2019 International Conference on Networked Systems (NetSys)

Energy-Efficient Adaptation Logic for HTTP Streaming in Mobile Networks

Christian Moldovan, Florian Wamser, Tobias Hoßfeld
Chair of Communication Networks, University of Würzburg, Würzburg, Germany
Email: {christian.moldovan, florian.wamser, tobias.hossfeld}@uni-wuerzburg.de

***Abstract*—The requirements for video streaming have changed drastically during the past years. In today's Internet, high definition resolutions are considered default for videos, even in mobile settings, and with 4G penetration reaching 90 percent in the US, this no longer poses a big problem. However, while mobile bandwidth has increased, the battery life time of mobile devices has not increased significantly. Furthermore, current data plans are still not large enough to regularly stream movies during the commute. Users still resort to downloading media before travel.**

In this paper we propose a new HTTP adaptive streaming algorithm that delivers videos in high quality while avoiding stalling events, schedules the download of video segments so that a energy conserving idle state is often reached and keeps the buffer low at points in the video where many viewers abandon the video to save data. While most adaptive streaming algorithms optimize quality and stalling, this is the first attempt to use an adaptive streaming algorithm to reduce energy consumption. Since video streaming providers mostly care about the Quality of Experience when watching videos, energy efficiency is left to the device manufacturers. Therefore, both parties have little incentive to cooperate in this regard. But on the Internet of tomorrow, where most videos are watched on mobile devices, energy efficiency and the Quality of Experience must go hand in hand.

***Index Terms*—HTTP adaptive streaming, energy efficiency, user engagement, adaptation algorithm, mobile networks**

I. Introduction

Video streaming is one of the most popular applications in the Internet. With higher bandwidth coverage, mobile video streaming has become commonplace. According to a Cisco white paper [1] in 2016, 9% of total video traffic was mobile traffic and it is assumed to increase to 21% in 2021. Furthermore, Ericsson predicts that in 2024 over 1.4 billion devices will be subscribed to 5G which is estimated to offer bit rates of 1-10 Gbit/s. Mobile video data is estimated to grow by 35% per year. By 2024 video is forecast be make up 74% of total mobile traffic [2]. In times where 4G penetration reached 90% in the US[1] and 5G is starting to be deployed, stalling events and poor video quality are less of an issue. Mobile bandwidth grows exponentially. In contrast, the energy density of Lithium-Ion cells has only quadrupled since they became commercially available in 1991 and may soon hit a limit [3]. The average monthly data volume per mobile Internet subscription in Germany was 850 MB in 2017[2]. In comparison, one hour of HD (720p) video uses about 900 MB. Currently, there exists no energy and data efficient adaptive streaming mechanism. The last attempt was in [4] where a data conserving algorithm is presented for non-adaptive streaming.

In this paper, we present a new adaptation algorithm for HTTP Adaptive Streaming (HAS) that is based on KLUDCP [5]. The algorithm uses audience retention statistics of the video that is currently watched to keep the buffer low during periods where many users abandon the video to not waste data. Otherwise, the algorithm downloads video segments in an on/off pattern based on eSchedule [4]. This way a device that is used for video streaming can enter the Radio Resource Control (RRC) IDLE state more frequently, leading to lower energy consumption. In practice this algorithm is easy to implement for video streaming providers since they have access to these statistics. We compare our algorithm with its baseline (KLUDCP) in terms of application layer QoS (i.e. number of stalling events, average quality, and number of quality switches) and in terms of energy savings and data savings. For this prupose, we conduct a simulation which uses viewing statistics[3] from real YouTube videos, real mobile bandwidth traces (WiFi, 3G [6] and LTE [7]) and appropriate energy models. In the simulation, users watch five consecutive videos and may skip ahead in a video or abandon it based on the audience retention statistic of each video. This is the first time that real user behavior is used in an adaptation algorithm and in its evaluation.

The remainder of this paper is structured as follows. Section III discusses the used energy models, the audience retention. Section II further provides background and an overview of related work. In Section IV we present the new adaptation algorithm. Simulation results are presented in Section V, while concluding remarks and outlook are given in Section VI.

II. Related Work

A. HTTP Adaptive Streaming

In modern video streaming applications, video segments are downloaded into a buffer and later played from it. During an Internet video stream, the throughput of the utilized network can vary. If the video is played with a higher rate than it is

[1] https://opensignal.com/reports/2018/02/state-of-lte

[2] https://www.statista.com/statistics/469121/mobile-internet-monthly-data-volume-per-user-germany/

[3] https://support.google.com/youtube/answer/1715160?reftopic=3029003

978-1-7281-0568-0/19/$31.00 ©2019 IEEE

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The definitive version of this paper has been published in 2019 International Conference on Networked Systems (NetSys)

downloaded, the buffer empties and stalling occurs until the buffer is refilled. To prevent this, in adaptive video streaming the video bit rate can be reduced if a low buffer or a low network throughput is detected. If the network conditions improve, the video bit rate can be increased.

User studies have shown that the factors that affect the user experience the most include the frequency and duration of stalling events and the video quality [8]. There is a disagreement whether the number of quality switches has a significant impact [9], [10] or not [11]. However, all agree that the number of switches is not important as long as it is not too high. The ITU-T standard from 2017 [12] does not include the number of switches in its QoE model.

The heuristics that decide when the video quality should be changed rely on buffer thresholds or throughput thresholds of the last few segments. As one of the earlier academic algorithms, KLU [5] considers a single user in a mobile environment using single layer content. KLU takes three input parameters into account: the current buffer level as ratio compared to the maximum buffer level, the throughput measured while the last segment is downloaded and the average playback bitrate of each quality level. The adaptation strategy calculates the available bandwidth as percentage of the last measured throughput to select the next quality level. The higher the buffer level, the higher is the calculated percentage of the last measured throughput. The calculated throughput is compared to the bitrate of each quality level. The highest quality level below the calculated bitrate is selected as next quality level. The algorithm presented in this paper is based on KLU and KLU is used as a baseline for comparison. There are modern adaptation algorithms that use machine learning approaches to predict the best decision. One successful example is Pensieve [13] which is trained to directly optimize certain QoE metrics. Others are based on Markov models such as CQBS [14] which increases the video buffer when channel quality is high and consumes the buffer when channel quality decreases in a cost-efficient manner. While their system works with the assumption that videos are watched completely, in this paper, we go one step further, and investigate the effect of realistic user abandonment behavior. Furthermore, we develop an adaptation strategy that considers the user behavior and the energy cost of the current channel to increase the data and energy efficiency of video streaming.

The primary goal of adaptation algorithms is to avoid stalling, since it has a high negative impact on the quality of experience. The secondary goal is to keep the video quality high while not switching between quality levels too frequently, since this is considered annoying. Bandwidth should be conserved since most videos are abandoned after a few seconds. This can be done by limiting the maximum buffer size or limiting the download rate. For example, YouTube has a maximum buffer size to limit the unnecessarily downloaded data if users abandon a video. If there were no limit, in a high bandwidth setting, the whole video would be downloaded quickly. If users abandon the video early, a large share of the downloaded data is wasted. This leads to unnecessary transmission cost for the involved ISPs and YouTube. In a mobile setting, the data a client can use in a month is often limited by the data plan the user has signed with his ISP. Therefore, data is particularly expensive in mobile environments. Since the battery life of mobile devices is also limited, energy efficiency is considered very important. Current adaptive streaming algorithms manage to minimize stalling events and the number of quality switches while maintaining a high average video quality. However most do not consider data and energy efficiency.

One issue that is addressed little in current research is adaptive streaming for multiple users that share a bottleneck. In such a scenario interactions between the clients happen on application layer and network layer. In particular with QUIC, there are many problems in a multi user scenario, since it uses UDP instead of TCP. For example [15] found that in a shared bottleneck scenario with one QUIC flow and two or four TCP flows, QUIC consumes more than 50% of the available bandwidth. Currently, there are no adaptation approaches that are used to resolve this issue since this is a new and difficult topic. The authors of [16] determine the optimal resource allocation for multiple users who watch the same video at the same time, e.g. live streaming of a popular show or event. Their quadratic program optimizes the average quality and the number of quality switches while stalling is completely avoided. The authors of [17] extend this scenario to different videos that may be watched at different times by multiple users. Their quadratic program is extended to include the fairness of the resource allocation between the users. Different types of fairness are discussed that may be employed.

B. Energy and Data Efficiency in Mobile Video Streaming

Qian et al. [18] investigate different RRC inactivity timers in video streaming over 3G cellular networks. They discover a performance inefficiency due to tail effects and state promotion overhead. They find that each application has its optimal value for the inactivity timer. In [19], they present a framework that optimizes tail times resulting in a significant reduction of energy consumption for various Internet applications. Hoqoe et al. [20] discover that video streaming platforms use different streaming techniques for different devices, players, and video qualities. They discover that there is room for optimization in terms of energy efficiency for every technique.

Seufert et al. [21] use throughput traces of 2G, 3G, 4G and WiFi networks to investigate how effective it is to offload mobile traffic to WiFi hotspots. They find that the low throughout of WiFi networks leads to lower QoE and higher energy consumption compared to 4G. WiFi offloading is recommended if only 2G or 3G networks are available but not for 4G. In this paper, we conduct a similar simulation, which utilizes 3G and LTE network traces to simulate HAS in a mobile environment. Further, we simulate user behavior and evaluate the energy consumption of different adaptation strategies and different devices to evaluate HAS with similar network traces and different tail times.

Schwartz et. al. [22] evaluated four different video streaming mechanisms, in respect to QoE, energy consumption, User

Equipment (UE), and wasted traffic. They show, that their streaming mechanism, which uses a buffer with two thresholds, offers the best trade-off between energy consumption and wasted traffic. They use basic probability distributions to model the user behavior while we use real user statistics in our evaluation. Furthermore, their work does not discuss adaptive streaming mechanisms since it was written in a time before adaptive streaming was popularized.

Siekkinen et al. [23] measured 20% energy savings in mobile video streaming when shaping the traffic received from 3G and LTE networks into traffic bursts. Energy could not be reduced further due to YouTube's background traffic, which was interfering with the traffic shaping and causes unexpected transitions of the RRC. Traffic shaping also provides a good balance between saved energy and signaling overhead. In [4], they developed a scheduling algorithm, which relies on viewing statistics to reduce the energy consumption and traffic overhead in mobile video streaming. They defined a scenario, where users can abandon the playback at any time and developed an algorithm, which predicts the user behavior based on the viewing statistics. Depending on the wireless interface that is used, the algorithm calculates the energy and traffic optimal download schedule. We adapt their scheduling algorithm to develop a HAS adaptation strategy, which optimizes the quality, energy consumption, and traffic waste. Furthermore, we follow a similar idea and utilize audience retention statistics to perform a user centric traffic shaping. However, our strategy is built around HAS since it is very common in video streaming applications. In addition, we have more complex video sessions, where multiple videos are watched, and video content may be skipped.

It was observed that YouTube downloads segments first in a low resolution and later, if there is more bandwidth available, in a higher resolution [24]. While this may be beneficial in mobile networks with high variance, it leads to wasted data. The authors of [25] find that instead of re-downloading segments, the quality of 20% of the videos investigated in their study could be downloaded in a higher quality level. Furthermore, 94% of all stalling events that occurred during their study could have been avoided. This demonstrates how much potential for optimization there is and how necessary data-conserving mechanisms are in HAS.

III. System Model

A. Energy Model

Huang et al. [26] investigated the performance and power characteristics of LTE networks. They measured the energy consumption and timing of 3G, LTE and WLAN interface in smart phones. They found that LTE possesses the highest tail time and power consumption. WLAN has the smallest tail time and energy consumption, and the smallest DRX cycle and promotion delay. The 3G wireless interface has the biggest promotion delay. The LTE interface possesses less promotion delay than 3G, but a significant higher promotion energy. Additionally, they investigated the send and receive power consumption. They conducted several experiments and developed an energy consumption model. Also, LTE possesses the biggest base power consumption, but the least power consumption per download throughput. WLAN possesses the least amount of base energy consumption, but the biggest energy consumption per download throughput. The tail energy can be limited using Fast Dormancy (FD) [27]. As the model does not provide a specification for the 3G wireless interface with FD, we specify the missing model using the FD configuration from [4]. The FD timer is set by the device to reduce the tail time[4][5]. To reduce the tail time, we set the FD timer to 5 seconds. We use the same 3G power model to simulate the same device, which supports FD. Further, the tail time is reduced to the half of the LTE wireless interface's tail time. Li et al. [28] investigated the energy consumption of video decoding in smartphones. They conducted several experiments, watching different videos and built an empirical model of the video decoding power consumption. They compared the energy consumption with the empirical model, which shows less than 10 percent error. The smartphones both possess a display resolution of $800x480$. The video decoding energy consumption is different for up scaling and down scaling to the same resolution. The different devices possess a difference in energy consumption of 300 mW for smaller and 500 mW for larger videos.

B. Audience Retention

Most studies, such as [29], consider a video to be completely watched when benchmarking adaptation algorithms, but on actual video platforms users frequently interact with videos. Users abandon the video playback or skip some playback and resume at another playback position. Statistics about the user's video playback behavior show, that users skip playback time and abandon the playback. For example, YouTube provides audience retention statistics[6] to the owner of the video channel. These statistics describe which part of a given video is watched by what share of users. Figure 1 shows an example for a video, which possesses a duration of 3 minutes and 43 seconds. The audience retention starts at 100%, indicating that all users started to watch the video at its beginning. About 30% of all users abandoned the video or skipped ahead during the first 7 seconds. Due to the user's possibility to skip playback before the start of playback, the audience retention can also start with less than 100%. Users, who skip backwards to the beginning of the video, result in higher audience retention than 100%. Forward and backward skipping lets the audience retention rise and fall during the playback time.

Video segments with a low user retention, are watched by few users. Always downloading them results in a lot of wasted data and energy. We make use of this fact by delaying the download of these segments until necessary. This leads to a lower expected data consumption and energy consumption

[4]https://www.gsma.com/newsroom/wp-content/uploads/2013/08/TS18v1-0.pdf

[5]http://www.3glteinfo.com/fast-dormancy-in-3gpp/

[6]https://support.google.com/youtube/answer/1715160?hl=en

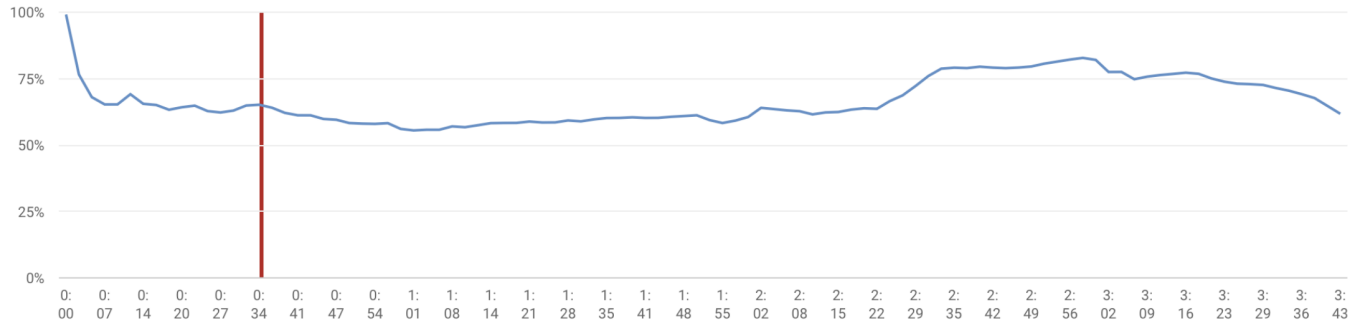

Figure 1. Example for viewer abandonment statistic: audience retention of video segments compared to total views of YouTube video ABSIFBFIOS. Abandonment is typically high at the beginning. An increase in the curve indicates that users skipped ahead in the video.

Variable	Definition
P_0	last segment that was downloaded
P_1	first segment of next batch that is downloaded
P_2	last segment of next batch that is downloaded
n	size of batch in segments
bw^+	optimistic bit rate suggestion for next batch
bw^+	pessimistic bit rate suggestion for next batch
tl	tail time
er	predictability of the network throughput
$r_{dl}(P_0)$	throughput during download of last segment
bl^+	optimistic buffer level estimation
bl^-	pessimistic buffer level estimation
$Q(i)$	quality layer in which segment i is downloaded
$br(Q(i))$	encoding rate of quality layer $Q(i)$
t_i	remaining play time of segment i
con	constraints to avoid stalling and guarantee high quality
$p_{abd}(j)$	abandonment prob. at segment j
D	last segment played before next download
E_{tail}	energy consumption per second during tail
E_{rx}	energy consumption of bit rate

Table I
NOTION OF VARIABLES.

during video streaming. The details of this approach are described in Algorithm 1.

IV. ADAPTATION ALGORITHM

In this section, we present the adaptation algorithm and its components. Our algorithm is a combination of the buffer- and bandwidth-based algorithm from [5] and an energy-efficient schedule for the download of video segments from [4] that we extend for adaptive streaming. We determine the bit rate bw^+ of the following segments P_1 to P_2 according to KLUDCP, adding an error variable to avoid overestimating future bandwidth. Since we want to select a quality layer for a batch of segments instead of a single segment, several changes had to be made to the basic algorithm. To avoid stalling when downloading large batches of segments, we estimate the bit rate after the download of each segment. An overview of the used notion is given in Table I.

We define an error $er \in [0, 1]$, which describes the highest percentage, which we expect the bandwidth to sink/rise per downloaded segment. We call this optimistic bandwidth estimation for a rising throughput and pessimistic bandwidth estimation for a shrinking throughput. Equation 1 shows the optimistic bandwidth estimation, where the error value rises per segment download and the resulting estimated bandwidth rises per segment download. Equation 2 shows the pessimistic bandwidth estimation, where the error value rises per segment download and the resulting estimated bandwidth shrinks. The buffer level is used to calculate the quality level, so we calculate the expected buffer level after each segment download. The throughput prediction can be replaced by more refined methods, compare [13].

As we consider optimistic and pessimistic bandwidth, we also need an optimistic and pessimistic buffer level estimation. Equation 3 calculates the expected pessimistic buffer level, while Equation 4 calculates the expected optimistic buffer level. Both consist of three parts: the first part sums the already buffered time. The second part sums additionally buffered time after the download is completed, in respect to the pessimistic/optimistic bandwidth estimation. The third part represents the tail time, which is spent before the download started.

$$bw^+(P_1, P_2, n, tl) = \quad (1)$$
$$\begin{cases} r_{dl}(P_0) \cdot (1-er)^{n-1} \cdot 0.3 & \text{if } 0 \le bl^+ < 0.15 \\ r_{dl}(P_0) \cdot (1-er)^{n-1} \cdot 0.5 & \text{if } 0.15 \le bl^+ < 0.35 \\ r_{dl}(P_0) \cdot (1-er)^{n-1} & \text{if } 0.35 \le bl^+ < 0.5 \\ r_{dl}(P_0) \cdot (1-er)^{n-1} \cdot (1 + 0.5 \cdot bl_i) & \text{if } 0.5 \le bl^+ < 1 \end{cases}$$

$$bw^-(P_1, P_2, n, tl) = \quad (2)$$
$$\begin{cases} r_{dl}(P_0) \cdot (1-er)^{n-1} \cdot 0.3 & \text{if } 0 \le bl^- < 0.15 \\ r_{dl}(P_0) \cdot (1-er)^{n-1} \cdot 0.5 & \text{if } 0.15 \le bl^- < 0.35 \\ r_{dl}(P_0) \cdot (1-er)^{n-1} & \text{if } 0.35 \le bl^- < 0.5 \\ r_{dl}(P_0) \cdot (1-er)^{n-1} \cdot (1 + 0.5 \cdot bl_i) & \text{if } 0.5 \le bl^- < 1 \end{cases}$$

For the video encoding rate $br(Q(i))$ of quality $Q(i)$ of segment i, the buffer level is determined as

$$bl^+ = \sum_{i=P_1}^{P_2} t_i + \sum_{i=P_2+1}^{P_2+n} t_i \cdot (1 - \frac{br(Q(i))}{bw^+}) - tl \quad (3)$$

$$bl^- = \sum_{i=P_1}^{P_2} t_i + \sum_{i=P_2+1}^{P_2+n} t_i \cdot (1 - \frac{br(Q(i))}{bw^-}) - tl. \quad (4)$$

The equations determine an optimistic and pessimistic bandwidth and buffer estimation. Equation 5 contains multiple constraints for optimistic and pessimistic bandwidth and buffer estimation. The buffer constraint checks, if the buffer may run empty or full. The bandwidth constraint checks, if the estimated bandwidth is not less than the quality level's bit rate. It also checks, if the optimistic bandwidth estimation is not bigger than the next quality level's bit rate. Both constraints must be met to ensure a smooth playback.

$$con = \begin{cases} 1, & \text{if } bw^+ \geq br(Q) \wedge bw^- < br(Q+1) \\ & \quad \wedge bl^+ > 0 \wedge bl^- < 1 \\ 0, & \text{else} \end{cases} \tag{5}$$

$$\mathbb{E}[B_{waste}(P_2, D, n)] = \sum_{i=D+1}^{P_2+n} \sum_{j=D+1}^{i} p_{abd}(j) \cdot t_i \cdot br \tag{6}$$

$$\mathbb{E}[E_{waste}(P_m, P_2, D, n)] = \tag{7}$$
$$E_{tail} * \sum_{i=P_1}^{D} \max(t_i, t) + \frac{\mathbb{E}[B_{waste}(P_2, D, n)]}{r_{dl}} \cdot E_{rx}(r_{dl})$$

The abandonment probability $p_{abd}(j)$ is the probability that a user abandons the video right before starting to watch segment j. The algorithm starts by selecting the quality level before the algorithm enters three interleaved loops. The first loop iterates over the segments, which can be downloaded. The loop begins at the first not yet buffered segment and ends after the limiter is exceeded. The second loop also begins at the first, not yet buffered segment and ends at the current position of the first loop. The third loop iterates over the buffered segments. Then the algorithm calculates the expected buffer state for each combination. The expected buffer state is determined using the expected buffer state of the download, which downloads all previous segments, or the current buffer state, if there is no previous download. The expected wasted Energy depends on the abandonment probability $p_{abd}(j)$ and the current buffer state, compare Equations 6 - 7. If there are no constraint violations and the expected wasted energy is less than the expected wasted energy of the stored buffer state, this buffer state is stored. The algorithm saves one buffer state per segment. The comparison is performed to the stored buffer state, which download ends at the same segment. After all combinations are evaluated, the algorithm performs a back trace. Therefore, the algorithm starts at the stored expected buffer state, which downloads the last segment. The next expected buffer is searched, which ends before the buffer state's first downloaded segment. This process continues, until the back trace reaches the current buffer. The traced downloads are returned as schedule in reversed order together with the determined quality level. The download schedule can be incomplete, which should not affect the overall behavior, because the schedule must be reevaluated after each download. If there is no optimal download without adaptation or stalling, the algorithm falls back by downloading the next segment at the selected quality level.

Algorithm 1: Energy-Efficient Batch Adaptation (EE)

Data: $P_1, P_2, buffersize$
// $buffersize$ is the maximum buffer size
Result: $S, quali$
$E_n(P_2) = 0$
$P_1(P_2) = P_1$ // first segment in buffer
$P_2(P_2) = P_2$ // last segment in buffer
forall $q = P_1$ *to* P_2 **do**
  if *br(p) < minbw* **then**
    quali = q(p) // quality of last segment which has lower bit rate than bandwidth
forall $i = P_2 + 1$ *to* $P_2 + buffersize$ // segments that may be downloaded without overextending the buffer size
do
  forall $j = P_2 + 1 : i$ // subsets of consecutive segments beginning with the next segment
  do
    forall $k = P_1 - 1 : j$ // segments of buffer and subset
    do
      $E_{cur} = E_n(j) + \mathbb{E}[E_{waste}(P_1(i), P_2(i), k, i-j)]$ // expected wasted energy when downloading segments until j, considering abandonment rate and tail energy
      $P_{minn} = P_1(j-1)$
      forall $l = P_1(j-1) : P_2(j-1)$ **do**
        $P_{minn} += (t_l - (t_l \cdot enc/dl))$
      $P_{minn} -= tail$
      if $E_{curr} < E_n(i) \wedge con(P_{minn}, j-1, i-j, tail, quali) == 0)$ // check if expected wasted energy will be lower compared to downloading one less segment and if buffer and bandwidth constraints are fulfilled
      then
        $E_n(i) = E_{cur}$
        $lastchange(i) = j$
        $P_2(i) = j - 1$
        $P_1(i) = P_{minn}$
        $P - x(i) = i$
        // schedule segments $P_2 + 1$ to j to be downloaded as a batch
$end = PL_x + buffersize$
while $(end > PL_x)$ **do**
  $S = (end - lastchange(end), S)$
  $end = lastchange(end)$

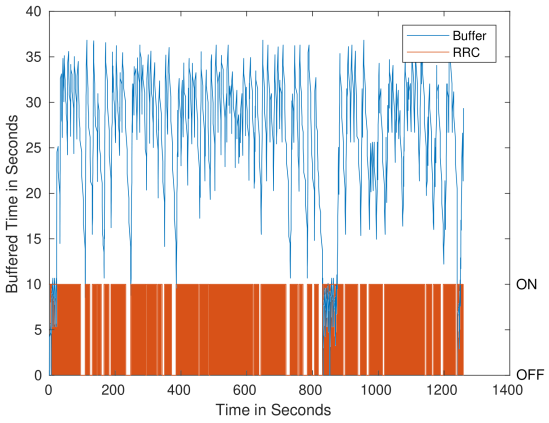

Figure 2. Video browsing session in which five videos are watched of which three are abandoned early. At the beginning of a video the abandonment rate is high, so the buffer is kept low to save data. The RRC state is turned off frequently to save energy with EE.

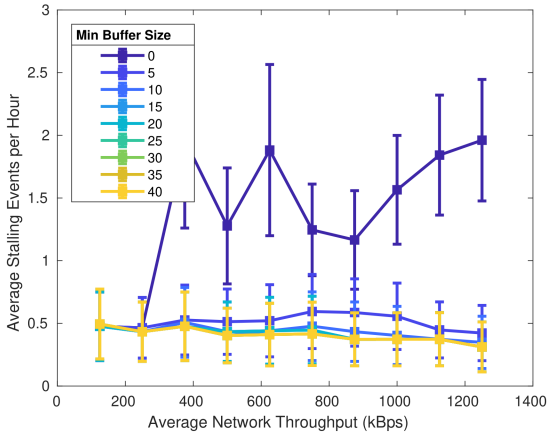

Figure 3. Impact of the minimum buffer size on the frequency stalling events for constant bandwidth.

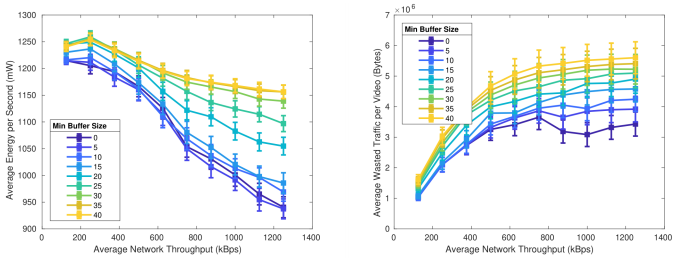

Figure 4. Impact of the minimum buffer sized on energy and data saved for constant bandwidth using an LTE interface.

As shown in Figure 2, the adaptation strategy does not keep the same buffer size and the RRC transits to the lower state for several times. The buffer fluctuates between 10 and 30 seconds. According to the audience retention, the buffer is filled and depleted. During the depletion phase, the RRC transits to the lower state. After a long tail duration, the adaptation strategy continues the download to avoid stalling. Another aspect of the buffer behavior is that the buffer stays small during the first few seconds of playback. The audience retention statistics usually possess high abandon probabilities at the video's beginning. Therefore, the first segments are carefully downloaded. This approach shows the desired buffer and RRC behavior, which potentially saves energy and traffic.

V. Results

A. Methodology

We use user abandonment statistics from 21 popular videos from six cooperating YouTube channels. We download the corresponding videos as mp4-file in every available resolution. We use three kinds of goodput traces: constant bandwidth, real vehicular 3G traces [6], real vehicular LTE traces [7] We simulate video streaming on application layer. We download videos and decode them into their frame sequence using *ffprobe* to determine the location of key frames which we use as segment start in our simulation. We simulate a video browsing session, where the user watches five videos in a row, compare Figure 2. The probability that the user abandons the video or skips ahead in the video is given through the viewer abandonment statistic for the video he is currently watching. We do not consider skipping back to a previous position of the video since it is too difficult to determine from viewer statistics. If a video is finished or abandoned, the next video is selected randomly from the remaining pool of videos. Each experiment was repeated 100 times. All confidence intervals in the figures are given with 95% confidence.

B. Impact of Buffer Size

First, we investigate the impact of the buffer size on QoS, energy consumption and data consumption for our new algorithm. The minimum buffer size determines the lower threshold that should never be underpassed to avoid stalling. A minimum buffer size lower than 10 seconds leads to more frequent stalling events, as can be seen in Figure 3. A larger minimum buffer size also leads to higher average quality and fewer quality switches. However, a higher min buffer means that more data will be wasted in the case of an abandonment event and more energy is used since idle periods cannot last as long, compare Figure 4. We therefore use a balanced min buffer size of 15 seconds to avoid too many stalling events.

The (maximum) buffer size defines how much video content may be downloaded into the buffer at most before it is played. In Figure 5,6 and 7 we see the impact of the buffer size on energy consumption and wasted data. A large buffer of 80 s means that we can download a lot of video content into the buffer and then pause the download until the buffer is almost depleted. This pause is very energy efficient since the idle RRC state can be reached for a long period of time. However, if the user abandons the video when the buffer is filled up, a lot of data is wasted. This means that the buffer size is a trade-of between energy and data. Since it depends on the scenario whether data or energy is more valuable, we use three parameters for the buffer size in the following: 30s, 50s, 80s.

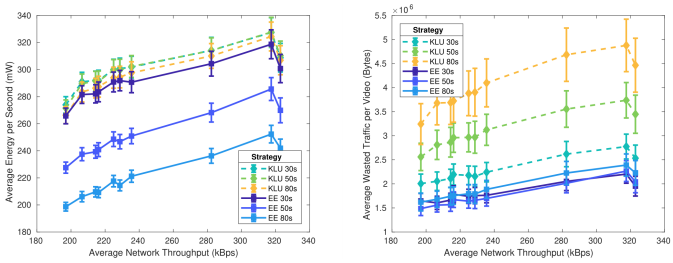

Figure 5. Resourcefulness of KLU and EE in constant WiFi scenario

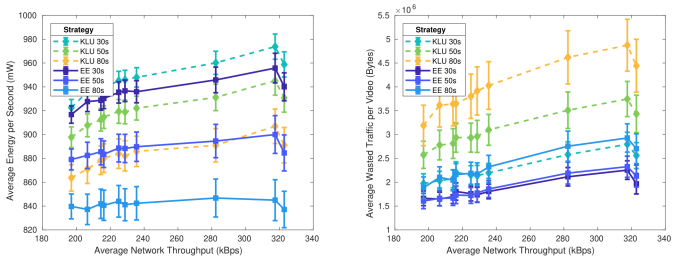

Figure 6. Resourcefulness of KLU and EE in vehicular 3G scenarios

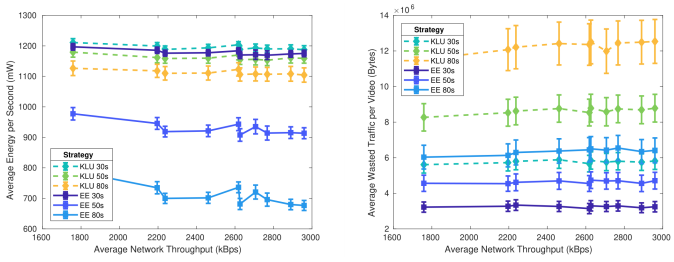

Figure 7. Resourcefulness of KLU and EE in vehicular LTE scenarios

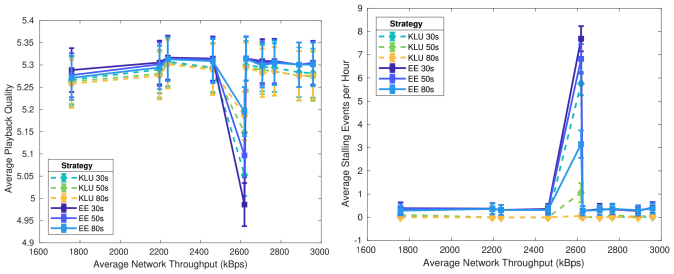

Figure 8. Application layer QoS for KLU and EE in a vehicular LTE scenario

C. Impact of Network on Saved Energy

Next, we investigate how much energy can be saved in different scenarios. In a WiFi scenario (Figure 5) the consumed energy can be reduced by about 25% with a large buffer. While less data is wasted with EE than with KLU, data is usually not limited in WiFi settings compared to other mobile settings where data is limited through data plans. In conclusion, EE leads to higher energy efficiency in a WiFi setting. In a 3G scenario (Figure 6) EE leads to $10 - 12\%$ of energy saved when comparing a 30s buffer to an 80s buffer. In terms of data, it becomes visible that especially for high buffer sizes, EE is much more efficient than KLU since it makes use of user abandonment statistics. In absolute numbers however, in average only about 2MB are saved for each video that is started. In contrast, savings are much higher in LTE networks (Figure 7). For an 80s buffer, we can reduce the average energy consumption by $35 - 40\%$ and the wasted traffic by about 50%. Here, the saved data is larger, but also only lies between 3-6 MB which corresponds to 12-24 seconds of 720p video content. From the energy perspective it makes the most sense to use EE in LTE scenarios where it can result in much longer battery time. Furthermore, it is visible that a large buffer always increases the battery time significantly while the saved data is insignificant. Even with a large buffer only little data is wasted since the buffer is not filled up during scenes in the video where many users abandon.

D. Performance in Terms of Application-Layer QoS

Next, we compare the QoS for KLU and EE in the LTE scenario. The average playback quality is only about 1% higher in KLU, c.f. Figure 8. With KLU the average number of quality switches per hour lies between 7 and 50 depending on the scenario and is about 50% - 100% higher than with EE. The number of quality switches is within acceptable values for both strategies. The number of stalling events is higher for EE with 0.4 stalls per hour, but acceptable for mobile scenarios. This is because KLU constantly stays at high buffer, while EE has ON/OFF phases to conserve energy. An exception can be observed for the subway scenario. In this scenario, the user passes through many long tunnels where no connection to the network is possible. If such a tunnel cannot be predicted, stalling can only be avoided if the buffer is very high when we enter the tunnel. So, only KLU with 80 s buffer size can avoid some stalling events. Such scenarios are discussed in [30] where first solutions for outage prediction and handling are presented. We therefore recommend to choose a large buffer to reduce the number of stalling events in these rare events.

To sum up the results, we see that EE is much more resourceful than KLU while maintaining a similar QoS on application layer. An exception is only observed in a scenario with very bad connectivity where energy and data efficiency are a secondary concern. When such a scenario is detected, the player should switch to a conservative adaptation strategy that maintains a high buffer such as KLU.

VI. Conclusion

The popularity of video streaming has increased considerably on mobile devices in the past years. Consumers do not only care about the quality of the content and the quality of the service, but also about the energy efficiency and the data efficiency of their device while they use the service. We identified adaptation algorithms as a good point in the service chain for optimizing the expected energy consumption and the expected data consumption.

In this paper, we combine an adaptive streaming heuristic and an energy efficiency scheme for video streaming using

user behavior statistics into a new adaptive streaming heuristic. Furthermore, we extend the heuristic with a data efficiency scheme by comparing the abandonment probability with tail energy that can be saved to determine a schedule for downloading the next segments. We investigate the performance of our heuristic by simulating a video session where users watch five consecutive videos. Simulated users abandon videos or skip through parts of a video based on viewer retention statistics of real YouTube videos and users.

Our results show that being aware of the user behavior and scheduling segment downloads efficiently can reduce the energy consumption of the video download by about 25% in WiFi networks and by $35-40\%$ in LTE networks with a buffer size of 80s. The wasted traffic that results from video segments that are downloaded but never viewed can be reduced by about 50% in the latter scenario. The gain in terms of energy and data comes at no cost in terms of quality and stalling compared to the baseline KLUDCP, except for scenarios with very long connection interruptions. For this case, we plan to investigate a Markov Chain based adaptation algorithm that performs very well in these situations [14] and possibly replace the KLUDCP component of our heuristic. Furthermore, we plan to extend the heuristic for 360-degree videos using head movement statistics of videos to enable long term viewport prediction.

ACKNOWLEDGMENT

This work was partly funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/1-2 and KE 1863/6 TR 257/43-1 ZI 1334/2-1. The authors would like to thank Markus Meier for working on the algorithm, implementing the simulation and conducting the evaluation.

REFERENCES

[1] V. Cisco, "Cisco visual networking index: Forecast and methodology 2016–2021.(2017)," 2017.

[2] A. Ericsson, "Ericsson mobility report," *Nov*, 2018.

[3] J. Janek and W. G. Zeier, "A solid future for battery development," *Energy*, vol. 500, no. 400, p. 300, 2016.

[4] M. Siekkinen, M. A. Hoque, and J. K. Nurminen, "Using viewing statistics to control energy and traffic overhead in mobile video streaming," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1489–1503, 2016.

[5] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over http in vehicular environments," in *Proceedings of the 4th Workshop on Mobile Video*. ACM, 2012, pp. 37–42.

[6] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Dataset: Hsdpa-bandwidth logs for mobile http streaming scenarios," 2012.

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

[8] C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista, and M. Mattavelli, "Automated QoE evaluation of dynamic adaptive streaming over http," in *Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*. Ieee, 2013.

[9] M. Zink, J. Schmitt, and R. Steinmetz, "Layer-encoded video in scalable adaptive streaming," *IEEE Transactions on Multimedia*, vol. 7, no. 1, 2005.

[10] L. Yitong, S. Yun, M. Yinian, L. Jing, L. Qi, and Y. Dacheng, "A study on quality of experience for adaptive streaming service," in *International Conference on Communications Workshops (ICC)*. IEEE, 2013.

[11] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner, "Assessing effect sizes of influence factors towards a QoE model for http adaptive streaming," in *Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2014.

[12] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, "A bitstream-based, scalable video-quality model for http adaptive streaming: Itu-t p. 1203.1," in *Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2017.

[13] 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*. ACM, 2017, pp. 197–210.

[14] J. W. Kleinrouweler, B. Meixner, J. Bosman, H. van den Berg, R. van der Mei, and P. Cesar, "Improving mobile video quality through predictive channel quality based buffering," in *30th International Teletraffic Congress (ITC 30)*. IEEE, 2018.

[15] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, "Taking a long look at quic: an approach for rigorous evaluation of rapidly evolving transport protocols," in *Proceedings of the 2017 Internet Measurement Conference*. ACM, 2017, pp. 290–303.

[16] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying QoE optimal adaptation of http adaptive streaming based on subjective studies," *Computer Networks*, vol. 81, 2015.

[17] C. Moldovan, L. Skorin-Kapov, P. E. Heegaard, and T. Hoßfeld, "Optimal fairness and quality in video streaming with multiple users," in *30th International Teletraffic Congress (ITC 30)*. IEEE, 2018.

[18] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3g networks," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 137–150.

[19] ——, "Top: Tail optimization protocol for cellular radio resource allocation," in *Network Protocols (ICNP), 2010 18th IEEE International Conference on*. IEEE, 2010, pp. 285–294.

[20] M. A. Hoque, M. Siekkinen, J. K. Nurminen, and M. Aalto, "Dissecting mobile video services: An energy consumption perspective," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*. IEEE, 2013, pp. 1–11.

[21] M. Seufert, V. Burger, and F. Kaup, "Evaluating the impact of wifi offloading on mobile users of http adaptive video streaming," in *Globecom Workshops (GC Wkshps), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[22] C. Schwartz, M. Scheib, T. Hoßfeld, P. Tran-Gia, and J. M. Gimenez-Guzman, "Trade-offs for video-providers in lte networks: Smartphone energy consumption vs wasted traffic," in *Energy Efficient and Green Networking (SSEEGN), 2013 22nd ITC Specialist Seminar on*. IEEE, 2013, pp. 1–6.

[23] M. Siekkinen, M. A. Hoque, J. K. Nurminen, and M. Aalto, "Streaming over 3g and lte: How to save smartphone energy in radio access network-friendly way," in *Proceedings of the 5th Workshop on Mobile Video*. ACM, 2013, pp. 13–18.

[24] C. Sieber, P. E. Heegaard, T. Hoßfeld, and W. Kellerer, "Sacrificing efficiency for quality of experience: YouTube's redundant traffic behavior," in *IFIP Networking 2016 Conference (Networking 2016)*, Vienna, Austria, May 2016.

[25] C. Moldovan, C. Sieber, P. Heegaard, W. Kellerer, and T. Hoßfeld, "Youtube can do better: Getting the most out of video adaptation," in *Teletraffic Congress (ITC 28), 2016 28th International*, vol. 3. IEEE, 2016.

[26] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 225–238.

[27] F. Dormancy, "Fast dormancy best practices," *GSM association, network efficiency task force*, 2010.

[28] X. Li, Z. Ma, and F. C. Fernandes, "Modeling power consumption for video decoding on mobile platform and its application to power-rate constrained streaming," in *Visual Communications and Image Processing (VCIP), 2012 IEEE*. IEEE, 2012, pp. 1–6.

[29] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "Qoe-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 2, p. 28, 2016.

[30] E. Liotou, T. Hoßfeld, C. Moldovan, F. Metzger, D. Tsolkas, and N. Passas, "Enriching http adaptive streaming with context awareness: A tunnel case study," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016.