EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems

Research Article EAI.EU

QoE-Energy Consumption Optimization for End-User Devices in Adaptive Bitrate Video Streaming Using the Lagrange Multiplier Method

Tien Vu Huu[*,1], Thao Nguyen Thi Huong[2]

[1]Faculty of Multimedia, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam; tienvh@ptit.edu.vn
[2]Faculty of Electronics Engineering, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam; thaonth@ptit.edu.vn

Abstract

The reduction of greenhouse gas emissions in the Internet and ICT sectors has become a critical challenge. According to recent research, the key contributors to greenhouse gas emissions in Internet include high energy consumption factors such as data centers, transmission network devices, and end-user devices. Among Internet services, video streaming is one of the services having the highest traffic volume and number of users. Consequently, developing energy-efficient solutions for video streaming networks, particularly for end-user devices, is an urgent research priority. Reducing energy consumption in end-user devices in a video streaming system often requires compromises in parameters that impact the quality of user experience (QoE). Therefore, achieving an optimal trade-off between minimizing energy consumption and maintaining an acceptable QoE is a key objective. In this study, a cost function that integrates QoE and energy consumption is developed using the Lagrange multiplier method. Based on this function, an adaptive bitrate algorithm is proposed to select optimal video segments for video players, ensuring maximum QoE while minimizing energy consumption. The performance of the proposed method is evaluated using various types of video samples under varying network bandwidth conditions. Experimental results show that the proposed method reduces energy consumption of end-user devices by up to 5.9% and enhances QoE by 3.9% compared to previous methods.

Received on 01 February 2025; accepted on 30 March 2025; published on 14 April 2025

Keywords: Video streaming, Adaptive Bitrate, QoE, Energy consumption

doi:10.4108/eetinis.v12i3.8587

1. Introduction

According to recent research by the global climate consulting organization Carbon Trust (UK) on CO2 emissions [1], the Information and Communications Technology (ICT) sector is one of the ten sectors having the highest emissions beside industries such as steel, plastic and cement production. In the ICT sector, three components contribute to the process of emissions including data centers, transmission networks and end-user devices.

Data centers are places where centralized servers are located to process and store data for all activities on the Internet. Research in [1] show that these data centers consume about 208TWh of electricity, equivalent to emitting about 2% of global CO2 emissions per year. Besides data centers, data transmission networks also contribute to CO2 emissions. Although the amount of electricity consumed per GB gradually decreases each year due to the development of transmission equipment manufacturing technology, the amount of data on the network increases each year. Therefore, the total energy consumption of the network still increases year by year. The third factor, which is a major contributor to CO2 emissions, is terminal devices including computers and

*Corresponding author. Email: tienvh@ptit.edu.vn

mobile devices with more than 50% of the emissions of the entire ICT network. Among applications at end-user, video content traffic accounts for more than 70% of total Internet traffic and is forecast to grow more in the near future [2]-[3]. This explosion in traffic is mainly due to the rapid increase in video streaming services (video streaming, video conferencing, video on demand), the increasement of video quality (2K, 4K, 8K) and the increasement in the number and diversity of terminal devices (TV, PC, mobile devices). It can be said that video streaming is one of the services that significantly contribute to CO2 emissions on the Internet. Hence, proposing solutions to reduce energy consumption (EC) with the goal of reducing CO2 emissions is one of the urgent tasks today.

Recently, many studies in the field of video streaming have focused on reducing energy consumption for components in this network. In [4], energy consumption models for the entire network are proposed. The research indicated that the total energy consumption of a video sequence at server side is sum of the encoding energy, the decoding energy and energy consumed for storing that video in server. The energy consumption model of the transmission network encompasses the energy usage of all network components involved in the video content delivery process. The energy consumption model at client side equals to the sum of energy consumption in process of sending requests, downloading and playing back video content.

On the server side, authors in [5] analyzed energy consumption for x265 encoder at different bitrate values. To measure energy consumption, authors used tools to measure energy consumption of each part of the system, CPU, DRAM and cache. The encoder is configured at different coding modes to measure energy consumption. The results show that when the bitrate in coding modes is decreased, the energy consumption is increased and vice versa. In addition, the results also indicate that CPU account for 95% of the total energy consumption, followed by DRAM at 3% and the cache memories consumes the lowest energy at 2%.

On the client side, based on the given energy consumption models, several studies have focused on reducing energy consumption for end-user devices [6]-[7]. Along with the issue of energy consumption, the quality of user experience (QoE) at client side in video streaming system is also an issue that needs to be considered. Quality of experience is defined as the level of satisfaction or dissatisfaction of users when using a service or application [8]. Due to the abundance of video sources, users will easily stop consuming the video content if its quality is not as expected [9]. Therefore, to satisfy users' expectations, content providers also need to guarantee an acceptable QoE level besides the target of reducing EC. Additionally, previous researches also showed that energy consumption is strong correlated to QoE. High-quality video means a large amount of data to transmit, leading to end-user's devices need more energy to process. Consequently, achieving a balance between reducing energy consumption and maintaining acceptable QoE at the client side in video streaming systems remains a critical challenge. In [10], a survey is conducted to investigate user preferences regarding energy saving in video streaming. Specifically, the study examines whether users are willing to compromise QoE to save energy and the extent to which they are willing to reduce video streaming quality for energy efficiency. The survey results indicate that reducing QoE to an acceptable level, rather than maintaining maximum QoE, is a viable approach to lowering energy consumption.

In [6], an Arduino based low-cost platform is designed to measure the energy consumption of the smart mobile device. The experiment results of this method show that the energy consumption at end-user device depends on radio interface and bitrate of network. In particular, the end-user device using WiFi has higher energy efficiency than one using LTE network. In addition, the higher quality level of network results in the lower energy consumption for end-user device. In [7], a QoE model is proposed, which defines QoE metric as the perceived video quality reduced by impairments caused during data transmission and quality fluctuations during playback. In addition, an energy consumption model is formulated by estimating a quadratic function to represent the relationship between energy consumption, video bitrate, and wireless signal strength. Based on these models, an optimization problem is formulated with the objective of minimizing energy consumption while maximizing QoE.

To date, many methods for optimizing QoE and EC have been proposed. However, to enhance system performance by maximizing QoE and minimizing EC, the QoE and EC estimation models must be accurate. Therefore, in this study, a new QoE and EC estimation model is proposed. Specifically, an EC model is developed based on the bitrate levels of video segments downloaded by the client. Then, a Lagrange cost function for QoE and EC is constructed based on the two QoE and EC models. Based on the Lagrange cost function, an adaptive bitrate selection algorithm is proposed to determine the optimal bitrate level for each video segment.

The structure of this paper is organized as follows. Section 2 introduces the background and related works including HTTP Adaptive Streaming (HAS) technology, QoE and EC models as well as joint optimization between QoE and EC algorithms for video streaming system. Section 3 describes the proposed EC

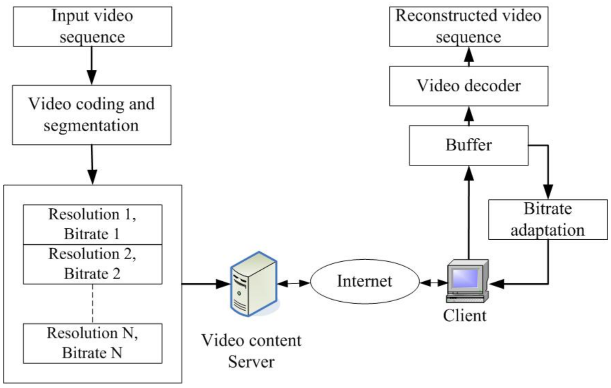

Figure 1. HTTP Adaptive Streaming System Architecture.

model, QoE-EC optimal function and adaptive bitrate selection algorithm. Afterward, Section 4 describes the experiments and evaluates the system's performance. Finally, Section 5 presents the conclusions.

2. Related works

2.1. HTTP Adaptive Birate Video Streaming

In recent years, The most popular technique for streaming video over the Internet has been adaptive streaming via the HTTP Hypertext Transfer Protocol or HAS (HTTP Adaptive Streaming). HAS offers several benefits and partially addresses cost-related challenges. Using HTTP, providers can use web servers instead of expensive equipment to reduce costs significantly. Furthermore, HAS allows media packets, particularly input videos, to pass through firewalls and network address translators (NAT) efficiently and seamlessly.

Figure 1 provides an overview of the HAS architecture. On the server side, the original video sequence is divided into equal time segments and encoded at different quality levels corresponding to each resolution and bitrate. The higher the resolution and bitrate, the better the video quality. On the client side, the video player selects video segments at an appropriate quality level, sends requests to the server, and downloads them using the HTTP protocol. Once downloaded, the client stores the received video segments in a buffer, decodes the segments sequentially, and displays them on the user's device.

As illustrated in Figure 1, the content server provides N different quality versions of the video. The adaptive bitrate (ABR) algorithm on the client side selects the most suitable segment based on network conditions, buffer status, or energy consumption, and sends a request to the server accordingly. For instance, when the network bandwidth is in good condition, the client can request higher-quality segments of the video. Conversely, when the network bandwidth is poor, the client may request lower-quality segments. This approach helps prevent the client's buffer from depleting, thereby avoiding playback interruptions.

Currently, several protocols have been developed based on the HTTP platform, including Apple's HTTP Live Streaming (HLS) [11], Microsoft's Smooth Streaming [12], and Adobe's HTTP Dynamic Streaming [13]. However, due to the lack of synchronization among these protocols, the Moving Picture Experts Group (MPEG) and the International Organization for Standardization (ISO) introduced the MPEG-DASH standard in 2012 [14]. The second version of the DASH standard was released in 2014, with further updates and new features introduced in 2017.

2.2. QoE Model

In general, parameters affecting QoE may include initial delay, stalling frequency, stalling duration, video quality, switching frequency, and rebuffering frequency [15] [16]. However, these parameters in a video streaming system are interdependent. For instance, a method aimed at enhancing video quality may increase stalling duration because high video quality typically corresponds to a higher bitrate and resolution. Therefore, to address the trade-offs between parameters, the proposed methods attempt to establish a QoE model that integrates multiple parameters and then maximize the QoE value.

To date, many QoE estimation methods have been proposed and standardized, but two main approaches exist: using Mean Opinion Score (MOS) and using utility score. The MOS-based model, also known as a non-reference model because it is based on subjective evaluation of the viewer and does not need to be compared with the original video. In [17], subjective experiments are conducted to derive the impairment function for factors including initial delay, stall and quality level variation. Then, a comprehensive user experience model is formulated from the derived functions. In [18], a model was developed to explore scene statistics of luminance coefficients to quantify image distortion. The advantage of this method is that it accurately reflects the perception of the human visual perception. However, this method is difficult to implement at end-user device due to its high computational complexity. On the other hand, the methods using utility score are more widely proposed in ABR algorithms due to their real-time QoE estimation capability. In [19], the utility score is estimated by the sum of three parameters: bitrate, quality level change, and rebuffering. The formula for calculating the utility score for a set of M consecutive video chunks is given as follows.

$$Utility = \sum_{n=1}^{M} Q(R_n) - \mu T - \xi \sum_{n=1}^{M-1} |Q(R_{n+1}) - Q(R_n)| \quad (1)$$

where $Q(R_n)$ is the function that measures user-perceived quality based on the bitrate R_n of the n^{th} chunk. $Q(R_n)$ is the penalty incurred after each change in quality level. μ and ξ are the weights for the penalties associated with rebuffering and quality variability, $\mu = 3000$ and $\xi = 1$. T denotes the total rebuffering time. In the calculation of the utility score, the first element represents the bitrate utility, while the second and third elements correspond to the penalties for rebuffering and bitrate changes, respectively. Various ABR algorithms incorporate different variations of this QoE calculation model, with optimized bitrate computations and penalty weights [20] - [22]. In [23], a QoE model is built based on bitrate as following function:

$$QoE = a \times r^b + c \quad (2)$$

In which r denotes bitrate while a, b, and c are the coefficients varied according to video resolutions. Although Eq. (2) has the advantage of computational simplicity, it models QoE solely as a function of bitrate, without accounting for critical influencing factors such as buffer depletion or quality fluctuations. In contrast, Eq. (1) incorporates three key QoE components: user-perceived quality, rebuffering penalty, and quality fluctuation penalty. By incorporating such both short-term factors (instantaneous quality) and long-term factors (quality stability and buffering impact), Eq. (1) is commonly used in various methods for estimating QoE values [24], [25].

2.3. Energy Consumption Model for End-User Devices

As mentioned above, the energy consumption of user devices is one of the key factors contributing to the overall energy consumption of the network. Therefore, reducing the energy consumption of user devices can significantly lower the total energy consumption of the online video streaming system. To achieve this reduction, an energy consumption estimation model for user devices must be developed. Previous studies have employed two primary approaches to estimate energy consumption: hardware-based method [6] and software-based method [26]. In [6], a low-cost device based on an Arduino board was designed to measure the energy consumption of a smartphone. In this device, the energy consumption P is modeled in LTE and Wifi network as follow:

$$P = r_d \times R + r_t + \nu \quad (3)$$

where r_d is the energy consumption rate for data transfer, measured in $mJoule/kbit$, r_t is the energy consumption per unit of time in $mWatt$, R is the average bitrate of the video in $kbps$ and ν is a constant value. An advantage of the above model is that it accounts for signal strength across different network types. However, a limitation of the model lies in its assumption that power consumption is a linear function of bitrate. This simplification may not accurately reflect the nonlinear processing behavior of devices in adaptive bitrate video streaming networks.

Instead of using hardware, a software-based approach is introduced in [26], where an application called eLens is presented. In this application, the energy consumption cost of each instruction in the source code is estimated. The proposed method demonstrates that the estimation accuracy exceeds 90%, and its runtime is acceptable. The advantage of this method is that it can be adapted across various platforms, provided that an energy cost model per instruction is available, thereby increasing its versatility. However, it requires a detailed energy model that maps each instruction type (or API call) to its corresponding energy cost.

2.4. QoE-EC Model

In some previous methods, QoE and EC are typically considered independent values. Therefore, the objectives of these methods are either to maximize QoE or to minimize energy consumption. As a result, the ABR algorithm on the client side tends to request higher bitrate for video chunks to enhance QoE. This may lead to an increase the energy consumption of the user device due to the more processing requirements associated with larger data volumes. To address the growing energy challenges in video streaming systems, the QoE-EC optimization algorithms have been proposed in [27]-[29]. The goal of QoE-EC optimization algorithms are to maximize the QoE and to minimize the energy consumption simultaneously. To achieve this target, a cost function incorporating both QoE and EC is formulated, and factors related to QoE and EC are adjusted to minimize the cost function. In [27], an algorithm is proposed to optimize QoE and EC by adjusting the quality and brightness of videos at the server. Then, on the client side, the proposed algorithm selects video chunks with predetermined quality and brightness levels based on network bandwidth. In this algorithm, although the trade-off between QoE and EC is considered, sudden changes in quality and brightness between video chunks are not accounted for. In [28], an algorithm for optimizing QoE and EC in mobile video systems is proposed. In this algorithm, QoE is measured in MOS while the energy consumption model is based on coding time and energy consumption at each coding layer. Based on these QoE and EC models, the particle swarm optimization (PSO) method is employed to solve the optimization problem. Although this algorithm significantly reduces energy consumption, it also results in a considerable reduction in QoE compared to an approach that selects bitrate based solely on the QoE model..

To address the limitations of the above algorithms, in [29], method called E-WISH introduces a total cost function incorporating throughput cost, buffer cost, quality cost, and energy cost. The optimal bitrate is then determined for each video segment by minimizing the total cost function. In particular, the total cost function is computed as follows:

$$C(i) = \alpha C_t(i) + \beta C_b(i) + \gamma C_q(i) + \delta C_e(i) \tag{4}$$

where i represents the index of the bitrate level. $C_t(i)$, $C_b(i)$, $C_q(i)$, and $C_e(i)$ represent the throughput cost, buffer cost, quality cost, and energy cost of a video chunk with bitrate R_i, respectively. The weights α, β, γ, and δ are computed as 0.074, 0.203, 0.723, and 0.1, respectively. In this method, the EC model is estimated as a linear function of frame rate, resolution, and bitrate:

$$C_e(i) = w_1 f_i + w_2 r_i + w_3 R_i \tag{5}$$

where f_i and r_i are frame rate and resolution of video chunk. The w_1, w_2, and w_3 are selected as 0.19, 6.29×10^{-8}, and 3.522×10^{-4}, respectively.

In Eq. (4), the cost function is designed as a linear combination of QoE-related costs (throughput cost, buffer cost, and quality cost) and energy consumption cost in Eq. (5). The bitrate of video segment is then selected to minimize this overall cost function. However, in this study, the constraint between QoE and EC is not explicitly considered. Furthermore, the cost function in this approach is nonlinear with respect to bitrate, necessitating the use of differentiation to determine its extrema. To effectively solve the constraint between QoE and EC, we employ the Lagrange multiplier method to derive the optimal bitrate that minimizes the cost function. This method is particularly suitable for optimization problems involving extrema of nonlinear cost functions under constraints.

3. Proposed method

3.1. Architecture of the proposed framework

Figure 2 illustrates the architecture of the proposed framework. In this framework, the QoE model and EC model are designed to estimate QoE and EC values corresponding to different bitrate levels. These estimated QoE and EC values, along with the buffer occupancy value—which represents the duration of video content stored in the buffer—are provided to the adaptive bitrate selector. The adaptive bitrate selector aims to determine the optimal bitrate that maximizes QoE while minimizing energy consumption. Based on the input parameters and the QoE estimation model, the adaptive bitrate selector estimates QoE values for all available bitrate options. The bitrate that yields the highest QoE value is selected for the next video chunk. Subsequently, a request containing the chunk index and its corresponding bitrate is sent to the video server.

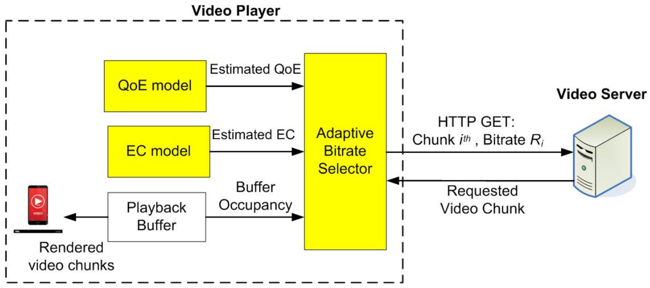

Figure 2. Architecture of the proposed framework.

3.2. QoE Estimation Model

As mentioned above, the QoE metric is evaluated based on several factors, including initial delay, stalling frequency, stalling duration, switching frequency, video quality. However, a video streaming system cannot achieve the best conditions of all factors simultaneously because of constraints between factors. For example, a video system that attempts to improve video quality under limited bandwidth conditions may experience an increase in stalling frequency. Therefore, in QoE evaluation, only a subset of these factors is considered, depending on the strategy of each method. In this article, the model from [19] is used to estimate the QoE metric based on factors such as stalling frequency, switching frequency, and video quality for adaptive chunk selection. By integrating these factors, the QoE model provides a more accurate estimation of QoE compared to simpler models that only rely on bitrate. This makes the model well-suited for the bitrate optimization framework proposed in the paper, where the goal is not only to maximize visual quality but also to minimize rebuffering and ensure smooth playback, all of which are essential to delivering high-quality user experience. Specifically, the QoE estimation model in this proposed method is computed as follows:

$$QoE(R) = R - \mu\left(\frac{R \times L}{C} - B_c\right) - |R - R_{n-1}| \tag{6}$$

where R is the bitrate of the current video chunk being considered for download. L is length of the video chunk, C is the tredicted network bandwidth, B_c is the current buffer occupancy, R_{n-1} is the bitrate of the

Table 1. Energy consumption at different bitrates for the "Big Buck Bunny Video" sequence

Bitrate (kbps)	Energy consumption (mWs)
100	3063
200	3071
300	3209
—	—
11000	3551
12000	3672

previously downloaded chunk. In this work, the simple sliding percentile algorithm in ExoPlayer [32] is used to predict network bandwidth.

3.3. Energy Consumption Model

In general, the energy consumption estimation model for user devices may depend on various factors, including screen brightness, signal strength, hardware processing capability, frame rate, bitrate and resolution [29], [30], [31]. However, apart from the available bitrate and resolution levels of video segments on the server, which the video player can select, the remaining factors are external and cannot be adjusted by the video player. Therefore, in this study, the EC estimation model is established based on variations in bitrate and resolution, while other factors remain constant. To establish the EC model as a function of bitrate, the video sequence Big Buck Bunny at five resolutions including 448×252, 592×332, 768×432, 1280×720, and 1920×1080 is encoded with different bitrates from $100Kbps$ to $12Mbps$. Since energy consumption is not significantly affected by a specific video content, the estimated EC model can be applied to other video sequences that having the same parameters as the tested Big Buck Bunny video sequence.

In this experiment, a Samsung Galaxy Note 10 smartphone running Android 12 is used to measure energy consumption at each bitrate level. In addition, the open-source ExoPlayer [32] is used video playback and the Android API BatteryManager [33] is used to measure energy consumption. To collect data, each video sequence is played at 12 different bitrate levels and energy consumption is measured during each playback session. Table 1 presents the data points for the Big Buck Bunny video sequence, which has a duration of two minutes.

After testing with five video sequences, the least-squares regression method was employed to derive the fitted curve representing the relationship between energy consumption (EC) and bitrate (R). The data points and the fitted curve are illustrated in Figure 3

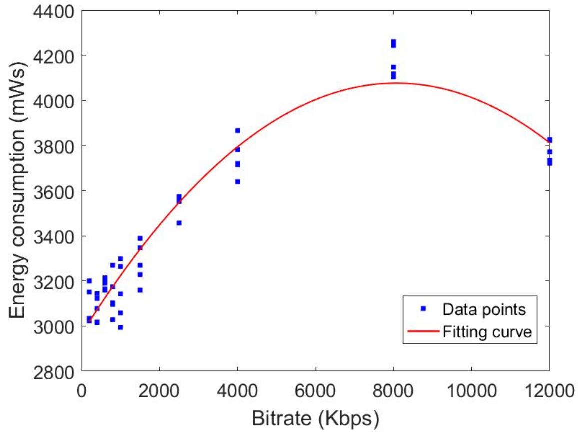

Figure 3. The fitting curve of EC function.

with the estimated energy function $EC(R)$ as follows:

$$EC(R) = -2 \times 10^{-5} \times R^2 + 0.3 \times R + 2965 \tag{7}$$

where EC denotes energy consumption, and R represents the bitrate of the video segments. The accuracy of the fitted curve was assessed using the R-squared (R^2) metric, which yielded a value of 0.93, indicating a high degree of fit.

The results presented in Fig.3 demonstrate that energy consumption is directly proportional to the bitrate when the bitrate is below $8000Kbps$, and inversely proportional when the bitrate exceeds $8000Kbps$. This trend arises from the fact that, at lower bitrates, the video player requires a longer duration to download video segments, keeping the device in an active state and thereby increasing energy consumption. Furthermore, the power consumption of the video decoder rises with increasing bitrate of video segments, contributing to higher overall consumption in the low-bitrate range. In contrast, when the bitrate exceeds $8000Kbps$, video segments are downloaded more rapidly, leading to faster buffer saturation. As a result, the video player transitions to an idle state sooner, thereby reducing energy consumption for downloading video segments. Consequently, beyond this bitrate threshold, the overall power consumption decreases in comparison to lower bitrate conditions.

3.4. QoE – Energy Consumption Optimization Algorithm

In an ABR video streaming system, the main task of the video player is to select the bitrate for each video chunk to maximize the QoE value. However, when considering both QoE and EC, the task of the video player is to select bitrate that maximizes QoE while minimizing energy consumption. To solve this problem, the Lagrange multiplier method is used to determine the optimal bitrate value at which QoE is maximized and EC is minimized.

Assuming that $QoE(R)$ is a function of QoE with respect to bitrate, and $EC(R)$ is a function of EC with respect to R. To maximize $QoE(R)$ and minimize $EC(R)$, the Lagrange cost function J is formulated as follows:

$$J = \frac{1}{QoE(R)} + \lambda EC(R) \tag{8}$$

in which λ is the Lagrange multiplier. The above problem can be reformulated as finding the solutions for R that minimize J. In Lagrange multiplier method, the cost function can be formulated in both maximal or minimal functions. However, this method often minimizes a cost function rather than maximizing it because, in many applications, we seek to minimize error, energy, or cost. Therefore, in this study, the minimizing J cost function is used because our objective is to minimize $EC(R)$.

In the extrema problem, the minimum of the J cost function is obtained by setting its derivative to zero, i.e.,

$$\frac{\partial J}{\partial R} = 0 \tag{9}$$

From (8) and (9), the Lagrange multiplier can be derived as:

$$\lambda = -\frac{\frac{\partial \frac{1}{QoE(R)}}{\partial R}}{\frac{\partial EC(R)}{\partial R}} \tag{10}$$

Substitute (6) and (7) into (10), the Lagrange multiplier is derived as:

$$\lambda = \begin{cases} -\frac{(2-\mu\frac{L}{C})}{((2-\mu\frac{L}{C})R+\mu B_c- R_{n-1})^2(-4.10^{-5}R+0.3)} & if\ R < R_{n-1} \\ -\frac{1-\mu\frac{L}{C}}{((1\ -\mu\frac{L}{C})R+\mu B_c)^2\left(-4.10^{-5}R+0.3\right)} & if\ R = R_{n-1} \\ \frac{\mu\frac{L}{C}}{(-\mu\frac{L}{C}R+\mu B_c+R_{n-1})^2\left(-4.10^{-5}R+0.3\right)} & if\ R > R_{n-1} \end{cases} \tag{11}$$

Substitute (12) into (8), the Lagrange cost function can be computed as:

$$J = \begin{cases} \frac{1}{(2-\mu\frac{L}{C})R+\mu B_c-R_{n-1}} - \lambda EC(R) & if\ R < R_{n-1} \\ \frac{1}{(1-\mu\frac{L}{C})R+\mu B_c} - \lambda EC(R) & if\ R = R_{n-1} \\ \frac{1}{-\mu\frac{L}{C}R+\mu B_c+R_{n-1}} - \lambda EC(R) & if\ R > R_{n-1} \end{cases} \tag{12}$$

Based on the above Lagrange cost function, an EC-based adaptive bitrate (EC_ABR) algorithm is proposed as shown in the pseudocode in Algorithm 1.

Algorithm 1 EC-based adaptive bitrate algorithm

Input: R_{n-1}: Bitrate of the previous downloaded chunk;
R_i: Bitrate at quality level i^{th} of the video chunk index n^{th};
J^i_{cost}: The Lagrange cost at bitrate R_i;
N: The number of quality levels of video sequence;
L: The length of video chunk;
B_c : The current buffer occupancy;
C: The predicted bandwidth;
J_{min} := Integer.MAX_VALUE.
Output: R^*: Optimal bitrate for video chunk n^{th}

for $(i = 0; i >= N; i++)$ **do**

$$J^i_{cost} = \begin{cases} \frac{1}{(2-\mu\frac{L}{C})R_i+\mu B_c-R_{n-1}} - \lambda EC(R_i) & if\ R_i < R_{n-1} \\ \frac{1}{(1-\mu\frac{L}{C})R_i+\mu B_c} - \lambda EC(R_i) & if\ R_i = R_{n-1} \\ \frac{1}{-\mu\frac{L}{C}R_i+\mu B_c+R_{n-1}} - \lambda EC(R_i) & if\ R_i > R_{n-1} \end{cases}$$

  if $(J^i_{cost} < J_{min} \& R_i \leq C)$ **then**
    $J_{min} = J^i_{cost}$;
    $R^* = R_i$;
  else
    $R^* = R_0$;
  end if
end for
return R^*;

In the proposed EC_ABR algorithm, the objective of the loop is to determine the optimal bitrate R^* for chunk n^{th} among the available bitrates at different quality levels to achieve the highest QoE metric while minimizing energy consumption. For each bitrate value, from the lowest to the highest, corresponding to the lowest to the highest video quality levels, the algorithm computes Lagrange cost J_{cost} as in (12). If J_{cost} is minimal and the bitrate is lower than predicted network bandwidth, the considered bitrate is selected for downloading the next video chunk. Otherwise, the lowest bitrate is selected.

4. Experiment and Performance Evaluation

4.1. Experiment Setup

In this experiment, a testbed for an HTTP-based video streaming system is built on a Node.js web server. To simulate different scenarios, the network bandwidth between the server and the client is shaped using Traffic Control tool [34] which is used for managing and manipulating packet transmission. On the client side, ExoPlayer is employed to implement the proposed adaptive bitrate selection algorithm.

To validate the performance of the proposed method, four video sequences representing four video categories—sports, animation, action films, and documentaries—are utilized in a dataset conforming to the latest version of the MPEG-DASH standard [35]. The experiments are conducted under four different network bandwidth scenarios, including low-speed ($200Kbps$ and $500Kbps$), medium-speed

Table 2. Resolutions and bitrates of video test sequences

Video sequences	Type	Duration (second)	Resolutions	Bitrate (Mbps)
Big Buck Bunny	Animation	300	480x270, 640x360, 1280x720,1920x1080	0.257, 0.512, 0.815, 1.5, 2.4,3.0, 4.1, 6.1, 10.1
Tears of Steel	Action movie	300	480x270, 640x360, 1280x720, 1920x1080	0.257, 0.512, 0.815, 1.5,2.4, 3.0, 4.1, 6.1, 10.1
Red Bull Playstreets	Sport	300	320x240, 480x360, 854x480,1280x720, 1920x1080	0.102, 0.152, 0.202, 0.252, 0.302,0.401, 0.501, 0.701, 0.9, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0
Of Forest And Men	Documentary	300	320x240, 480x360, 854x480, 1024x576	0.48, 0.94, 0.138, 0.189, 0.234, 0.280, 0.370, 0.470, 0.561, 0.652, 0.837, 1.0, 1.3, 1.5, 1.8, 2.2, 2.6, 3.3, 3.9

Table 3. Comparison of estimated energy consumption models

Video sequences	Benchmark	E-WISH		p_ECM	
	EC (mWs)	EC (mWs)	MAE	EC (mWs)	MAE
Big Buck Bunny	3930	1230	**2126**	3681	**344**
Tears of Steel	3979	2378		3627	
Red Bull Playstreets	3271	1760		3829	
Of Forest and Men	3839	1147		3620	

(1000$Kbps$ and 2000$Kbps$), and high-speed (5000$Kbps$ and 10000$Kbps$). These bandwidth levels are commonly employed to simulate Wi-Fi, 3G, and 4G network conditions for video streaming systems [3].

To evaluate the accuracy of the proposed energy consumption model (p_ECM), four video sequences listed in Table 2 are used. These source videos are encoded at standard resolutions range from 480p (480270) to 1080p (19201080) corresponding to bitrates from 0.1$Mbps$ to 10$Mbps$ [36]. The encoded video sequences are then divided into equal-length segments in duration of 2 seconds, and stored in .m4s files according to MPEG-DASH standard.

4.2. Energy Consumption Evaluation

To evaluate the accuracy of the proposed energy consumption model (p_ECM), four video sequences listed in Table 2 are used. First, the benchmark energy consumption is measured using the Android BatteryManager API. Then, the four video sequences are played again to estimate energy consumption using Eq. (5) and Eq. (7). Table 3 presents a comparison between the p_ECM and the E-WISH model. As the results show, the Mean Absolute Error (MAE) of the energy consumption in E-WISH is 2126, while that of p_ECM is 344. The lower accuracy of E-WISH's method is due to its energy consumption model being computed using a simple linear combination of resolution, bitrate, and frame rate. In contrast, the p_ECM establishes the energy consumption model through a regression method based on real measured energy consumption data.

4.3. QoE-Energy Consumption Performance Evaluation

As mentioned above, QoE and EC are interdependent quantities. To evaluate the effectiveness of QoE and EC optimization methods, the Bjontegaard metric [37]-[38] is used in this work. This metric is commonly applied to assess the performance of video encoders using two

Table 4. Comparison of QoE-EC performance of methods

Video	Resolution	Exo		E-WISH		EC_ABR		EC_ABR vs. Exo		EC_ABR vs. E-WISH	
		EC (mWs)	QoE	EC (mWs)	QoE	EC (mWs)	QoE	BD-EC	BD-QoE	BD-EC	BD-QoE
Big Buck Bunny	200 Kbps	5210	31902	5221	31503	5181	32102	-3.28	611.66	-0.07	1774.30
	500 Kbps	5185	32250	5061	32370	5090	33480				
	1 Mbps	4944	33540	4610	33980	4214	35910				
	2 Mbps	4727	36090	4627	36910	4686	37350				
	5 Mbps	4692	37940	4592	38210	4487	39700				
	10 Mbps	4613	38250	4521	38376	4413	39853				
Tears of Steel	200 Kbps	4997	43571	4802	44160	4589	46153	-6.05	3.76	-5.44	-96.86
	500 Kbps	4943	43840	4710	44020	4544	46570				
	1 Mbps	4880	47550	4720	47710	4230	48580				
	2 Mbps	4413	48215	4319	49250	4010	51740				
	5 Mbps	4079	48800	3968	49320	3810	53750				
	10 Mbps	4012	49517	3910	49730	3798	53910				
Red Bull Playstreets	200 Kbps	5138	38190	5019	38279	5121	38429	-14.91	617.17	4.02	5128.45
	500 Kbps	5043	38210	4990	38800	4810	41280				
	1 Mbps	4985	37940	4815	39240	4887	39980				
	2 Mbps	4357	59050	4157	60314	4156	61200				
	5 Mbps	4345	66280	4045	67439	4300	66180				
	10 Mbps	4289	66428	4016	67830	3989	66472				
Of Forest And Men	200 Kbps	4897	49865	4728	51012	4659	52892	0.62	389.64	0.06	494.25
	500 Kbps	4866	50200	4721	51231	4644	52940				
	1 Mbps	4536	51460	4415	52212	4313	54050				
	2 Mbps	4381	52600	4291	53719	4157	55750				
	5 Mbps	4256	55320	4131	57421	3978	61060				
	10 Mbps	4132	55810	4078	57987	3873	61984				
Average								**-5.90**	**405.56**	**-0.36**	**1825.03**

values: video quality (measured by PSNR in dB) and bitrate (measured in Kbps or Mbps). In the Bjontegaard metric, BD-PSNR and BD-Rate are used to quantify how much one codec improves over another in terms of PSNR and bitrate. In this work, the Bjontegaard metric is applied to compare the performance of the proposed EC_ABR method with ExoPlayer and E-WISH in terms of QoE and EC. Similar to RD performance evaluation in video coding, a positive BD-QoE indicates that the QoE of the proposed method is higher than that of the other methods, while a negative BD-EC indicates that the energy consumption of the proposed method is lower than that of the other methods.

Table 4 presents the QoE-EC performance comparison of three methods: ExoPlayer, E-WISH and EC_ABR. As the results show, compared to ExoPlayer, the average BD-EC is −5.9, meaning that EC_ABR reduces energy consumption by 5.9% compared to ExoPlayer while BD-QoE is 405.7 higher than that of ExoPlayer (Approximately a 0.9% increase in average QoE across all bandwidth cases). Compared to E-WISH, the energy consumption of EC_ABR is lower by 0.4 while the BD-QoE is higher by 1825. This means that EC_ABR can reduce energy consumption by 0.4% and improve QoE by 3.9% over E-WISH for the same bandwidth value.

Figure 4 illustrates the average QoE and energy consumption of four video sequences at six bandwidth values. The results indicate that QoE values tend to be directly proportional to network bandwidth, whereas

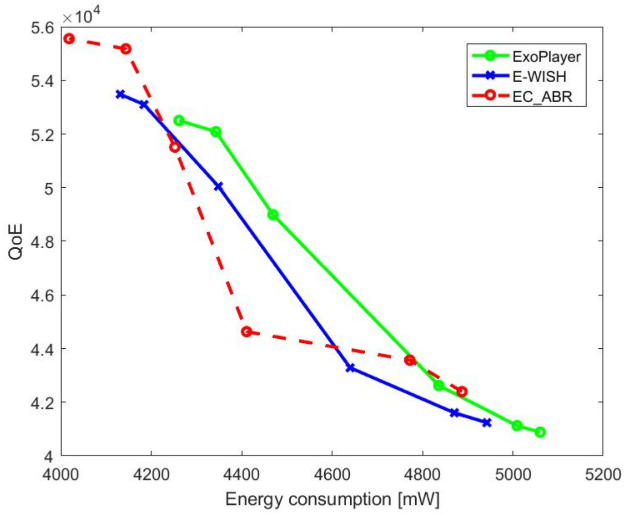

Figure 4. QoE-EC comparison between methods.

EC values exhibit an inverse relationship with network bandwidth. At the highest bandwidth level ($10Mbps$), the QoE values of three methods are highest, while EC is lowest. Conversely, at the lowest bandwidth level ($0.2Mbps$), the QoE values of three methods are lowest, whereas EC reaches its highest for all three methods. The reason is that when the network

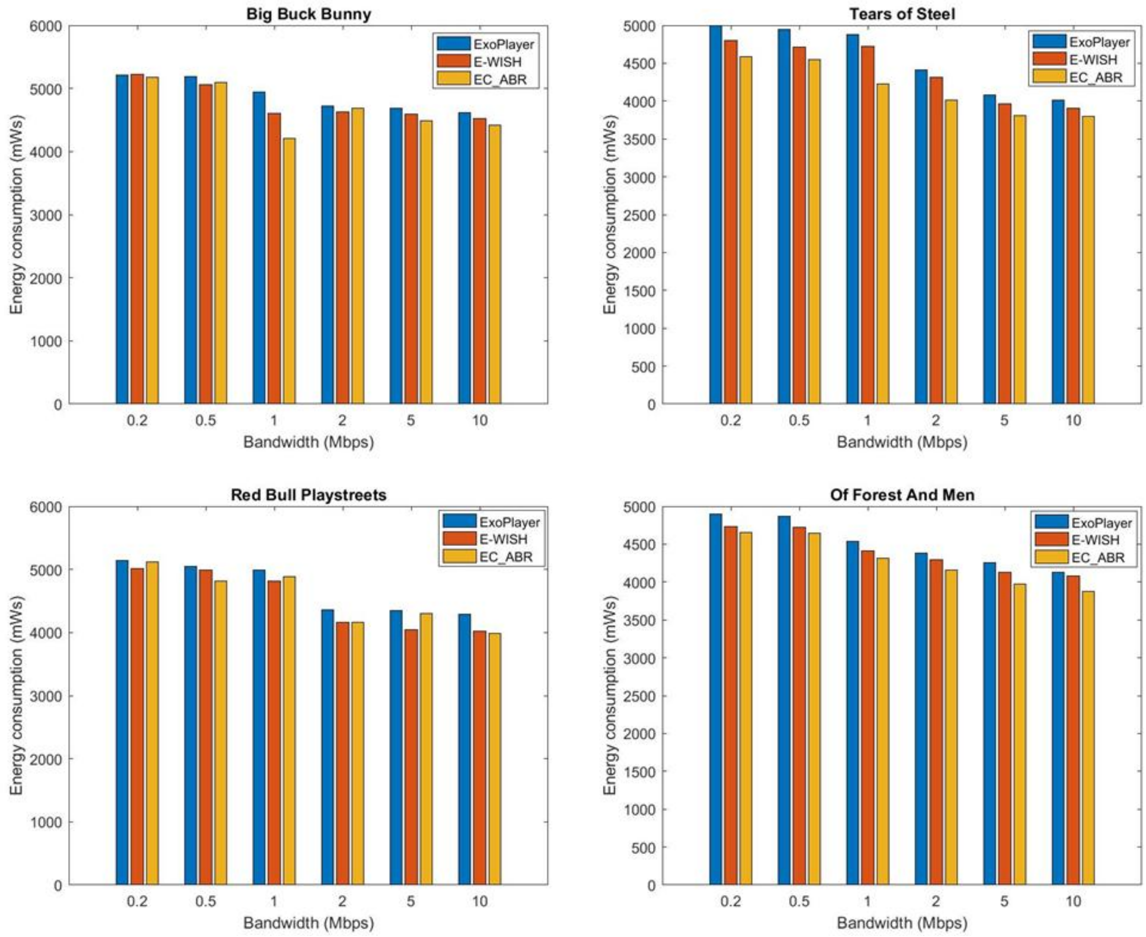

Figure 5. Comparison of energy consumption across different methods for four video sequences.

bandwidth is at its maximum value of $10Mbps$ (upper-left corner points of the curves), the ABR algorithms of all three methods are capable of selecting higher-bitrate video segments for download. According to the QoE formulations in Eq.(4) (used by E-WISH) and Eq.(6) (used by ExoPlayer and EC-ABR), higher bitrates lead to higher QoE values. Moreover, at high bandwidth levels, video segments are downloaded more quickly, allowing the video player to remain in an active state for a shorter duration. Consequently, energy consumption is reduced. Conversely, when the network bandwidth is at its lowest value of $0.2Mbps$ (lower-right corner points of the curves), the ABR algorithms of the methods tend to select lower-bitrate video segments. As a result, QoE decreases while energy consumption increases due to the prolonged active state of the video player.

Figure 4 also shows that at the same bandwidth value, the proposed method EC_ABR can achieve lower energy consumption and higher QoE compared to the other methods. In particular, at the highest bandwidth level of $10Mbps$, the energy consumption of EC_ABR is approximately $4000mWs$, while that of E-WISH and ExoPlayer is approximately $4100mWs$ and $4200mWs$, respectively. At the same bandwidth level, the QoE of EC_ABR reaches the highest value of approximately 55554, whereas the QoE of E-WISH and ExoPlayer are lower, at approximately 53480 and 52500, respectively. Similarly, at the lowest bandwidth level of $0.2Mbps$, the EC and QoE of EC_ABR are approximately 4887 and 42394, respectively. Meanwhile, the EC of E-WISH and ExoPlayer is higher, at approximately 4942 and 5060, respectively. QoE values of E-WISH and ExoPlayer are lower, at approximately 41238 and 40882, respectively.

Figure 5 illustrates the energy consumption of three methods at different bandwidth levels. The results show that the proposed method has the lowest energy consumption, followed by E-WISH and ExoPlayer. This is because energy consumption is not considered in the adaptive bitrate algorithm of ExoPlayer. To improve the adaptive bitrate algorithm, energy consumption is taken into account in the QoE model of the E-WISH algorithm. However, in this method, the energy consumption model is a linear function of resolution, bitrate, and framerate, which has lower accuracy than the quadratic polynomial model of EC_ABR. Consequently, the proposed EC_ABR can select a more optimal bitrate compared to the other methods, achieving lower EC while maintaining higher QoE.

Figure 5 also shows that when the available bandwidth decreases, the ABR algorithm tends to select lower-quality video segments corresponding to lower bitrates, and the download speed is also reduced. Consequently, the playback buffer is depleted more rapidly, leading to playback stalls. As a result, the video player must repeatedly send requests to the server or frequently operates with an empty buffer, both of which contribute to increased energy usage. Conversely, when the bandwidth increases, the ABR algorithm selects higher-quality video segments, and the download rate to the buffer improves. As the buffer is replenished more effectively, the likelihood of playback stalls is reduced, leading to improved QoE. Furthermore, higher bandwidth conditions eliminate the need for repeated server requests and prevent interruptions in segment downloading, thereby reducing the frequency of buffer depletion. As a result, the energy consumption of the video player is significantly reduced.

5. Conclusion

In this work, we propose a method to reduce energy consumption and enhance QoE simultaneously based on the Lagrange multiplier method. In particular, an energy consumption function of bitrate is estimated. Then, a cost function incorporating both QoE and energy consumption is formulated using the Lagrange multiplier method. Based on the cost function, an adaptive bitrate algorithm is proposed to estimate the optimal bitrate for the next video chunk, allowing the video player to achieve the highest performance in terms of energy consumption and QoE. The results show that the proposed method achieves up to a 5.9% reduction in average energy consumption and a 3.9% increase in average overall QoE compared to some previous methods.

References

[1] Cacbon Truth, "Carbon impact of video streaming", [Online]. Available: https://www.carbontrust.com/our-work-and-impact/guides-reports-and-tools/carbon-impact-of-video-streaming (last accessed : Dec. 2024).

[2] Ericsson. (2022)., "Mobile data traffic outlook," [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast (last accessed : Dec. 2024).

[3] C. Bezerra, A. De Carvalho, D. Borges, N. Barbosa, J. Pontes and E. Tavares, "QoE and energy consumption evaluation of adaptive video streaming on mobile device," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2017, pp. 1-6, doi: 10.1109/CCNC.2017.8016294.

[4] C. Herglotz, M. Kränzler, R. Schober, and A. Kaup, "Sweet Streams Are Made of This: The System Engineer's View on Energy Efficiency in Video Communications [Feature]," IEEE Circuits Syst. Mag., vol. 23, no. 1, pp. 57–77, 2023, doi: 10.1109/mcas.2023.3234739.

[5] D. Silveira, M. Porto, and S. Bampi, "Performance and energy consumption analysis of the X265 video encoder," 25th European Signal Processing Conference, EUSIPCO 2017, vol. 2017-Janua. pp. 1519–1523, 2017. doi: 10.23919/EUSIPCO.2017.8081463.

[6] L. Zou, A. Javed, and G.-M. Muntean, "Smart mobile device power consumption measurement for video streaming in wireless environments: WiFi vs. LTE," in 2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2017, pp. 1–6. doi: 10.1109/BMSB.2017.7986151.

[7] X. Chen, T. Tan, G. Cao, and T. F. La Porta, "Context-Aware and Energy-Aware Video Streaming on Smartphones," IEEE Trans. Mob. Comput., vol. 21, no. 3, pp. 862–877, 2022, doi: 10.1109/TMC.2020.3019341.

[8] K. Brunnström et al., Qualinet White Paper on Definitions of Quality of Experience. 2013.

[9] F. Dobrian et al., "Understanding the impact of video quality on user engagement," SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 362–373, Aug. 2011, doi: 10.1145/2043164.2018478.

[10] G. Bingol, S. Porcu, A. Floris, and L. Atzori, "An Analysis of the Trade-Off Between Sustainability and Quality of Experience for Video Streaming," 2023, pp. 1600–1605. doi: 10.1109/ICCWorkshops57953.2023.10283614.

[11] HTTP Live Streaming, [Online]. Available: https://developer.apple.com/streaming/ (last accessed Dec., 2024).

[12] Microsoft Smooth Streaming, Online]. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-sstr/ (last accessed : Dec. 2024).

[13] Adobe HTTP Dynamic Streaming, [Online]. Available: https://helpx.adobe.com/adobe-media-server/dev/dynamic-streaming.html (last accessed : Dec. 2024).

[14] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," in IEEE MultiMedia, vol. 18, no. 4, pp. 62-67, April 2011, doi: 10.1109/MMUL.2011.71.

[15] 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 Commun. Surv. Tutorials, vol. 17, no. 1, pp. 469–492, 2015, doi: 10.1109/COMST.2014.2360940.

[16] Konstantoudakis, K., Breitgand, D., Doumanoglou, A. et al. Serverless streaming for emerging media: towards 5G network-driven cost optimization. Multimed Tools Appl 81, 12211–12250 (2022). doi.org/10.1007/s11042-020-10219-7.

[17] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and Validating User Experience Model for DASH Video Streaming," IEEE Trans. Broadcast., vol. 61, no. 4, pp. 651–665, 2015, doi: 10.1109/TBC.2015.2460611.

[18] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," IEEE Trans. Image Process., vol. 21, no. 12, pp. 4695–4708, 2012, doi: 10.1109/TIP.2012.2214050.

[19] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 2015, pp. 325–338. doi: 10.1145/2785956.2787486.

[20] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-Optimal Bitrate Adaptation for Online Videos," IEEE/ACM Trans. Netw., vol. 28, no. 4, pp. 1698–1711, 2020, doi: 10.1109/TNET.2020.2996964.

[21] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2018. Oboe: auto-tuning video ABR algorithms to network conditions. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18). Association for Computing Machinery, New York, NY, USA, 44–58. doi.org/10.1145/3230543.3230558.

[22] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in Proceedings of the Conference of the ACM Special Interest Group on Data Communication, 2017, pp. 197–210. doi: 10.1145/3098822.3098843.

[23] M. Mu et al., "A Scalable User Fairness Model for Adaptive Video Streaming Over SDN-Assisted Future Networks," IEEE J. Sel. Areas Commun., vol. 34, no. 8, pp. 2168–2184, 2016, doi: 10.1109/JSAC.2016.2577318.

[24] Schwenzer M., Ay M., Bergs T. et al., "Review on model predictive control: an engineering perspective," Int J Adv Manuf Technol 117, 1327–1349 (2021). doi.org/10.1007/s00170-021-07682-3.

[25] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, Feng Qian, and Zhi-Li Zhang, "A variegated look at 5G in the wild: performance, power, and QoE implications," In Proceedings of the 2021 ACM SIGCOMM 2021 Conference (SIGCOMM '21). Association for Computing Machinery, New York, NY, USA, 2021, 610–625. doi.org/10.1145/3452296.3472923

[26] S. Hao, D. Li, W. G. J. Halfond and R. Govindan, "Estimating mobile application energy consumption using program analysis," 2013 35th International Conference on Software Engineering (ICSE), San Francisco, CA, USA, 2013, pp. 92-101, doi: 10.1109/ICSE.2013.6606555.

[27] B. Varghese, G. Jourjon, K. Thilakarathne, and A. Seneviratne, "e-DASH: Modelling an energy-aware DASH player," in 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017, pp. 1–9. doi: 10.1109/WoWMoM.2017.7974320.

[28] S. Zhou, M. Ran, and Z. Lu, "Adaptive energy-efficient and QoE-aware optimization method for mobile video services," in 2016 16th International Symposium on Communications and Information Technologies (ISCIT), 2016, pp. 388–392. doi: 10.1109/ISCIT.2016.7751657.

[29] D. Lorenzi, M. Nguyen, F. Tashtarian, and C. Timmerer, "E-WISH: An Energy-aware ABR Algorithm For Green HTTP Adaptive Video Streaming," in Proceedings of the 3rd Mile-High Video Conference, 2024, pp. 28–33. doi: 10.1145/3638036.3640802.

[30] C. Herglotz, W. Robitza, M. Kränzler, A. Kaup and A. Raake, "Modeling of Energy Consumption and Streaming Video QoE using a Crowdsourcing Dataset," 2022 14th International Conference on Quality of Multimedia Experience (QoMEX), Lippstadt, Germany, 2022, pp. 1-6, doi: 10.1109/QoMEX55416.2022.9900886.

[31] M. Ghasempour, H. Amirpour and C. Timmerer, "Real-Time Quality- and Energy-Aware Bitrate Ladder Construction for Live Video Streaming," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 15, no. 1, pp. 83-93, March 2025, doi: 10.1109/JETCAS.2025.3539948.

[32] Exoplayer, "Exoplayer", [Online]. Available: https://developer.android.com/reference/androidx/media3/exoplayer/ExoPlayer (last accessed : Dec. 2024).

[33] Android, "BatteryManager", [Online]. Available: https://source.android.com/docs/core/power/device (last accessed : Dec. 2024).

[34] "Linux Traffic Control", [Online]. Available: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9 (last accessed : Dec. 2024).

[35] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in Proceedings of the 3rd Multimedia Systems Conference, 2012, pp. 89–94. doi: 10.1145/2155555.2155570.

[36] J. Vlaović, S. Rimac-Drlje and D. Žagar, "Influence of Segmentation Parameters on Video Quality in Dynamic Adaptive Streaming," 2020 International Symposium ELMAR, Zadar, Croatia, 2020, pp. 37-40, doi: 10.1109/ELMAR49956.2020.9219029.

[37] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD-curves," 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:61598325 (last accessed: Dec. 2024).

[38] J. J. S. Pateux, "An Excel Add-in for Computing Bjontegaardmetric and its Evolution," ITU-VCEG, VCEG-AE07, vol. 2007, [Online]. Available: https://github.com/tbr/bjontegaard_etro (last accessed : Dec. 2024).