Measuring and Optimizing Video Player Streaming Profile Algorithm.

Ing. G. BOUILLON
Ing. D. BOSCHLOOS
Ing. P. DEKIMPE

ECAM – Bruxelles

Le streaming Over The Top (OTT) est le moyen le plus utilisé pour accéder à du contenu media distant sur Internet. Au cours des dernières décennies, de nombreuses entreprises se sont concentrées sur l'amélioration de la qualité de leurs services pour leurs clients. Aujourd'hui, les attentes des utilisateurs en matière de streaming à distance sont énormes (rapidité de changement, qualité, disponibilité…) et l'optimisation du streaming OTT est plus que nécessaire. Ce travail se concentre sur la mesure et l'optimisation de l'actuelle solution de streaming proposée par Proximus (appelée Pickx), du point de vue du contenu vidéo mais aussi du système de sélection automatique de sa qualité.

Mots-clefs : informatique, algorithme, Proximus, Pickx, ABR, VMAF, Sabre, simulation, réseau, internet, vidéo, stream, streaming, OTT

Over The Top (OTT) Streaming is the most used way of consuming remote media content over the Internet. During past decades, a lot of companies focussed on increasing the quality of experience for their customers. Today, users' expectations for remote content streaming are huge (loading speed, quality, availability…) and OTT streaming optimisations are necessary. This work focuses on measuring and optimizing the current Proximus streaming solution (called Pickx), from a video content point of view to a player automatic quality selection system.

Keywords: IT, algorithm, Proximus, Pickx, ABR, VMAF, Sabre, simulation, network, internet, video, stream, streaming, OTT

1. Introduction

Online video streaming is becoming the most popular way of consuming remote content. By the end of 2022, according to *Cisco* studies [1], online video streaming will represent more than 82% of all consumer internet traffic. This is 15 times higher than in 2017 [1]. Video streaming has been gaining popularity and today end-users' expectations have strongly increased. The end-user Quality of Experience (QoE)[1] is a central concern for anyone who wants to do business with remote content streaming. Studies show that after clicking "play" in a player, roughly 6% of the audience leaves every second if the video doesn't start up in the two first seconds. This means that after 10 seconds of waiting, more than half of the audience has left [2]. Moreover, it is shown that a 1% rebuffering[2] during the playback time induces a watch time decrease of ±5%. A study made by *Conviva* found out that poor quality in 15 minutes videos causes a 33% of the viewers to depart immediately. After 10 minutes, less than 10% of viewers are still watching [3]. These quality problems decrease the QoE and thus discourage remote content streaming. This proves that quality level selection alongside content delivery speed are key features that all streaming content companies should pay attention to.

The goal of this work is to propose solutions to optimize *Proximus* video player speed and quality. Considering that *Proximus* has control over the entire end-to-end chain from player to network connectivity and stream encoding, solutions can be suggested across all these domains.

Today, automatic video quality selection algorithms perform quite well (in term of quality and responsiveness) especially thanks to the technology evolution and associated solutions that emerge each year. Optimizing such an algorithm would be outside the scope of this work as there is a lot of innovation in this domain and such a task would require further investigation given the wide range of possibilities. A first possible solution would thus be to choose one of the existing algorithms provided by related work (form research, papers, internet, etc.) and implement it to the *Proximus* streaming solution (called Pickx TV). The essential part here is to choose the correct one that fits with the Pickx use cases.

For *Proximus*, the solution cannot be limited to choose the best selection algorithm. The solution must work in their environment and must be compatible with their chosen players. Their current working solution needs to be analysed to improve some other streaming parameters such as the streaming content itself. In that case the so-

[1] Quality of Experience (QoE) is the measure of how easy it is to use a specific service.
[2] A rebuffering is a loading event that appears when the player buffer is empty. This means the consumer must wait for the player to download the video sequence.

lution would be more suitable for Pickx, and different teams may be interested. Solving a problem by the roots is often more effective and maintainable. In this case it means optimizing the way content is delivered starts with looking at how the content is created (encoded).

The structure of this article will be as follows: In the next section, we will delve into the concept of Adaptive Bitrate Streaming and explain its mechanism for non-streaming expert readers. This will allow them to understand the core focus of this work. In the third section, we will analyse the network parameters and factors that can significantly impact the quality of the video stream. The fourth section will present the metrics used to measure the video quality of encoding and the new quality metric developed by *Netflix* that incorporates human perception. The fifth section will go into the selection methodology and present some simulations of the most used ABR algorithms. Finally, in the sixth and seventh sections, we will summarize our work and provide some points for future considerations.

2. Adaptive Bitrate Streaming

In recent years watching remote content over internet (i.e., OTT streaming) has gained great popularity while non-HTTP protocols have been left behind due to complexity of use and performance upgrades[3]. OTT streaming is based on HTTP protocols and runs on TCP connections. Thanks to TCP communications network packets will be assured to reach their destination even if it requires more time due to retransmissions. As described in the Introduction, the importance of streaming quickness, smoothness and responsiveness has led to the development of a new technology called Adaptive Bitrate (ABR) streaming.

With ABR technologies the player can quickly start the video playback and increase or decrease the displayed quality. This switching decision is directed by what is called the ABR decision algorithm. This algorithm takes multiple parameters into account (such as network throughput, buffer size, playback position, etc.) to decide whether to proceed with a quality up or down-scaling.

As ABR streaming technology is essential to this work, a small description on how it works is required to understand the following developed steps.

[3] TCP communication was designed for small data transfers because of its acknowledgement mechanism. However, as computer performance continues to increase, this solution has become affordable for everyone's computer. In addition, HTTP protocols do not require the installation of additional software, unlike many other non-HTTP protocols.

There are multiple ABR implementations designed by different organizations. For instance, Dynamic Adaptive Streaming over HTTP (DASH), developed by the DASH Industry Forum. It is used by many media companies such as *Netflix*, *Google*, *Microsoft*, etc. *Apple* has also developed its own application that is natively supported on its devices. Even if there are multiple implementations, the working procedure is the same for all providers. Setting up such an operation requires server-side and player-side processing.

2.1. Server-Side Processing

The original stream goes through an encoder that is responsible for encoding the stream at different bitrates. Those bitrates are the available quality levels described by a predefined ABR ladder. The development of such a ladder will be discussed in Section 5. Then to ensure an adaptable playback the encoded streams are divided into small 3- to 10-seconds video chunks called **fragments**. Those fragments can be downloaded by the player, regardless of their quality and order. Finally, the encoder lists streams fragments download order as well as their encoded bitrate in a file called **manifest**.

2.2. Player-Side Processing

Once the player loads a stream, it starts by downloading a manifest file that contains the streams' information such as the available resolution and bitrate levels as well as the fragments' length and location. Then, the player downloads the first desired fragment depending on the ABR strategy decision. While playing the first fragment the ABR algorithm might decide to start downloading the following fragment at a higher bitrate level if the network throughput is sufficient, see Figure 1. The fragment will then be placed in a **buffer** waiting to be played by the player.

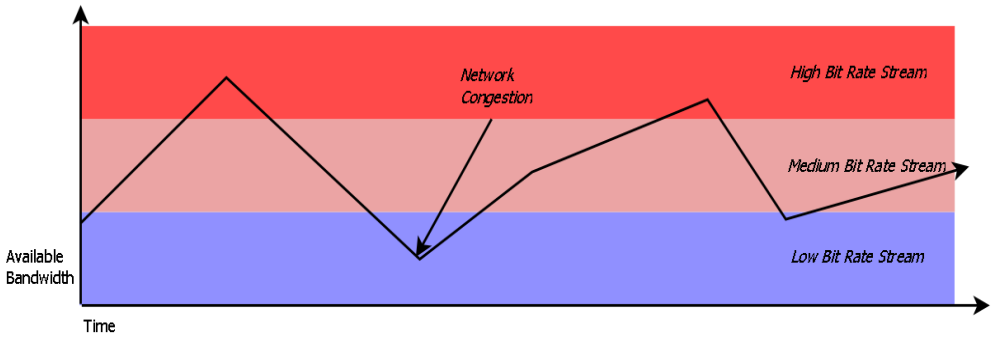


Figure 1: ABR selection in action.

3. Network Identification

This chapter will cover an important part of the problem which is the network analysis. The internet is in fact the limiting factor for remote content transmissions. Indeed, network packets need to be routed and it might take some time to reach the final destination i.e., the end-user device. Most of the time the problem is that there is no control on how those packets will be delivered as it depends on many global factors such as network congestion, router speed, resource location, etc.[4] For a content provider of any kind the main and easiest factor that can be controlled is the packet data size. The smaller the data the faster the destination is reached. That's why, among other things, almost all internet traffic is compressed to be as small as possible. This also applies to media streaming with ABR fragmentation, see Section 2.

3.1. Quality Metrics and Player Options

The most common factors known to influence the quality of a network connection are:

- The **latency**, the time a data packet takes to travel from a point to its destination across the network then back to its origin;
- The percentage of **retransmitted packets** due to packet loss;
- The latency deviation for different packets to be received, called **jitter**;
- The available **throughput** that represents the amount of data that can be transmitted on the network per seconds.

Most of the time, these factors cannot be changed easily yet improving one of them would increase the Quality of Experience (QoE) for the end-users.

Players have some key parameters that can influence the playback quality. Those player configuration options must be chosen by the development team according to the QoE objectives. The following configurations must be carefully chosen to ensure a good streaming experience.

- The **quality level ladder** is the predefined list of available resolutions and/or qualities. This is required by almost all web media player as it allows ABR streaming. Moreover this list is supposed to be defined by a media delivery dedicated team because it influences the streaming smoothness. See Section 4 for more details on ABR ladder determination.

[4] Nowadays more and more homes no longer have a simple modem to connect to the internet. The internet connection setup has many more components like routers, proxy, VPN, power line, etc. Those are placed in-between modem and user devices and have an impact on the latency as well as network stability.

- The **buffer size** is also a significant parameter while streaming with ABR. It is the memory space in playback seconds (s) that the player is allowed to use in order to plan a video fragment download. The buffer is filled in during playback to allow higher quality level download.
- The **ABR algorithm** is responsible for the automatic quality level selection. The algorithm takes multiple parameters as input such as a network throughput estimation, the maximum buffer size, the buffer level, the quality level ladder, etc. Depending on the video player, different algorithm can be chosen. Each algorithm has its advantages and drawbacks, see Section 5 for different algorithm details.

Player metrics describe how good a stream is playing. They highly depend on the network factors quality and configuration options described earlier. These are the only measurements that the player can consider for selecting the best quality level and take a crucial role in the QoE definition.

- The **selected quality levels**. These are the selected quality levels of the previous video fragment. This metric describes how the network state has been evolving up to a moment in time. With this, it can predict the average network quality. The selected fragment quality size is written $\sigma_{(n,l)}$ with n being the fragment number and l the selected quality level.
- The **buffer fill-in level** represents the playback video that is ready to be displayed at time t, written $B(t)$. When video fragments are downloaded, if one that comes before is still playing, the upcoming fragment will be placed in the buffer. If the buffer is full, this means the network can potentially assume higher fragment quality download. The ABR algorithm could decide to remove the last buffered fragment to replace it with a higher quality fragment.
- The **fragments download time**. This is the time it took for the previous fragments to be downloaded, formulated as DT_n with n the fragment number. Combined with the previous selected fragments quality level, it is possible for the algorithm to compute the available network throughput (T_n) at the n-th fragment download time with the ratio:

$$T_n = \sigma_{(n,l)}/DT_n$$

- The **join time** is the time for the first video frame to appear on the player, written JT. This covers the time for the fragment to be downloaded, sent to the player and rendered on screen. This is one of the most important metric as it is responsible for up to 6% audience departure every second after an initial two second wait time [2]. The join time must be as short as possible to avoid users to leave, which implies loading a lower fragment quality and then scaling it up to a higher quality, see Section 5.

- The amount of **rebuffering events**. A rebuffering event is taking place when a fragment finishes its playback and the buffer is empty. Basically a n-th fragment duration (written τ_n) must be the maximum time for fragment $n+1$ to be downloaded to ensure a smooth playback.[5] Rebuffering occurs when the buffer is not filled at the end of the latest fragment played time such as:

$$B\left(\sum_{i=0}^{n}\tau_i\right) < \tau_{n+1}$$

3.2. Bad Network Conditions

It is not straightforward to define a "bad network condition". Such a subjective observation depends on the usage. For example, requirements for reading news on internet are not the same as for watching 4K videos on YouTube. In the case of OTT media streaming for Pickx, we need to define what the criteria are for an "acceptable network condition". Then conditions that do not meet these criteria will be considered as "bad network conditions".

We need to define what is the minimal QoE for the users, which are the reasons that make them leave. Those reasons depend on multiple subjective factors. One way to know how important these factors are, would be to ask the Pickx consumer directly through a market survey. But it is slightly out of the scope of this work as it focuses on a technical and quantifiable solution. Therefore, we suggest using common sense and propose to define a "good streaming experience", based on different studies ([2], [4], [5]), as follows:

- The optimal selected resolution should be at least 720p for mobile & tablets and 1080p for larger screens with a minimal 1.5Mbit/s bitrate. Today 720p HD streams are a minimum necessity as 67% of viewers say video quality is the most important factor when watching a stream [4]. We will see in Section 4 that depending on the content type, the 1.5Mbps quality level can be considered as the minimal required quality for action scenes.
- The join time should be less than 3s. As the studies suggest, after 3s of waiting, roughly 20% of the audience is gone [5]. Three seconds seems the maximum limit for keeping user satisfaction.
- The optimal number of rebuffering events should be less than 2% of the watch time. According to studies, an occurrence of two rebuffering events induces a 10% watch time decrease. [2]

[5] This can be improved by downloading fragments earlier for instance, see Section 5 with different ABR algorithms.

To find the network factors that produce this good streaming experience, let's compute some equations and summarize the notations in Table 1.

Usually, the first fragment is the lightest one that only contains video headers which is a negligible size compared to the others, the equations are:

$$\sigma_{(2,1)} \approx \sigma_{(1,1)} + \sigma_{(2,1)}$$
$$N = 1 + (V_T/\tau_n)$$

Except the first one, other fragment playback durations are constant and set to 4s to avoid too large and unstable downloads or too many small fragments download.

$$\tau_n = 4s \quad \forall n \in \{2, \dots, N\}$$

Note that latency plays a significant role in this download time, with LAT = 100ms and a constant join time of 3s:

$$DT_1 + LAT < 3$$
$$\Rightarrow DT_1 < 2.9\text{s}$$

With previous equations a good network throughput can be defined as follows:

$$T > \sigma_{(2,1)}/DT_1 \qquad \text{with} \quad \sigma_{(2,1)} = \beta_1 * \tau_2$$

Then, with L = 1 and the required bitrate β_1 = 4Mbit/s:

$$T > (4 * 4)/2.9$$
$$\Rightarrow T > 5.52\text{Mbit/s}$$

While playing $B(t) > 0$ and the n-th fragment must be downloaded before the end of the $(n-1)$-th fragment playback (written DL_n) to meet the third defined condition.

To summarize the different limiting factors that represent an "acceptable network condition" can be described as:[6]

- A maximum latency of 100ms;
- A minimum network constant throughput of 5.52Mbit/s;
- A maximum packet loss of 5%[7].

These limiting network factors combined with the network technology on Table 2 will be used for the network environment simulation to compare ABR technology in Section 5.

[6] Note that these limiting factors do not consider the computer speed and usage or the server latency.

[7] The packet loss limit is set to 5% to stress the network a little more. [6]

Notation	Definition
T	The available network throughput
V_T	Video duration
N	Number of video fragments
L	Number of video quality level
n	n-th fragment
l	Selected quality level
$\sigma_{(n,l)}$	n-th fragment size at l-th level
β_l	l-th level bitrate
DT_n	n-th fragment download time
τ_n	n-th fragment duration
$B(t)$	Buffer fill-in level
DL_n	n-th fragment download deadline
LAT	Network latency

Table 1: Notations

Mobile network name	Download throughput (Mbit/s)	Latency (ms)
3G	4	100
4G	15	50
4G+	45	50
Wi-Fi	80	12
5G	150	1

Table 2: Typical mobile network throughput and latency. [7]–[9]

4. Streams Bitrate & Quality Analysis

When it comes to streaming optimization, the first idea would be to optimize the player or its ABR algorithm. The original video - made by a movie studio (or other) - is not the data stream that is sent to the video player; it is much too large. Meaning that a stream will first be encoded (and reduced in size) before transport over the network, see Section 2. This encoding measurement is important as that is where first efficiency losses may appear. That is the purpose of this section, namely, how to measure the encoding quality (encoding loss) and what optimization to do to increase the streaming QoE.

The first section of this chapter is about exposing the most used Full Reference (FR) quality measurement metrics[8] for video encoding. These are mainly DMOS, PSNR and SSIM. Then in the second section we will introduce VMAF, a new FR metric proposed by *Netflix* that seems to better represent human quality perception. We will then capture and plot streams' quality metrics for a more in-depth analysis. Finally, in the last section we will discuss some ABR quality ladder improvements that *Proximus* could implement for Pickx.

4.1. DMOS

DMOS stands for Differential Mean Opinion Score. As the name suggests it is an average opinion score based on human subjective judgments. The differential part means that comparison assessment is made between an original video and the same video with distortion. This differentiation is made by subjects rating the quality based on an Absolute Category Rating (ACR) that maps ratings between "bad" and "excellent" to numbers between 1 and 5, see Table 3.

ACR	Scale
Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

Table 3: ACR ratings associated with DMOS scale.

With N subjects and S the DMOS scale value, the DMOS score is computed as follows:

$$DMOS = \frac{1}{N}\sum_{i=1}^{N} S_i$$

DMOS is the best predictive metric as it represents what the average subject voted for. The bigger the N the more accurate the score is. The problem is to run this metric at scale as it requires more viewers. Increasing N means increasing the time needed to run the tests and at the end increasing the cost. Another DMOS drawback is the mathematical bias induced while averaging on an ordinal scale[9].

[8] A Full Reference (FR) metric means that the quality measurement works by a comparison of two samples: an original and a distorted one.

[9] An ordinal scale is a scale where intervals are not constant. For instance, with the DMOS scale, something "poor" is not twice as good as something "bad". Which means computing the average is meaningless considering the centre of the scale is not equal to the actual metric centre.

4.2. PSNR

PSNR stands for Peak Signal to Noise Ratio. It is widely used for lossy signal compression measurements[10]. In video encoding context it is used for detecting errors that were induced while compressing the signal. PSNR is based on a Mean Square Error (MSE) measurement, which is a comparison between an original signal to its distorted version. The Mean Square Error (MSE) equation for a discrete source signal $s[k]$ and its distorted version $d[k]$ $\forall k \in \{1, \dots, K\}$ is defined as follows:

$$MSE_s = \frac{1}{K}\sum_{k=1}^{K}(s[k] - d[k])^2$$

With the maximum quantification level (written M_s) of the source signal, the PSNR equation is defined as:

$$PSNR = 10 * \log\frac{M_s^2}{MSE_s}$$

PSNR can be applied to images and videos of size $W\text{x}H$ by increasing its MSE_s dimensions.

PSNR value is a mathematical ratio expressed in dB between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [10]. The higher the value the higher the quality is. This signal quality indicator is not as perfect as human overall quality perception. The problem with PSNR metric is that the score represents a pixel average comparison. There is no global quality perception on the image. In some cases, PSNR scores the same for different noise perturbations, see Figure 2.

To overcome this problem PSNR-HVS and PSNR-HVS-M metrics are PSNR extensions that incorporate human visual perception measurements. Those extensions include contrast perception and PSNR-HVS-M also includes visual masking effects[11] detection.

[10] A lossy compression is a compression where quality is altered.
[11] Visual masking effects appear when images' information is hidden by another image called mask.

4.3. SSIM

SSIM stands for Structural Similarity Index Measure. It is also a FR metric that describes the perceived quality of images and videos. The structural similarity is the way SSIM differs from PSNR by considering pixel interdependency. The closer the pixels the more dependent they are. This metric is much more accurate regarding the perceived quality by the Human Visual System (HVS) because it compares chunks of pixels close to one another on both images rather than comparing each single pixel. On Figure 2 a comparison between PSNR and SSIM can be observed. SSIM metric is a value between 0 and 1 where 0 means no structural similarity in the images, and 1 means the images are identical.

Instead of using traditional error summation methods, the SSIM uses a combination of three factors that are loss of correlation, luminance distortion, and contrast distortion. The way it works is typically by using small image chunks of pixel size 8x8 on both images and computing the similarity between the chunks (x and y).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Where:

- μ are average values;
- σ are variance values;
- c are constants.

Multi-Scale SSIM (MS-SSIM) is an improvement of the SSIM metric that allows the sub-sampling of multiple image stages. It generally outperforms the SSIM algorithm. [11]

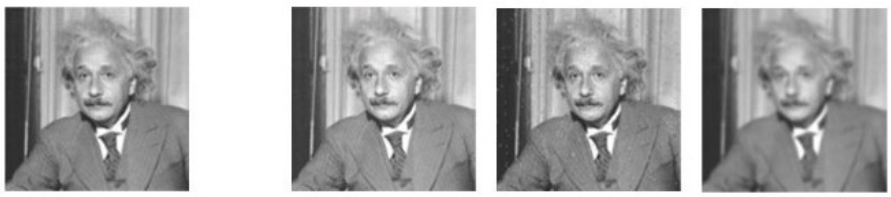


Figure 2: SSIM values vs PSNR values for different noise sources. [10]

4.4. VMAF

At *Netflix*, they noticed that the existing metrics described above did not fully represent human visual perception or were not a solution to run at scale. They decided to launch VMAF (stands for Video Multi-Method Assessment Fusion), a tool that predicts image and video quality resemblance level [12]. The VMAF tool produces a VMAF score which describes the quality resemblance between an original video and the same altered video. It scores this comparison between 0 and 100[12]. This metric tries to estimate human relative image quality level based on Machine Learning (ML) predictions. The project started in 2014 in collaboration with the University of Southern California and is still ongoing as an open-source project maintained by *Netflix* and the VMAF community. They created a C/C++ library called *libvmaf* that is implemented in multiple ways such as Python scripts, command line interface or FFMPEG[13] filter.

On Figure 3 a comparison between VMAF and Daala_PSNRHVS[14] prediction scores shows that VMAF seems to better fit the DMOS evaluation[15]. The same-coloured dot represents the same video source but with a different encoding configuration.

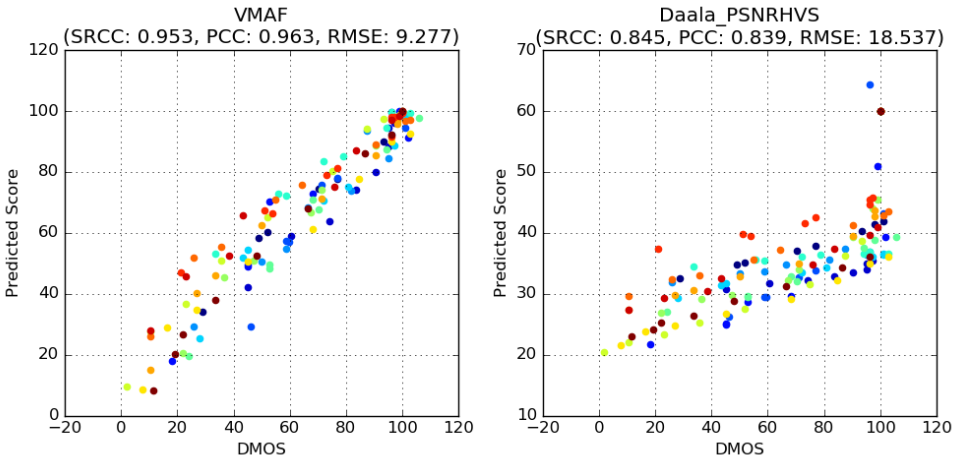


Figure 3: VMAF scores (left) vs. PSNR-HVS scores (right) for NFLX dataset. [12]

[12] With VMAF < 20 the distorted video is described as "very bad" and with VMAF > 80 the distorted video is considered as excellent.
[13] FFMPEG is an open-source software used for video manipulation such as transcoding, scaling, post-production, etc.
[14] Daala_PSNRHVS is a PSNR-HVS measurement on a Daala compression video.
[15] The dataset was divided into 2 sub-sets: NFLX-TRAIN and NFLX-TEST to avoid overfitting. The graph shows NFLX-TEST predictions.

Viewing distance system

On Table 4, the table indicates the optimal viewing distance for a typical resolution. This table is based on the human eye contrast sensitivity[16]. For example, for a resolution of 1280x720 the subject needs to be placed at roughly five times the screen height ($\approx 5H$) to see the content sharpness. A subject watching a 4K stream on a 4K monitor must be placed at $1.6H$. With a 4K monitor, the subject can be closer because there are much more pixels than in a 720p TV of the same height.

This raises the question for mobile devices, usually the mobile viewing distance is $\approx 4H \pm 1H$. Table 4 would recommend a resolution of 720p (4,8H) or 1080p (3,2H). Which from a required download bitrate is quite different. According to this table, it is therefore difficult for the users to appreciate the difference between 1080p and 720p unless they are really close to the monitor.

Could VMAF provide an answer if for mobile a higher resolution has value for the subject? To prove this point, they trained a Machine Learning (ML) model for phones based on the same video sequence as the environment setup. Except that subjects were watching contents on a mobile phone with resolution 1920x1080, and the fixed viewing distance was replaced by a distance subjects were comfortable with.

On Figure 4, for the same encoding, perceptual qualities on phone devices are higher than on TV displays. That's because the viewing distances were left to the subjects' appreciation and artifacts were less visible on small devices. On the same figure, the perceptual quality for 720p and 1080p on phone devices are quite the same. That's because of the viewing distance, which means that maybe using 1080p for devices is not efficient as it takes a lot of resources for the same human perception. We will continue this discussion in Section 6.

Image system	Aspect ratio	Optimal viewing distance
640 x 480	4:3	7 H
720 x 576	4:3	6 H
1028 x 720	16:9	4.8 H
1920 x 1080	16:9	3.2 H
3840 x 2160	16:9	1.6 H
7690 x 4320	16:9	0.8 H

Table 4: The VMAF viewing distance setup. [12]

[16] A typical human acuity is 40-50 cycles per degree. For 1° in the vision field, human can perceive 40 to 50 repetitive pattern cycles.

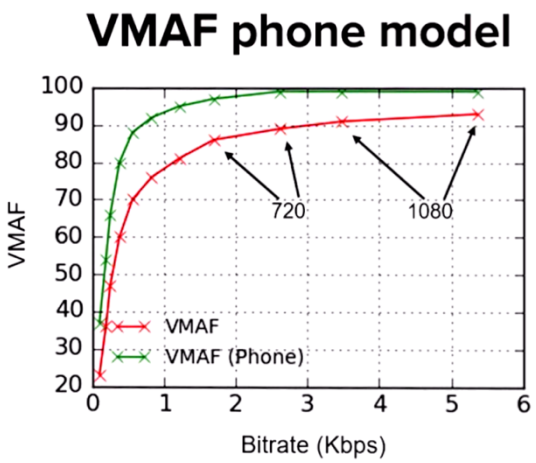


Figure 4: VMAF scores for phone model vs default model. [12]

4.5. Streams Quality Comparison

We used public Creative Commons streams provided by the Blender Foundation[17] to create a pool of different streams types. For analysing movie quality, we decided to use Big Bucks Bunny (BBB)[18], Tears of Steel (TOS)[19] and Elephant Dream (ED)[20] to compare VMAF scores with other metrics described in Section 4. These media sources are quite different regarding their content types. Big Bucks Bunny (see Appendix 9.1) is an animated movie that has a bright tone which can represent the content watched by children. Tears of Steel (see Appendix 9.2) is a science fiction film with a lot of dynamic shots, representing a typical action movie. Elephant Dream (see Appendix 9.3) is a short computer animated fantasy movie with a constant dark tone which is interesting for analysing dark video frames.

For each of these streaming sources we isolated two different sequences of 15 seconds duration. The first is an "action" scene with series of dynamic shots and movements. The second one focuses on "still" images and has one or two long traveling shots. For each sequence we downscale the encoding quality level[21] to fit the Pickx

[17] The Blender Foundation is a non-profit developing organization that maintains a 3D content creation program called Blender. They distribute multiple Creative Commons movies since 2006.
[18] https://dash.akamaized.net/dash264/TestCasesHD/2b/qualcomm/2/MultiRes.mpd
[19] https://media.axprod.net/TestVectors/Cmaf/clear_1080p_h264/manifest.mpd and
[20] https://dash.akamaized.net/dash264/TestCasesHD/2a/qualcomm/1/MultiResMPEG2.mpd and
[21] Note that the original video bitrates were respectively: 7941kbps for BBB, 6225kbps for TOS, 7952kbps for ED.

predefined quality ladder (500kbps, 1000kbps, 1500kbps and 6000kbps)[22]. Figure 5 shows for the ED "action" sequence, a comparative video image for each encoding levels. The difference between the 500kbps and the 6000kbps is quite visible.

Action scenes (with a lot of dynamic frames) may require higher quality level to reach the same QoE as quieter scenes. For instance, still frames on Figure 6 have the same encoding ladder as dynamic frames on Figure 5. This shows that for motionless frames, the highest bitrate is not required even if the network quality allows it. Frames on Figure 6 are also much darker than those on the Figure 5. This means that less information is displayed and thus subjects are less likely to see quality differences. Removing ABR top-ladder quality levels for streams with high VMAF score at low bitrate might make sense, see quality ladder suggestion in the end of this section.

Differences between original and 6000kbps frames are not noticeable in case of Figure 6 and Figure 5. As well as subjective judgment, comparison metrics confirm that the human perception is nearly the same. Appendices 9.4, 9.5 and 9.6 prove that 6000kbps have huge resemblances with original streams. 6000kbps streams are therefore considered as original streams and are the maximum quality level to reach. Those excellent results mean that 6000kbps level can be the highest available quality for future ABR ladder development.

With a VMAF average score of 86.1 for the 1500kbps streams (Appendices 9.4, 9.5 and 9.6), differences become significant[23]. With approximately 4 times the sequence size, the VMAF score increases by ≈7. The differences between those streams are on average noticeable for “action” scenes, see gaps between 1500kbps and 6000kbps streams on the Appendices. It seems that, for action movies, the last two quality levels (1500kbps and 6000kbps) are not perfectly chosen as the perceptual changes are quite noticeable. We will discuss how to reduce these gaps in the suggestion section.

For non-animated movies on Appendix 9.5, the visual differences between 500kbps and 1000kbps streams are significant. Increasing the bitrate by a factor of 2 would result approximately to a VMAF 30 points increase, which result in a relatively higher perceived video quality.

[22] The 6000kbps configuration is only used for the Apple TV app.
[23] A VMAF score ≤ 90 means that frames’ difference starts to be noticeable.

For still and animated movies Appendices 9.4, and 9.6 shows that 500kbps level produces a medium streaming quality. Quality starts to be poor for 500kbps "action" streams. A VMAF score below 30 is considered as very poor and frames appear to be really distorted. This is due to the difficulty to encode high information rates at low bitrates. The fact that the original frame has a lot of motion blur indicates the large amount of information to display between two following frames. If the bitrate decreases, the quality will also decrease. In the improvement section we will discuss if there is a benefit to keep such a bad quality level for Pickx.

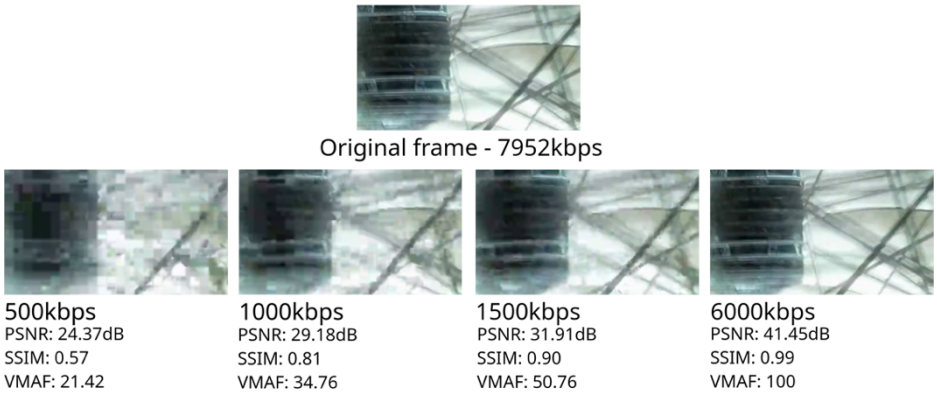


Figure 5: ED metrics on action scene from 500kbps to original with a factor 5 zoom.

Figure 6: ED metrics on still scene from 500kbps to original with a factor 5 zoom.

4.6. Streams Improvement Suggestion

As described in the previous section, 500kbps streams quality are really "poor" in case of action shots. Action movies are made up of many action shots and it is trivial

that poor quality action shots will result in poor quality action movies[24]. On the other hand, 500kbps bitrate level is quite sufficient for still shots and cartoon movies. Moreover, even if the quality is considered “really bad” in some cases, to keep low resolution in ABR ladder is meaningful as older devices will still be supported.

On Figure 5 and Figure 6, the differences between the 1500kbps and the 6000kbps frames may be light or more significant depending on the content. This shows that the current ladder has its benefits as 1500kbps is sufficient for still and animated movies. Adding bitrates on top of the 1500kbps level can only be interesting for action and dynamic movies.

Adding a 6000kbps quality level to the Pickx web platform ABR ladder can really increase the users’ QoE as they will enjoy excellent quality streams if the network throughput allows it[25]. As shown for the action scenes in Appendices 9.5 and 9.6, VMAF gaps between 1500kbps and 6000kbps quality levels are large. This would force the ABR algorithm to jump to a very high bitrate level that might not be stable. The lack of "middle" quality levels for action movies shows that it might be interesting to include them as well for a better user experience.

Therefore, it might be more convenient to handle a more "standard" ABR ladder for Pickx streams. For example, by using six different quality levels. It seems to be a good trade-off between the ten levels used by *Netflix* (and other industries) and the 3 levels currently used for Pickx (web and mobile). Table 5 is based on the *Netflix* default ABR ladder, adapted to the different Pickx usage.[26]

Quality level	Bitrate (kbps)	Resolution
1	500	480x270
2	1000	640x360
3	1500	1280x720
4	3000	1280x720
5	4300	1920x1080
6	6000	1920x1080

Table 5: Suggested ABR ladder adapted for Picks.

[24] VMAF measurements are particularly slow. The average measurement framerate is 5 FPS. For measuring a 90min movie it can take up to 7h. The VMAF community is currently working on a way to increase the processing speed by allowing users to specify the number of frames to skip every seconds.

[25] Recall that the 6000kbps streams are not accessible for web and mobile users. It is only used for the Apple TV app.

[26] The original ladder uses a 4:3 resolution ratio at low bitrates for very old devices compatibility.

On Appendix 9.7, the new ABR ladder is applied to the streams (BBB, TOS, and ED). The quality gaps seem to be reduced thanks to the new levels.

This ladder seems quite effective but could be tailored to the stream content type. For instance, VMAF results for BBB show that 1500kbps level might be sufficient for still and animated movies. Thus, higher quality levels (3000kbps, 4300kbps and 6000kbps) might overload network traffic for no reason. Furthermore for "action" scenes at 500kbps for TOS and ED streams produce a “really bad” VMAF score, and streams might not be usable as the required QoE would not be met.

This reflection shows that it can be interesting to adapt the current ABR ladder with a more dynamic approach. That is what *Netflix* has developed in 2015 under the name of Per Title Encoding (PTE). The idea is to adapt the ABR quality ladder to the content type. As described above, in some cases there is no need for high quality levels (3000kbps, 4300kbps and 6000kbps) as the quality score reaches its limit and the user will not perceive any difference. With action movies it is interesting to add more quality steps between the 1500kbps and the 6000kbps levels while for animation or cartoon movies 1500kbps or lower bitrates are sufficient. We recommend using such a technique. See the original *Netflix* blog post at Reference [13] for a full demonstration.

5. Tuning the Selection Methodology

In this section, we will explore the various commonly used ABR algorithms and evaluate their pros and cons. Our aim is to suggest the best algorithm for Pickx, based on performance analysis. To evaluate the performance of these algorithms, we will simulate them using a specific tool and compare the results under different circumstances.

5.1. Algorithm Improvement

The Pickx default algorithm (called **v2**[27]) performs quite well for the predefined network configurations. But when it comes to adapt video quality with a discontinuous 4G+ network configuration (Appendix 9.8), the algorithm keeps switching between available qualities. This decreases the QoE as the user will perceive a lot of quality updates. To improve the Pickx player selection algorithms (**v1** and **v2**) there is only one way; tuning the different configuration settings proposed by the player SDK. Due to the lack of documentation and the complexity of this task we will not follow

[27] The Pickx player also have an algorithm called **v1** that is working but not used anymore because of its outdated technology.

this direction. Moreover, **v1** and **v2** are closed source which means reverse engineering the code is not allowed although it would be feasible. We will simulate other ABR algorithms with *Sabre* (ABR simulation tool [14]).

5.2. Alternative Algorithm

There are different related works for ABR algorithm. Some of them are already implemented in different players and can be simulated with *Sabre*. A few others are promising but still under development.

THROUGHPUT

THROUGHPUT is a simple algorithm that maps the available network throughput to the video bitrate quality level. It typically switches quality level to the highest available bitrate $\leq 90\%$ than the available throughput [14]. This algorithm performs very well in case of an empty buffer (at start-up, when seeking[28] or when rebuffering events occur).

BOLA

BOLA stands for Buffer Occupancy based Lyapunov Algorithm. It is a buffer occupancy-based algorithm released in 2016 that doesn't need network prediction [15]. BOLA seems more stable in case of network throughput fluctuation as the quality is guaranteed as much as possible. The longer the buffer maximal duration the better the algorithm performs. BOLA focuses on decreasing rebuffering events. BOLA is used in production by a lot of video providers such as *Akamai*, *BBC*, *Orange*, etc. and is already implemented in *Sabre*. [14]
On Appendix 9.9, the algorithm produces a higher average quality level despite many quality changes especially for low network throughput configurations.

BOLAE

BOLAE stands for BOLA Enhancement and was developed in 2019. This algorithm focuses on decreasing rebuffering events as well as quality oscillations [14]. This algorithm was also implemented in *Sabre*. On Appendix 9.10, the average played qualities are lower than BOLA in all cases but for slow network condition (3G and 4G) bitrate changes are reduced by approximately a factor of 3. In the 3G and 4G network conditions the algorithm tries to reduce the fragments download time i.e., the quality, in anticipation of potential future disruptions. For higher network conditions, BOLAE seems to slowly increase the quality level, again for anticipation. This means that BOLAE may be efficient in case of poor network conditions.

[28] Seeking is a time movement from a playback point to another.

DYNAMIC

DYNAMIC is a simple decision algorithm that uses BOLA and THROUGHPUT depending on the buffer occupancy level [14]. Typically for small buffer occupancy ($\leq$ 10s) THROUGHPUT is selected. In the case of Pickx, the buffer length is set to 8s which means that the selected algorithm would always be THROUGHPUT. This algorithm is effective because it combines BOLA and THROUGHPUT performances. For start-up, seeking or rebuffering event the THROUGHPUT algorithm is used to quickly download the video fragment to reduce the playback latency. When the buffer is full or greater than 10s BOLAE is used for optimizing the quality.

PENSIVE

This algorithm uses Reinforcement Learning (RL) to select the optimal streaming bitrate based on network predictions [16]. The Machine Learning (ML) model is trained during the playback session by recording past network states. It does not rely on a pre-trained model as many ML applications do today. PENSIVE is not implemented in *Sabre*. It can be interesting for further research to simulate PENSIVE to compare results with other algorithms on Pickx streams.

5.3. Improvement Suggestion

As a result of these simulations, we can conclude that depending on the available network characteristics and the desired QoE, it might be interesting to use different algorithm.

DYNAMIC seems to be the most efficient algorithm in any network situation (Appendix 9.11) but it needs the buffer size to be larger than 10s. For Pickx the buffer size is set to 8s by *Proximus* business. This algorithm can therefore only be used if the buffer size policy is removed[29]. This is a trade-off between streaming smoothness and live latency that needs to be carefully considered by Pickx leaders.

However, for an effective streaming session with the maximum selected quality level, BOLA seems to perform the best. BOLA is not recommended for poor network configurations (3G or 4G) as a lot of quality switches will occur. This algorithm is to be used if the number of quality changes does not have an important role in the QoE definition or if the network quality is relatively high (4G+ or Wi-Fi).

On the other hand, for poor network configurations (3G or 4G) if the number of quality changes is to be as small as possible, BOLAE might be the best algorithm to choose. The problem is the average selected quality level. After a network disruption, the selected quality level might take a long time to settle down to a higher level.

[29] Increasing the buffer size may have an impact on live latencies which may not be acceptable in some cases.

To use this algorithm is to accept a loss of average quality. If the network throughput is very high (with a Wi-Fi connection for example) BOLAE might be sufficient even if the average quality is lower than BOLA.

The decision between BOLA and BOLAE is typically a trade-off between amount of quality switches and playback average selection quality. This decision must also be carefully made by Pickx leaders to improve the QoE.

6. Conclusion

During this work multiple themes were covered, from a network and stream quality analysis to suggestions for the improvement of automatic video quality selection algorithm for *Proximus* Pickx solution.

The network environment description gives the fundamental factors to understand how OTT video playback can be altered. A "bad" network configuration can be defined by assuming the importance of these factors. With network simulation tools it is possible to simulate such a "bad" network environment for streaming contents.

Stream quality analysis methods are used to benchmark the current Pickx streams quality ladder. Comparative quality measurement metrics are described in Section 4 such as PSNR, SSIM and VMAF. VMAF is the new measurement method that outperforms the others regarding HVS predictions. It is shown in that section that the main drawback for the Pickx ladder is the difference between successive quality levels and the lack of high-quality streams, especially for very dynamic scenes within action movies. By using the *Netflix* and the current Pickx quality ladders, a new 6-levels ladder is suggested which should increase the users' QoE. It is also suggested to take an interest in dynamic quality ladder definition called Per Title Encoding (PTE). PTE can tailor the quality ladder to the streaming content type. An animated movie does not require the same amount of information as an action movie for the same perceived quality. In this case it is not always necessary to have the higher quality level as differences between the maximum and lower levels are in most case not noticeable, plus it can save network throughput, see next section.

The ABR selection algorithm oversees whether to switch the video quality by analysing the player environment (network throughput and/or buffer occupancy). With a tool for Simulating ABR Environments (*Sabre*) and a Python script it is possible to quickly plot multiple ABR algorithm benchmarks with different network conditions. Results show that Pickx current algorithm (**v2**) does not surpass BOLA nor DYNAMIC algorithms. DYNAMIC seems to be the most effective if the buffer maximum size is carefully chosen. BOLA and BOLAE algorithms may have advantages for different network conditions.

7. To Go Further

It is inevitable that in the coming years the expectations of users regarding the network speed and latency will increase. As a result, the required QoE for streaming providers will become more and more sophisticated. With the current 5G deployment in the word, the network limiting factors described in Section 3 might change as old devices will progressively be replaced by new devices that support more powerful technologies. The current *Proximus* 5G deployment in Belgium is undefined so far and the network coverage map shows that there is still time to get there [17]. The arrival of 5G will also affect the ABR quality ladder as algorithms will no longer bother to change to low quality level. Thus, lower quality profiles will slowly be replaced by higher quality streams that allow playing 4K videos for example. Even if the 4K streams on smartphones do not make sense due to the screen size and distance, according to Section 4, it makes sense to add these streams' qualities for new high resolution TV monitors.

Concerning the video quality analysis, VMAF is still under development and the tool might have interesting new improvements such as the temporal features measurement optimization and the colour space measurement that will allow capturing potential chroma artifacts. The Per Title Encoding (PTE) strategy study and development might be a continuation of this work and results might be interesting in the future. It might also be interesting to properly compare *Proximus* streams by accessing multiple content types such as live TV programs, action movies, cartoons, etc.

For the ABR quality selection, there is still an interest in analysing and simulating new proposed algorithms such as OBOE [18] and HINDSIGHT [19] or in using a Machine Learning approach with PENSIVE [16]. The tool *Sabre* might need some improvements regarding the documentation and the compatibility with new incoming algorithms.

A final topic that must be explored is the environmental sustainability. According to Reference [20], the total internet carbon footprint in 2020 was higher with a factor of 2 than the worldwide air travel. By the end of 2022, video streaming over internet will represent more than 82% of all the consumer internet traffic [1]. Which means that video processing steps from the encoding to the screen display has a huge ecologic impact that must not be neglected. It may be interesting to adapt or change the way people consume video contents. For instance, decreasing the default selected video quality in ABR algorithms might be a good first solution. That's what *YouTube* did in 2020 during the COVID-19 pandemic to help lessen broadband strain [21]. Results of such an action have not yet been published by the firm. But it might be interesting to explore.

8. References

[1] Cisco, ‘Forecast and Trends, 2017–2022’. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] S. S. Krishnan and R. K. Sitaraman, ‘Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs’, *Proceedings of the 2012 Internet Measurement Conference*, no. 14, pp. 211–224, 2012, doi: 10.1109/TNET.2013.2281542.

[3] Conviva, ‘Uk consumer survey report 2015 dont break the spell’. http://www.conviva.com/conviva-customer-survey-reports/uk-consumer-survey-report-2015-dont-break-the-spell/

[4] J. Santora, ‘75 Live Streaming Stats Every Marketer Should Know in 2022’. http://www.conviva.com/conviva-customer-survey-reports/uk-consumer-survey-report-2015-dont-break-the-spell/

[5] Ydraw, ‘Video Abandonment-How to Stop It!’ https://ydraw.com/whiteboard-video/video-abandonment-how-to-stop-it/

[6] A. Lamberti, ‘How to Measure Network Performance: 9 Network Metrics’. https://obkio.com/blog/how-to-measure-network-performance-metrics/

[7] Kenstechtips, ‘Download Speeds: What Do 2G, 3G, 4G and 5G Actually Mean?’ https://kenstechtips.com/index.php/download-speeds-2g-3g-and-4g-actual-meaning

[8] S. Fenwick, ‘Mobile Network Experience Report’. https://www.opensignal.com/reports/2020/03/belgium/mobile-network-experience

[9] Speedtest, ‘Belgium’s Mobile and Fixed Broadband Internet Speeds’. https://www.speedtest.net/global-index/belgium#mobile

[10] D. Monsters, ‘A Quick Overview of Methods to Measure the Similarity Between Images’. https://medium.com/@datamonsters/a-quick-overview-of-methods-to-measure-the-similarity-between-images-f907166694ee

[11] Wikipedia, ‘Structural similarity’. https://en.wikipedia.org/wiki/Structural_similarity

[12] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, 'Toward A Practical Perceptual Video Quality Metric'. https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652?gi=6a615618c98a

[13] J. Ozer, 'How Netflix Pioneered Per-Title Video Encoding Optimization'. https://streaminglearningcenter.com/articles/how-netflix-pioneered-per-title-video-encoding-optimization.html

[14] K. SPITERI, 'From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player'. https://dl.acm.org/doi/fullHtml/10.1145/3336497#sec-23

[15] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, 'BOLA: Near-Optimal Bitrate Adaptation for Online Videos', *University of Massachusetts*, 2020.

[16] H. Mao, R. Netravali, and M. Alizadeh, 'Neural Adaptive Video Streaming with Pensieve', *MIT Computer Science and Artificial Intelligence Laboratory*, 2017.

[17] Proximus, 'Frequently asked questions about 5G'. https://www.proximus.be/support/en/id_sfaqs_5g_global/self-employed-and-small-companies/support/telephony/mobile-phone-and-sim-card/set-up-your-mobile-phone/frequently-asked-questions-about-5g.html

[18] Z. Akhtar and Y. S. Nam, 'Oboe: Auto-tuning Video ABR Algorithms to Network Conditions', *University of Southern California and Purdue University*, 2018.

[19] T.-Y. Huang, C. Ekanadham, A. J. Berglund, and Z. Li, 'Hindsight: Evaluate Video Bitrate Adaptation at Scale', *Netflix*, 2019.

[20] A. Popescu, *Greening Video Distribution Networks*. 2018.

[21] J. Alexander, 'YouTube is reducing its default video quality to standard definition for the next month'. https://www.theverge.com/2020/3/24/21192384/youtube-video-quality-reduced-hd-broadband-europe-streaming#:~:text=By%20default%2C%20videos%20will%20start,must%20manually%20select%20that%20option.

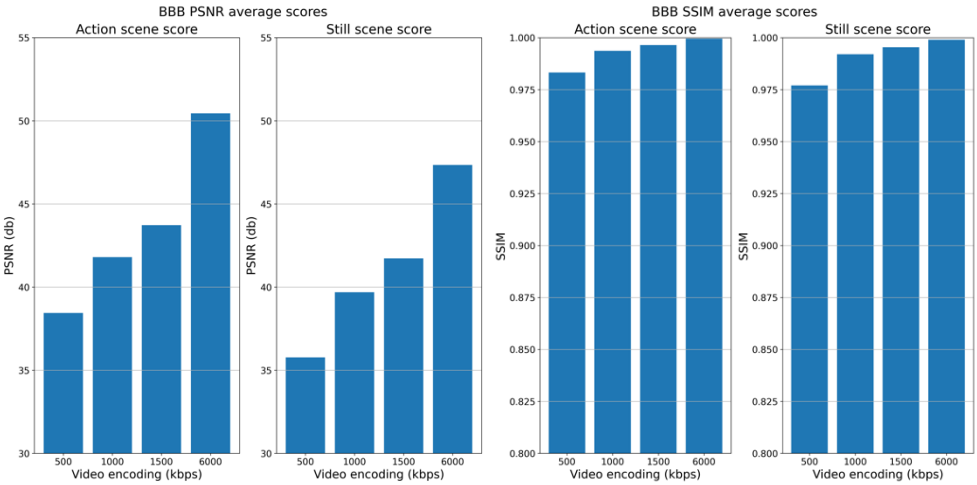
9. Appendices

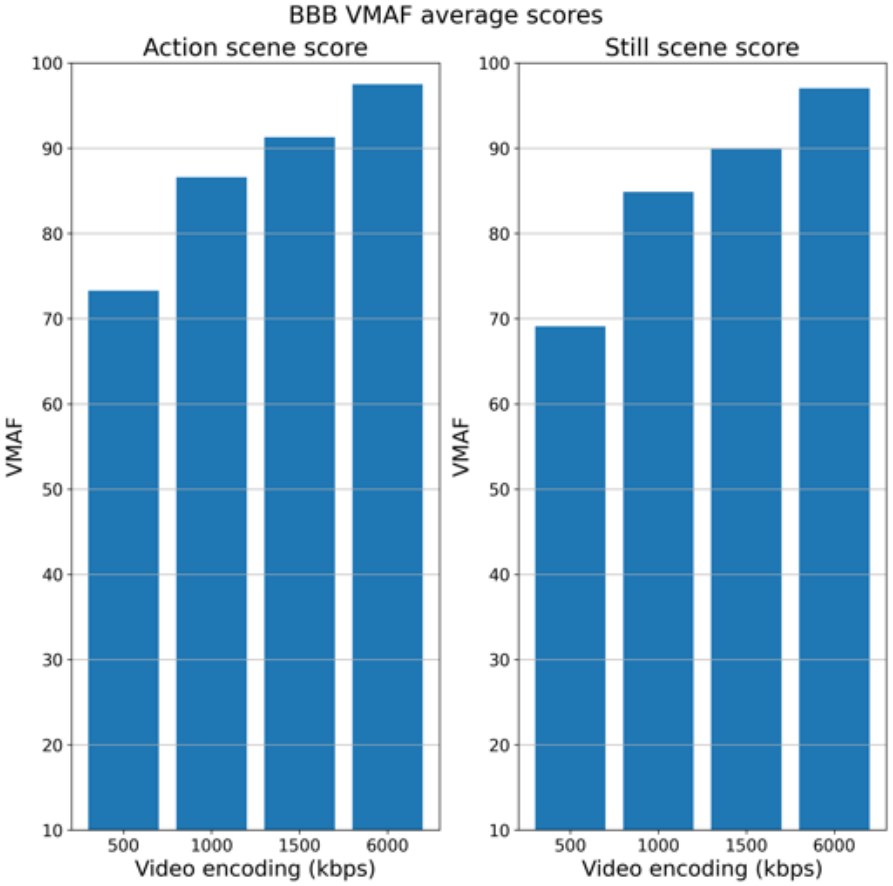
9.1. Big Buck Bunny frame

9.2. Tears of Steel frame

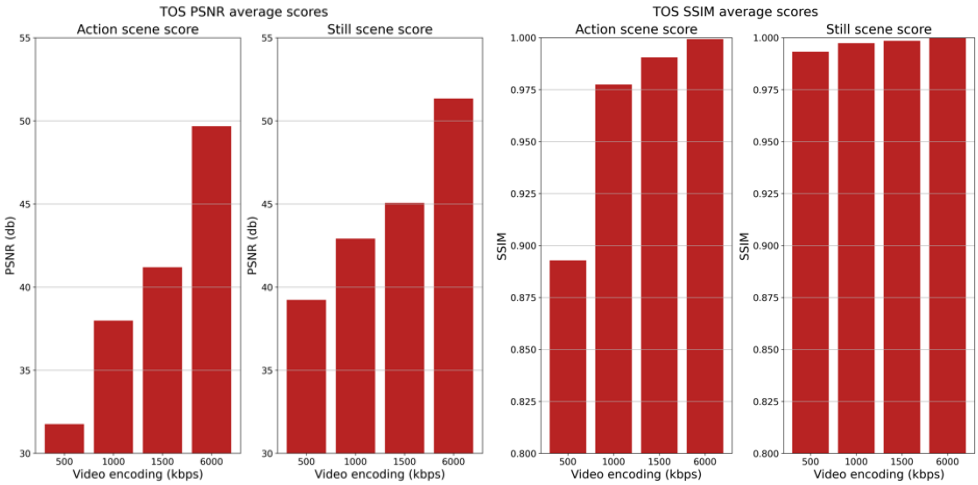
9.3. Elephant Dream frame

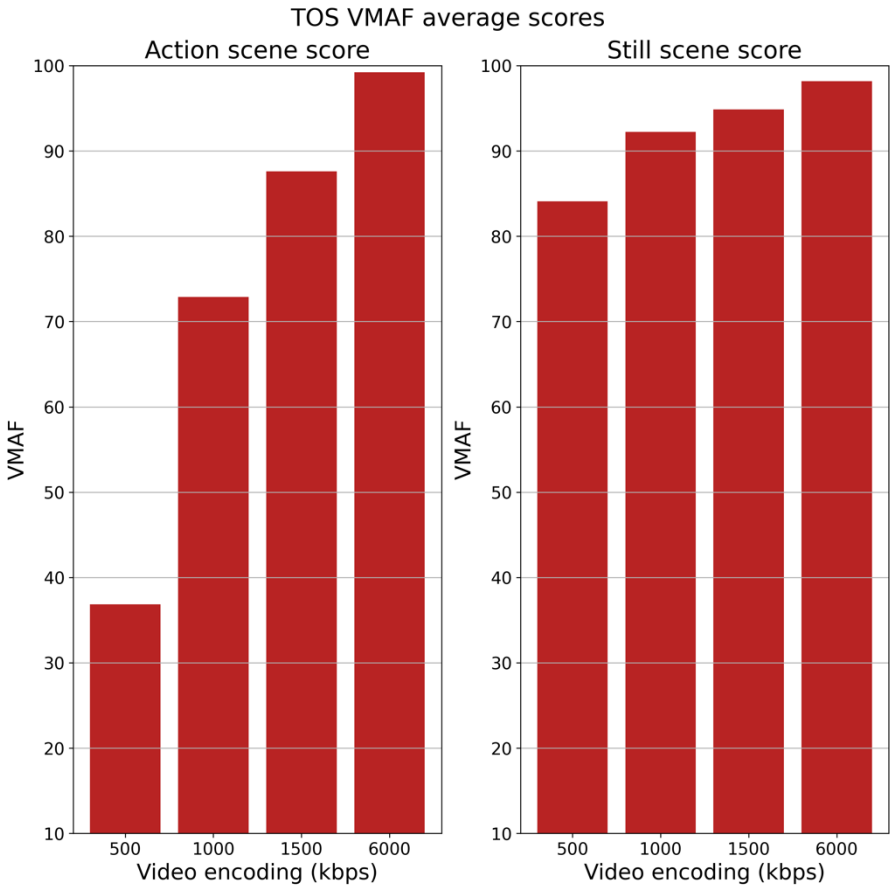
9.4. BBB measurement metrics



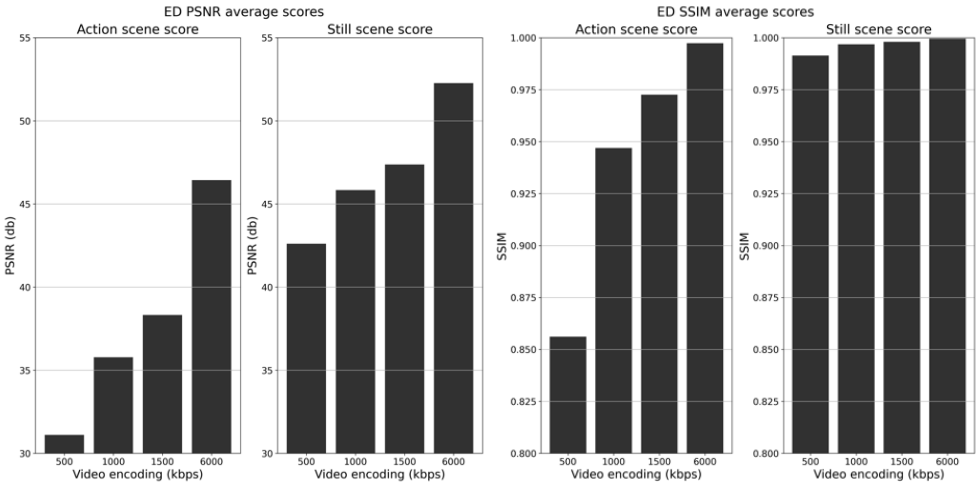


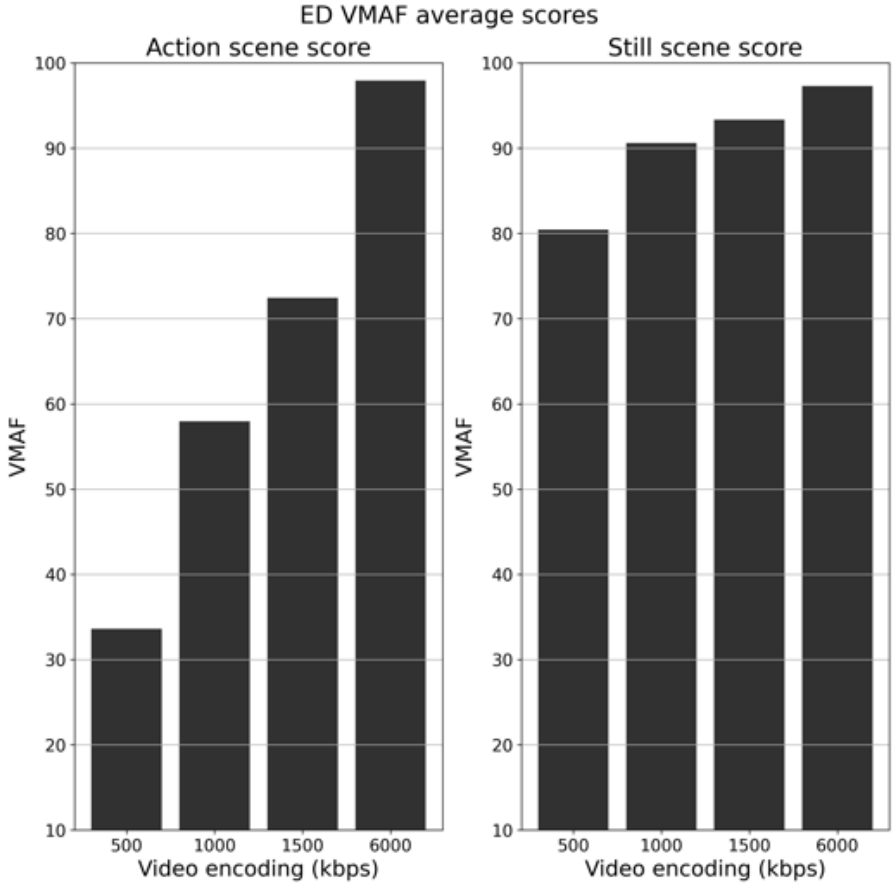
9.5. TOS measurement metrics



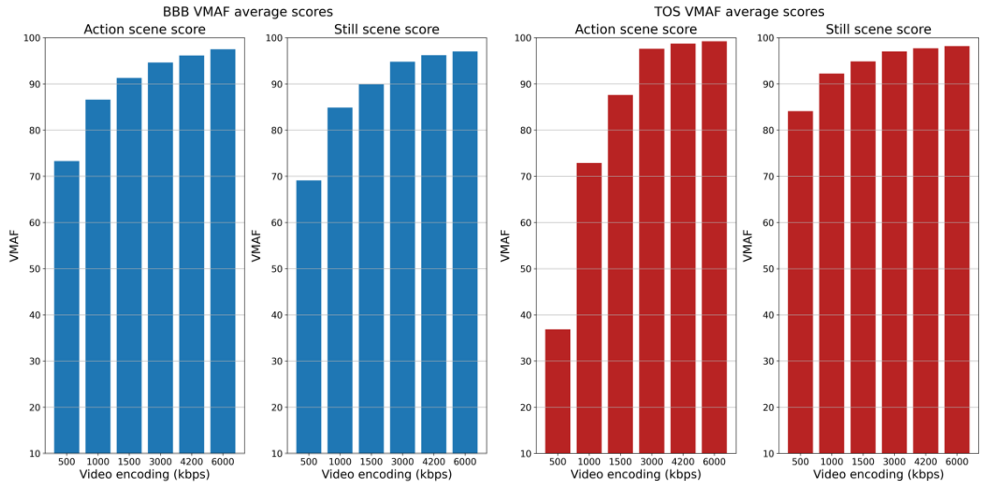


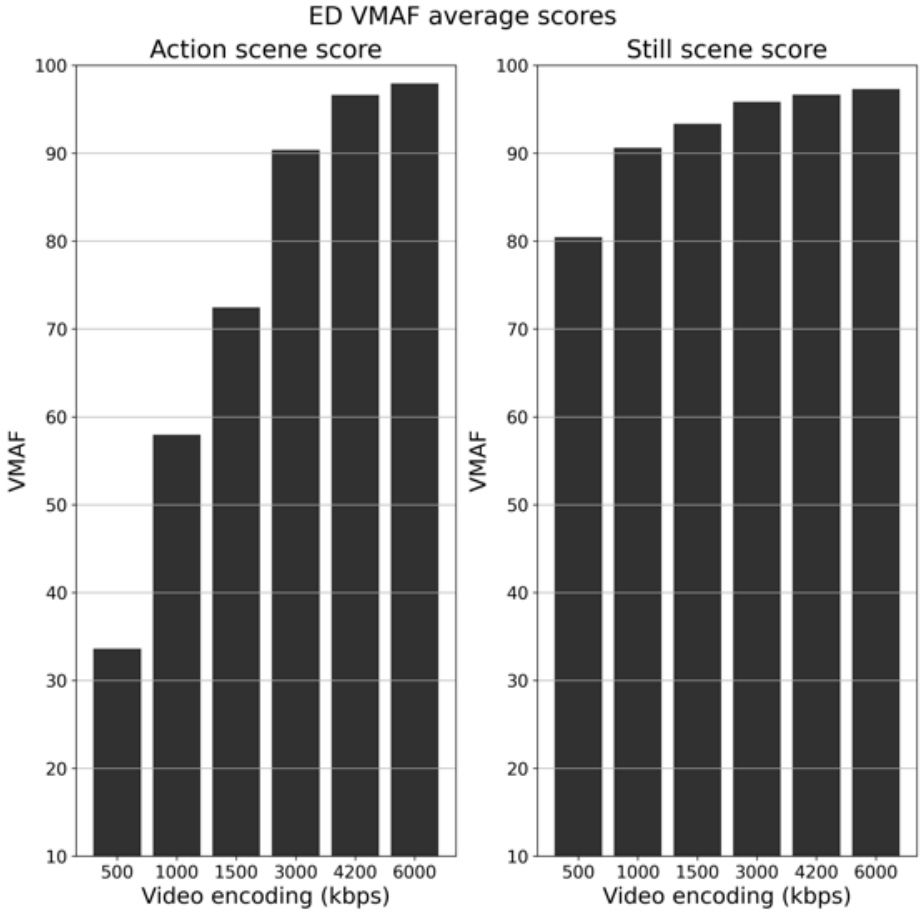
9.6. ED measurement metrics



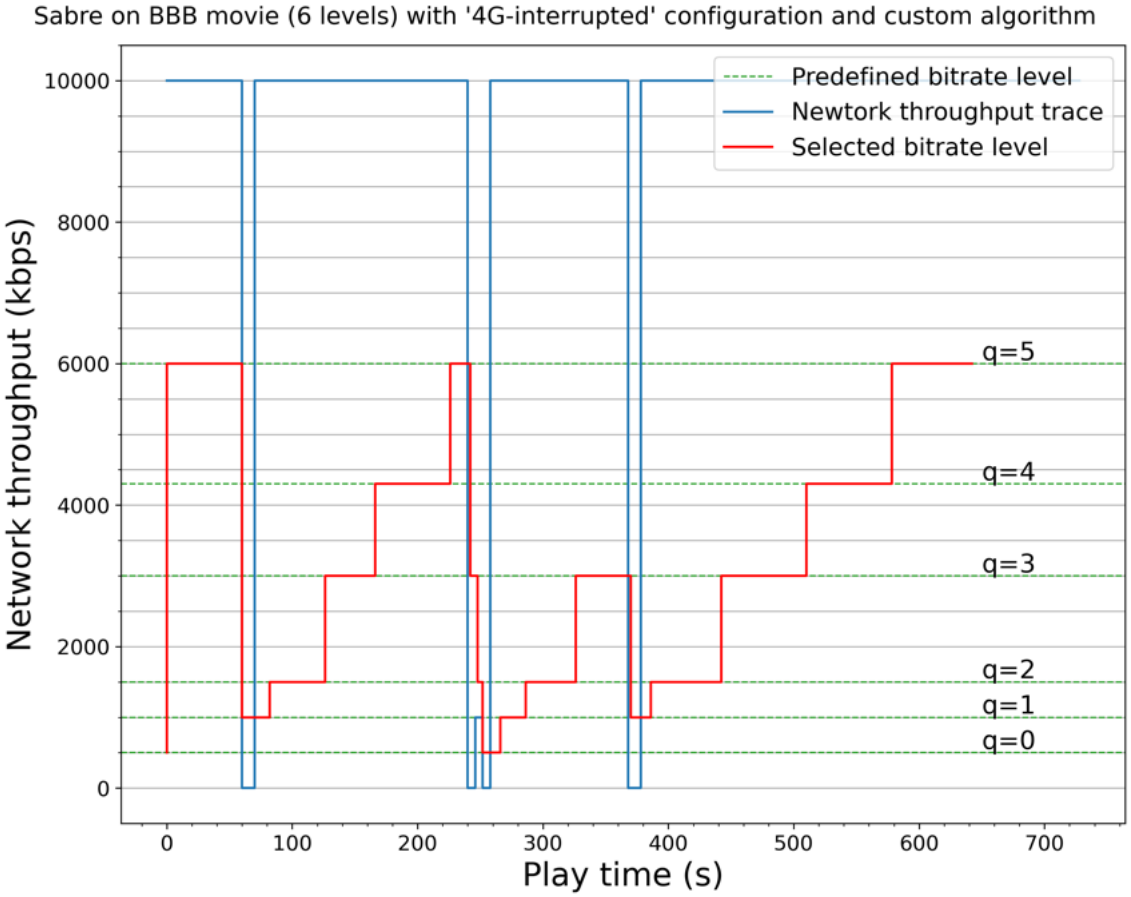


9.7. New ABR ladder VMAF measurement

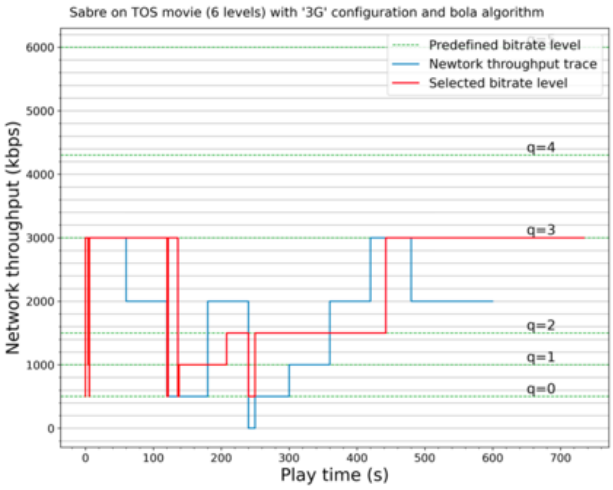




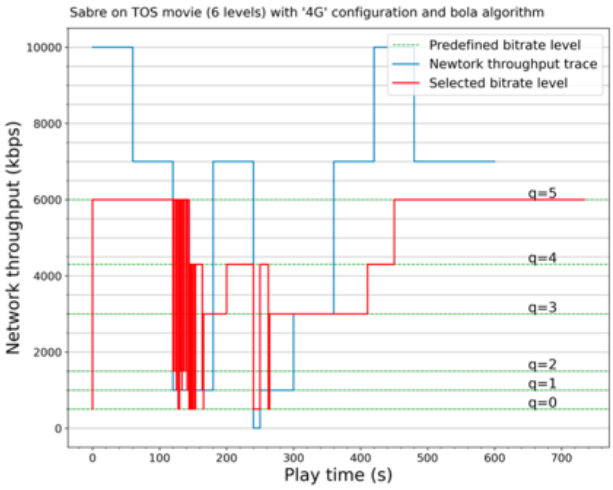
9.8. Default algorithm simulation



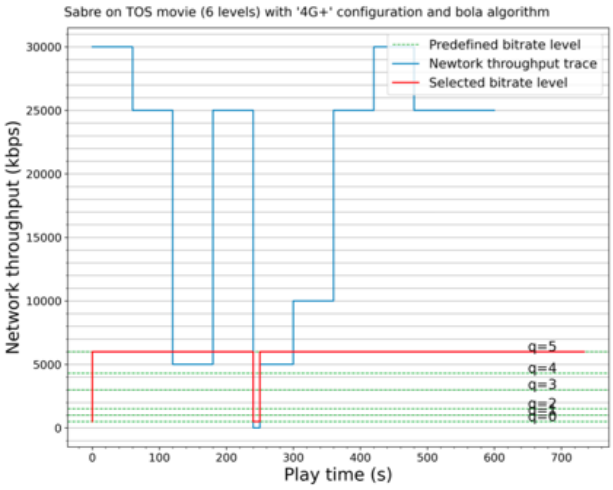
9.9. BOLA simulations



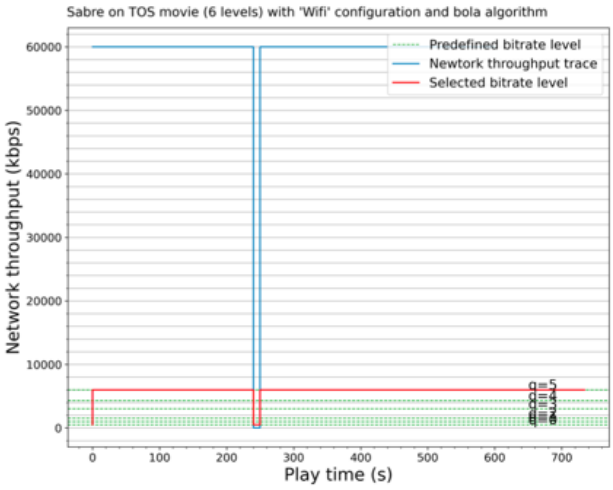
(a) 3G network trace



(b) 4G network trace

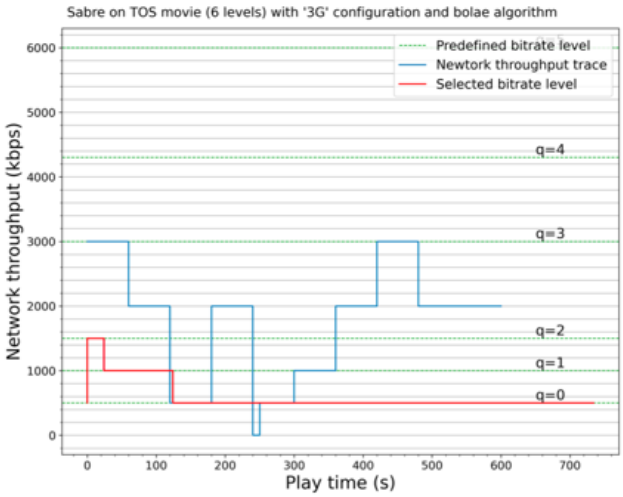


(c) 4G+ network trace

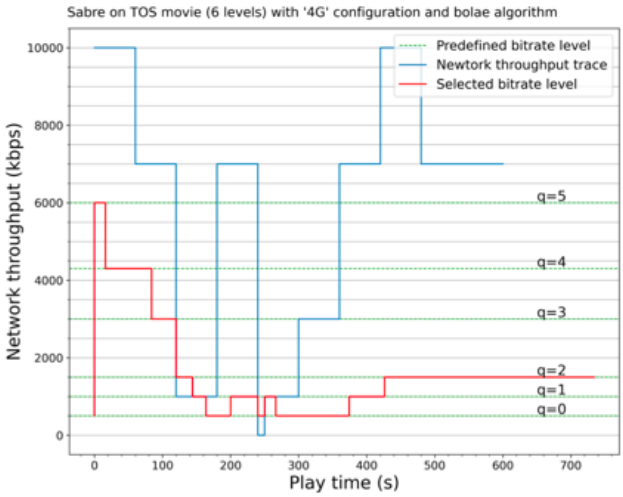


(d) Wi-Fi network trace

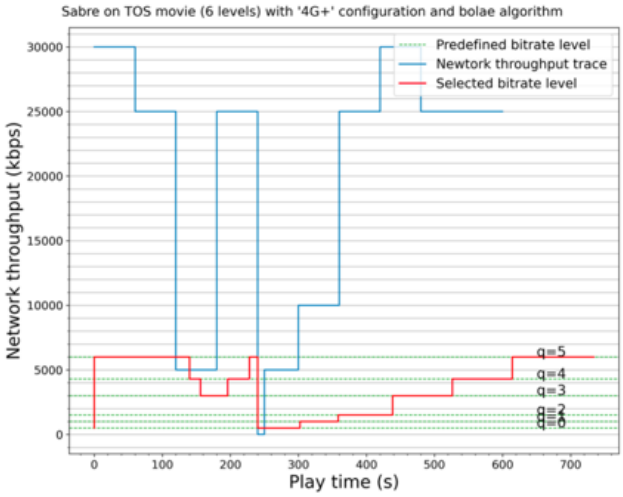
9.10. BOLAE simulations



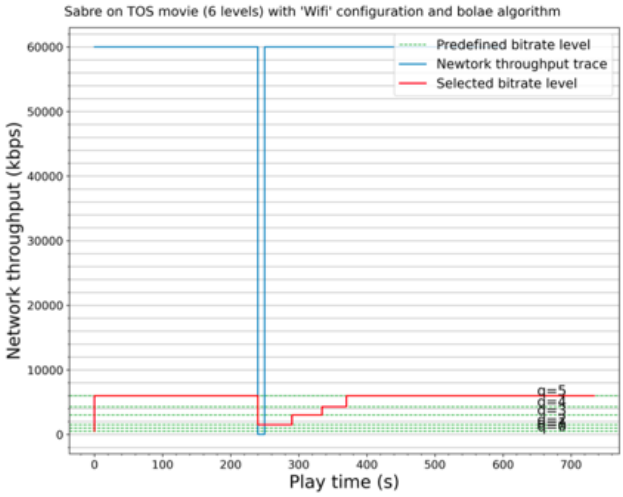
(a) 3G network trace



(b) 4G network trace

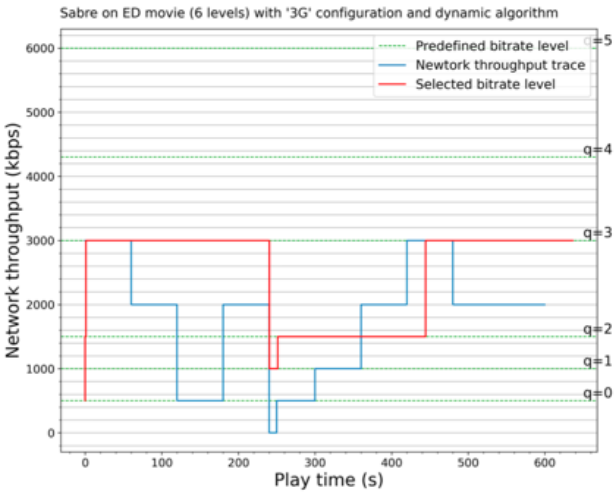


(c) 4G+ network trace

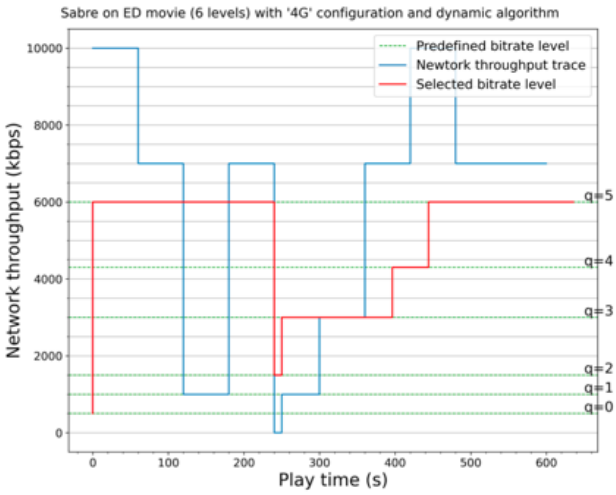


(d) Wi-Fi network trace

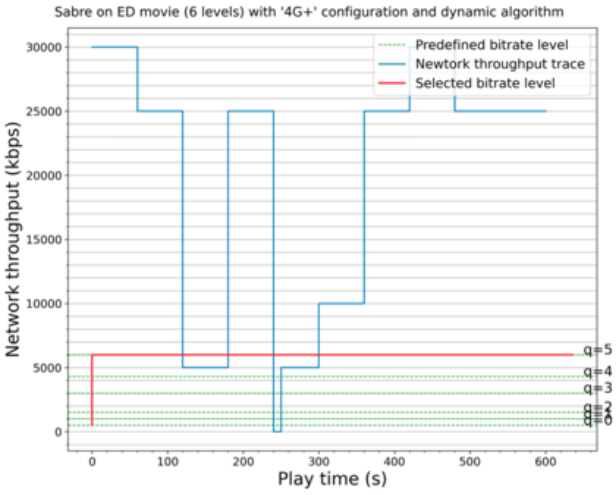
9.11. DYNAMIC simulations



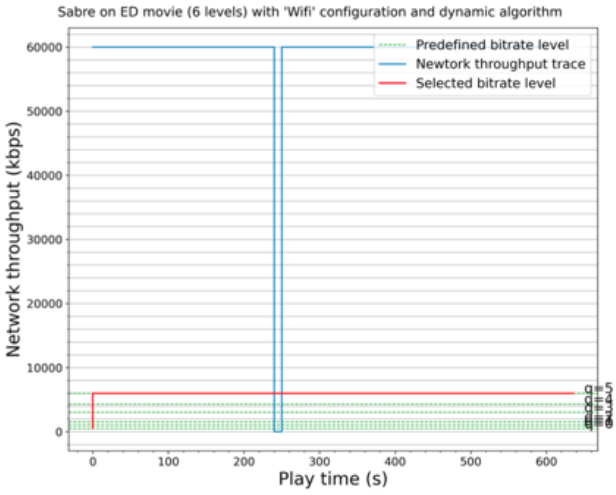
(a) 3G network



(b) 4G network



(c) 4G+ network



(d) Wi-Fi network