A Hybrid Energy-Aware Video Bitrate Adaptation Algorithm for Mobile Networks

Fábio Araújo*, Denis Rosário*, Eduardo Cerqueira*, and Leandro A. Villas†
* Federal University of Pará (UFPA), Belém – Brazil
† Institute of Computing, University of Campinas (UNICAMP), Campinas – Brazil
Email: {fabioraraujo, denis, cerqueira}@ufpa.br, leandro@ic.unicamp.br

***Abstract*—The number of mobile devices that use video streaming applications has been steadily rising year after year. Platforms responsible for providing multimedia service face great challenges in delivering high-quality content for mobile users. One main problem in sharing video flows is the high energy consumption in mobile devices, which reduces their lifetime. A video adaptation approach with Quality of Experience (QoE) support is a key issue to increase the user experience while watching videos as well as reduce the energy consumption in mobile devices. In this paper, we propose a hybrid energy-aware video bitrate adaptation algorithm to deliver videos with high QoE and energy-efficiency for mobile users. Simulation results show the efficiency of the proposed algorithm compared to existing adaptation video bitrate algorithms, reducing the number and duration of stalls, as well as saving energy.**

I. Introduction

In recent years, video streaming applications are increasingly present in our daily life, due to the popularization of video applications and the advances in video adaptation techniques. According to Cisco forecasts [1], the traffic from mobile devices transmitted over wireless networks will account for more than 63% of total IP traffic by 2021. Specifically, video traffic will represent 82% of all Internet traffic, and video-on-demand (VoD) services, such as YouTube, Netflix, will nearly double by 2021 [1]. This creates a demand for content delivery mechanisms to provide Quality of Experience (QoE) support during multimedia dissemination such as HTTP streaming [2].

In HTTP streaming schemes, the video is divided into multiple chunks/segments, and each one can be requested with a different bitrate version. In general, they can be categorized into two type: *Fixed video bitrate streaming* and *Adaptive Streaming*. The fixed bitrate considers that users fix a video bitrate, from the beginning to the end of video download, which is the simplest way of implementation. On the other hand, HTTP Adaptive streaming (HAS) services allow users to request each segment with a different birate version. In the beginning, the client makes an HTTP request to obtain the meta-data of the different audio and video representations, which is described by the Media Presentation Descriptor (MPD) file [2]. The client requests an appropriate bitrate version based on the available transmission resources and device processing capabilities. The main goal of HAS systems is to choose the adequate bitrate level for the next segments to deliver the video with QoE support, *i.e.*, maximizing bitrate while minimizing the stall probability and avoiding bitrate switches [3]. This is because poor QoE results in the viewer abandoning the video completely [4].

In HAS systems, adaptation techniques are divided into two categories: server-side approaches and client-side approaches [5]. Server-side HAS approaches are the most effective adaptation techniques for scenarios where users want to watch the entire video. On the other hand, mobile devices waste scarce network and storage resources by downloading the video that the users do not watch the whole video. As reported by Finamore et al. [6], 60% of videos are watched for less than 20% of their duration. Based on this, our study is focused on client-side HAS approaches, since the mobile devices are the best network entity to quickly detect the QoE degradation during the video player, and quickly react to the network changes [3].

Client-side HAS approaches can be classified into rate-based, buffer-based, and hybrid-based [7]. Rate-based algorithms rely on throughput estimation from previous chunks to request a higher video bitrate for the following chunks. Alternatively, buffer-based algorithms take into account buffer occupancy, where they consider different thresholds to request the video bitrate in order to keep the buffer occupancy at a given level. Finally, hybrid-based algorithms combine both approaches to adapt the video bitrate, obtaining better results due to the capacity of synchronizing the buffer occupancy and the available network bandwidth.

In hybrid-based HAS systems, it is important to predict the buffer empty and overload, which causes stall events and the waste of network resources. For instance, the buffer will be empty, as soon as the throughput of the video streaming application is lower than the video bitrate. In this way, the video playback cannot continue as soon as there is insufficient data available in the buffer, resulting in a poor QoE. It is also important to predict the connection throughput in order to estimate the average transfer capacity and the instantaneous throughput for VoD services.

Additionally, energy consumption is also a key metric for video adaptation, since multimedia services over battery-powered mobile devices is an energy-intensive task [8]. However, current HAS schemes do not consider energy as an important factor to adapt videos' bitrate. As a result, the QoE, in general, is poor in scenarios that mobile devices do not have enough energy to playback a video in an acceptable bitrate

ISBN 978-3-903176-13-3

level, due to the device turning off during video playback [9]. For instance, users prefer a less than excellent but acceptable video quality when offered higher energy saving [9].

In this paper, we propose the Energy-Saving Based Adaptation Algorithm (ESBA). It considers the client-side and hybrid-based techniques for video adaptation and takes into account devices characteristics (devices' energy and maximum supported resolution), connection throughput (estimated by a Machine Learning Process) and a probability of stalls (computed based on buffer information) to request video segments with different bitrates. In this sense, ESBA delivers videos with high QoE and energy-efficiency for mobile users. Simulation results demonstrate the efficiency of the ESBA in transmitting adapted video with energy efficiency 50% higher than videos transmitted by existing adaptation mechanisms for mobile and static clients. Moreover, results also show that ESBA delivers videos with QoE support, *i.e.*, ESBA has equal normalized bitrate and lower number and duration of stalls compared to existing adaptation mechanisms in a scenario composed of mobile and static clients.

The rest of this paper is organized as follows. Section II outlines existing works and their main drawbacks to providing video transmission QoE and energy-efficiency support. Section III describes the proposed the ESBA algorithm. Section IV discusses the simulation description and results. Finally, Section V introduces the conclusions and future works.

II. Related Work

Amour et al. [10] proposed a framework named Optimized Quality of DASH (OQD), which considers QoE to adapt the video bitrate. OQD implemented a Machine Learning-based Method to predict users Mean Optimal Score (MOS) by considering three important networks and application QoE Influence Factors, *i.e.*, bandwidth, video quality, and video dropped frame. OQD selects the suitable video quality segment based on the predicted MOS in order to maximize the bandwidth usage. However, they do not use a strategy to minimize stall events, causing a negative impact on QoE. In addition, they do not consider energy for decision making.

Saamer et al. [11] examined how Smooth Streaming, Netflix and Adobe OSMF react to persistent and short-term changes in the available bandwidth in the underlying network. In this way, they proposed an adaptation algorithm, referred as Adaptech Streaming, which aims to detect persistent and short-term available bandwidth variations in a timely manner in order to provide smooth bitrate variations and avoid video freezes.

Coelho et al. [12] presented an HAS strategy, named of Va-QoE-Adapt. It considers a pre-stall state for the decision process about the video bitrate adaptation based on the available transmission resources. In this way, a low bitrate representation is accessed only under special conditions, *i.e.*, pre-stall state, in order to mitigate stalls events. In addition, Va-QoE-Adapt takes decisions about bitrate switching based on QoE related parameters to minimize playback stalls.

Zhao et al. [13] proposed a dynamic adaptive algorithm in order to keep a high QoE for the average user's experience. Authors formulated QoE optimization that can be utilized in bandwidth and buffer-based approaches. The estimated bandwidth was captured by the weighted sliding window based bandwidth estimator named Sliding Percentile (SP). Their results were obtained by their empirical network traces and show that their approach works stably under different network conditions.

Dubin et al. [14] proposed a new crowd algorithm that maximizes QoE in video streaming services. Their algorithm estimates the current segment download path based on the client's location and speed. It predicts the future path network bandwidth conditions based on the client's playout buffer and the crowd estimated bandwidth. Authors used an estimated Mean Opinion Score (eMOS) [15] formula to measure the QoE and their geo-predictive adaptive logic algorithm tries to maximize it in order to obtain a better QoE.

Ayad et al. [16] investigated the detailed operations of the different players by code level analysis and through reverse engineering. Specifically, they presented the pseudo codes of three open source players and devise a method to obtain the detailed operation information, e.g., bitrate and buffer amount, of popular streaming players whose source codes are not publicly available. They conducted extensive experiments on their testbed and provided suggestions based on the behaviors of these players, including the repeated over-estimation of the available bandwidth, unfair bitrate selection when multiple players compete for the bandwidth, and insensitivity of Quick UDP Internet Connections (QUIC) protocol to the varying network bandwidth.

Based on our analysis of the state-of-the-art, we conclude that an effective strategy that provides a good QoE and also saves energy is not yet present in literature. In order to obtain an effective strategy, it is necessary to use parameters that are directly related to the QoE and new techniques to predict the quality of the network, as well as to consider the device power to provide a low frequency and time of the player stalls, and to avoid waste of energy.

III. ESBA Algorithm

This section describes ESBA, which is a client-side and hybrid-based video bitrate adaptation algorithm. ESBA takes into account information about energy, mobile devices, network, and buffer to request video segments with different bitrate level in order to provide energy-efficiency and QoE support. In this way, Figure 1 illustrates ESBA modules, namely, adaptation, buffer, network, and energy modules, as well as the communication flow. Initially, the client requests the video content from the HTTP server. Next, the server sends the MPD, which is an XML file that contains the information about video segments (R_i), as well as their relationships and information required for the client to choose between them. Afterward, the client requests the next segment in an appropriate bitrate version based on the decision taken by the ESBA adaptation module. In the following, we explain each module.

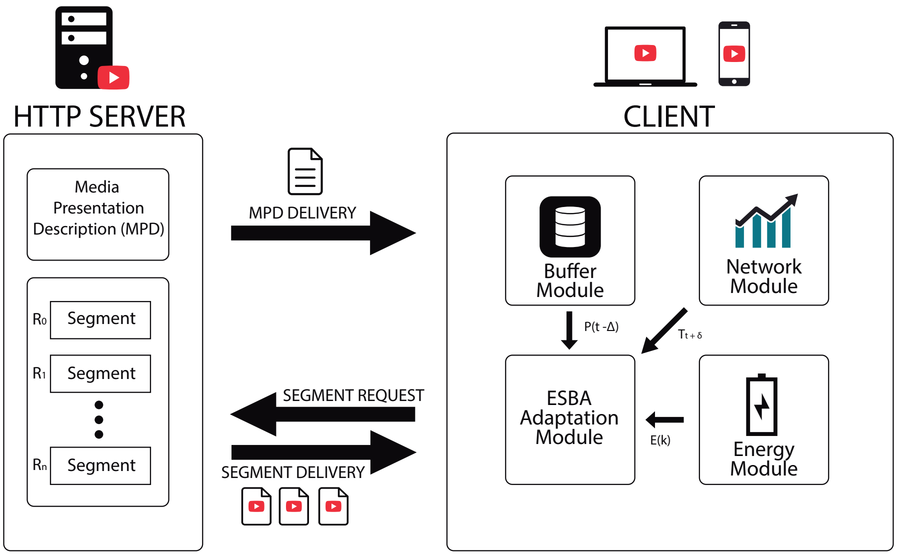


Fig. 1: ESBA Modules

The energy module is responsible for checking the current remaining energy of the mobile device (D_{energy}), and computing whether it has enough energy to playback the video at a specific bitrate level k. Specifically, the energy module evaluates the energy required to playback the video $E(K)$ in a given bitrate level based on Eq. (1). The energy module considers the number of segments (N), the energy power ($Power_k$) required to stream a video segment in the kth bitrate level, and also the battery voltage used in the devices ($Battery\ Voltage$). As result, higher the bitrate level requires more energy, since the mobile device will spend more energy to streaming the video at higher bitrate levels.

$$E(k) = \frac{\sum_1^N Power_k * 10^3}{Battery\ Voltage} \frac{(mW)}{(V)} \quad (1)$$

As shown in Algorithm 1, ESBA compares the energy required to playback the video $E(K)$ in the current bitrate level with the device remaining energy D_{energy}. As soon as the remaining energy is not enough to playback the next video segment in the current bitrate level (R_{cur}), ESBA reduces the bitrate level R_{cur-1} to evaluate whether the device has enough energy to playback the video segment with the bitrate level R_{cur-1}. This happens until it reaches the minimum acceptable bitrate level R_{min}. Otherwise, ESBA also evaluates buffer and network conditions to determine R_{next}, as soon as the device has the required energy to playback the video with the current bitrate level.

The buffer module is responsible for computing the stall probability ($P(t-\Delta)$) based on the buffer levels recorded in the last Δ seconds. In this sense, it checks the current buffer occupancy level by means of three thresholds, named as b_s , b_{cont} and b_{ov}. Particularly, the buffer module aims to capture the possibility of stalls, since the player stops the video playback as soon as the buffer level reaches the minimal buffer threshold (b_s). In addition, the buffer overload threshold (b_{ov}) captures the situation, where the player stops to request segments since the buffer is overloaded. Finally, the continuity threshold (b_{cont}) means the probability of session continuity over the next second.

The number and duration of stalls are the most important factor that impacts the QoE, due to the impact they directly cause on the continuity of the video session for the user [17]. In this way, the imminence of the video reproduction interruption can be given by the lack of energy, by the buffer occupation, or by the insufficient available bandwidth. In order to mitigate stalls, buffer module calculates the probability of stalls according to the Eq. (2), which presents the probability of stalls based on the buffer levels recorded in the last Δ seconds. The probability is calculated by the ratio of the number of times Nb_s the buffer level was below its minimum threshold b_s and the total number of observations made to the buffer N_B.

$$P(t-\Delta) = \frac{Nb_s(t-\Delta)}{N_B(t-\Delta)} \quad (2)$$

The network module is responsible for estimating the connection throughput. It is important to highlight that the connection throughput is highly non-linear and varies with time, and also changes abruptly when entering or leaving a congestion hour. We must devise an accurate prediction model to predict the dynamic nature of the traffic data. In this way, We use artificial neural networks (ANN), with their remarkable ability to learn from examples and derive meaning from complicated or imprecise data, to extract patterns and detect trends of connection throughput.

An ANN builds itself through the process of learning from the previous experience, which is called ANN training. By online learning, the ANN model can take into account the changes in the environmental conditions and adapt itself to the changes. These characteristics of the ANN have made it a potential solution for the prediction of connection throughput, which presents complicated short and long-range temporal dependence [10]. An ANN is formed from hundreds of single units, artificial neurons or processing elements (PE), connected with coefficients (weights), which constitute the neural structure and are organized in layers. The power of neural computations comes from connecting neurons in a network. Each PE has weighted inputs, transfer function, and one output. The behavior of a neural network is determined by the transfer functions of its neurons, by the learning rule, and by the architecture itself. The weights are the adjustable parameters and, in that sense, a neural network is a parameterized system.

In our study, we used a Multilayer Perceptron (MLP) to estimate the connection throughput and used the squashing function as the non-linear transfer function based on Eq. (3). An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation algorithm for training the network. Based on setup experiments, our MLP consists of 5 inputs (*i.e.*, previous connection throughputs), 3 hidden layers with 3 nodes each, and 1 output (*i.e.*, connection throughput).

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

One problem in the construction of an ANN model is the tradeoff between prediction accuracy and cost, where the cost includes training and prediction costs, which are related to the number of input parameters. In general, the prediction accuracy is better with more input parameters. However, more input parameters will lead to a higher cost in terms of processing and time. In this way, selecting an appropriate training size for network bandwidth prediction is another problem in ANN construction. In our setup experiments, our training and validation sets were generated through 26 simulations performed in NS3 in different types of scenarios (described in section IV), where 80% of the data was used to train a model and 20% to validate a model.

At a specific time t, the network module computes the average connection throughput $\hat{T}_{t-i\delta}$ while downloading each segment i accessed in the last $i\delta$ seconds based on the observed traffic data, where δ is the duration of a segment and uses them as the input of the trained ANN. As a result, the trained ANN estimates the connection throughput of the next δ seconds ($T_{t+\delta}$). It is important to note that the average connection throughput is calculated based on the number of bits received while downloading a video segment at a specific interval of time and not based on the video segment's bitrate.

Finally, ESBA adaptation module is responsible for receiving information from other modules and collecting information from the mobile devices in order to apply the bitrate adaptation algorithm. Due to the limited energy resources of mobile devices, sometimes they may not have enough energy to playback an entire video in a given bitrate level. In this sense, as shown in Algorithm 1, ESBA adaptation module evaluates the energy issue to select the adequate bitrate level (lines 3, 5 and 13). In case of insufficient energy to playback a video, ESBA reduces the bitrate level until the lowest acceptable bitrate in order to request the bitrate quality for the next segment (R_{next}). As a consequence, it enables users to watch videos as long as possible in an acceptable bitrate level, improving energy-efficiency and QoE. In addition, reducing the bitrate level increases the available throughput in the network, which potentially increases the bitrate level of existing VoD transmissions. In order to avoid waste of energy, ESBA adaptation module does not select bitrate level higher than the client display resolution, avoiding unnecessary download of segments that will not be used.

Algorithm 1: ESBA Algorithm

Result: R_{next}
1 $R_{next} \leftarrow R_{cur}$
2 $E_{cur} \leftarrow (Eq.1)$
3 **if** $D_{energy} > E_{cur}$ **then**
4   **if** $b_{cont} < B(t) < b_{ov}$ **then**
5   **if** $T_{t+\delta} < f(R_{cur})$ *and* $R_{cur} < R_{max}$ *and* $D_{energy} > E_{cur+1}$ **then**
6       $R_{next} \leftarrow R_{cur+1}$
7   **end**
8   **else**
9   **if** $b_s < B(t) < b_{cont}$ **then**
10   **if** $T_t < f(R_{cur})$ *and* $R_{cur} > R_{min}$ **then**
11         $R_{next} \leftarrow R_{cur-1}$
12   **else**
13   **if** $T_t < f(R_{cur})$ *and* $R_{cur} < R_{max}$ *and* $D_{energy} > E_{cur+1}$ **then**
14           $R_{next} \leftarrow R_{cur+1}$
15   **end**
16   **end**
17   **else**
18   **if** $B(t) \leq b_s$ *and* $R_{cur} > R_{escape}$ **then**
19   **if** $P(t-\Delta) < \Theta$ **then**
20           $R_{next} \leftarrow R_{min}$
21   **else**
22           $R_{next} \leftarrow R_{escape}$
23   **end**
24   **end**
25   **end**
26   **end**
27 **else**
28   **if** $R_{cur} > R_{min}$ **then**
29     $R_{next} \leftarrow R_{cur-1}$
30   **end**
31 **end**

ESBA takes into account the probability of stalls based on the buffer occupancy level $B(t)$ (lines 4 and 9), as well as the estimated throughput $T_{t+\delta}$ (lines 5, 11 and 14) computed by the Buffer module and Network module, respectively. In case of the buffer occupancy reaches an interval equal to or less than a minimum threshold (b_s) (line 18), the frequency of such event occurs is checked (line 19). If that event happens for at

ISBN 978-3-903176-13-3 © IFIP

least the number of times of a pre-stall threshold (Θ) (line 21), then the return of the ESBA algorithm R_{next} will have a bitrate equal to a low-quality bitrate level, named escape mode, to avoid stall events (line 22). Otherwise, the bitrate of the next segment will be R_{min}, the minimum acceptable bitrate, which higher than escape mode representation (line 20).

IV. Evaluation

This section describes the methodology and metrics used to evaluate ESBA, Adaptech, and VA-QoE HAS protocols. Afterward, we evaluate the impact of the different number of static and mobile clients on QoE and energy efficiency.

A. Scenario Description

We implemented ESBA on the network simulator 3 (NS-3) environment and conducted 33 simulations runs with different randomly generated seeds, and the results show the values with a confidence interval of 95%. The NS-3 simulation environment is responsible for controlling parameters related to the scenarios (*i.e.*, simulation area, number of nodes, mobility, etc.) and to the network (*i.e.*, bandwidth, loss model, power model, etc.). Table I summarizes the main simulation parameters.

We deployed 20 and 30 static or mobile nodes in order to evaluate the impact of node density and mobility. Each client is equipped with an 802.11n radio in a shared a wireless channel of 120 Mbps. We have conducted simulations with three adaptation algorithms, namely, ESBA, VaQoE-Adapt [12], and Adaptech [11]. Specifically, ESBA performs the adaptation decisions by using the low-quality representation and taking into account devices characteristics (devices' energy and maximum supported resolution), connection throughput (estimated by an MLP) and the probability of stalls (calculated by the Eq. 2), such as introduced in Section 3. On the other hand, VaQoE-Adapt does not consider devices' characteristics and only takes into account connection throughput (calculated by an exponential moving average of the flow), the probability of stalls and low-quality representation. Finally, Adaptech only takes into account connection throughput (also calculated by moving average) and the probability of stalls.

TABLE I: Simulation Parameters

Parameters	Values
Simulation Area	150 m^2
Number of Clients	20 and 30
Mobility Model	Static nodes and random way point
Node Speed	1-3 m/s
Request Distribution	Zipf distribution
Videos Sequence	Big Buck Bunny
Video Duration	280-630 seconds
Representation rates	1, 2.5, 5, 8 Mbps

Clients consider the scootplayer to request videos from the HTTP server, and also to watch the videos. The HTTP server distributes video content for each client request, where we considered seven versions of Big Buck Bunny video downloaded from the video library with different video duration, varying from 280s to 630s. We encoded the videos using H.264 codec at representation rates of 2.5 Mbps, 5 Mbps, 8 Mbps, and an additional representation rate of lower quality 1 Mbps. The different numbers of video's duration represent scenarios where a user watch the whole video, and also scenarios that a user watch a part of the video. The client request ratio follows a Zipf-like distribution, where the video of shorter duration is more priority than those of longer duration. The relative probability of requesting the ith most popular video is proportional to $\frac{\Omega}{i^{\alpha}}$, where Ω is a normalizing constant, and α is an exponent that varies from a trace to trace. That is, the request distribution does not follow the strict Zipf's law (with $\alpha = 1$), but follows a more general Zipf's like distribution.

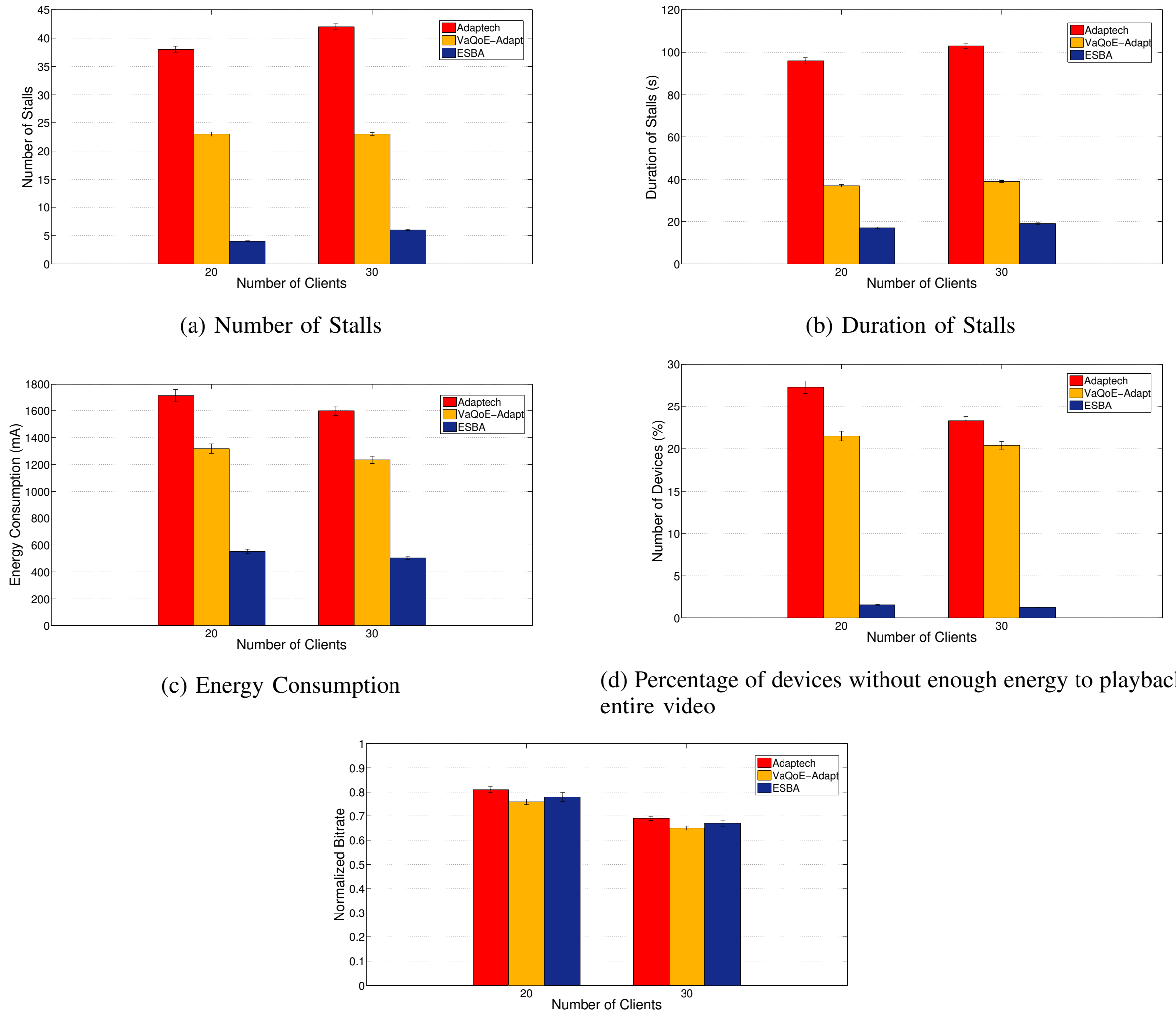
We applied well-known objective QoE metrics, namely, normalized bitrate, number of stall events and their duration. In addition, we considered energy metrics, *i.e.*, average current consumption and the number of devices without energy to playback the entire video in order to evaluate the performance of ESBA, Adaptech, and VA-QoE algorithms in scenarios composed of static and mobile clients. These metrics have a significant influence on the QoE, where unexpected values could result in the viewer abandoning the video service completely. The stall events indicate the number of times the buffer was emptied, and thus the video stopped playback. The stall duration measures the time duration when the playback of the video is temporarily stopped. In addition, it is important to analyze the normalized bitrate μ_n, which is computed based on Eq. (4). This is because it is possible to identify whether there is or not a waste of energy while the client is requesting a video segment/chunk.

$$\mu_n = \frac{\sum_{i=0}^{n}(R_i * D_i)}{T_s * R_{mb}} \tag{4}$$

where, i represents the i-th segment, R_i is the function that returns the segment bitrate i, D_i represents the segment length i, T_s represents the duration of the session and R_{mb} the maximum bitrate supported by the playback device.

B. Results

Figure 2 shows the QoE and energy results for video delivered by ESBA, Adaptech, and VaQoE-Adapt in a scenario composed of 20 and 30 static clients. By analyzing the results of Figures 2a and 2b, we conclude that ESBA significantly reduced the number and duration of stalls events, *i.e.*, ESBA reduced from 38 and 23 to 4 stall events compared to Adaptech and VaQoE-Adapt in the scenario with 20 static clients, while it reduced from 42 and 23 to 6 in the scenario with 30 static clients. On the other hand, ESBA reduced the duration of stall events from 100 s and 40 s to 20 seconds compared to Adaptech and VaQoE-Adapt in the scenario with 20 static clients, and similar behavior happens for the scenario with 30 static clients. The performance of ESBA is due to it combines information about the mobile device, network, buffer, and energy issues for decision making, which increases the available throughput without compromising the QoE. For



(a) Number of Stalls

(b) Duration of Stalls

(c) Energy Consumption

(d) Percentage of devices without enough energy to playback the entire video

(e) Normalized Bitrate

Fig. 2: QoE and energy results for scenario with 20 and 30 static clients

instance, devices with low energy to playback videos must request the minimum acceptable bitrate of 2.5 Mbps, increasing the available throughput in the network and reducing the stall events for other clients. Moreover, ESBA predicts the connection throughput by considering an ANN. On the other hand, VaQoE-Adapt and Adaptech consider moving average for throughput estimation, which is less accurate than the ANN considered by ESBA.

In terms of energy consumption, ESBA provides energy-efficiency compared to Adaptech and VaQoE-Adapt, as shown in Figure 2c. Specifically, ESBA reduced the energy consumption by 60% and 70% compared to VaQoe-Adapt and Adaptech, respectively. This is because ESBA considers a module responsible for checking the device battery as the first step of the adaptation algorithm, decreasing or limiting the quality of the segment if necessary. This is because users prefer a less than excellent but acceptable video quality when offered higher energy saving [9]. On the other hand, VaQoe-Adapt and Adaptech do not consider energy issues for decision making, thus in most situations, they download a larger number of packets (sometimes unnecessarily) for higher bitrate level, which consumes more energy. As a consequence, using ESBA only 3% of devices did not have enough energy to playback the entire content, while 29% and 21% of devices are without enough energy to playback the entire video using Adaptech and Va-QoEAdapt algorithms, as shown in Figure 2d.

Figure 2e shows the normalized bitrate for each video delivered by ESBA, Adaptech, and VaQoE-Adapt in a scenario composed of 20 and 30 static clients. Observing that, we conclude all algorithms deliver video segments with almost the same bitrate levels. This is because they consider similar metrics to request video segments with a given bitrate level. For instance, some clients using Va-QoEAdapt or ESBA requested video segments with the minimal bitrate level, *i.e.*, escape mode representation, as soon as they detect that there is not enough connection throughput to request video segments with higher bitrate level. In addition, ESBA considers energy information to make decisions, and thus the clients requests low-quality bitrate segments as soon as a given device do not have enough energy to play a whole video.

Figure 3 shows the normalized bitrate for each video segment received by a specific client for a video duration of 300s delivered by ESBA, Adaptech, and VaQoE-Adapt in a scenario composed of 30 static clients. Values closer to 1 means content delivery with a high QoE, while values above 1 indicate a waste of energy since there is no need to request a higher segment quality than the supported device, consuming unnecessary network resource shared with other clients. By analyzing the results of Figure 3, we can conclude that VaQoE-Adapt and Adaptech are not aware of the video quality supported by the device since some segments have the normalized bitrate higher than 1. This causes waste of energy, network, and CPU resources since the client requested segments with the bitrate higher than the supported by client's device. Moreover, we can also observe that ESBA and VaQoE-Adapt delivered some segments with the minimal bitrate level, *i.e.*, escape mode representation. This is because the available bandwidth is not favorable to request high bitrate level for such segments, and thus the minimal bitrate level is requested.

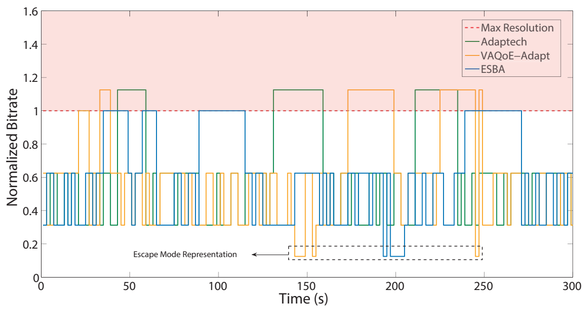


Fig. 3: Normalized bitrate for each video segment in a scenario composed of 30 static clients

Figure 4 shows the QoE and energy results for video delivered by ESBA, Adaptech, and VaQoE-Adapt in a scenario composed of 20 and 30 mobile clients. By analyzing the results, we conclude that ESBA also outperforms Adaptech and VaQoE-Adapt algorithms to deliver adapted videos with better QoE and energy-efficiency. In addition, the scenario with 30 clients has also worst QoE and energy performance than the scenario with 20 clients, since there are more clients requesting videos and consuming network resources. By comparing the QoE results of Figures 2 and 4, *i.e.*, scenarios with and without node mobility, we observe that node mobility reduced the number and duration of stalls events in 20% regardless the adaptation algorithm. This is because node mobility leads to more packet loss ratio, reducing the accuracy of any mechanism to predict the connection throughput, which is an important metric for the adaptation mechanisms. It is important to highlight that the connection throughput prediction used by ESBA is still performing well in the scenario with node mobility since it considers an ANN to learn from the previous experience. On the other hand, comparing the energy results of Figures 2 and 4, we observe that results are almost the same. This is because although the mobile scenario has larger packet retransmissions, the quality of the segments is generally lower, which requires a smaller number of packets to download a video segment.

From our performance evaluation analysis, we conclude that ESBA significantly reduces the energy consumption, number, and duration of stalls to deliver multimedia content using adaptive bitrate streaming techniques compared to Adaptech and VaQoE-Adapt algorithms. In addition, ESBA keeps the normalized bitrate level similar to Adaptech and VaQoE-Adapt algorithms.

V. Conclusion

In this paper, we introduced a hybrid energy-aware video bitrate adaptation algorithm to deliver videos with high QoE and energy-efficiency for mobile users, named ESBA. It takes into account devices characteristics (devices' energy and maximum supported resolution), connection throughput (estimated by a Machine Learning Process) and the probability of stalls (computed based on buffer information) to request video segments with different bitrates. In this way, ESBA improves the energy-efficiency, while keeps a high QoE in an HAS system. Simulation results demonstrate the energy-efficiency of the ESBA, especially in scenarios where the user's device has not enough power to playback multimedia content in the video quality requested. For instance, ESBA transmitted adapted video with energy efficiency 50% higher than videos transmitted by Adaptech and VA-QoE. Moreover, results also demonstrate that ESBA delivers videos with satisfactory QoE, *i.e.*, ESBA has a high normalized bitrate and lower number and duration of stalls compared to Adaptech and VA-QoE.

A further investigation is necessary to look into other scenarios, using a higher number of nodes, and comparing with other strategies such as cache strategies, in order to define which one is better in a different set of situations.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

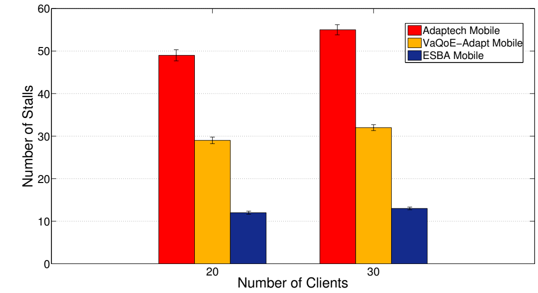
References

[1] Cisco, "Global mobile data traffic forecast update, 2016–2021 white paper, accessed on nov 2, 2017."

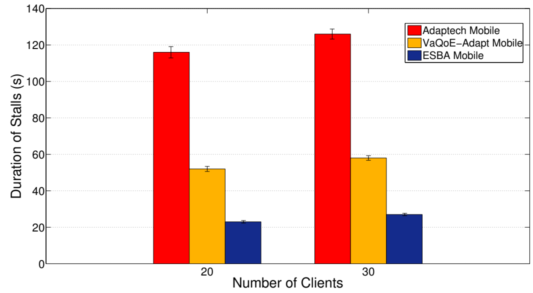
[2] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[3] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proceedings of the ACM SIGCOMM*. New York, NY, USA: ACM, 2015, pp. 325–338.

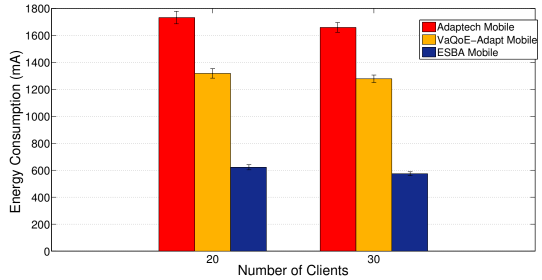
[4] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 401–418, 2016.



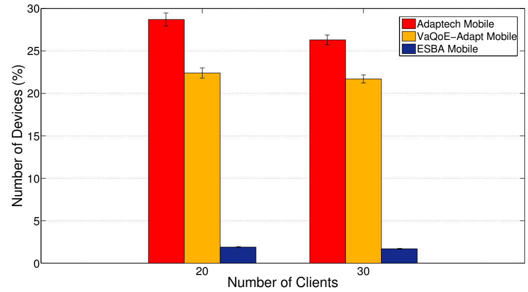
(a) Number of Stalls



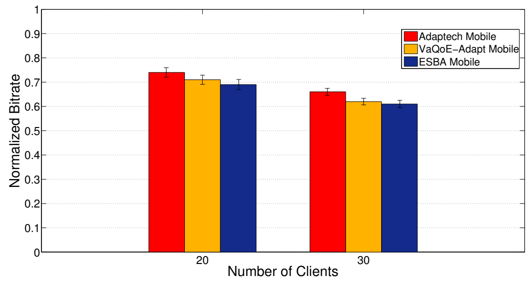
(b) Duration of Stalls



(c) Energy Consumption



(d) Percentage of devices without enough energy to playback the entire video



(e) Normalized Bitrate

Fig. 4: QoE and energy results for scenario with 20 and 30 mobile clients

[5] M. A. Hoque, M. Siekkinen, J. K. Nurminen, M. Aalto, and S. Tarkoma, "Mobile multimedia streaming techniques: Qoe and energy saving perspective," *Pervasive and Mobile Computing*, vol. 16, pp. 96–114, 2015.

[6] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 345–360.

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

[8] M. A. Hoque, M. Siekkinen, and J. K. Nurminen, "Energy efficient multimedia streaming to mobile devicesâĂŤa survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 579–597, 2014.

[9] C. Singhal and S. De, "Energy-efficient and qoe-aware tv broadcast in next-generation heterogeneous networks," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 142–150, 2016.

[10] L. Amour, M. S. Mushtaq, S. Souihi, and A. Mellouk, "Qoe-based framework to optimize user perceived video quality," in *proceedings of the IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE, 2017, pp. 599–602.

[11] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over http," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 271–287, 2012.

[12] M. Coelho, C. Melo, and N. Fonseca, "Versioning-aware and qoe-oriented strategy for adaptative bitrate streaming," in *proceedings of the IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

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

[14] R. Dubin, A. Dvir, O. Pele, O. Hadar, I. Katz, and O. Mashiach, "Adaptation logic for http dynamic adaptive streaming using geo-predictive crowdsourcing for mobile users," *Multimedia Systems*, vol. 24, no. 1, pp. 19–31, 2018.

[15] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design and optimisation of a (fa) q-learning-based http adaptive streaming client," *Connection Science*, vol. 26, no. 1, pp. 25–43, 2014.

[16] I. Ayad, Y. Im, E. Keller, and S. Ha, "A practical evaluation of rate adaptation algorithms in http-based adaptive streaming," *Computer Networks*, vol. 133, pp. 90–103, 2018.

[17] Conviva, "Viwer experience report, 2015 white paper, accessed on sep 7, 2017."