Energy-Aware and Context-Aware Video Streaming on Smartphones

Xianda Chen, Tianxiang Tan and Guohong Cao
Department of Computer Science and Engineering
The Pennsylvania State University
E-mail: {xuc23, txt51, gxc27}@psu.edu

***Abstract*—Although streaming video at a higher bitrate (resolution) can lead to better Quality of Experience (QoE), a larger amount of data will have to be downloaded and processed on smartphones and thus consuming more energy. On a moving bus where the wireless signal is weak, more energy will have to be spent on maintaining high bitrate video streaming than at a static environment such as at home or a cafe where the wireless signal is strong. On the other hand, the user perceived QoE does not increase too much by watching high bitrate videos in a vibrating environment (i.e., a moving vehicle), because the perception of video quality is affected by the environment such as the vibration or shaking on a moving bus. To address this problem, we propose to save energy by considering the context (environment) of video streaming. To model the impact of context, we exploit the embedded sensors (e.g., accelerometer) in smartphones to record the vibration level during video streaming. Based on quality assessment experiments, we collect traces and model the impacts of video bitrate and vibration level on QoE, and model the impacts of video bitrate and signal strength on power consumption. Based on the QoE model and the power model, we formulate the energy-aware and context-aware video streaming problem as an optimization problem. We present an optimal algorithm which can maximize QoE and minimize energy. Since the optimal algorithm requires perfect knowledge of future tasks, we further propose an online bitrate selection algorithm. Through real measurements and trace-driven simulations, we demonstrate that the proposed algorithm can significantly outperform existing approaches when considering both energy and QoE.**

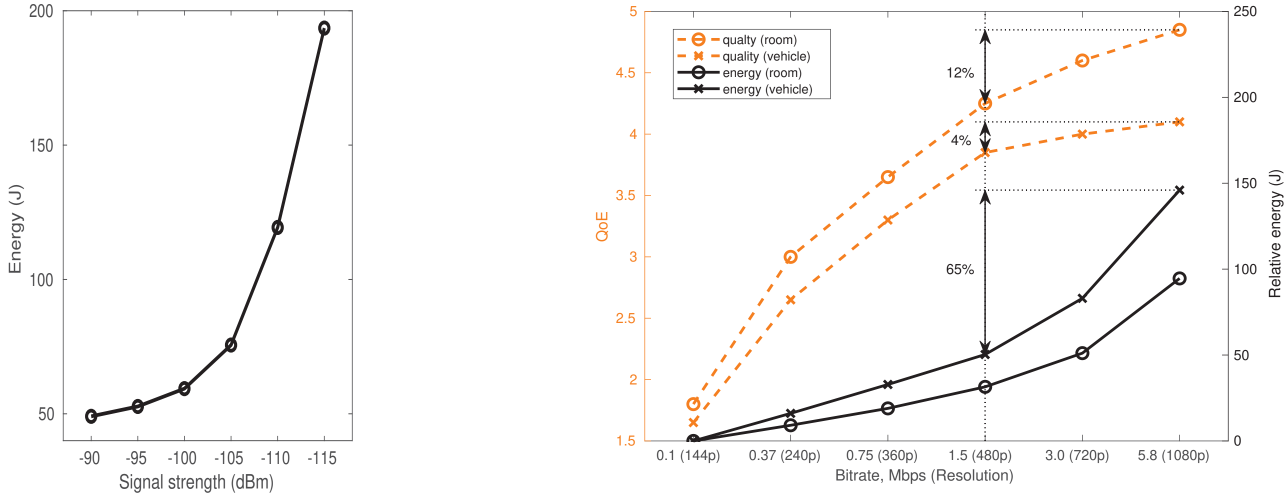
I. Introduction

Nowadays, video streaming has become the most popular application on smartphones. It is expected that mobile video traffic will account for over 78% of the mobile data traffic by 2021 [1]. While mobile networks offer very high peak bandwidth, video streaming over mobile networks still suffers from rapid and significant network fluctuations. To adapt for various network conditions, the Dynamic Adaptive Streaming over HTTP (DASH) protocol has been widely adopted for video streaming. With DASH, at the server side, the video is cut into a sequence of video segments, each of which is encoded with different bitrates, corresponding to different resolutions. At the client side, based on the estimated network bandwidth, the video player can dynamically determine the right bitrate for each segment, such that the video segment can be successfully downloaded before it is played to maintain good *Quality of Experience* (*QoE*).

Based on DASH, many existing bitrate adaptation algorithms [2, 3, 4, 5] focus on accurately predicting the network bandwidth, and then use higher bitrates to maintain better QoE. Although streaming video at a higher bitrate can lead to better QoE, a larger amount of data will have to be downloaded and processed on smartphones and hence consuming more energy. Since smartphones are powered by battery, energy efficiency is an important issue and energy-efficient video streaming has received considerable attention. Researchers have proposed various techniques [6, 7, 8] to reduce the power consumption of the wireless interface during video streaming, and have proposed techniques [9, 10, 11, 12] to reduce the energy consumption of video processing on smartphones.

Different from these energy-saving techniques, we propose to save energy by considering the context (environment) of video streaming; i.e., watching video on a moving bus/train or in a quiet room may have different QoE requirements. On a moving vehicle where the wireless signal is weak, it costs much more energy to maintain high bitrate video streaming than at a static environment such as at home or a cafe where the wireless signal is strong. On the other hand, the user perceived QoE may not increase too much by watching high bitrate videos in a vibrating environment such as on a moving vehicle. This is because the perception of video quality is affected by the environment such as the vibration or shaking on a moving vehicle. The vibration of the smartphone causes discomforts during video watching, which results in QoE degradation even with a high resolution. As a result, in such vibrating environments, reducing the bitrate of video streaming may significantly reduce the energy consumption, without degrading the QoE too much, and hence it is important to find a better tradeoff between QoE and energy considering the context of video streaming.

To support energy-aware and context-aware video streaming on smartphones, we have the following challenges: (1) How to model the impact of context on QoE? (2) How to select the right bitrate to minimize energy and maximize QoE? To answer the first question, we recruit twenty users to watch and rate videos under two contexts; i.e., on a moving vehicle and in a quiet room. To model the impact of context, we exploit the embedded sensors (e.g., accelerometer) in smartphones [13] to record the vibration level during video streaming. Based on these quality assessment experiments, we collect traces and model the impacts of video bitrate and vibration level on QoE, and model the impacts of video bitrate and signal strength on power consumption. Based on the QoE model and the power



(a) (b)

Fig. 1: (a): Total energy consumption to download 100MB data under various network conditions. (b): Perceived QoE and energy consumption as functions of bitrate under different environments (a quiet room or a moving vehicle).

model, we formulate the energy-aware and context-aware video streaming problem as an optimization problem. We first present an optimal algorithm which can maximize QoE and minimize energy. Since the optimal algorithm requires perfect knowledge of future tasks (which will not be available in practical scenarios), we further propose an online algorithm for video streaming.

In summary, this paper has the following contributions.

- We are the first to study the impact of context on QoE and energy consumption for video streaming on smartphones.
- We formulate the energy-aware and context-aware video streaming problem as an optimization problem. We first propose an optimal solution, which provides a performance upper bound. We then propose an online bitrate selection algorithm.
- We evaluate the proposed solutions with extensive trace-driven simulations. The evaluation results show that our online bitrate selection algorithm can significantly reduce the energy consumption while maintaining good QoE.

The rest of the paper is structured as follows. We introduce the background and motivation in Section II. We present the system model and the problem formulation in Section III. Section IV presents our energy-aware and context-aware video streaming algorithm. In Section V, we present the evaluation results. Section VI discusses related work and Section VII concludes the paper.

II. Background and Motivation

In DASH, the video is broken into a sequence of small HTTP-based segments, where each segment is encoded into multiple copies with various bitrates. Based on the network quality, the segment with the right bitrate is used for streaming.

Streaming video at a higher bitrate leads to better quality, but it requires to download and process more data, and thus consuming more energy. When the network condition becomes worse, much more energy will be consumed. Fig. 1(a) shows the energy consumption of downloading 100MB data under various network conditions. The measurement was done using an LG Nexus 5X smartphone using T-Mobile LTE network. In the measurement, we focus on the power consumption of the wireless interface. As can be seen, when the signal strength decreases, the energy consumption significantly increases. For example, the energy consumption increases from 49J to 193J as the signal strength changes from -90 dBm to -115 dBm.

One way to reduce the energy consumption is to use the video segment with low bitrate. However, this may reduce the video quality and affect the QoE under different contexts. To understand the impact of context on QoE and energy consumption for video streaming on smartphones, we conducted some experiments. With approval by our Institutional Review Board (IRB), twenty subjects were recruited to watch Youtube videos of various bitrates (resolutions) under two different contexts (environments): in a quiet room and on a moving vehicle. After watching each video, the subjects rate the perceived quality using the nine-grade numerical quality scale (9 denotes "excellent" and 1 denotes "bad") based on the ITU-T Recommendation P.910 [14], which is then transformed to the five-level rating scale, using $\quad = 1 + 4 \cdot \frac{\quad}{9}$. The value is used to represent the QoE. The energy consumption is calculated using the power models that will be detailed in Section III-C.

As shown in Fig. 1(b), the QoE does not improve too much when the resolution is very high (e.g., 720p) for smartphones. We also see that the QoE varies with context. When the resolution drops from 1080p to 480p, the QoE degrades much slower on a moving vehicle (4%) than in a quiet room (12%). On the other hand, the network condition on a moving vehicle is worse than that in a quiet room. As a result, reducing the resolution from 1080p to 480p can save 65% energy when watching videos on a moving vehicle, while only degrading the QoE by 4%. Thus, it is important to consider energy, context, and QoE together for video streaming.

To model QoE considering the context of video streaming, we can exploit the embedded sensors (e.g., accelerometer) in smartphones [13]. Based on these sensors, we can differentiate whether the user is in a quiet room or on a moving vehicle. We will present the detail of the QoE model and formulate the context-aware video streaming problem in the next section.

III. System Model and Problem Formulation

In this section, we introduce the video streaming model, the QoE model, the power model, and the problem formulation.

A. Video Model

We follow the DASH video streaming process. The video is broken into a sequence of n HTTP-based segments, where each segment contains L seconds of video. Each segment has V versions of copies corresponding to V different bitrates on the server side. Based on the network quality, the client requests the segment with the right bitrate level from the server. More specifically, the video streaming process is modeled as n data transmission tasks, corresponding to transmitting n video segments. Let T_i denote the i^{th} task, and let T_i^j denote the i^{th} task where the video segment is encoded with bitrate index j ($j \in \{1, 2, ..., V\}$).

Since many users may skip or early quit during video streaming, to save bandwidth, similar to Youtube, we use a buffer threshold (β) to limit the amount of the video data to be downloaded. β is defined in terms of seconds. Before the buffered data reaches β, i.e., the video length of downloaded but not yet viewed video in the buffer is less than β, the video player can download the next segment. When the buffered data reaches β, the player stops the downloading process. It will wait until the video length of the buffered video is less than β before downloading the next segment. Let $B_i \in [0, \beta]$ denote the amount of video data in the buffer when the client requests the i^{th} segment. To avoid stall events (or rebuffering), the i^{th} segment should be completely downloaded before B_i is drained out by the video player at the client.

B. QoE Model

As explained in Section II, the QoE model should consider the context of video steaming; i.e., watching video on a moving vehicle or in a quiet room may have different QoE requirements. The QoE can be modeled with Equation 1, where Q is the user perceived QoE, Q_o is the "original" quality without considering any quality loss, I_t is the quality impairment during data transmission, and I_v is the quality impairment resulting from vibration during video playback. Similar to [15], Q_o is modeled with a Michaelis-Menten function, as shown in Equation 2, where b is the bitrate, and c_1 and c_2 are the model parameters which are determined by the subjective quality assessment experiments.

$$Q = Q_o - I_t - I_v \tag{1}$$

$$Q_o = max(1, min(5, 1 + 4 \cdot \frac{c_1 \cdot b}{c_2 + b})) \tag{2}$$

$$I_t = f_r \cdot I_r + f_b \cdot I_b \tag{3}$$

$$I_v = c_3 + c_4 \cdot exp(c_5 \cdot b \cdot v) \tag{4}$$

Similar to [16, 17], the QoE impairment during data transmission can be modeled by Equation 3, considering the impact of rebuffering events and bitrate changes. Mok *et. al* [16] has defined three levels (low, medium, and high) of rebuffering impacts on QoE by choosing I_r, and we use their base level (i.e., $I_r = 0.742$), where the rebuffering duration is less than one second. The rebuffering frequency is defined as $f_r = \frac{(S_i/R_i - B_i)_+}{B_i}$, where S_i is the segment data size, R_i is the downloading throughput, and $(x)_+ = max\{x, 0\}$. For the impact of bitrate change, we adopt the results from [17]; i.e., reducing the bitrate of a segment by 3 Mbps has the same penalty as one second rebuffering. Thus, we have $I_b = 0.742$ and $f_b = \frac{(b_{i-1} - b_i)_+}{3.0}$, where b_i and b_{i-1} are the bitrate of the i^{th} and $(i-1)^{th}$ video segment, respectively.

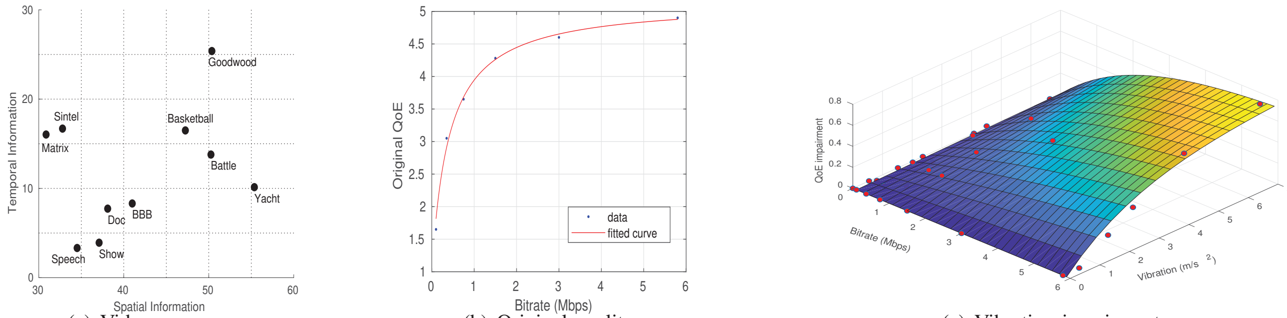
To model the impact of context, we exploit the embedded sensors (e.g., accelerometer) in smartphones [13] to record the vibration level during video streaming. Since we are only interested in the acceleration associated with the vibration of the smartphone, we subtract the force caused by gravity from the raw accelerometer data. The vibration level is formulated with Equation 5, where m_a is the average value of M acceleration samples in the sampling window and f_a is the average variation between consecutive acceleration samples. Here we consider m_a and f_a equally; i.e., $\alpha = 0.5$. With the vibration level, we model the impact of vibration impairment with Equation 4, where b is the bitrate, v is the vibration level, and c_3, c_4 and c_5 are the model parameters.

$$v = \alpha \cdot m_a + (1 - \alpha) \cdot f_a \tag{5}$$

$$m_a = \frac{1}{M} \sum_{m=1}^{M} \sqrt{a_{m,x}^2 + a_{m,y}^2 + a_{m,z}^2} \tag{6}$$

$$f_a = \frac{1}{(M-1)} \sum_{m=2}^{M} \sqrt{\sum_{l=\{x,y,z\}} (a_{m,l} - a_{m-1,l})^2} \tag{7}$$

To determine parameters c_1, c_2, c_3, c_4, c_5, in Equation 2 and Equation 4, we performed subjective quality assessment experiments with 20 users following ITU-T Recommendations. The subjects are asked to watch 10 videos downloaded from Youtube under two contexts: in a quiet room and on a moving vehicle. Table I summaries the characteristics of the videos. To demonstrate that the chosen videos cover a wide range of different types and genres following the standard ITU-T



(a) Video sequences (b) Original quality (c) Vibration impairment

Fig. 2: (a): Average spatial and temporal information of the test videos. (b): The "original" quality of a video as a function of bitrate. (c): The QoE impairment due to vibration.

TABLE I: The test videos.

Genre	Explanation	Genre	Explanation
Speech	Speech on TV	Matrix	A fight scene in The Matrix (movie)
Show	Allen show	Battle	A battle scene in The Hobbit (movie)
Doc	Documentary	Basketball	Sport
BBB	Big Buck Bunny (animation)	Yacht	Moving yacht
Sintel	Sintel (movie)	Goodwood	Horseracing

TABLE II: The resolution and bitrate for video dataset.

Resolution	Bitrate (Mbps)
1080p	5.80
720p	3.00
480p	1.50
360p	0.75
240p	0.375
144p	0.10

P.910 [14], we calculate the temporal and spatial information of the videos which are shown in Fig. 2(a), where a video with higher temporal information has more changing scenes, and a video with higher spatial information has more spatial details in video frames. As can be seen, the chosen videos cover a wide range of different types and genres. The videos are encoded into different bitrate versions at 30 fps, and Table II shows the bitrate for each resolution. After watching a video, the subjects rate the perceived quality as discussed in Section II. In this quality assessment experiment, the videos are locally cached in the smartphone. Then, there will not be any data transmission, and we can focus on quantifying the impact of the vibration factor.

We first consider video watching in a quiet room. The perceived quality of watching video in a quiet room is used as the "original" quality (i.e.,), and the results are shown in Fig. 2(b). As can be seen from the figure, increases with the increase of the video bitrate. When the bitrate becomes very high, further increasing the bitrate will not lead to significant increase in the QoE, which is consistent with the results reported in [18, 19]. With the least squares regression method, we can get the fitted curve, where the parameters and are shown in Table III.

To see the impact of vibration on video quality, we measure

TABLE III: The parameters of QoE model.

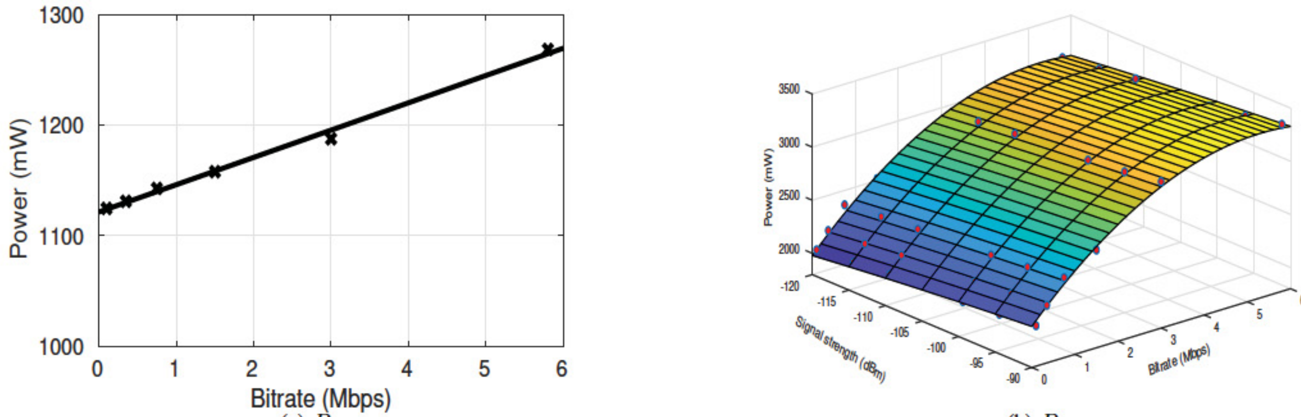
Coefficient					
Value	1.036	0.429	0.782	-0.782	0.0648

the quality impairment ( ) caused by vibration; i.e., the QoE difference between watching the same video with the same bitrate in a quiet room and on a moving vehicle. Based on the accelerometer data collected by the smartphone during video watching, and Equation 5, we can calculate the vibration level ( ). The relationship among QoE impairment ( ), vibration level ( ), and video bitrate ( ) is shown in Fig. 2(c). As shown in the figure, when the bitrate is very small (e.g., 0.1 Mbps), the QoE is very poor regardless of the context and thus the vibration impairment is almost zero. When the vibration level is very low (i.e., in a quiet room), the quality impairment is also very small. The quality impairment significantly increases when the bitrate and the vibration level increase. For example, when the vibration level increases from 2 to 6, the quality impairment grows from 0.049 to 0.184 for 1.5 Mbps videos, and the impairment grows from 0.174 to 0.549 for 5.8 Mbps videos. With the least squares regression method, we can get the fitted surface, where the parameters are shown in Table III.

C. Power Model

To model the power consumption during video streaming, we collect real measurement data by watching a short video "Everything Wrong With Transformers" from Youtube with different bitrates (i.e., 144p, 240p, 360p, 480p, 720p and 1080p) at different signal strengths. We use a rooted LG Nexus 5X smartphone running Android 7.0, which uses T-Mobile LTE network. Tcpdump is used to collect the network trace, and the Monsoon power monitor is used to measure the power level during video streaming. We also collect the wireless signal strength during video streaming by using an Android ADB shell *dumpsys telephony.registry* at the background. Using the collected data traces, we build two different power models based on whether there is data transmission (video downloading) or not.

When there is no data transmission, i.e., the smartphone plays the buffered video segment, the power consumption



(a) P_b (b) P_t

Fig. 3: The power model considering signal strength and video bitrate.

TABLE IV: The power models.

State	Power (mW)
Without data trans.	$P_b(b) = 1121.5 + 24.71b$
With data trans.	$P_t(b, s) = 2301.2 + 439.6b$ $41.57b^2 \quad 2.96s \quad 0.047s^2$

(denoted by P_b) is only affected by the video bitrate. For high bitrate video, more data will be processed such as decoding and rendering, and hence more power will be consumed. Since P_b increases with the video bitrate, P_b is modeled as a linear function of the video bitrate. With the least squares regression method, we can get the fitted curve as shown in Fig. 3(a).

With data transmission, the power consumption (denoted by P_t) is affected by both video downloading and video processing, and hence it is affected by the bitrate (b) and the wireless signal strength (s). To model the relationship between power consumption and these two factors, we use the quadratic function [20] for curve fitting. Fig. 3(b) shows the fitted surface, where the R-square value of the fitting is 0.9866, which means a very high accuracy. The results are summarized in Table IV, where the video bitrate b is in terms of Mbps, the signal strength s is in terms of dBm and the power level is in mW.

Energy Consumption of A Task: Based on our power models, we can calculate the energy consumption of video streaming which consists of a set of tasks. We calculate the energy of task T_i^j based on if there is rebuffering.

Case (a) (No rebuffering): When there is no rebuffering, the energy consumption of task T_i^j depends on the relationship among L, β, B_i, and the downloading time $\frac{S_i}{R_i}$, where S_i is the segment data size and R_i is the downloading throughput.

If the video length of the downloaded but not yet viewed video after downloading the i^{th} segment is less than the buffer threshold β, i.e., $(B_i - \frac{S_i}{R_i} + L) < \beta$, the video player starts to download the next segment when T_i^j is completed. In this case, the energy of task T_i^j is the energy consumed during the period of downloading the i^{th} segment.

If the buffered data after downloading the i^{th} segment reaches β, i.e., $(B_i - \frac{S_i}{R_i} + L) \geq \beta$, the player stops the downloading process and waits until the buffered video is less than β before downloading the next segment. In this case, the energy of task T_i^j includes the energy consumed during the period of downloading the i^{th} segment, and the energy consumed during the waiting period before starting to download the next segment.

Since the buffered data when the player downloads the i^{th} segment is B_i, there are $k = \lceil B_i/L \rceil$ video segments in the buffer, where the $(i-k)^{th}$ video segment is being played. Thus, we use $P_t(b_{i-k}, s)$ to calculate the energy consumed during the period of downloading the i^{th} segment. If more than one video segments will be played during the downloading period, since the power consumption P_t varies with video bitrate, we calculate the energy consumed during the downloading period by summing up the energies consumed when different bitrate videos being played during the downloading period. Since the waiting time is less than L, i.e., the $(i-k)^{th}$ video segment is being played during the waiting period, we use $P_b(b_{i-k})$ to calculate the energy consumed during the waiting period. P_t and P_b can be calculated based on Table IV.

In summary, the energy of task T_i^j when there is no rebuffering can be calculated with Equation 8, where $(x)_+ = max\{x, 0\}$.

$$
\begin{aligned}
E_a(T_i^j) = & P_t(b_{i-k}, s)(\frac{S_i}{R_i} - (\lceil S_i/R_i/L \rceil - 1)L) \\
& + \sum_{g=1}^{\lceil S_i/R_i/L \rceil - 1} P_t(b_{i-k+g}, s)L \\
& + P_b(b_{i-k})(B_i - \frac{S_i}{R_i} + L - \beta)_+
\end{aligned} \tag{8}
$$

Case (b) (Rebuffering): When there is rebuffering, i.e., $B_i < \frac{S_i}{R_i}$, we can divide the video downloading into two parts: B_i and $\frac{S_i}{R_i} - B_i$. During the first part, the buffered video data is played out; while there is no video playback during the second part, i.e., rebuffering. We calculate the energy of T_i^j by summing up the energy consumptions of these two parts.

Since there is no video playback during the rebuffering, we use $(0\ \ )$ to calculate the energy consumed by downloading video during the rebuffering. Thus, the energy consumption of when there is rebuffering can be calculated with Equation 9.

$$
\begin{aligned}
(\ \ ) = &\ (\ \_\ \ )(\ \ - (\ \ -1)\ \ ) \\
&+ \sum^{-} (\ \_\ \ \ ) \\
&+ (0\ \ )(\text{—} - \ \ )
\end{aligned}
\tag{9}
$$

In summary, the energy of task can be calculated with Equation 10.

$$
(\ \ ) = \begin{cases} (\ \ ) & \text{if } \frac{\ i}{\ i} \leq \\ (\ \ ) & \text{otherwise} \end{cases}
\tag{10}
$$

D. Problem Formulation

In this subsection, we formalize the energy-aware and context-aware video streaming problem. To determine the right bitrate for each video segment (task), we introduce a binary variable for bitrate selection, where $= 1$ if the video segment in task is encoded with bitrate index , i.e., ; otherwise, $= 0$. Since only one bitrate is selected for each task, $\sum \ \ = 1$.

In our energy-aware and context-aware video streaming, the goal is to minimize the energy consumption and maximize the QoE, and this can be achieved by selecting the right bitrate for each video segment downloaded in each task. This is a multi-objective optimization problem, and we apply the weighted sum method [21] to formalize it. For task , the QoE (( )) can be calculated with Equation 1, and the energy consumption (( )) can be calculated with Equation 10. Then, the optimization problem can be formulated as follows.

$$
\begin{aligned}
\text{minimize} \quad & \sum\sum \ \ (\ \frac{(\ \ )}{(\ \ )} - (1 - \ \ )\frac{(\ \ )}{(\ \ )}) \\
\text{subject to} \quad & \sum \ \ = 1 \qquad \forall
\end{aligned}
\tag{11}
$$

The objective of this optimization problem is to minimize energy and maximize QoE under the constraint that only one bitrate is selected for each task, $\sum \ \ = 1$. Since ( ) and ( ) are measured using different units, they are normalized with the highest bitrate (i.e., ( ) and ( )). is a weighting factor. With a smaller , the optimization problem puts more weight on maximizing QoE; with a larger , minimizing energy is more important.

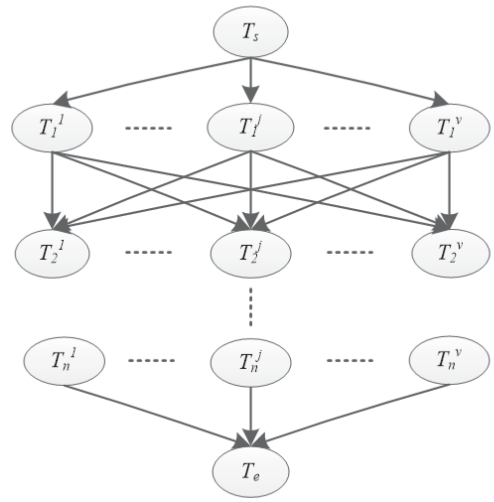


Fig. 4: Mapping the energy-aware and context-aware video streaming problem to the shortest path problem.

IV. Energy-Aware and Context-Aware Video Streaming

In this section, we present our energy-aware and context-aware video streaming algorithms. We first present an optimal algorithm which requires knowledge of all future tasks. Since it is impossible to have such knowledge in practice, we present an online bitrate selection algorithm by removing such assumption.

A. The Optimal Algorithm

For a video streaming consisting of tasks, our goal is to find bitrate decisions to minimize the energy and maximize the QoE. The process of determining the right bitrate for each video segment (each task) can be mapped to the shortest path problem as shown in Fig. 4.

Let node and denote the start and end of the video streaming process, respectively. Since there are bitrates available, we add nodes for each task ( ), where corresponds to the video segment downloaded in and it is encoded with bitrate index . As shown in Fig. 4, we add edges from to each node of , and from each node of to . We add an edge from each node of task to all nodes of the next task. By considering both energy consumption and QoE, the weight of an edge is defined as $(\ \frac{\ ^{j}_{i}}{V_i} - (1 - \ \ )\frac{\ ^{j}_{i}}{V_i})$. For edges from nodes of to , their weights are defined as 0.

In this graph, we consider all bitrate cases of each task and all possible schedule paths between tasks, so each path from to maps to a bitrate selection sequence for the video streaming, and vice versa. As a result, the shortest path from to corresponds to the optimal bitrate selection of all tasks in the video streaming process such that the QoE is maximized and the energy is minimized.

Based on the graph, we can use the Dijkstra's algorithm to find the shortest path. Given a video streaming process consisting of tasks, the graph has ( ) nodes and ( ) edges. As the time complexity of the Dijkstra's algorithm is

$((\ \ +\ \ )\ \ \ )$, where is the number of nodes and is the number of edges, the time complexity of the optimal algorithm is $(\ \ \ \ (\ \ ))$.

B. The Online Bitrate Selection Algorithm

Since the optimal algorithm requires the complete knowledge of all future tasks, which is impossible in practice, it can only be served as a performance upper bound. In this subsection, we propose an online bitrate selection algorithm, which first estimates the available bandwidth and the vibration level, and then determines the right bitrate for each video segment.

To estimate the network bandwidth, similar to [2], we use the harmonic mean of the downloading throughputs of the past several segments to estimate the network bandwidth. Since the network condition varies widely during video streaming, especially when the user is on a moving vehicle, some downloading throughputs can be much higher or lower than others among the past several segments. The harmonic mean is used to eliminate the impacts of these fluctuations. More bandwidth estimation methods can be found in [3, 22, 23], which is out of the scope of this paper.

The vibration level is estimated based on the collected accelerometer data in a time window from the current time to the previous $0\ 2*$, where is small, i.e., 30 seconds. This vibration level is used to determine the right bitrate for the video segment. Since the player buffer threshold ( ) is very small, the downloaded video segment will be played after very short time. Since the time interval between the video downloading and the video playback is very small, the vibration level when downloading video can be used to estimate the vibration level when the downloaded video is played.

Based on the estimated bandwidth and the vibration level, the online bitrate selection algorithm can calculate the energy consumption (( )) and the user perceived QoE (( )) when downloading the video segment encoded with bitrate . To minimize energy and maximize QoE, we apply the same objective function of Equation 11, i.e., $\frac{\ _i^j}{\ _i^V} - (1-\ )\frac{\ _i^j}{\ _i^V}$.

The online bitrate selection algorithm is described in Algorithm 1. Due to network variations, a sudden large bitrate increase may result in more rebuffering events and frequent bitrate changes, and a sudden large bitrate drop may lead to severe QoE impairment. To deal with these problems, the online algorithm first computes a *reference* bitrate for the video segment (line 4) based on Equation 11. Then, it determines the final bitrate, based on the relationship between the reference bitrate and the bitrate of the previous video segment.

If the reference bitrate is higher than that of the previous video segment, the algorithm selects the bitrate which is one level higher than the bitrate of the previous video segment (line 5-6). This gradual bitrate change reduces the QoE impairment caused by frequent bitrate changes due to network variations. If the network bandwidth is consistently high, i.e., the reference bitrate is higher than the bitrate of the previous video segment for consecutive video segments, the video bitrate will gradually increase to the reference bitrate.

Algorithm 1: The Online Bitrate Selection Algorithm

Data: , , , the bitrate of previous segment _

Result: Bitrate level for the video segment

```
1  Procedure Bitrateselection( )
2      ← estimated bandwidth
3      ← vibration level calculated with Eq. (5)
4      ← argmin ∈{ }( ─── − (1− )─── )
5      if  _ then
6          ←  _ +1
7      else if  _ then
8          ← max{ | ∈{  _ }&( ── ≤  )}
9          //  : data size if video is encoded with bitrate
10     end
11     ←
```

If the reference bitrate is lower than that of the previous video segment, the algorithm does not immediately drop to the reference bitrate to reduce the QoE impairment. Instead, for each bitrate decrease, the online algorithm searches from the bitrate of the previous video segment to the reference bitrate, and finds the first bitrate which can be used to successfully download the video segment before the buffered data is drained out (line 7-9). If the network bandwidth is consistently low, i.e., the reference bitrate is lower than the bitrate of the previous video segment for consecutive video segments, the video bitrate will eventually decrease to the reference bitrate.

V. Performance Evaluations

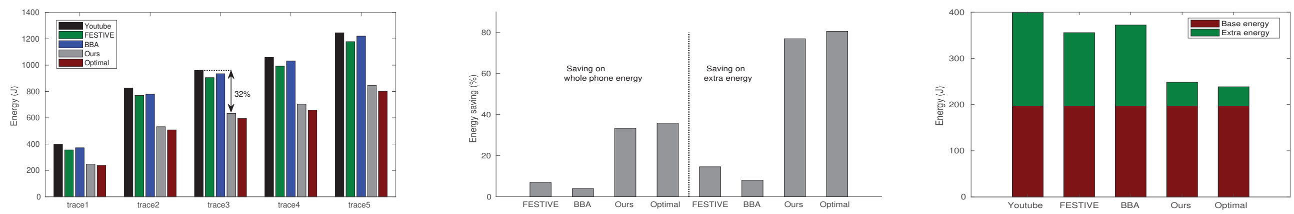
In this section, we evaluate the performance of the proposed algorithms through extensive trace-driven simulations and compare it with other existing algorithms.

A. Simulation Setup

We collect traces by watching videos from Youtube using LG Nexus 5X smartphone. Subjects watch five videos with various video length and data size, and under different contexts, as shown in Table V. We collect three kinds of traces: the network trace by using Tcpdump to extract the downloading time and the downloading data size, the signal strength trace by using a ADB shell, and the accelerometer data in the smartphone. For simulations, each video is cut into 2 seconds segments and is encoded with fourteen bitrates, i.e., {0.1, 0.2, 0.24, 0.375, 0.55, 0.75, 1.0, 1.5, 2.3, 2.56, 3.0, 3.6, 4.3, 5.8} Mbps. We set the buffer threshold $= 30$ seconds. We equally consider minimizing energy and maximizing QoE; i.e., $= 0\ 5$.

Based on these traces, we compare the performance of the following approaches.

- `Youtube`: Video streaming with the original Youtube app at a bitrate of 5.8 Mbps (i.e., with resolution of 1080p).



(a) (b) (c)

Fig. 5: (a): Comparison of different approaches on energy consumption. (b): Energy saving compared to Youtube. (c): The base energy and the extra energy for trace 1.

TABLE V: Video traces.

Video ID	Length (sec)	Data size (MB)	Avg. vibration
1	198	65.1	6.83
2	371	123.8	2.46
3	449	140.6	6.61
4	498	152.2	6.41
5	612	173.1	5.23

- `FESTIVE [2]`: A throughput-based bitrate adaptation approach, which uses the harmonic mean of the last 20 throughput measurements to estimate the available bandwidth, and then selects the highest available bitrate that is just below the estimated bandwidth.
- `BBA [24]`: A buffer-based bitrate adaptation approach. BBA uses throughput to control video bitrate at the startup phase. After reaching the steady state, BBA maps the current buffer level to bitrate selection using a linear function.
- `Ours`: The online bitrate selection algorithm, which selects the bitrate that minimizes energy and maximize QoE.
- `Optimal`: The optimal algorithm, which is impossible in practice, so it only provides a performance upper bound.

B. Energy Comparison

The energy consumption is calculated based on the power models shown in Section III-C. As shown in Fig. 5(a), we compare the energy consumption of different approaches based on the measured traces. As can be seen in the figure, compared to Youtube which downloads video segments at bitrate of 5.8 Mbps and consumes the most energy, other four approaches can save energy due to the bitrate adaptations. FESTIVE always downloads video segments encoded at the highest available bitrate that is just below the estimated bandwidth, and consumes much energy. BBA is more aggressive to download higher bitrate segments after the buffer reaches the steady state, i.e., BBA requests the highest bitrate after the buffered data is larger than the pre-defined upper threshold, thus consumes much more energy compared to FESTIVE. Compared to FESTIVE and BBA, our approach can save more energy, since our approach takes into account the context of video streaming and download low bitrate videos when the vibration level is high. As shown in Fig. 5(b), our approach can save 33% energy on average, which is very close to Optimal (36%), and is much higher than FESTIVE (7%) and BBA (4%)

TABLE VI: The power model validation.

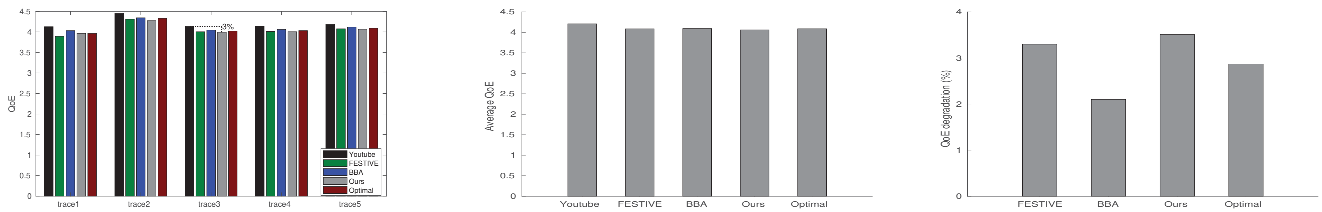
Bitrate (Mbps)	Measured energy (J)	Calculated energy (J)	Error ratio
5.8	708.13	713.59	0.77%
3.0	648.69	658.62	1.53%
1.5	637.36	622.55	2.32%
0.75	615.69	609.79	0.96%
0.375	608.04	597.75	1.69%
0.1	597.02	589.38	1.28%

The energy saving will be more significant for smartphones with smaller screen size. As shown in Fig. 5(c), we separate the total energy consumption during video streaming into two parts: base energy and extra energy consumption. The base energy is the energy consumed when all video segments are encoded with the lowest bitrate, which includes the energy consumed by the screen and the energy consumed by data transmission and video processing. Thus, the base energy is the minimum energy consumption for video streaming. All video streaming approaches will try to choose bitrates higher than the lowest bitrate to improve QoE, at the cost of more energy consumption. The extra energy is the energy difference between the energy consumed by the selected video streaming approach and the base energy. As shown in Fig. 5(b), when only considering the extra energy, our approach can save 77% energy which is very close to Optimal (80%), and it is much higher than FESTIVE (15%) and BBA (8%).

Power model validation: To validate the power models, we compare the energy consumption measured by the Monsoon power monitor and the energy consumption calculated using the power models. For each video, we first identify the download periods from the packet trace obtained by Tcpdump, and obtain the signal strength during each download period from the signal strength trace. We then calculate the energy consumption using the power models shown in Section III-C. Here we show an example when the signal strength is -90 dBm. The energy consumption for different bitrate videos based on the real power trace is shown in Table VI. The calculated energy consumptions are very close to the real measurements, which indicates that our power models are pretty accurate. The error ratio is consistently less than 3%, with an average of 1.43%.

C. QoE Comparison

Fig. 6(a) compares the QoE of all approaches. As can be seen, Youtube outperforms other approaches for all traces



(a) (b) (c)

Fig. 6: (a): Comparison of QoE for each trace. (b): The average QoE for each approach. (c): QoE degradation compared to Youtube.

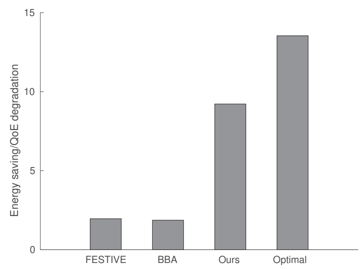


Fig. 7: The ratio of energy saving over QoE degradation.

because Youtube requests all video segments at the highest bitrate, and it suffers no quality impairment from bitrate changes. However, the QoE gap between Youtube and others is very small. This is because the QoE for video streaming on smartphones does not improve too much when the video bitrate is very high, and the perceived quality is highly affected by the vibrating environment. Compared to other traces, the QoE for trace 2 is much better for all approaches due to its low vibration level.

The overall QoE for all approaches is shown in Fig. 6(b). It can be seen that our approach can achieve very high QoE. This is because our approach considers the network bandwidth and the vibration level when selecting video bitrate for each video segment. Our approach requests video segments encoded at the most suitable bitrate under the vibrating environment, and thus gains high QoE. After reaching the steady state, BBA is aggressive to request higher bitrate videos, and thus gains higher QoE at the cost of higher energy consumption. Fig. 6(c) shows the QoE degradation compared to Youtube. Compared to Youtube, the average QoE degradation of our approach is 3.5%, similar to FESTIVE (3.3%) and BBA (2.1%).

D. Considering both Energy and QoE

For energy-aware and context-aware video streaming, the goal of our approach is to minimize energy and maximize QoE. As shown in Fig. 5(a) and Fig. 6(a), for trace 3, our approach save up to 32% energy at the cost of only 3% QoE degradation. We use the ratio of energy saving over QoE degradation to evaluate the overall performance, and the result is shown in Fig. 7. Since the QoE does not improve too much when the bitrate is very high for video streaming on smartphones and is highly affected by the vibrating environment, FESTIVE and BBA waste too much energy on maintaining high bitrate video streaming in the vibrating environment. As can be seen, on average, our approach achieves better performance compared to FESTIVE (4.8X) and BBA (5.1X).

VI. Related Work

QoE. There has been considerable research on maximizing QoE for DASH based video streaming [2, 5, 16, 17, 24, 25, 26, 27]. Researchers [16, 25] have proposed to model QoE for DASH based video streaming with multiple metrics: average video bitrate, bitrate changes between successive segments, and rebuffering event. FESTIVE [2] balances stability and efficiency, and provides fairness among video players. Rather than estimating network bandwidth, researchers [5, 24] have also proposed to use the current buffer level to determine the video bitrate. Li *et al.* [26] propose a probe and adapt bitrate adaptation scheme. Yin *et al.* [17] propose a control-theoretic model, which optimizes the user perceived QoE by considering throughput and buffer information. In Pensieve [27], a neural network is trained to select the video bitrate of future video segments. These methods try to maximize the QoE by considering the quality impairment caused by data transmission, and none of them considers the quality impairment caused by vibrations.

Energy Consumption. Researchers have proposed various techniques [6, 8, 28] to reduce the energy consumption of the wireless interface during video streaming. Hu *et al.* [6] proposed techniques to save energy based on whether the user tends to watch video for a long time, skip, or early quit. Hoque *et al.* [8] proposed a download scheduling algorithm base on crowd-sourced viewing statistics. Wu *et al.* [28] proposed an energy efficient video streaming scheme over heterogeneous networks. There has been some research [7, 29, 30] on reducing the tail energy in cellular networks during data transmission. Researchers have also proposed techniques [9, 10, 11, 12, 31, 32] to reduce the energy consumption of video processing on smartphones. Yang *et al.* [9] proposed to save energy for video streaming by adaptively adjusting the CPU frequency. Geng *et al.* [10] proposed an energy-efficient computational offloading scheme for energy-intensive video processing on multicore-based mobile devices. He *et al.* [31] proposed to save energy through dynamic resolution scaling based on the user-screen distance. Other researchers [11, 12, 32] focused on dynamically adapting the video bitrate and the display brightness to save energy. Complementary to these energy-saving techniques, we take a different approach to save energy by considering the context of video streaming.

VII. Conclusions

In this paper, we proposed to save energy by considering the context of video streaming; i.e., watching video on a moving vehicle or in a quiet room may have different QoE requirements. Based on quality assessment experiments, we collected traces and modeled the impacts of video bitrate and vibration level on QoE, and modeled the impacts of video bitrate and signal strength on power consumption. Based on the QoE model and the power model, we formulated the energy-aware and context-aware video streaming problem as an optimization problem. We presented an optimal algorithm which can maximize QoE and minimize energy. Since the optimal algorithm requires perfect knowledge of future tasks which is not available in practice, we further propose an online bitrate selection algorithm. Through real measurements and trace-driven simulations, we demonstrate that our online bitrate selection algorithm can significantly reduce the energy consumption (33%) with only small QoE degradation (3.5%). The results also show our algorithm can significantly outperform other existing approaches when considering both energy and QoE.

Acknowledgement

This work was supported in part by the National Science Foundation under grants CNS-1526425 and CNS-1815465.

References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021 White Paper*. http://goo.gl/DXWFyr.

[2] J. Jiang, V. Sekar, and H. Zhang. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With FESTIVE. *IEEE/ACM Trans. on Networking*, 22(1):326–340, 2014.

[3] A. H. Zahran, D. Raca, and C. Sreenan. ARBITER+: Adaptive Rate-Based InTElligent HTTP StReaming Algorithm for Mobile Networks. *IEEE Trans. on Mobile Computing*, 2018.

[4] Y. Qin, R. Jin, S. Hao, K. R. Pattipati, F. Qian, S. Sen, B. Wang, and C. Yue. A Control Theoretic Approach to ABR Video Streaming: A Fresh Look at PID-Based Rate Adaptation. In *IEEE INFOCOM*, pages 1–9, 2017.

[5] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. Bola: Near-Optimal Bitrate Adaptation for Online Videos. In *IEEE INFOCOM*, pages 1–9, 2016.

[6] W. Hu and G. Cao. Energy-Aware Video Streaming on Smartphones. In *IEEE INFOCOM*, pages 1185–1193, 2015.

[7] Y. Yang and G. Cao. Prefetch-Based Energy Optimization on Smartphones. *IEEE Trans. Wireless Commun.*, 17(1):693–706, 2018.

[8] M.A. Hoque, M. Siekkinen, , and J.K. Nurminen. Using Crowd-Sourced Viewing Statistics to Save Energy in Wireless Video Streaming. In *ACM MobiCom*, pages 377–388, 2013.

[9] Y. Yang, W. Hu, X. Chen, and G. Cao. Energy-Aware CPU Frequency Scaling for Mobile Video Streaming. *IEEE Trans. on Mobile Computing*. To appear. DOI: 10.1109/TMC.2018.2878842.

[10] Y. Geng, Y. Yang, and G. Cao. Energy-Efficient Computation Offloading for Multicore-Based Mobile Devices. In *IEEE INFOCOM*, pages 1–9, 2018.

[11] J. S. Leu, M. C. Yu, C. Y. Liu, A. P. Budiarsa, and V. Utomo. Energy Efficient Streaming for Smartphone by Video Adaptation and Backlight Control. *Computer Networks*, 113:111–123, 2017.

[12] Z. Yan and C. W Chen. RnB: Rate and Brightness Adaptation for Rate-Distortion-Energy Tradeoff in HTTP Adaptive Streaming over Mobile Devices. In *ACM MobiCom*, pages 308–319, 2016.

[13] X. Sun, Z. Lu, W. Hu, and G. Cao. Symdetector: Detecting Sound-Related Respiratory Symptoms Using Smartphones. In *ACM UbiComp*, pages 97–108, 2015.

[14] *ITU-T P.910: Subjective video quality assessment methods for multimedia applications*. https://www.itu.int/rec/T-REC-P.910-200804-I.

[15] K. Yamagishi and T. Hayashi. Parametric Quality-Estimation Model for Adaptive-Bitrate-Streaming Services. *IEEE Trans. on Multimedia*, 19(7):1545–1557, 2017.

[16] R. K. Mok, E. W. Chan, and R. K. Chang. Measuring the Quality of Experience of HTTP Video Streaming. In *IFIP/IEEE Symposium on Integrated Network Management (IM)*, pages 485–492, 2011.

[17] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *ACM SIGCOMM*, pages 325–338, 2015.

[18] G. Cermak, M. Pinson, and S. Wolf. The Relationship among Video Quality, Screen Resolution, and Bit Rate. *IEEE Trans. on Broadcasting*, 57(2):258–262, 2011.

[19] B. Belmudez and S. Moller. An Approach for Modeling the Effects of Video Rresolution and Size on the Perceived Visual Quality. In *IEEE Int'l Symposium on Multimedia*, pages 464–469, 2011.

[20] *Quadratic curve*. http://mathworld.wolfram.com/QuadraticCurve.html.

[21] I. Y. Kim and O. L. De Weck. Adaptive Weighted Sum Method for Multiobjective Optimization: a New Method for Pareto Front Generation. *Structural and Multidisciplinary Optimization*, 31(2):05–116, 2006.

[22] X. Xie, X. Zhang, S. Kumar, and L. E. Li. piStream: Physical Layer Informed Adaptive Video Streaming over LTE. In *ACM MobiCom*, pages 413–425, 2015.

[23] C. Yue, R. Jin, K. Suh Y. Qin, B. Wang, and W. Wei. Linkforecast: Cellular Link Bandwidth Prediction in LTE Networks. *IEEE Trans. on Mobile Computing*, 17(7):1582–94, 2018.

[24] T. Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *ACM SIGCOMM*, pages 187–198, 2014.

[25] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao. Deriving and Validating User Experience Model for DASH Video Streaming. *IEEE Trans. on Broadcasting*, 61(4):651–665, 2015.

[26] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE J. Sel. Areas Commun.*, 32(4):719733, 2014.

[27] H. Mao, R. Netravali, and M. Alizadeh. Neural Adaptive Video Streaming with Pensieve. In *ACM SIGCOMM*, pages 197–210, 2017.

[28] J. Wu, B. Cheng, M. Wang, and J. Chen. Energy-Efficient Bandwidth Aggregation for Delay-Constrained Video over Heterogeneous Wireless Networks. *IEEE J. Sel. Areas Commun.*, 35(1):30–49, 2017.

[29] Y. Yang, Y. Geng, and G. Cao. Energy-Aware Advertising through Quality-Aware Prefetching on Smartphones. In *IEEE MASS*, pages 144–152, 2017.

[30] 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 *ACM MobiSys*, pages 225–238, 2012.

[31] S. He, Y. Liu, and H. Zhou. Optimizing Smartphone Power Consumption through Dynamic Resolution Scaling. In *ACM MobiCom*, pages 27–39, 2015.

[32] B. Varghese, G. Jourjon, K. Thilakarathne, and A. Seneviratne. e-DASH: Modelling An Energy-Aware DASH Player. In *IEEE WoWMoM*, pages 1–9, 2017.