kept at a low level.

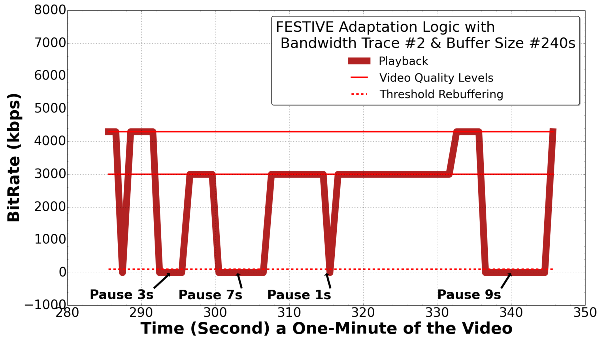
#### 5.4.5.2 Behavior with FESTIVE

Figure 5.22 through mini-figures (a), (b) and (c) illustrate the behavior of the FESTIVE algorithm with the same traces, at the same time range where rebuffering occurs with SBA. We observe that the rebuffering is much more frequent than for SBA. Actually, the rebuffering events happened (3, 5 and 5) times for { (18, 3 and 2), (3, 7, 1 and 9) and (3, 3, 6, 3 and 6) } seconds respectively for the first, second and third set of bandwidth traces.

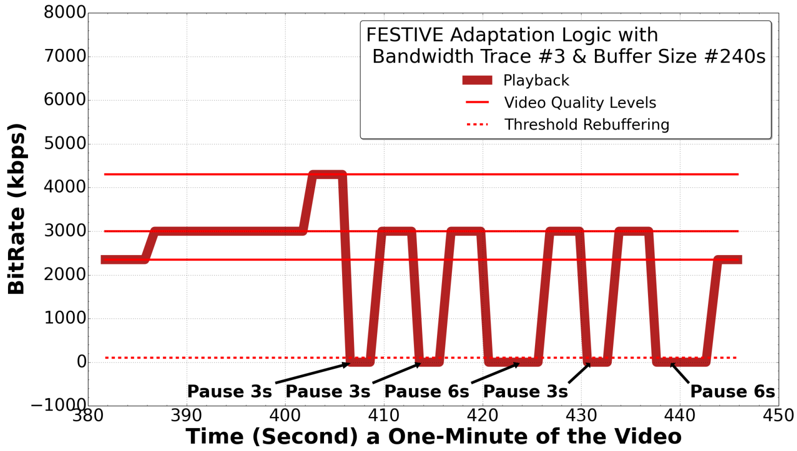
Figure 5.23 through mini-figures (a), (b) and (c) show the estimation bandwidth by the FESTIVE adaptation algorithm using the first, second and third bandwidth traces related to Figure 5.22.




(a)




(b)




(c)

**Figure 5.22:** *The situation of video playback during using FESTIVE adaptation algorithm at the same pauses places that happened during using our SBA mode.*

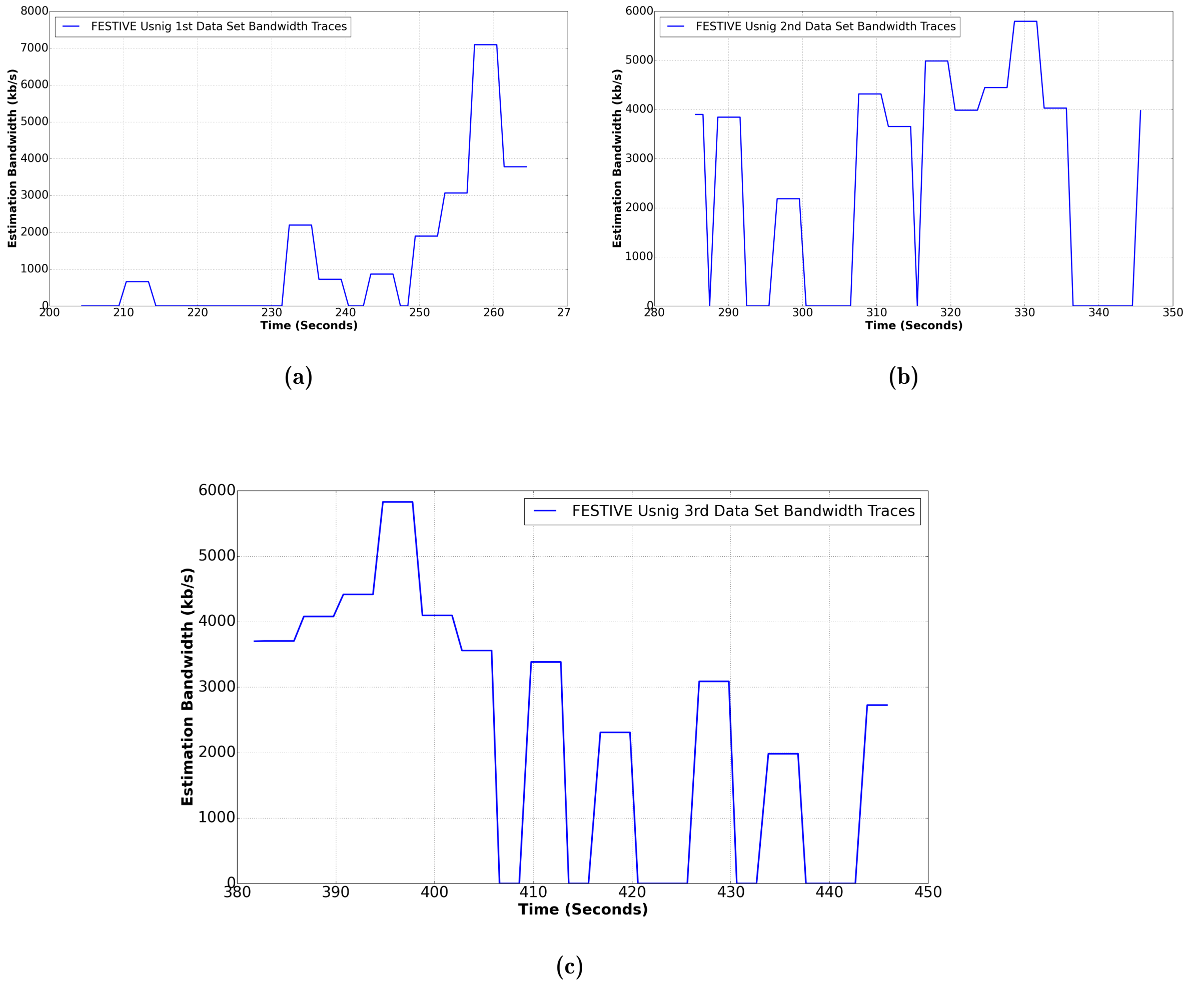
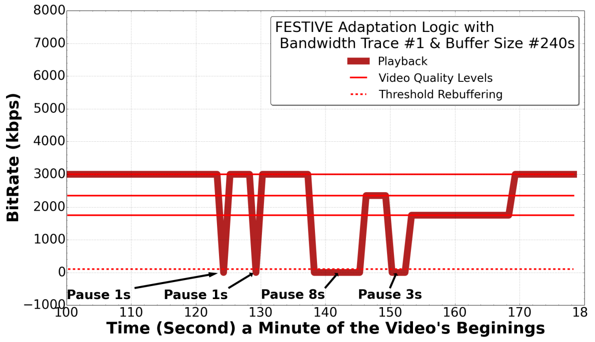
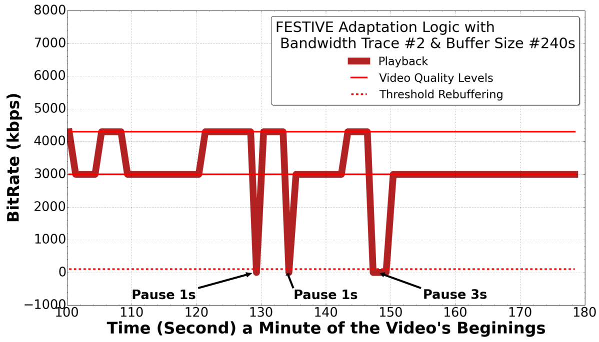


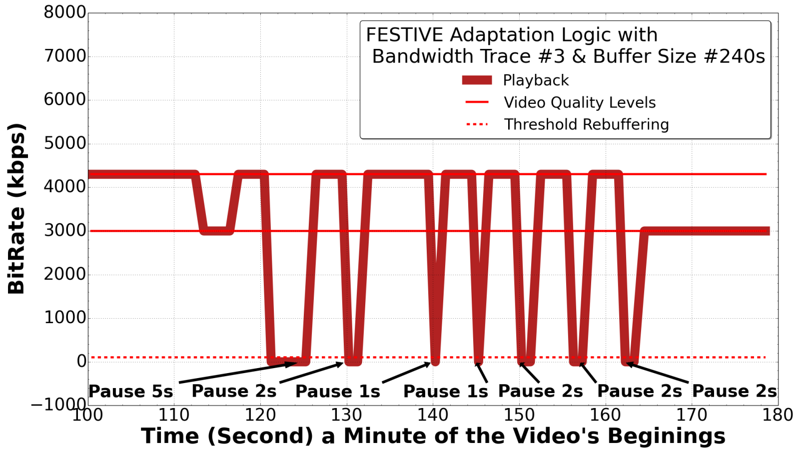

**Figure 5.23:** *The estimation bandwidth by the FESTIVE adaptation algorithm.*




(a)




(b)




(c)

**Figure 5.24:** *The situation of video playback at the beginning the video session during using FESTIVE adaptation algorithm.*

Figure 5.24 through mini-figures (a), (b) and (c) give the behavior of FESTIVE at the beginning of the video playback, where we can observe also frequent bitrate switching.

#### 5.4.5.3 Behavior with OSMF

Figure 5.25 through mini-figures (a), (b) and (c) illustrated the behavior of the OSMF algorithm with the same traces, at the same time range where rebuffering occur with SBA. We observe both frequent rebuffering and bitrate switching.

Figure 5.26 through mini-figures (a), (b) and (c) show the estimation bandwidth by the OSMF adaptation algorithm using the first, second and third bandwidth traces related to Figure 5.25.

Figure 5.27 through mini-figures (a), (b) and (c) give the behavior of OSMF at the beginning of the video playback, where we can observe also frequent bitrate switching.

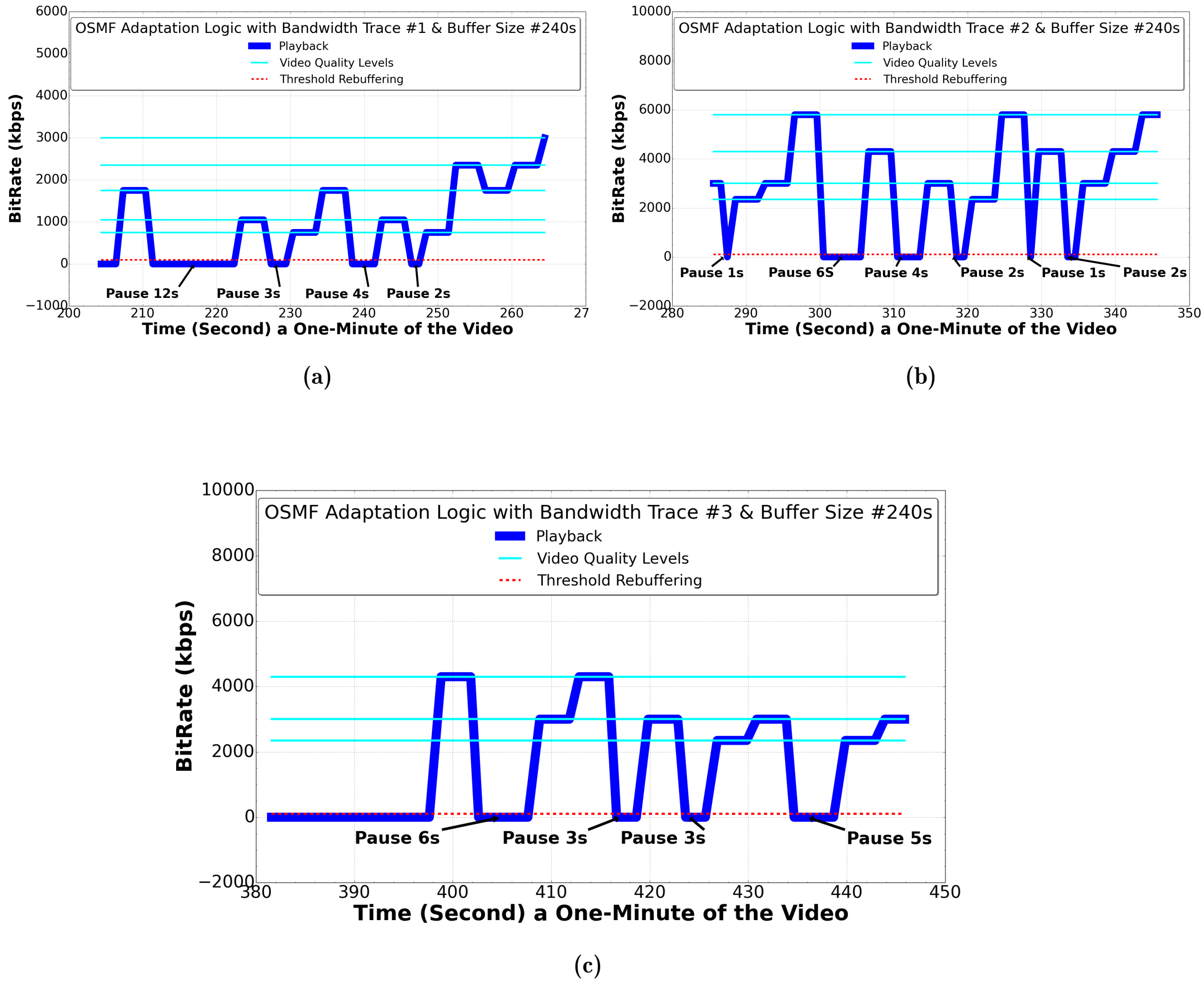
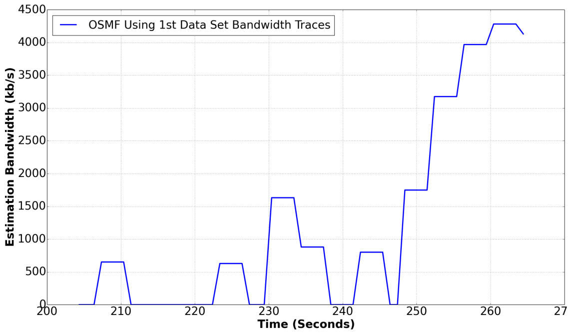
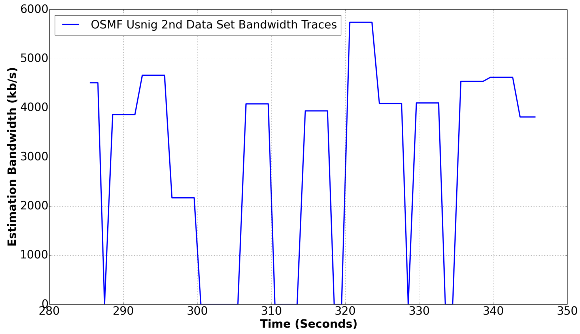


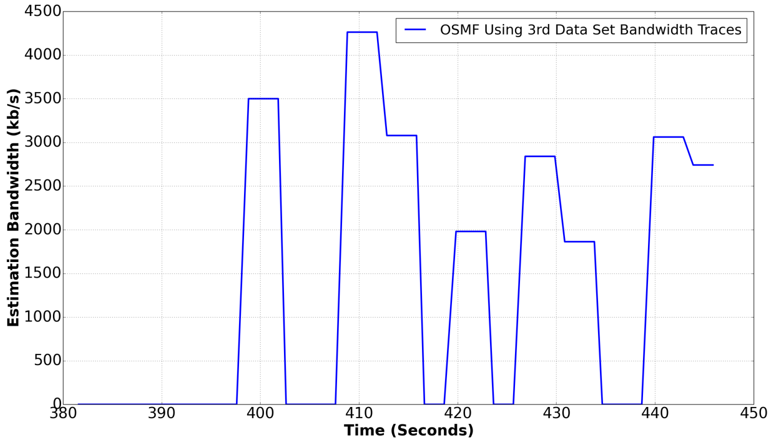

**Figure 5.25:** *The situation of video playback during using OSMF adaptation algorithm at the same pauses places that happened during using our SBA mode.*




(a)

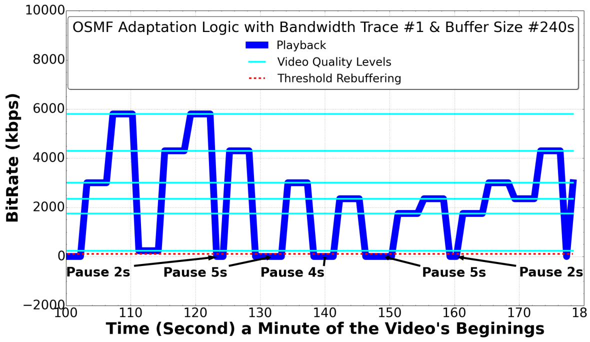



(b)

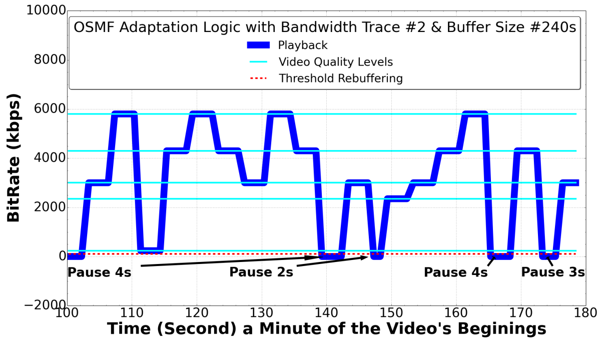



(c)

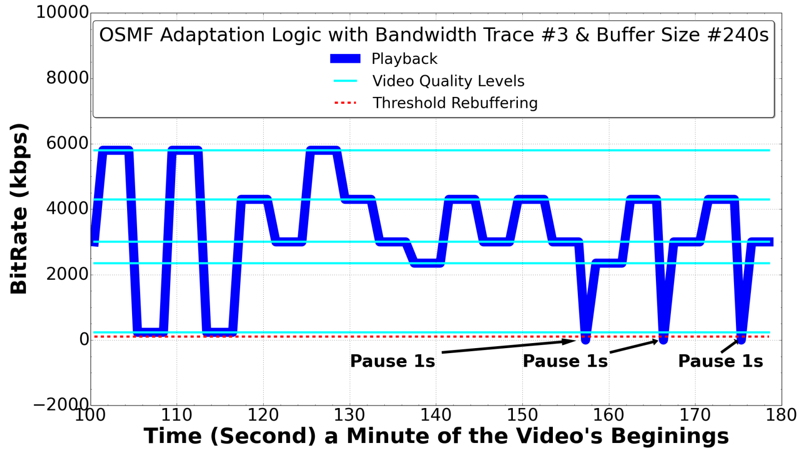
**Figure 5.26:** *The estimation bandwidth by the OSMF adaptation algorithm.*




(a)




(b)




(c)

**Figure 5.27:** *The situation of video playback at the beginning the video session during using OSMF adaptation algorithm.*

#### 5.4.5.4 Summary comparison

We summarize the here-before detailled results through two figures.

Figure 5.28 gives the *cumulative* measures of rebuffering duration for the 10 tested traces. This shows clearly the bad behavior of FESTIVE and OSMF.

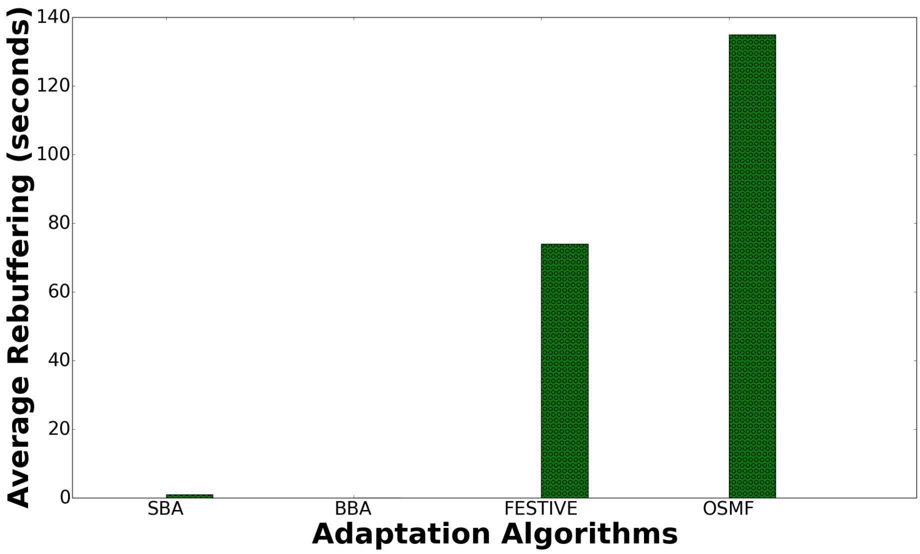



**Figure 5.28:** *The average of rebuffering events during a video session.*

Figure 5.29 gives the respective duration of frozen and unfrozen video playback.

### 5.4.6 Conclusion of focused study on rebuffering

One of the most critical video streaming metrics versus QoE is the rebuffering duration for each event. These detailed and focused comparative studies confirm one of the design goal of our framework: our SBA has a better behaviour on the rebuffering phenomenon versus FESTIVE and OMSF.

## 5.5 Conclusion

This chapter, together with the previous chapter, confirm the validity and efficiency of our framework, which consists in giving a prominent role to the video-quality metric as a key parameter for video streaming adaptation.

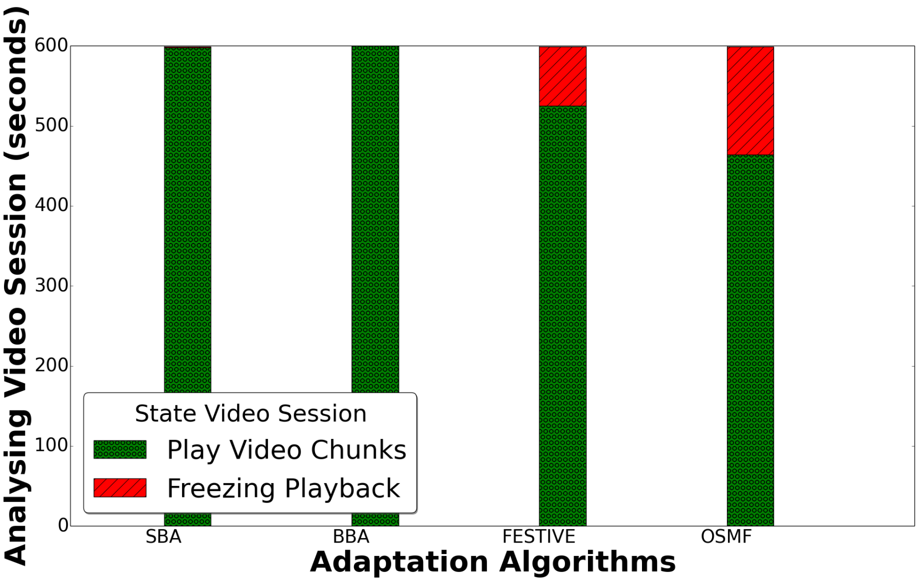



**Figure 5.29:** *Analysis of the video playback for video streaming sessions.*

CHAPTER 6

# Conclusion and future work

In this thesis, we presented a novel framework for video adaptation mechanism (ABR) for the DASH based video streaming.

This work is mainly motivated by our believe on the the prominent role that objective video quality metric should and could play in ABR mechanism.

After analysis of the existing works (Chapter 2), we proposed a generic framework for DASH adaptation, termed as Video-Quality Metrics Based-Adaptation algorithm (VQBA) (cf. § 4.3). This framework is designed for any objective Video-Quality-Metrics (VQM). In practice, we have tested it with the following metrics: SSIM, PSNR and VMAF.

In order to validate this framework, and also to access its performance, we conducted numerous experimental studies with real network traffic traces and test video sequences (cf. § 4.4). The experimentation is also carried with

i. We first apply the VQBA framework to SSIM metric, termed as SSIM Based Adaptation (SBA). This first set of results confirm our design goal with the SSIM metric (cf. § 4.5).

ii. We also investigated the choice of SSIM threshold (cf. § 4.6).

iii. We enlarge our experimental studies with a second set of network traffic traces, namely the (Ghent, Belgium) 4G network traces, to check the performance of the SBA (cf. § 4.7).

iv. We then extend our studies to the use of PSNR and VMAF video quality metrics as adaptation metrics, namely the PBA and VBA mechanism. We conducted comparative studies among them (SBA, PBA and VBA) (cf. § 5.1). We also made a more large comparative study between SBA and PBA against various QoE metrics and several non video-quality-aware adaptation ABR mechanisms (cf. § 5.2)

v. We also made a focused study on the rebuffering phenoma under SBA (§ 5.4)

We tested our generic framework VQBA using some objective VQM such as (SSIM, PSNR and VMAF), where our studies were conducted with comparison to some non video-quality-aware ABR (BBA, FESTIVE and OSMF). These studies show that our framework does achieve our design objectives.

The demand for mobile video streaming services is increasing, and video end-user expectations for the good viewing experience QoE it also goes up.

At the time that this thesis ends, under COVID-19, video streaming is taking a more and more prominent place in our everyday life.

This thesis focuses on the mechanism of ABR streaming algorithms, with the general idea of giving objective video quality metric a prominent role in the adaptation mechanism. Our experimental results validated this approach with three metrics, namely SSIM, PSNR and VMAF.

One direction of our future consists in exploring this roadmap, which we believe to be promising, with deeper studies with these usual metrics under more video sequences and/or networking contexts. We are also identifying other candidate video quality metrics.

We hope the design principle derived from this thesis will contribute to deal with the increasingly challenging environment in the future.

CHAPTER 7

# Appendix

> *Success is not final, failure is not fatal: it is the courage to continue that counts.* — WINSTON CHURCHILL.

## 7.1 Capture of Mobile Network Traffic

In order to obtain a realistic networking case, we have captured several real traffic traces over the 4G mobile network of a big network provider. The bandwidth traces were collected from different areas and periods in Paris to insure a large coverage of the traffic pattern. Figure 7.1 illustrate one of such traces.

For this, we used the `iPerf` tool with the command: `iperf3 -c iperf.scottlinux.com`.

Typical $iPerf$ output contains a time-stamped report of the amount of data transferred and the throughput measured, under the following form:

| [ID] | Interval | Transfer | Bandwidth |
|---|---|---|---|
| [4] | 0.00-1.00sec | 65.4 KBytes | 534 Kbits/sec |

where:

i. Interval: specifies the time duration for which the data is captured.

ii. Transfer: the captured (transferred) data's size.

iii. Bandwidth: the rate at which the data is transferred.

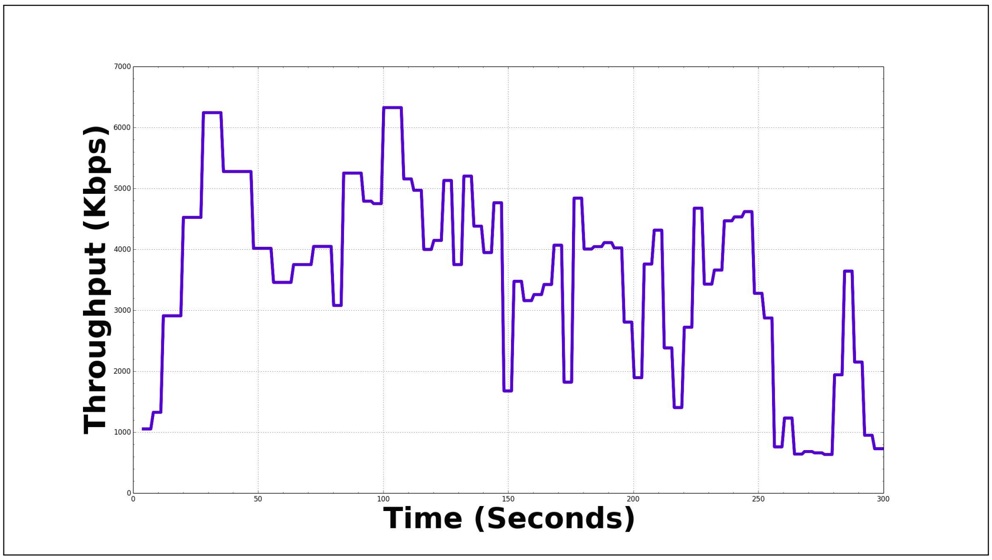



**Figure 7.1:** *The variation of throughput of one set that we collected from different sets of bandwidth traces.*

## 7.2 Video Streaming Source Preparation

In what follows, we will describe the steps that we followed to prepare the video sequences we used for our tests.

### 7.2.1 Video Sequences

Recall that we have chosen these three videos **Animation** (*Big Buck Bunny*, 9min56sec) [108], **Documentary** (*Of Forests and Men*, 7min33sec) [109] and **Sport** (*The World's Best Bouldering in Rocklands, South Africa*, 13min18sec) [110]. These ones are available at high resolution (1920x1080). We downloaded these high resolution version as our basic source.

### 7.2.2 Multi-resolution Encoding

We then used the `FFmpeg` encoder to recode the source video at ten resolutions different using the set of <bitrate, resolution> shown in Table 7.1, which is the one used by Netflix.

More specifically, these videos are encoded at 24 frames/s (FPS) by using the H.264 codec of FFMPEG. We give below the exact commands for the *Big Buck Bunny* sequence:

```
ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 235k -s 320*240 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-235k.mp4

ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 375k -s 384*288 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-375k.mp4
```

**Table 7.1:** *Netflix Resolutions*

| Bitrate (kbps) | Resolutions |
|---|---|
| 235 | 320*240 |
| 375 | 384*288 |
| 560 | 512*384 |
| 750 | 512*384 |
| 1050 | 640*480 |
| 1750 | 720*480 |
| 2350 | 1280*720 |
| 3000 | 1280*720 |
| 4300 | 1920*1080 |
| 5800 | 1920*1080 |

```
ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 560k -s 512*384 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-560k.mp4

ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 750k -s 512*384 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-750k.mp4

ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 1050k -s 640*480 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-1050k.mp4

ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 1750k -s 720*480 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-1750k.mp4

ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 2350k -s 1280*720 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-2350k.mp4
```

```
ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 3000k -s 1280*720 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-3000k.mp4
```

```
ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 4300k -s 1920*1080 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-4300k.mp4
```

```
ffmpeg -i Big_buck_Bunny_Video_Source.mp4 -an -b:v 5800k -s 1920*1080 -vcodec libx264
-x264opts 'keyint=24:min-keyint=24:no-scenecut' -r 24 bbb-5800k.mp4
```

These commandes produced the following 10 files

```
bbb-235k.mp4, bbb-375k.mp4, bbb-560k.mp4, bbb-750k.mp4, bbb-1050k.mp4,
bbb-1750k.mp4, bbb-2350k.mp4, bbb-3000k.mp4, bbb-4300k.mp4, bbb-5800k.mp4
```

### 7.2.3 Making of chunks

We then proceed to cut each video sequence into chunks (each chunk has a duration of 4 seconde) [113]. For this, we use the `MP4Box` sofware, with the following command:

```
MP4Box -dash 4000 -rap -bs-switching no -profile live -out bbb.mpd
bbb-235k.mp4 bbb-375k.mp4 bbb-560k.mp4 bbb-750k.mp4 bbb-1050k.mp4
bbb-1750k.mp4 bbb-2350k.mp4 bbb-3000k.mp4 bbb-4300k.mp4 bbb-5800k.mp4
```

Once these steps are done, we save the different chunks in a folder

We developed a python script to get and store the size of each chunk. Figure 7.2 illustrates chunk sizes at three different bitrate levels (235, 560 and 750 kbps) for the video sequence *Big Buck Bunny*

### 7.2.4 Objective video quality computation

The last step is to compute the Objective video quality metric associated with each chunk at each bitrate level.

For SSIM, we proceed as follows. We compare, for each chunk at each bitrate level, the original video and the encoded one. As an example, the command for *Big_buck_Bunny* is

```
qpsnr -a avg_ssim -o blocksize=16:fps=24 -r Big_buck_Bunny_Video_Source.mp4 bbb-235k.mp4
bbb-375k.mp4 bbb-560k.mp4 bbb-750k.mp4 bbb-1050k.mp4 bbb-1750k.mp4 bbb-2350k.mp4
bbb-3000k.mp4 bbb-4300k.mp4 bbb-5800k.mp4 > bbb_SSIM.csv
```

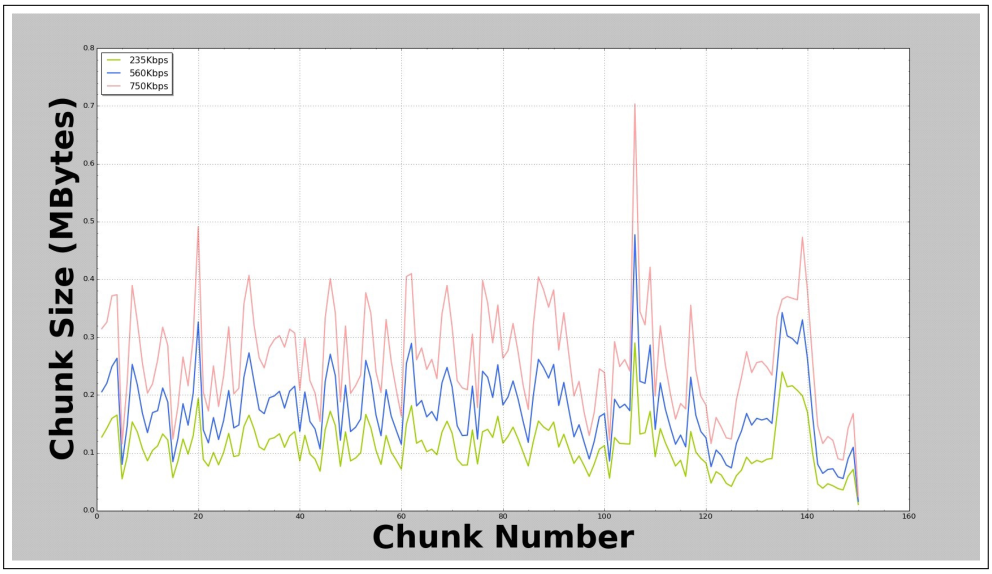



**Figure 7.2:** *Chunk size at three bitrate levels (235, 560 and 750 kbps for the video sequence* Big Buck Bunny

We get, for each chunk at each bitrate level, a value between 0 and 1. The closer the value is to 1, the closer the quality of the encoded video to the original one. These results will also be stored to be explored by our adaptation framework.

The result of the order next is saved in the file "ssimtest.csv" in the form of a table in which records the value of the correlation between the original frame (24 frames) and the frame of the video encoded at all times for the 10 resolutions:

Figure 7.3 illustrates SSIM for three different bitrate levels (235, 560 and 5800 kbps) of the sequence *Big Buck Bunny*

The PSNR values are obtained with similar commands:

```
qpsnr -a avg_psnr -o blocksize=16:fps=24 -r Big_buck_Bunny_Video_Source.mp4 bbb-235k.mp4
bbb-375k.mp4 bbb-560k.mp4 bbb-750k.mp4 bbb-1050k.mp4 bbb-1750k.mp4 bbb-2350k.mp4
bbb-3000k.mp4 bbb-4300k.mp4 bbb-5800k.mp4 > bbb_PSNR.csv
```

For VMAF, we used the following tool and command

```
vmafossexec.exe yuv420p 1920 1080 "C:\Users\Mustafa OTHMAN\Downloads\bbb.yuv"
"C:\Users\Mustafa OTHMAN\Downloads\bbb3000.yuv"
"C:\Users\Mustafa OTHMAN\Downloads\vmaf-master\vmaf
master\model\vmaf_v0.6.1.pkl" --psnr --ssim --ms-ssim --log  MZ3000.csv
```

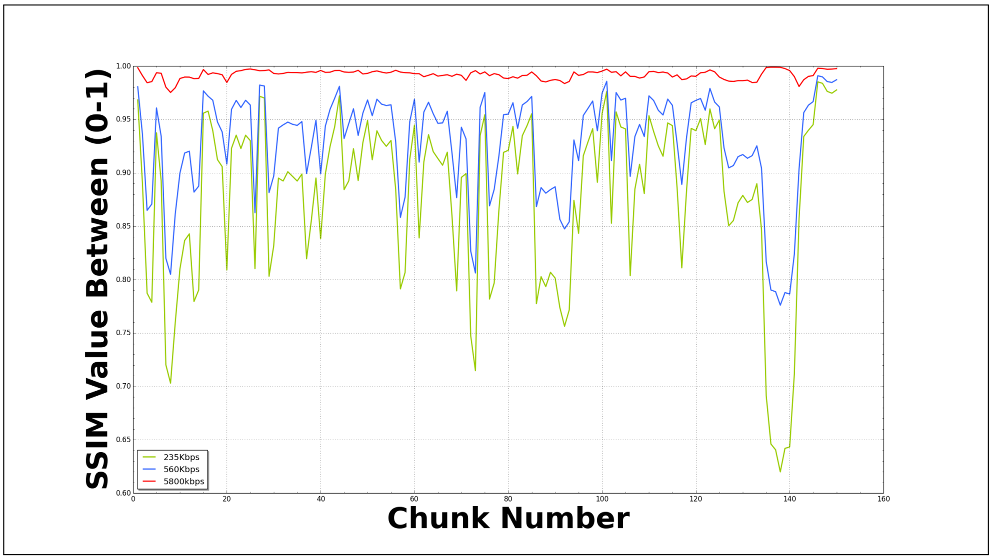



**Figure 7.3:** *SSIM metric of three different copy of the Video (235, 560 and 5800) kbps.*

# Bibliography


[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (vni), complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018.

[2] F. WANG, Z. FEI, J. WANG, Y. LIU, and Z. WU, "Has qoe prediction based on dynamic video features with data mining in lte network," *Information Sciences*, vol. 60, no. 042404, pp. 1–042 404, 2017.

[3] F. Hartung, S. Kesici, and D. Catrein, "Drm protected dynamic adaptive http streaming," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 277–282.

[4] Z. Wang, "Applications of objective image quality assessment methods [applications corner]," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 137–142, 2011.

[5] Z. Zhu, S. Li, and X. Chen, "Design qos-aware multi-path provisioning strategies for efficient cloud-assisted svc video streaming to heterogeneous clients," *IEEE transactions on multimedia*, vol. 15, no. 4, pp. 758–768, 2013.

[6] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, vol. 52, no. 2, pp. 587–600, 2013.

[7] H. Nam, K.-H. Kim, and H. Schulzrinne, "Qoe matters more than qos: Why people stop watching cat videos," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[8] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[9] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 362–373, 2011.

[10] M. Othman, K. Chen, and A. Mokraoui, "A user-experience driven ssim-aware adaptation approach for dash video streaming," CENTRIC 2019 (SoftNet 2019), Valencia, Spain, 2019.

[11] ——, "A study of qoe-aware adaptation mechanism for dash video streaming based on objective visual quality assessment," in *2020 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE, 2020, pp. 81–86.

[12] I. Recommendation, "E. 800: Terms and definitions related to quality of service and network performance including dependability," *ITU-T August*, vol. 1994, 1994.

[13] I. T. S. Sector and O. ITU, "Quality of experience requirements for iptv services," *ITU-T Recommendation G*, vol. 1080, 2008.

[14] P. Le Callet, S. Möller, A. Perkis *et al.*, "Qualinet white paper on definitions of quality of experience," *European network on quality of experience in multimedia systems and services (COST Action IC 1003)*, vol. 3, no. 2012, 2012.

[15] I. Recommendation, "1203.3,"parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport-quality integration module,"," *International Telecommunication Union*, 2017.

[16] D. Tsolkas, E. Liotou, N. Passas, and L. Merakos, "A survey on parametric qoe estimation for popular services," *Journal of network and computer applications*, vol. 77, pp. 1–17, 2017.

[17] B. García, M. Gallego, F. Gortázar, and A. Bertolino, "Understanding and estimating quality of experience in webrtc applications," *Computing*, vol. 101, no. 11, pp. 1585–1607, 2019.

[18] R. Trestian, I.-S. Comsa, and M. F. Tuysuz, "Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?" *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 945–977, 2018.

[19] S. Yuan, T. Lin, S. Bai, and S. Ci, "User-oriented qoe-driven server selection for multimedia service provisioning in content distribution networks," in *2015 IEEE 26th Annual Inter-*

*national Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 2077–2082.

[20] P. Brooks and B. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," *IEEE network*, vol. 24, no. 2, pp. 8–13, 2010.

[21] K. De Moor, I. Ketyko, W. Joseph, T. Deryckere, L. De Marez, L. Martens, and G. Verleye, "Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting," *Mobile Networks and Applications*, vol. 15, no. 3, pp. 378–391, 2010.

[22] K. U. R. Laghari and K. Connelly, "Toward total quality of experience: A qoe model in a communication ecosystem," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 58–65, 2012.

[23] Y. Chen, K. Wu, and Q. Zhang, "From qos to qoe: A tutorial on video quality assessment," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2014.

[24] L. Skorin-Kapov and M. Varela, "A multi-dimensional view of qoe: the arcu model," in *2012 Proceedings of the 35th International Convention MIPRO*. IEEE, 2012, pp. 662–666.

[25] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi *et al.*, "Qualinet white paper on definitions of quality of experience," 2013.

[26] P. Reichl, S. Egger, S. Möller, K. Kilkki, M. Fiedler, T. Hoßfeld, C. Tsiaras, and A. Asrese, "Towards a comprehensive framework for qoe and user behavior modelling," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2015, pp. 1–6.

[27] N. Barman and M. G. Martini, "Qoe modeling for http adaptive video streaming–a survey and open challenges," *IEEE Access*, vol. 7, pp. 30 831–30 859, 2019.

[28] M. T. Diallo, "Quality of experience and video services adaptation," Ph.D. dissertation, Institut National des Télécommunications, 2015. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01239849

[29] I. Martinez-Yelmo, I. Seoane, and C. Guerrero, "Fair quality of experience (qoe) measurements related with networking technologies," in *International Conference on Wired/Wireless Internet Communications*. Springer, 2010, pp. 228–239.

[30] C. Wang, H. Kim, and R. Morla, "Qoe driven server selection for vod in the cloud," in *2015 IEEE 8th International Conference on Cloud Computing.* IEEE, 2015, pp. 917–924.

[31] K. Piamrat, C. Viho, J.-M. Bonnin, and A. Ksentini, "Quality of experience measurements for video streaming over wireless networks," in *2009 Sixth International Conference on Information Technology: New Generations.* IEEE, 2009, pp. 1184–1189.

[32] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal processing: Image communication*, vol. 19, no. 2, pp. 121–132, 2004.

[33] W. Song and D. W. Tjondronegoro, "Acceptability-based qoe models for mobile video," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 738–750, 2014.

[34] I. Rec, "P. 800: Methods for subjective determination of transmission quality," *International Telecommunication Union, Geneva*, p. 22, 1996.

[35] Z. Akhtar and T. H. Falk, "Audio-visual multimedia quality assessment: A comprehensive survey," *IEEE Access*, vol. 5, pp. 21 090–21 117, 2017.

[36] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE transactions on broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.

[37] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, 2010.

[38] K. Okarma, "Video quality assessment using the combined full-reference approach," in *Image Processing and Communications Challenges 2.* Springer, 2010, pp. 51–58.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[40] A. Zvezdakova, S. Zvezdakov, D. Kulikov, and D. Vatolin, "Hacking vmaf with video color and contrast distortion," *arXiv preprint arXiv:1907.04807*, 2019.

[41] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-dash: A deep q-learning framework for dash video streaming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 703–718, 2017.

[42] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.

[43] M. Rohani, A. N. Avanaki, S. Nader-Esfahani, and M. Bashirpour, “A reduced reference video quality assessment method based on the human motion perception,” in *2010 5th International Symposium on Telecommunications*. IEEE, 2010, pp. 831–835.

[44] R. Soundararajan and A. C. Bovik, “Rred indices: Reduced reference entropic differencing for image quality assessment,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 517–526, 2011.

[45] ——, “Video quality assessment by reduced reference spatio-temporal entropic differencing,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 684–694, 2012.

[46] D. Brunet, E. R. Vrscay, and Z. Wang, “On the mathematical properties of the structural similarity index,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1488–1499, 2011.

[47] S. Cicalo, N. Changuel, R. Miller, B. Sayadi, and V. Tralli, “Quality-fair http adaptive streaming over lte network,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 714–718.

[48] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, “Towards network-wide qoe fairness using openflow-assisted adaptive video streaming,” in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. ACM, 2013, pp. 15–20.

[49] S. Pudlewski, A. Prasanna, and T. Melodia, “Compressed-sensing-enabled video streaming for wireless multimedia sensor networks,” *IEEE Transactions on Mobile Computing*, no. 6, pp. 1060–1072, 2012.

[50] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hoßfeld, “Impact of frame rate and resolution on objective qoe metrics,” in *2010 second international workshop on quality of multimedia experience (QoMEX)*. IEEE, 2010, pp. 29–34.

[51] T. Zinner, O. Abboud, O. Hohlfeld, T. Hossfeld, and P. Tran-Gia, “Towards qoe management for scalable video streaming,” in *21th ITC Specialist Seminar on Multimedia Applications-Traffic, Performance and QoE*. Citeseer, 2010, pp. 64–69.

[52] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2366–2369.

[53] S. Winkler and P. Mohandas, "The evolution of video quality measurement: From psnr to hybrid metrics," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, 2008.

[54] A.-N. Moldovan, I. Ghergulescu, and C. H. Muntean, "A novel methodology for mapping objective video quality metrics to the subjective mos scale." IEEE, 2014, pp. 1–7.

[55] ——, "Vqamap: A novel mechanism for mapping objective video quality metrics to subjective mos scale," *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 610–627, 2016.

[56] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, vol. 6, 2016.

[57] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on image processing*, vol. 15, no. 2, pp. 430–444, 2006.

[58] S. Li, F. Zhang, L. Ma, and K. N. Ngan, "Image quality assessment by separately evaluating detail losses and additive impairments," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 935–949, 2011.

[59] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2014.

[60] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 187–198.

[61] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 2001–2014, 2013.

[62] A. Biernacki and K. Tutschku, "Performance of http video streaming under different network conditions," *Multimedia Tools and Applications*, vol. 72, no. 2, pp. 1143–1166, 2014.

[63] T. Stockhammer, “Dynamic adaptive streaming over http– standards and design principles,” in *Proceedings of the second annual ACM conference on Multimedia systems*, 2011, pp. 133–144.

[64] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, “Deriving and validating user experience model for dash video streaming,” *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 651–665, 2015.

[65] A. El Essaili, D. Schroeder, D. Staehle, M. Shehada, W. Kellerer, and E. Steinbach, “Quality-of-experience driven adaptive http media delivery,” in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 2480–2485.

[66] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, “Sdnhas: An sdn-enabled architecture to optimize qoe in http adaptive streaming,” *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, 2017.

[67] A. Bentaleb, A. C. Begen, and R. Zimmermann, “Sdndash: Improving qoe of http adaptive streaming using software defined networking,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1296–1305.

[68] R. Pereira and E. G. Pereira, “Dynamic adaptive streaming over http and progressive download: Comparative considerations,” in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2014, pp. 905–909.

[69] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang, “Vivisecting youtube: An active measurement study,” in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2521–2525.

[70] C. B. Ameur, “Tcp protocol optimization for http adaptive streaming,” Ph.D. dissertation, Rennes 1, 2015.

[71] M. Allman, V. Paxson, W. Stevens *et al.*, “Tcp congestion control,” 1999.

[72] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 43–56, 2000.

[73] D. Z. Rodríguez, R. L. Rosa, E. C. Alfaia, J. I. Abrahão, and G. Bressan, “Video quality metric for streaming service using dash standard,” *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 628–639, 2016.

[74] D. Z. Rodríguez, R. L. Rosa, and G. Bressan, "No-reference video quality metric for streaming service using dash standard," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2015, pp. 106–107.

[75] Z. Duanmu, A. Rehman, and Z. Wang, "A quality-of-experience database for adaptive video streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 474–487, 2018.

[76] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 157–168.

[77] S. Zhao, Z. Li, D. Medhi, P. Lai, and S. Liu, "Study of user qoe improvement for dynamic adaptive streaming over http (mpeg-dash)," in *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2017, pp. 566–570.

[78] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "Qdash: a qoe-aware dash system," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012, pp. 11–22.

[79] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, and L. Martens, "Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching," *IEEE Transactions on Broadcasting*, vol. 59, no. 1, pp. 47–61, 2012.

[80] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 339–350, 2013.

[81] S. van Kester, T. Xiao, R. E. Kooij, K. Brunnström, and O. K. Ahmed, "Estimating the impact of single and multiple freezes on video quality," in *Human vision and electronic imaging XVI*, vol. 7865. International Society for Optics and Photonics, 2011, p. 78650O.

[82] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *2012 Fourth International Workshop on Quality of Multimedia Experience*. IEEE, 2012, pp. 1–6.

[83] Y. Qi and M. Dai, "The effect of frame freezing and frame skipping on video quality," in *2006 international conference on intelligent information hiding and multimedia*. IEEE, 2006, pp. 423–426.

[84] T. N. Minhas and M. Fiedler, "Impact of disturbance locations on video quality of experience," in *EuroITV 2011 Workshop: Quality of Experience for Multimedia Content Sharing.* EuroITV 2011 Workshop: Quality of Experience for Multimedia Content Sharing, 2011.

[85] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in *2011 IEEE International Symposium on Multimedia.* IEEE, 2011, pp. 494–499.

[86] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. De Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, 2012.

[87] M. Azumi, T. Kurosaka, and M. Bandai, "A qoe-aware quality-level switching algorithm for adaptive video streaming," in *2015 IEEE Global Communications Conference (GLOBECOM).* IEEE, 2015, pp. 1–5.

[88] Y. Xu, Y. Zhou, and D.-M. Chiu, "Analytical qoe models for bit-rate switching in dynamic adaptive streaming systems," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2734–2748, 2014.

[89] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications.* IEEE, 2016, pp. 1–9.

[90] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 389–400.

[91] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "Buffer-based smooth rate adaptation for dynamic http streaming," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference.* IEEE, 2013, pp. 1–9.

[92] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[93] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when http adaptive streaming players compete for bandwidth?" in *Proceedings of the 22nd*

*international workshop on Network and Operating System Support for Digital Audio and Video.* ACM, 2012, pp. 9–14.

[94] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.

[95] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 97–108.

[96] M. Riad, H. Abu-Zeid, H. S. Hassanein, M. Tayel, and A. A. Taha, "A channel variation-aware algorithm for enhanced video streaming quality," in *Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th.* IEEE, 2015, pp. 893–898.

[97] "The Open Source Media Framework, year = 2020, note = URL: https://dzone.com/refcardz/getting-started-web-video?chapter=1/ [retrieved: 2020-12-21] ."

[98] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados, and P. Chatzimisios, "Fdash: A fuzzy-based mpeg/dash adaptation algorithm," *IEEE Systems Journal*, vol. 10, no. 2, pp. 859–868, 2015.

[99] D. J. Vergados, K. Kralevska, A. Michalas, and D. D. Vergados, "Evaluation of http/dash adaptation algorithms on vehicular networks," in *2018 Global Information Infrastructure and Networking Symposium (GIIS).* IEEE, 2018, pp. 1–5.

[100] M. Çalı and N. Özbek, "Ssim-based adaptation for dash with svc in mobile networks," *Signal, Image and Video Processing*, pp. 1–8, 2020.

[101] L. Huo, Z. Wang, M. Xu, Z. Ding, and X. Tao, "Optimizing qoe of multiple users over dash: A meta-learning approach," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 3032–3036.

[102] X. Chen, Y. Zhao, and Y. Li, "Qoe-aware wireless video communications for emotion-aware intelligent systems: A multi-layered collaboration approach," *Information Fusion*, vol. 47, pp. 1–9, 2019.

[103] C. G. Bampis and A. C. Bovik, "Learning to predict streaming video qoe: Distortions, rebuffering and memory," *arXiv preprint arXiv:1703.00633*, 2017.

[104] R. Rassool, "Vmaf reproducibility: Validating a perceptual practical video quality metric," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2017, pp. 1–2.

[105] C. G. Bampis, Z. Li, and A. C. Bovik, "Continuous prediction of streaming video qoe using dynamic networks," *IEEE Signal Processing Letters*, vol. 24, no. 7, pp. 1083–1087, 2017.

[106] "4g/lte bandwidth logs,". [Online]. Available: https://users.ugent.be/~jvdrhoof/dataset/

[107] J. Van Der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "Http/2-based adaptive streaming of hevc video over 4g/lte networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.

[108] "big buck bunny homepage,". [Online]. Available: https://peach.blender.org/

[109] "of forests and men,". [Online]. Available: http://www.offorestsandmen.org/

[110] "the world's best bouldering in rocklands, south africa,". [Online]. Available: http://www.lt11.com

[111] "ffmpeg homepage,". [Online]. Available: https://www.ffmpeg.org

[112] "netflix encoding bitrate,". [Online]. Available: https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2

[113] "gpac homepage,". [Online]. Available: https://gpac.wp.imt.fr

[114] R. K. Mok, E. W. Chan, X. Luo, and R. K. Chang, "Inferring the qoe of http video streaming from user-viewing activities," in *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack*, 2011, pp. 31–36.

[115] P. Juluri, "Measurement and improvement of quality-of-experience for online video streaming services," Ph.D. dissertation, 2015.

[116] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 272–285.

[117] X. K. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, X. Jin, J. Rexford, and R. K. Sinha, "Can accurate predictions improve video streaming in cellular networks?"

in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, 2015, pp. 57–62.

[118] R. R. Pastrana-Vidal, J. C. Gicquel, C. Colomes, and H. Cherifi, "Sporadic frame dropping impact on quality perception," in *Human Vision and Electronic Imaging IX*, vol. 5292. International Society for Optics and Photonics, 2004, pp. 182–193.

[119] M.-N. Garcia, F. De Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnström, and A. Raake, "Quality of experience and http adaptive streaming: A review of subjective studies," in *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2014, pp. 141–146.

[120] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik, "Study of temporal effects on subjective video quality of experience," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5217–5231, 2017.

[121] K. Spiteri, R. Sitaraman, and D. Sparacio, "From theory to practice: Improving bitrate adaptation in the dash reference player," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 2s, pp. 1–29, 2019.

[122] M. B. Yahia, Y. L. Louedec, G. Simon, L. Nuaymi, and X. Corbillon, "Http/2-based frame discarding for low-latency adaptive video streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 1, pp. 1–23, 2019.

**Titre :** **Mécanismes d'adaptation basés sur des métriques objectives de la qualité vidéo pour la diffusion de vidéo selon DASH**

**Mots clefs :** Streaming Video; QoE; ABR; DASH; SSIM; PSNR; VMAF; Réseaux mobiles.

**Résumé :** La norme DASH (Dynamic Adaptive Streaming over HTTP) est largement adoptée pour la diffusion de vidéo. Le mécanisme d'adaptation du style ABR (Adaptive BitRate), qui est un des composants clé de DASH, n'est pas normalisée, car il doit prendre divers éléments en compte, notamment le contexte de la communication et du système, mais également la qualité perçue par les usagers, pour maximiser la QoE (Quality of Experience). De nombreux algorithmes ABR ont été proposés. Peu d'entre eux accordent une importance à la qualité perçue, et objectivement calculée, comme paramètre d'adaptation. Cette thèse propose un cadre générique, nommé VQBA (Video Quality Metric Based Adaptation algorithm), permettant d'intégrer une métrique objective de la qualité vidéo de son choix comme paramètre d'adaptation. Le principe consiste à maximiser l'utilisation efficace de la bande passante disponible en décidant d'adopter un débit plus élevé non seulement parce qu'il est faisable, mais aussi parce que cela apporte une amélioration visuelle significative. Nous avons mené de nombreux tests avec des séquences vidéo de diverses natures et en les plaçant dans de vraies situations de réseaux avec des traces issues des réseaux mobiles opérationnels. Ces tests, à travers trois métriques usuelles de la qualité vidéo, nommément SSIM (Structural Similarity Index Measurement), PSNR (Peak Signal to Noise Ratio) et VMAF (Video Multimethod Assessment Fusion), et en comparaison avec une sélection d'algorithmes ABR, montrent que la voie que nous avons explorée, c'est-à-dire, accorder une importance à la qualité vidéo comme paramètre d'adaptation, est une voie efficace pour une meilleure QoE.

**Title :** **Objective video quality metric aware Adaptation mechanisms for video streaming based on DASH**

**Keywords :** Video Streaming; QoE; ABR; DASH; SSIM; PSNR; VMAF; Mobile Networks.

**Abstract :** The DASH (Dynamic Adaptive Streaming over HTTP) standard is widely adopted for video streaming. The Adaptive BitRate (ABR) style adaptation mechanism, which is a key component of DASH, is not standardized, since it must take various elements into account, in particular the context of the communication and the system, but also the quality perceived by the users, to maximize the QoE (Quality of Experience). Many ABR algorithms have been proposed. Few of them attach importance to perceived, and objectively calculated, quality as an adaptation parameter. This thesis proposes a generic framework, called VQBA (Video Quality Metric Based Adaptation algorithm), allowing to integrate an objective metric of the video quality of one's choice as an adaptation parameter. The idea is to maximize the efficient use of the available bandwidth by deciding to switch to a higher speed not only because it is feasible, but also because it provides a significant visual improvement. We carried out numerous tests with video sequences of various kinds and by placing them in real network situations with traces from operational mobile networks. These tests, through three usual video quality metrics, namely SSIM (Structural Similarity Index Measurement), PSNR (Peak Signal to Noise Ratio) and VMAF (Video Multimethod Assessment Fusion), and in comparison with a selection of ABR algorithms, show that the path we explored, that is to say, giving importance to video quality as an adaptation parameter, is an effective path for better QoE.

**École doctorale Galilée, Université Sorbonne Paris Nord,**

**Laboratoire de Traitement et Transport de l'Information (L2TI).**

Villetaneusse, France