Alleviating Low-Battery Anxiety of Mobile Users via Low-Power Video Streaming

Guoming Tang[1], Kui Wu[2], Deke Guo[3,1], Yi Wang[1,4] and Huan Wang[2]
[1]Peng Cheng Laboratory, Shenzhen, China
[2]Department of Computer Science, University of Victoria, B.C., Canada
[3]Science & Technology on Information Systems Engineering Lab, National University of Defense Technology, China
[4]Southern University of Science and Technology, Shenzhen, China

***Abstract*—The pervasive low-battery anxiety (LBA) among modern mobile users has caused negative impacts on users' emotion and health, and such anxiety may directly lead to loss of customers in power-hungry applications, e.g., video streaming. Despite its importance, LBA has not been thoroughly investigated due to the difficulty in quantitatively measuring LBA. To fill the gap, we present a quantitative model to measure the LBA among mobile users and design a tailored mechanism to alleviate it via display energy saving in video streaming. In specific, we first conduct a large-scale user survey among 2000+ mobile users and strategically extract an empirical LBA model that captures the variation of user's anxiety degree along with the battery power draining. Then, by exploiting the emerging edge computing paradigm, we propose LPVS, a novel solution for low-power video streaming service at the network edge. It aims to minimize the LBA of mobile users, by integrating the extracted LBA model with the energy-saving image/video content transforming techniques. The emulation results using real-world video watching traces demonstrate that, LPVS can effectively alleviate mobile users' LBA and prolong the low-battery users' video watching time (i.e., customer retention) by 39%.**

I. Introduction

Background: Have you ever worried about the battery power before your smartphone dies? Have you ever unwillingly given up watching a video, just because of the low-battery status of your smartphone? If your answer is "yes", most probably you are suffering from the "low-battery anxiety" (LBA), i.e., the fear of losing mobile phone battery power, especially when the battery level is low (say below 20%). Believe it or not, ninety percent of mobile users showed the LBA "symptoms", according to an LG's survey in 2016 [1]. Worse still, based on our survey over 2000+ mobile users in 2019, the ratio of LBA suffering reaches 91.88% (§ III).

Evidences have shown that LBA can bring wide impacts on the behavior of mobile users. According to our survey, over 20% of the mobile audiences will drop video watching when the battery life remains 20% and the dropping rate quickly rises to 50% when only 10% battery energy left. This suggests that, saving mobile phone energy and prolonging its working time can not only release the suffering of LBA but also help customer retention in mobile video streaming services. For this purpose, we need to quantitatively investigate LBA and treat it as an important quality of experience (QoE) metric.

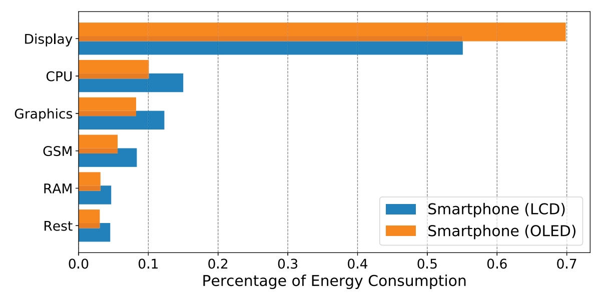


Fig. 1. Energy consumption of different hardware components of a smartphone during video playback (data for LCD smartphone is from [9], and data for OLED smartphone is estimated by comparing the power consumption of OLED and that of LCD [10]).

By looking into the power consumption splitting of the mobile platform during video streaming, we have found that the display module is the primary energy guzzler. As illustrated in Fig. 1, LCD and OLED, the mainstreams for smartphone displays, consume much higher energy than other components during the video playback. Although a great deal of efforts have been devoted to saving energy of the major components of mobile phones, e.g., CPU [2], [3] and communication module [4], [5], the energy consumption of the display has attracted relatively less attention.

Pushed by recent industrial developments, the percentage of energy consumption in displays may become even higher. First, with continuous hardware upgrade, both the size and resolution of mobile displays are increasing [6], making the displays more power hungry. Second, 5G communications and edge computing [7], [8] are around the corner. Together, they promise the mobile users a much improved experience of (HD, 4K or 8K) video watching, with faster speed, more stable connection, and lower latency than those of 4G networks. This in turn will further boom the already popular mobile video streaming services and lead to a much increased power consumption of displays.

Opportunities: It has been found that different multimedia

This work was supported in part by the National Natural Science Foundation of China (No. 61802421, No. U19B2024, No. 61872420), the Hunan Provincial Natural Science Foundation for Excellent Young Scholars (No. 2019JJ30029), the China Postdoctoral Science Foundation (No. 2019M663017), the project of "FANet: PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications" (No. LZC0019), and the Guangdong Key Research and Development Program (No. 2019B121204009). Corresponding authors: Yi Wang (wy@ieee.org), Deke Guo (dekeguo@nudt.edu.cn)

contents, represented by RGB pixel values, have quite different power dissipation on display (§ II-B for more details). This observation directly leads to the design of power-efficient image/video playing schemes using content transforming methods. With video transforming, a great portion of energy can be saved with negligible video quality distortion for human perception [11], [12]. While the image/video transforming can save display energy, it incurs extra CPU/GPU computation cost if performed on the mobile devices [11], [12]. Specifically, the transforming is operated on a per-pixel basis and thus computation intensive, especially for high-resolution display. In consequence, the expected energy saving on mobile devices can be offset or even negated.

The emerging edge computing paradigm is an efficient way to interact with mobile devices at the network edge [13], by providing time-sensitive mobile services with computation, storage and bandwidth resources close to the end users. Compared to remote cloud, edge computing is obviously a better choice for performing video transforming, since (1) video transforming depends on users' device types, e.g., backlight scaling for LCD and color transforming for OLED, and (2) edge server is much closer to mobile users. By leveraging edge computing for video transforming, we can save the display energy of mobile devices without the transforming overhead. We name such a refined service *low-power video streaming* (LPVS), which could be provided as a value-added service by the mobile service provider.

Challenges: There are two major challenges in the LPVS design. First, precisely quantifying the low-battery anxiety of mobile users is difficult and currently has no readily-available references. As the anxiety of mobile users belongs to one of the human feelings or emotions, it is not easy to measure (with some existing metric) nor optimize as we wished. Thus, without referred LBA metrics, *how can we measure and model the low-battery anxiety of mobile users in a quantitative way?* Second, due to limited computation resources at the edge environment, the video streaming provider may not be able to serve all but part of the audiences with video transforming. Taking the Nokia AirFrame open edge server [14] for example and referring to the video transcoding cost [15], one edge server can only process video streaming for about one hundred mobile devices simultaneously. Hence, under the resource constraint, *how can we choose the best subset of users to reduce their LBA?* To be specific, as most probably the LBA is not linearly increased with the draining of battery power, *how can we accommodate the quantified low-battery anxiety and choose the most cost-effective user groups for video transforming?*

Contributions: Addressing the above challenges, this is the first paper that systematically and quantitatively investigates the problem of alleviating LBA of mobile users. The innovation is to use LBA as a critical performance metric to guide video transforming at the edge. This paper includes the following contributions:

- We conduct a large-scale survey over 2000+ mobile users on low-battery anxiety, to model the quantitative relationship between the mobile users' anxiety degree and the mobile devices' battery status. This survey and corresponding empirical conclusions provide strong real-world evidences on the importance of this work.
- By incorporating the extracted quantitative model of LBA, we propose a new solution tailored for low-power video streaming, named LPVS, in which we explicitly depict the scenario and systematically model the energy saving and anxiety reduction by a joint optimization problem. By analyzing the hardness of the optimization problem, we further develop a two-phase heuristic method to solve it using information compacting and Bayesian inference.
- We develop the LPVS emulator and use a real-world *Twitch* dataset to evaluate the performance of LPVS. Extensive emulation results demonstrate that, with LPVS, the overall mobile users can save their devices' energy by up to 37% (thus much reducing the LBA of mobile users) and those low-battery users can prolong their video watching time by 39%.

II. Background and Related Work

A. Background of Low-Battery Anxiety

The anxiety caused by the low battery power of handheld devices has been noticed for a long time. It was initially investigated under the cover of *nomophobia*, which refers to the fear of being without a smartphone. In 2016, the low-battery anxiety was formally proposed mainly owning to an LG's survey among $2,000$ smartphone users in the US. The survey found that ninety percent of the mobile users would "feel panic" when their phone battery drops to 20 percent or lower [1]. Since then, the low-battery anxiety, i.e., LBA, has been widely known to the public.

LBA can bring widely and deeply negative impacts on mobile users' lives. According to the LG's survey, one in three people are likely to skip the gym, when it comes to choosing between hitting the gym and charging their smartphones. Furthermore, for those severely suffering from the LBA, it can cause strange behaviors, e.g., head home immediately, ask chargers from strangers, or secretly "borrow" other's charger. To be even worse, LBA is becoming the major trigger of nomophobia, which is commonly treated as one type of mental health problems.

Although LBA has shown negative effects upon mobile users' emotion, behavior and even health, it has not been investigated thoroughly. To be specific, there is no prior work measuring and quantifying LBA in a quantitative way, which is one of the major tasks in this work.

B. Background of Display Power Saving

Modern smartphones are typically equipped with one of the two display types: Liquid-Crystal Displays (LCD) or Organic Light-Emitting Diode (OLED) displays. The two types of displays work differently and have quite different power consumption characteristics.

TABLE I
REVIEW OF THE STATE-OF-THE-ART POWER SAVING STRATEGIES FOR LCD AND OLED, RESPECTIVELY.

Type	*Applied Strategy*	*Power Saving*
LCD	quality adapted backlight scaling [18]	27%-42%
	dynamic backlight scaling [19]	15%-49%
	dynamic backlight luminance scaling [20]	20%-80%
	brightness & contrast scaling [21]	$\leq 50\%$
	luminance dimming & compensation [22]	20%-38%
OLED	color and shape transforming [17]	25%-66%
	color transforming and darkening [23]	$\leq 60\%$
	color transforming with constraints [12]	$\leq 64\%$
	pixel disabling & resolution scaling [24]	$\leq 26\%$
	image pixel scaling [25]	38%-42%
	redundant subpixel shutoff [6]	$\leq 21\%$
	Average	13%-49%

1) Power Saving for LCD: The major power consumer of LCD is its backlight, which illuminates the liquid crystals of the display with various brightness levels. Its power consumption can be quite different at different brightness levels. Thus, by strategically scaling the backlight and regulating the image luminance, the original image displayed by a LCD can be rendered with much less energy and a distortion negligible/tolerable for human perception [11]. Accordingly, a broad spectrum of schemes based on backlight scaling (with luminance compensation) have been proposed to cut down the power consumption of LCD.

2) Power Saving for OLED: Compared to LCD, the OLED display is not only thinner and lighter, but also can support up to three orders of magnitude higher refresh rates [16]. For the OLED display, different RGB color sub-pixels generate lights with different energy efficiencies: the blue pixels consume about twice the power of green ones, while the red in between of the two [17]. Thus, the power consumption much relies on the displayed colors (rather than the brightness). On the other hand, the human visual system (HVS) has great perceptual flexibility, thus can tolerate small color changes [17]. Accordingly, various color transforming schemes have been developed for saving energy of OLED displays.

We summarize existing strategies for display power saving in Table I, for the LCD and OLED displays, respectively. Nevertheless, these strategies are "pixel-wise", i.e., they operate on a per-pixel basis, which incurs a non-negligible overhead for the mobile devices, especially those with high-resolution displays [11], [12].

C. Work Related to This Paper

Since our work aims at offloading the computation intensive image/video transforming from smartphones to edge servers to keep users from worrying about quick battery drainage, we only introduce the relevant work on proxy-based display energy saving for video streaming.

In [18], by analyzing the characteristics of video streaming services, the authors proposed a quality adapted backlight scaling scheme for LCD energy saving during video playback of handheld devices. In their prototype, a proxy server was setup for video transforming and over 40% of the energy saving was achieved. A similar idea was applied in the work of [26], where the adaptation of video transforming was shifted from the low-power device to a proxy middleware. In both work, the proxy server is a testing platform designed for one dedicated client device, which is different from our scenario where the video streaming service is provided for a group of users. Above all, neither of them takes the users' low-battery anxiety into consideration, which is another major difference from our work.

III. LBA SURVEY AND MODELLING

A. Data Collection

To learn the impact of low-battery anxiety and establish a quantitative model, we carefully designed an online survey (refer to [27] for the detailed questionnaire) and continuously collected the answers from mobile users for over three months. At the end, we collected $2,032$ effective answers after data cleansing. Refer to Table II in the Appendix for detailed information regarding the participants.

Based on the survey data, it is surprising that 91.88% ($1,867$ out of $2,032$) of the participants are suffering from the low-battery anxiety, more or less. This is consistent with the LG survey [1], but the percentage is even higher in our survey. It is also interesting to see that nearly half of the mobile users will give up watching an attractive video, once the battery level of mobile phone drops below 10%. These findings provide direct and strong support to the necessity of our work.

B. LBA Curve Extraction

In our elaborately designed questionnaire, one question the participants need to answer is: *At what battery level (in percentage from* 0 *to* 100%*) will you charge your mobile phone, when it is possible?* The answers provide us with an angle to infer at which energy level a user normally begins to worry about the battery life, i.e., experience the low-battery anxiety. Then, with all the collected answers, we are able to extract the LBA curve model: the anxiety degree (caused by the draining of battery power) vs. the battery energy status.

Specifically, a four-step procedure is conducted to obtain the LBA curve from the raw data: (1) initialize 100 empty bins, indicating the battery level from (almost) empty to full (i.e., $[1, 100]$); (2) for each answer, e.g., a (an integer in $[1, 100]$), add one to each of the bins in $[1, a]$; (3) conduct (2) for all the answers and obtain a declined discrete curve in the region of $[1, 100]$; (4) normalize the 100 cumulative numbers to the region of $[0, 1]$, denoting the anxiety degree, we obtain the LBA curve: anxiety degree vs. battery level. A similar approach was also adopted in [28], to identify the commonly used response time thresholds for service level objectives in cloud service.

Following the above extraction process, the resulting LBA curve from the surveyed $2,032$ users is shown in Fig. 2. From the illustrated LBA curve, we can observe that:

- The anxiety degree does not linearly increase with the decrease of energy level. If we use the linear function

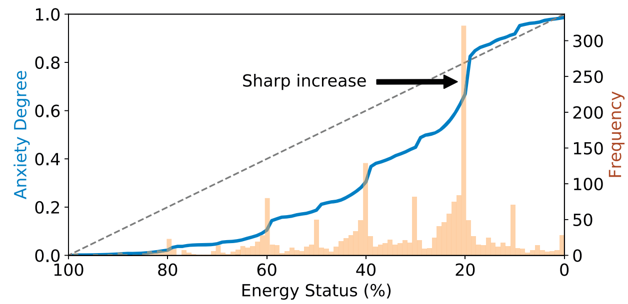


Fig. 2. Extracted anxiety curve from the survey data of $2,032$ mobile users.

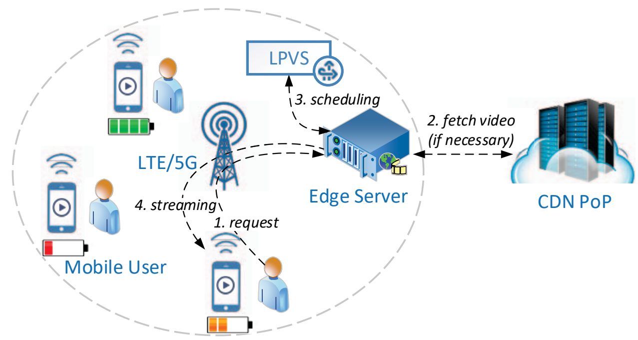


Fig. 3. The procedure of LPVS within a virtual cluster at the edge.

(the straight dashed line in Fig. 2) as a comparison, user's anxiety is a convex function of the energy level when the energy level is in $[20\%, 100\%]$, but is concave when the energy level drops to $[0, 20\%]$.
- A sharp increase of anxiety can be observed when the energy level drops to 20%. This is most probably caused by the color change of battery icon (e.g., the icon's face color changes to yellow or red) and the low-battery warning message.

C. Insights on LBA Alleviation

The non-linearity of the LBA curve indicates that, the user's sensitivity to the power draining (at different battery levels) is heterogeneous. This also implies that, when choosing a subset of mobile users for anxiety minimizing, following a random user selection strategy cannot be optimal, as those are currently not sensitive to the energy status may be selected (thus resulting in less performance gain). Instead, the mobile users that are sensitive to the battery power draining, e.g., those near the "sharp increase" area in Fig. 2, should be given a higher priority. This is also how our optimization scheme will take effect (§ IV-E).

Note that the extracted LBA curve was obtained with survey questions, with the assumption that participants' answers truthfully reflect their feelings and behaviors. This assumption may be challenged, and an alternative method to avoid this pitfall is to look into users' real behaviors [29], [30], which we leave as our future work.

IV. LPVS: Low-Power Video Streaming

A. Scenario Overview

As illustrated by Fig. 3, we assume the 5G mobile edge computing (MEC) [7] platform consisting of 5G base stations, edge servers, and CDN servers, where the 5G base stations and edge servers are deployed at locations close to end users and the CDN servers are located at the CDN Point of Presence (PoP). There is a content delivery strategy between the edge servers and the CDN servers [31], [32], which may prefetch a certain amount of video content from the CDN servers to the edge server, based on the (historical) video requests from the mobile users. This content prefetch strategy provides underlying support for and is independent of LPVS.

We assume that all the mobile devices within one region (e.g., the covered area of a base station) form one *virtual cluster* (VC), where they share the same edge server. Without loss of generality, assume that one VC hosts N mobile devices. We divide the time into slots of equal length (e.g., 5-minute interval in our implementation as per Remark 1). LPVS needs to make a (new) scheduling decision for video transforming at the beginning of each time slot (scheduling points in Fig. 4).

Usually, a complete video is split into a number of small chunks, but depending on different caching strategies [32], the edge server might not have the whole video chunks and the number of available video chunks may vary. For a video m, assume that the currently available number of video chunks, at the scheduling point, is K_m. With the above notations, we denote the video chunks played on device n in time slot t by $d_n(t), 1 \leq n \leq N, t \geq 1$, with:

$$d_n(t) := \langle VID, CID_1, \ldots, CID_{K_m} \rangle \tag{1}$$

in which VID stands for the video ID and $CID_i, i = 1, \ldots, K_m$, for the IDs of video chunks available for device n at the beginning of time slot t.

Overall, at the scheduling point, LPVS should make decisions on whether or not the edge server should perform video transforming for certain users in the VC, to save display power and alleviate their low-battery anxiety.

Remark 1. *We ignore the scenario where a user switches videos during one time slot. This omission is due to the periodical scheduling used in LPVS. The interval time should not be too small (e.g., in seconds) to avoid unnecessary computational overhead at the edge server. The empirical value of* 5 *minutes is used based on the facts that (1) battery level should not drop too much during this time and (2) people can tolerate a certain level of anxiety when battery level does not drop too much. If a user switches videos during one time slot, LPVS will keep the same decision (i.e., with or without video transforming) for this user until next scheduling point.*

B. Models for Power Consumption in Video Streaming

When a device plays a video, its *power rate*[1] may fluctuate up and down along with the played chunks, due to different brightness levels (for the LCD) or different color distributions

[1]The power rate is defined as the energy consumption of the mobile device in the time of playing one video chunk.

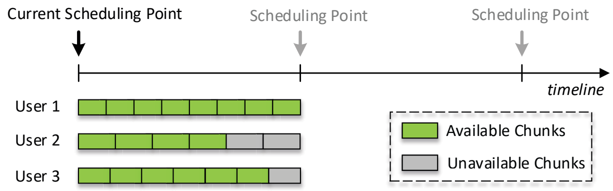


Fig. 4. Illustration of power rate estimating with the available video chunks for three users' mobile devices.

(for the OLED). When playing video m, we denote the power rate on the n-th device for the κ-th chunk by $p_{n,m}(\kappa), 1 \leq n \leq N, 1 \leq m \leq M, 1 \leq \kappa \leq K_m$. Note that, given the display's specification (e.g., type, size, and resolution) and the available video chunk κ, corresponding power rate $p_{n,m}(\kappa)$ can be estimated with existing power models for LCD [20] or OLED [17].

Also notice that, within one time slot, the requested video chunks may be not all available (due to different prefetching and buffering strategies), as illustrated for user 2 and user 3 in Fig. 4. Under such a situation, we only use the available video chunks for the estimation of $p_{n,m}(\kappa)$.

By applying the video transforming (given in § II) on the video $d_n(t)$, the *power reduction ratio*[2] of a specific device n during the time slot t can be represented as $\gamma_n(d_n(t))$ (γ_n for simplicity), where $0 < \gamma_n < 1$.

Remark 2. ***About*** γ_n*: The parameter represents an average power saving ratio on device n achieved over a bunch of video chunks during one time slot. Thus, it is not a fixed value and can only be learnt after playing the transformed video. We treat it as a random variable and update its value with the Bayesian inference, in § V-D. In the following part of modeling, we assume that we have already learnt the knowledge of γ_n.*

At the beginning of an arbitrary time slot t, due to limited edge server capacity, LPVS may only choose a subset of requested videos for transforming. We denote the decision variable of whether to perform video transforming for the n-th device during the entire time slot by x_n, with $x_n = 1$ indicating yes and $x_n = 0$ otherwise. Thus, for the time slot t, we have:

$$x_n \in \{0, 1\}, \forall n. \tag{2}$$

By determining the value of x_n and referring to the knowledge of $p_{n,m}(\kappa)$ and γ_n, we are able to infer the power rate of device n when playing the κ-th chunk of video m by:

$$\psi_{n,m}(\kappa) = x_n \cdot \gamma_n \cdot p_{n,m}(\kappa) + (1 - x_n) \cdot p_{n,m}(\kappa), \forall n, \forall \kappa. \tag{3}$$

C. Models for Energy Status and Low-battery Anxiety

Another important information in LPVS is the energy status of mobile devices, measured by the remaining energy of battery. We denote the energy status of the n-th device at the beginning of an arbitrary time slot by $e_{n,m}(\kappa)$, where m and κ are the video ID and the video chunk ID, respectively, requested by the n-th device.

[2]The power reduction rate is defined by the ratio of a device's power consumption with and without video transforming during one time slot.

Before the edge server determines to transform a video chunk requested by device n, the device should have sufficient energy to power the device (otherwise the transforming has no meaning). Hence, the following inequality should hold:

$$e_{n,m}(\kappa) \geq x_n \cdot \gamma_n \cdot p_{n,m}(\kappa) \cdot \Delta_\kappa, \forall n, \forall \kappa, \tag{4}$$

where Δ_κ denotes the time length of current video chunk. Thus, when $x_n = 0$, i.e., the video chunks requested by device n are not transformed, the above inequality takes no effect (always holds), as $e_{n,m}(\kappa)$ is non-negative.

Meanwhile, with the power rate of device n when playing the κ-th chunk of video m (given by $\psi_{n,m}(\kappa)$ in (3)), the energy status of the device (before playing the next video chunk $\kappa + 1$) can be predicted by:

$$e_{n,m}(\kappa+1) = e_{n,m}(\kappa) - \psi_{n,m}(\kappa) \cdot \Delta_\kappa, \forall n, 1 \leq \kappa \leq K_m - 1. \tag{5}$$

As we have mentioned, over 90% of the mobile users are suffering from the low-battery anxiety [1] (the figure is 91.88% in our survey). Usually, the less the battery energy, the higher the anxiety. Such relationship between the battery energy status and the low-battery anxiety have been captured by the anxiety curve shown in Fig. 2.

We use $\phi(\cdot)$ to denote the empirical function reflecting the *anxiety degree* of a user given the energy status of his device. To be specific, for device n with the energy status of $e_{n,m}(\kappa)$, the anxiety degree of its owner can be estimated by $\phi(e_{n,m}(\kappa))$. As illustrated by Fig. 2, the anxiety degree of a user is between 0 and 1, with energy status of her mobile device between 100% and 0, correspondingly.

As have been reported, low-battery anxiety can potentially have negative effect on user's video watching behavior (§ III-A). Thus, we believe that reducing user's low-battery anxiety is a critical aspect in improving the QoE of video streaming service, which is one key feature of our LPVS.

D. Video Streaming Capacity at the Edge

For the video chunks requested by device n and to be transformed at the edge during time slot t (i.e., $d_n(t)$), the computing resource that the video transforming needs is estimated by $g(d_n(t))$, where $g(\cdot)$ is the function reflecting the server's computing resource needed for video transforming. Similarly, the storage resource that the video transforming consumes can be estimated by $h(d_n(t))$, where $h(\cdot)$ is a function for storage space measuring during video transforming.

For the edge server attached to a VC, it usually has quite limited computing and storage resources (e.g., the *NOKIA* or *Inspur* edge server [14], [33]). We denote the *extra* computing and storage resources available at the edge server to perform video transforming by C and S, respectively. Then we have the following two capacity constraints:

$$\sum_{n=1}^{N} x_n \cdot g(d_n(t)) \leq C, \tag{6}$$

$$\sum_{n=1}^{N} x_n \cdot h(d_n(t)) \leq S. \tag{7}$$

E. Joint Optimization for Energy Saving & Anxiety Reduction

Given the edge server capacity constraints, for each time slot, our targets are i) to minimize the display energy consumption of all the mobile devices during video playback, and ii) to minimize the anxiety degree of all users simultaneously. The problem can be formulated as:

$$\min_{\mathbf{x}} \quad \sum_{n=1}^{N}\sum_{\kappa=1}^{K_m}\left(\psi_{n,m}(\kappa) + \lambda \cdot \phi(e_{n,m}(\kappa))\right) \tag{8a}$$

$$\text{s.t.} \quad (2) \sim (7), \tag{8b}$$

where λ is a *regularization parameter* that balances the two targets, and K_m is a constant representing the total number of chunks processed by the video streaming server in one time slot (for an arbitrary video m).

Remark 3. ***About*** λ: *In practice of LPVS, the regularization parameter λ should be determined or regulated by the video streaming service providers, based on their own policies and specific service-level agreements (SLAs) with the customers. We will show how this parameter can affect the results, i.e., the energy saving and anxiety reduction, in § VII.*

V. Solution Methodology

A. The Difficulties

It is nontrivial to solve the joint optimization problem given by (8), mainly due to three difficulties:

Difficulty-1: The device energy status ($e_{n,m}(\kappa)$) should be predicted after playing each video chunk during the whole time slot, for constraints (3), (4) and (5) and the objective function. For instance, with the effect of κ, the energy status update with (5) needs to be performed chunk by chunk. The constraints and the objective function hence twist together, making the optimization problem hard to solve.

Difficulty-2: The problem generally belongs to the integer programming (as $x_n(t)$ is binary), while whether it is linear or nonlinear solely depends on the function of $\phi(\cdot)$ (i.e., the anxiety curve) in the objective (8a). Referring to our extracted anxiety curve in Fig. 2, the function is obviously not linear and thus the problem we are facing belongs to the nonlinear integer programming, which is normally intractable.

Difficulty-3: Different devices may have different power reduction ratio γ_n, whose value is unknown in advance. As we have mentioned, γ_n is not a fixed value and may vary over different transformed videos. Hence, it causes a *circular argument*: to solve the problem, we need the value of γ_n as one of the inputs; on the other hand, we may have no information about the value of γ_n before the problem is solved.

We tackle the above three difficulties with i) information compacting, ii) a two-phase heuristic, and iii) Bayesian inference, respectively.

B. Information Compacting

We find that both the constraints and the objective can be compacted in a way that i) κ is marginalized and ii) the intermediate energy status ($e_{n,m}(\kappa)$) is eliminated. After information compacting, we can transform the problem into a neater form that renders an easy solution.

1) Information Compacting for the Constraints

First, we perform information compacting on the constraints of (4) and (5), as only these two constraints relate to $e_{n,m}(\kappa)$. We first summarize over κ for the inequality of constraint (4):

$$\sum_{\kappa=1}^{K_m} e_{n,m}(\kappa) \geq x_n \gamma_n \sum_{\kappa=1}^{K_m} p_{n,m}(\kappa) \tag{9}$$

Using (5), we rewrite the left-hand side of (9) as:

$$\sum_{\kappa=1}^{K_m} e_{n,m}(\kappa) \tag{10a}$$

$$= e_{n,m}(1) + e_{n,m}(2) + \cdots + e_{n,m}(K_m) \tag{10b}$$

$$= e_{n,m}(1) + (e_{n,m}(1) - \psi_{n,m}(1)) + + (e_{n,m}(1) - \psi_{n,m}(1) - \psi_{n,m}(2)) + \ldots \tag{10c}$$

$$= K_m \cdot e_{n,m}(1) - \sum_{\kappa=1}^{K_m}(K_m - \kappa)\psi_{n,m}(\kappa) \tag{10d}$$

Replacing the left-hand side of (9) with (10d), we have:

$$K_m \cdot e_{n,m}(1) - \sum_{\kappa=1}^{K_m}(K_m - \kappa)\psi_{n,m}(\kappa) \geq x_n\gamma_n \sum_{\kappa=1}^{K_m} p_{n,m}(\kappa). \tag{11}$$

In constraint (11), for given values of the decision variables $x_n, 1 \leq n \leq N$, all the other parameters are easy to calculate:

- K_m: the total number of video chunks delivered within one time slot, which is a known constant for each individual video;
- $e_{n,m}(1)$: the initial energy status of each device at the beginning of each time slot, which is reported at the scheduling point by each device;
- $p_{n,m}(\kappa)$: the power rate of the device playing each chunk (without transforming), which is known beforehand with existing power models;
- $\psi_{n,m}(\kappa)$: the power rate of the device playing each chunk under the given value of decision variable x_n, which can be computed with the knowledge of $p_{n,m}(\kappa)$ (refer to the definition in (3));
- γ_n: the power reduction ratio of each device after video transforming, which can be estimated and updated via a Bayesian approach (refer to the details in § V-D).

2) Information Compacting in Objective

Next, we perform information compacting for the objective function to avoid the computation of intermediate energy status $e_{n,m}(\kappa)$. With (5), we can derive the relationship between a device's predicted energy status (after playing a video chunk) and its initial energy status:

$$
\begin{aligned}
&e_{n,m}(\kappa) && \text{(12a)}\\
&= e_{n,m}(\kappa-1) - \psi_{n,m}(\kappa-1) && \text{(12b)}\\
&= e_{n,m}(\kappa-2) - \psi_{n,m}(\kappa-2) - \psi_{n,m}(\kappa-1) && \text{(12c)}\\
&\vdots\\
&= e_{n,m}(1) - \sum_{i=1}^{\kappa-1} \psi_{n,m}(i) && \text{(12d)}
\end{aligned}
$$

Thus, the objective function (8a) can be rewritten as:

$$
\sum_{n=1}^{N}\sum_{\kappa=1}^{K_m}\left(\psi_{n,m}(\kappa) + \lambda \cdot \phi(e_{n,m}(1) - \sum_{i=1}^{\kappa-1}\psi_{n,m}(i))\right) \quad (13)
$$

in which all the elements are either readily available or can be easily computed with the given decision variable of x_n.

We then apply the compacted constraint (11) to replace original constraints (4) and (5), and the transformed objective (13) to replace the original objective (8a). It is worth mentioning that the compacted form of the problem is equivalent to the original problem (8), as information compacting comes from (9) and (12), which only involve equalities.

C. A Two-phase Heuristic for Joint Optimization

To solve the nonlinear integer problem, we develop a two-phase heuristic method:

- **Phase-1**: We solve the following optimization problem:

$$
\begin{aligned}
\min_{x} \quad & \sum_{n=1}^{N}\sum_{\kappa=1}^{K_m}\psi_{n,m}(\kappa) && \text{(14a)}\\
\text{s.t.} \quad & (2), (6), (7), (11). && \text{(14b)}
\end{aligned}
$$

  Without considering the nonlinear function $\phi(\cdot)$, the above problem belongs to linear integer programming (ILP). Furthermore, with the compacted form, it can be directly fed into the off-the-shelf ILP solvers, such as CPLEX [34], Gurobi [35] or CVX [36]. By solving the problem, we actually obtain a subset (say, with number of N') of the mobile devices for video transforming, with which the energy consumption is minimized.
- **Phase-2**: To further cope with users' anxiety, we sort the mobile devices by their energy status, with which the anxiety degrees of mobile users are ranked. Then, we try to swap the selected N' devices in Phase-1 with the first $(N - N')$ devices whose owners have the largest anxiety degrees. The swapping is successful only when the objective value computed with (13) is reduced. The nonlinear function $\phi(\cdot)$ could use any theoretical anxiety-energy model (none to the best of our knowledge), or an empirical function such as that introduced in § III.

Note that the computational complexities of both Phase-1 and Phase-2 are much lower than solving the original nonlinear programming. For a VC including 1000 devices, our implementation on a low-end machine can find the optimal solution in Phase-1 in 5 seconds using the off-the-shelf ILP solver, and can finish the swapping process within 1 minute. Considering the scheduling time length is 5 minutes in our implementation, the computing overhead of LPVS is acceptable under "one-slot-ahead" scheduling working mode.

D. Determine γ_n with Bayesian Inference

When playing a transformed video, mobile devices of different specifications (LCD or OLED, size, etc.) may have different power reduction ratio γ_n. Nevertheless, before a transformed video is played on device n, the value of γ_n is unknown. In other words, at a scheduling point, we actually do not know the value of γ_n for the following time slot (since the video chunks have not been played yet).

Although we cannot know the value of γ_n for the current time slot t ($t > 1$), fortunately we do have the information of the previous time slot $t-1$ and can get the value of γ_n at the end of the previous time slot. This inspires us to update the value of γ_n with the obtained information (observation) from previous time slots. Such an idea can be naturally implemented with Bayesian inference, by treating γ_n as a random variable.

1) At the Beginning of the 1^{st} Time Slot

At the beginning of the first time slot, we initialize the probability distribution function (PDF) of power reduction ratio with a Gaussian distribution:

$$
p(\gamma_n) = \mathcal{N}(\mu,\sigma^2) \quad (15)
$$

where μ and σ^2 represent the mean and variance of the Gaussian distribution, respectively. μ can be initialized by:

$$
\mu = \frac{\gamma_L + \gamma_U}{2}, \quad (16)
$$

with γ_L and γ_U representing the lower and upper bounds of power reduction ratio, respectively (refer to Table I). As to the initialization of σ^2, we can choose a relatively big value due to the lack of confidence about the concentration of γ_n, e.g., $\sigma^2 = 12$ in our implementation.

2) At the End of Time Slot t

With the observation of power reduction when playing transformed video in time slot t, denoted by Δ_n, we update the PDF of γ_n for the next time slot by computing the posterior using the Bayesian rule:

$$
p(\gamma_n|\Delta_n) = \frac{P(\Delta_n|\gamma_n)p(\gamma_n)}{P(\Delta_n)}, \quad (17)
$$

where $p(\gamma_n)$ is the prior of γ_n used in time slot t; $P(\Delta_n|\gamma_n)$ is the likelihood of the observation under γ_n; $P(\Delta_n)$ is the marginal distribution of Δ_n over γ_n, i.e.,

$$
P(\Delta_n) = \int_{\gamma_L}^{\gamma_U} P(\Delta_n|\gamma_n)p(\gamma_n)d\gamma_n. \quad (18)
$$

3) At the Beginning of Time Slot $t+1$

With the above posterior of γ_n, we compute the expected value (or the expectation) of γ_n:

$$
\mathbf{E}_{p(\gamma_n|\Delta_n)} = \int_{\gamma_L}^{\gamma_U} \gamma_n p(\gamma_n|\Delta_n)d\gamma_n, \quad (19)
$$

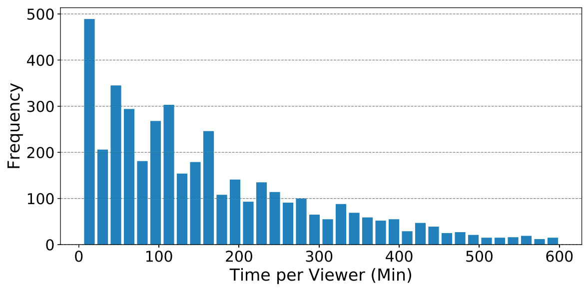


Fig. 5. The histogram of video session durations in our dataset.

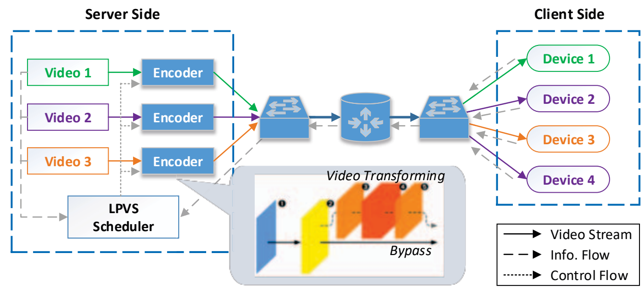


Fig. 6. The framework of LPVS emulator and its major building blocks.

and apply the obtained value for video transforming scheduling (i.e., $\gamma_n = \mathbf{E}_{p(\gamma_n|\Delta_n)}$) for time slot $t+1$.

Note that, as γ_n is assumed to follow the Gaussian distribution, the likelihood-prior pair in (17) is conjugate[3]. In this way, the update of γ_n can be computed precisely without any approximation.

VI. Implementations

A. Real-world Video Streaming Traces

We target at the *live video streaming service*, as it becomes extremely popular in recent years. It is reported by *Twitch*, a popular live video streaming platform, that in 2017 only, 355 billion minutes of live streams were watched and more than 2 million streamers had broadcast channels on the platform [37]. We then use a dataset from *Twitch* as input requests for our evaluations.

The dataset consists of traces from thousands of live streaming channels in 2014, with the sampling interval of 5 minutes. It includes the detailed information, such as the number of viewers in each channel, bitrates of each channel, and the duration of live channels. We filter the data and only keep the live channels that last for no more than 10 hours, which results in $1,566$ live channels and $4,761$ live video sessions. The histogram of video session durations is given in Fig. 5.

B. LPVS Emulation and Setups

To emulate the whole process of LPVS and validate its effectiveness, we develop an emulator with building blocks shown in Fig. 6. The major procedures include: information gathering, request scheduling and video transforming.

1) Information Gathering

Rationale: At each scheduling point of LPVS, along with their video (chunks) requests, the users (mobile devices) report the displays' information (e.g., size and resolution) as well as the energy status to the LPVS scheduler. Meanwhile, the video information (e.g., whether cached and currently available) is also collected from either the CDN PoP or edge streaming server. In addition, as introduced in § IV-B, the power rates of requested videos (chunks) are also estimated at the server side, aided by existing power modeling and profiling techniques for LCD in [20] and OLED in [17], respectively.

[3] A likelihood-prior pair is said to be conjugate if they result in a posterior that is of the same form as the prior.

Setups: The scheduling period (i.e., length of time slot) of LPVS is set to 5 minutes, which is consistent with the sampling interval of the *Twitch* dataset. A group of viewers in each channel of *Twitch* are selected and form a virtual cluster (VC) in our context. Some of the needed information to drive LPVS can be obtained from the dataset, e.g., the number of live channels/videos, the number of chunks in each time slot, and the resolutions of requested videos. For other information that cannot be learnt from the dataset, e.g., the power rate of live videos and the specifications of displays, we assign values for each of them by randomly choosing from available display resolutions under the supported bitrates. As to the energy status of the mobile devices, it is also not contained in the dataset, so we randomly assign values following a Gaussian distribution at the beginning of emulation.

2) Request Scheduling

Rationale: At the server side, with the gathered information of user requests ($d_n(t)$ defined by (1)), display specifications (inputs of the resource consumption functions $g(\cdot)$ and $h(\cdot)$ in § IV-D), energy status ($e_{n,m}(1)$), and video meta data (e.g., bitrates, power rates $p_{n,m}(\kappa)$), the LPVS scheduler performs the video request scheduling task. Specifically, by following the solution methodology introduced in § V, the scheduler is able to select the optimal subset of the requested videos for transforming, under the constraints of edge server capacity. This returns a scheduling strategy that results in the maximum display energy saving and user low-battery anxiety reduction simultaneously.

Setups: The LPVS scheduler works under the "one-slot-ahead" scheduling mode, i.e., during current slot the LPVS makes decision for the incoming requests in the next time slot. Then, at the scheduling point of next time slot, the obtained decision will be executed, with which the selected videos are sent to the encoders. Furthermore, we experiment on different values of λ to validate the effectiveness of LPVS w.r.t. balancing the power saving and anxiety reduction.

3) Video Transforming

Rationale: As shown in Fig. 6, all the requested videos will go through the encoder component at the server side. Within the video encoder, the selected videos (chunks) by the scheduler will be transformed using the techniques introduced in § II-B, and meanwhile the un-selected videos will bypass the transforming function. After the video transforming, the power saving ratio γ_n can be updated following our strategy

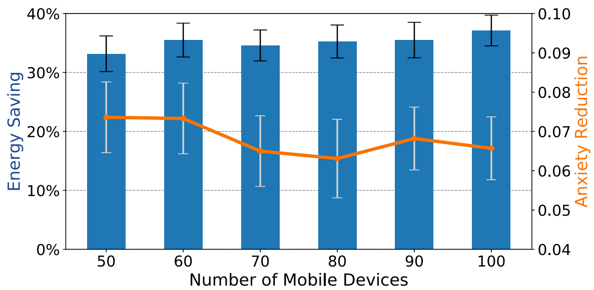


Fig. 7. Energy saving and anxiety reduction under sufficient edge resource.

in § V-D.

Setups: The amount of videos that can be transformed simultaneously depends on the capacity of edge streaming server. Referring to one commercial edge server model from Nokia (AirFrame Open Edge Server [14]) and resource consumption measurements of video transcoding [15], we estimate that one edge server can process video streaming (including transforming) for up to 100 mobile devices simultaneously. Moreover, at the beginning of the first time slot, we set the prior distribution of γ_n by a Gaussian distribution with $\mu = (0.13 + 0.49)/2 = 0.31$ (refer to the average upper/lower bounds given in Table I) and $\sigma = 12$.

VII. Performance Evaluations

In this section, we evaluate the performance of LPVS, under sufficient and insufficient edge capacity, respectively. Furthermore, we also investigate the impact of LPVS on the time per viewer, i.e., the time of individual viewers spending on watching videos. At last, the overhead of LPVS is evaluated w.r.t. the running time for the optimal scheduling.

A. LPVS with Sufficient Edge Resource

We first analyze the performance of LPVS under sufficient edge resource condition. As mentioned in our implementations, we choose the edge server with capacity supporting up to 100 mobile users' video transforming simultaneously. Thus, we look into the performance of LPVS for those VCs with no more than 100 mobile users (with group size ranging from 50 to 100 specifically).

1) Energy Saving of Mobile Devices: The results on energy saving can be found in Fig. 7, where the bar chart (in blue color) shows the percentage of energy saving after applying LPVS. The average energy saving ratio of different testing groups is 35.20%, with the maximum energy saving ratio of 37.13%. Thus, it can be concluded that LPVS can save a large portion of energy of users' mobile devices.

2) Anxiety Reduction of Mobile Users: The results on users' anxiety alleviation can also be found in Fig. 7, with the orange line chart at the right Y-axis. Specifically, the percentages of user anxiety reduction after applying LPVS are given, under different user group sizes. The average anxiety reduction ratio of different testing groups is 6.82%, with the maximum anxiety reduction ratio of 7.36%. We can see that, with LPVS, the mobile users' anxiety can be alleviated, while the effect is not significant as that of energy saving. This is mainly caused by our experimental setup that the energy status follows a Gaussian distribution, which leads to the majority with an energy status around 50% where the anxiety curve is relatively flat (refer to Fig. 2). Nevertheless, as will be shown in § VII-C, the impact of LPVS on the low-battery users is significant.

B. LPVS with Limited Edge Resource

We then look into the cases where the edge capacity is not enough, i.e., the computing and storage resources are insufficient to provide LPVS to all users but a selected subset. Specifically, we investigate the performance of LPVS in those VCs with user group size ranging from 100 to 500. In addition, since only a subset of users can be served with LPVS, the regularization parameter λ takes effect in balancing the energy saving and anxiety reduction.

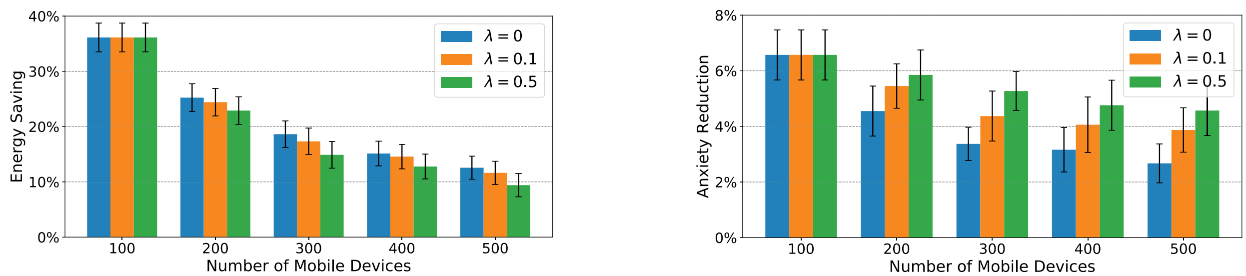
1) Energy Saving of Mobile Devices: The energy saving ratios of multiple user groups are shown in Fig. 8(a), under different settings of λ. We can see that, the energy saving ratio decreases with the increasing number of mobile users, as the portion of users that can take advantage of LPVS becomes smaller. In addition, with the increase of λ, the weight for energy saving in the objective (of problem (8)) becomes smaller, thus leading to the decrease of energy saving ratio in each VC group.

2) Anxiety Reduction of Mobile Users: The anxiety reduction ratios of mobile users under different VC groups are illustrated in Fig. 8(b). Based on the results, the anxiety reduction ratio decreases with the increase of user group size, which is caused by the insufficient edge capacity. Furthermore, with the increase of λ, the weight for anxiety reduction in the objective function becomes larger, thus resulting in increased anxiety reduction for each VC group. It is also worth mentioning that we only illustrate the effectiveness of λ in balancing the energy saving and anxiety reduction. How to set the value of λ in practice is determined by the LPVS provider (based on specific policies and SLAs), which is beyond the scope of this work.

C. Impact of LPVS on Low-battery Users

We have observed that the overall percentage in anxiety reduction with LPVS is not that obvious as that in energy saving, when the majority of users have relatively sufficient battery life. Nevertheless, when we shift our focus to the users who have low-battery status, the impact of LPVS is significant. To test the impact, we calculate the metric of time per viewer (TPV) for low-battery users, i.e., the time duration of individual users on video watching, under sufficient edge capacity condition. The TPV metric is inferred from our online survey question: *At what battery level (in percentage from* 1% *to* 100%*) will you give up watching a video you are interested in on your mobile phone?*

For each VC group we have tested, we collect data on those users who were i) selected for video transforming (served by



(a) Energy saving with LPVS under different values of λ.

(b) Anxiety reduction with LPVS under different values of λ.

Fig. 8. Energy saving and anxiety reduction under the condition of limited edge server capacity.

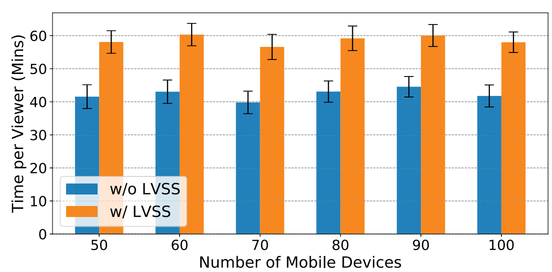


Fig. 9. Time per viewer increases with and without LPVS.

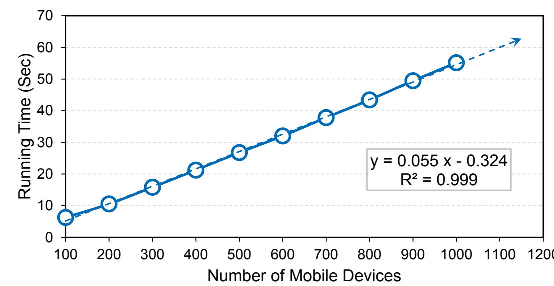


Fig. 10. Running times of LPVS scheduler with increase of VC group size; fitted by the linear regression function $y = 0.055x - 0.324$ with the R-squared value $R^2 = 0.999$.

LPVS) and ii) with energy status in $(0, 40\%]$ (so-called low-battery users) at the starting of LPVS. Then, we calculate the value of TPV for each low-battery user. For comparison, the TPV values without applying LPVS are also computed for these low-battery users.

The results are illustrated in Fig. 9 for the cases with and without applying LPVS, respectively. From the figure we can find that, without LPVS, the average value of TPV is 42.3 minutes, while with LPVS, the average TPV value increases to 58.7 minutes. This means that LPVS brings in an *extra* TPV of 16.4 minutes, which corresponds to 38.8% of the watching time duration of the low-battery users.

D. Overhead of LPVS and Impact on Other QoE Metrics

We treat LPVS as a value-added service upon the conventional video streaming and focus on evaluating the overhead of LPVS w.r.t. its running time. This is necessary when we adopt the "one-slot-ahead" working mode. If the scheduling cannot be finished in one time slot, it will affect the conventional video streaming service and may degrade other QoE metrics (e.g., increasing the video freezing time and frequency).

The average running time of LPVS resulting from our emulation is illustrated in Fig. 10, when performing optimal scheduling under different VC group sizes. We can observe that: i) with the increase of user group size (i.e., number of mobile devices), the running time of LPVS increases accordingly; ii) the increasing trend is approximately linear, indicating the low time complexity of our heuristic method (given in § V-C). Under such a linear increase trend, the maximum number of mobile devices that our LPVS scheduler can handle is over five thousand, within the scheduling time slot of five minutes.

Note that we did not consider the overhead of video content transforming for this analysis, as it is actually completed in the (conventional) video encoding phase (refer to details of video transforming in § VI-B). Therefore, the overhead of joint optimization for energy reduction and anxiety alleviation in LPVS can be well controlled following the "one-slot-ahead" working mode, thus making no impact on other QoE metrics (e.g., delay and jitter) of video streaming in practice. The perceptual impact of video transform has been well addressed (e.g., in [17]) and is beyond the scope of our work.

VIII. Conclusions

We proposed a novel solution for low-power video streaming services, LPVS, to save the ever increasing display energy consumption of mobile devices and alleviate mobile users' low-battery anxiety. In specific, we explicitly depicted the scenario where LPVS could apply, and modeled the energy saving and anxiety reduction at the edge by a joint optimization problem. Then, we analyzed the difficulties in solving the problem and developed a two-phase heuristic method accordingly, aided by information compacting and Bayesian inference. To help build LPVS, we conducted an online survey on low-battery anxiety and collected data from $2,032$ mobile users, which were used to extract the quantitative relationship between the anxiety degree and battery status. With an LPVS emulator and a real-world *Twitch* dataset, we investigated the benefits of our LPVS solution in energy saving and anxiety reduction. The results demonstrated that LPVS can save the overall mobile users' battery lives by up to 37% and prolong the low-battery users' video watching time by 39%.

REFERENCES

[1] LG, "'Low battery anxiety' grips 9 out of ten people," https://www.lg.com/us/PDF/press-release/LG_Mobile_Low_Battery_Anxiety_Press_Release_FINAL_05_19_2016.pdf, May. 2016.
[2] W. Yuan and K. Nahrstedt, "Energy-efficient soft real-time cpu scheduling for mobile multimedia systems," in *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5. ACM, 2003, pp. 149–163.
[3] D. Kadjo, R. Ayoub, M. Kishinevsky, and P. V. Gratz, "A control-theoretic approach for energy efficient cpu-gpu subsystem in mobile platforms," in *DAC*. ACM, 2015, p. 62.
[4] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *MobiSys*. ACM, 2010, pp. 285–298.
[5] T. Zhang, X. Zhang, F. Liu, H. Leng, Q. Yu, and G. Liang, "etrain: Making wasted energy useful by utilizing heartbeats for mobile data transmissions," in *ICDCS*. IEEE, 2015, pp. 113–122.
[6] Z. Yan and C. W. Chen, "Too many pixels to perceive: Subpixel shutoff for display energy reduction on oled smartphones," in *MM*. ACM, 2017, pp. 717–725.
[7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.
[8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.
[9] A. Carroll, G. Heiser *et al.*, "An analysis of power consumption in a smartphone." in *USENIX ATC*, vol. 14. Boston, MA, 2010, pp. 21–21.
[10] RTINGS.com, "Oled and led tv power consumption and electricity cost," https://www.rtings.com/tv/learn/led-oled-power-consumption-and-electricity-cost, May. 2017.
[11] Y. Liu, M. Xiao, M. Zhang, X. Li, M. Dong, Z. Ma, Z. Li, and S. Chen, "Gocad: Gpu-assisted online content-adaptive display power saving for mobile devices in internet streaming," in *WWW*. International World Wide Web Conferences Steering Committee, 2016, pp. 1329–1338.
[12] M. Dong and L. Zhong, "Chameleon: a color-adaptive web browser for mobile oled displays," in *MobiSys*. ACM, 2011, pp. 85–98.
[13] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1537–1562, 2019.
[14] NOKIA, "Airframe open edge server," https://www.nokia.com/networks/products/airframe-open-edge-server/, Jun. 2019.
[15] Wowza Streaming Engine Transcoder Benchmarks, https://www.wowza.com/resources/Wowza_Transcoder_Benchmarks_Transrate720p.pdf.
[16] 4D Systems, "Introduction to oled displays design guide for active matrix oled (amoled) displays," https://www.ruf.rice.edu/~mobile/elec518/readings/display/4D_AMOLED_Presentation.pdf, May. 2008.
[17] P. Stanley-Marbell, V. Estellers, and M. Rinard, "Crayon: saving power through shape and color approximation on next-generation displays," in *EuroSys*. ACM, 2016, p. 11.
[18] L. Cheng, S. Mohapatra, M. El Zarki, N. Dutt, and N. Venkatasubramanian, "Quality-based backlight optimization for video playback on handheld devices," *Advances in Multimedia*, vol. 2007, 2007.
[19] C.-H. Lin, P.-C. Hsiu, and C.-K. Hsieh, "Dynamic backlight scaling optimization: A cloud-based energy-saving service for mobile streaming applications," *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 335–348, 2012.
[20] N. Chang, I. Choi, and H. Shim, "Dls: dynamic backlight luminance scaling of liquid crystal display," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 12, no. 8, pp. 837–846, 2004.
[21] W.-C. Cheng and M. Pedram, "Power minimization in a backlit tft-lcd display by concurrent brightness and contrast scaling," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 25–32, 2004.
[22] I. Choi, H. Shim, and N. Chang, "Low-power color tft lcd display for hand-held embedded systems," in *ISLPED*. ACM, 2002, pp. 112–117.
[23] A. Bhojan, "Adaptive video content manipulation for oled display power management," in *MobiQuitous*. ACM, 2018, pp. 236–245.
[24] H.-Y. Lin, C.-C. Hung, P.-C. Hsiu, and T.-W. Kuo, "Duet: an oled & gpu co-management scheme for dynamic resolution adaptation," in *DAC*. IEEE, 2018, pp. 1–6.
[25] C.-H. Lin, C.-K. Kang, and P.-C. Hsiu, "Catch your attention: Quality-retaining power saving on mobile oled displays," in *DAC*. IEEE, 2014, pp. 1–6.
[26] S. Pasricha, S. Mohapatra, M. Luthra, N. D. Dutt, and N. Venkatasubramanian, "Reducing backlight power consumption for streaming video applications on mobile handheld devices." in *ESTImedia*, 2003, pp. 11–17.
[27] Questionnaire of the Low-battery Anxiety Survey, https://www.dropbox.com/s/xvwktw6odo2xuft/lba-survey.pdf?dl=0.
[28] J. Ding, R. Cao, I. Saravanan, N. Morris, and C. Stewart, "Characterizing service level objectives for cloud services: Realities and myths," in *IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, 2019, pp. 200–206.
[29] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures," in *MICRO*, 2009, pp. 168–178.
[30] A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda, "Characterizing and modeling user activity on smartphones: summary," in *SIGMETRICS*, 2010, pp. 375–376.
[31] C. Ge, N. Wang, W. K. Chai, and H. Hellwagner, "Qoe-assured 4k http live streaming via transient segment holding at mobile edge," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1816–1830, 2018.
[32] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
[33] PR Newswire, "Inspur releases OTII server for edge computing," https://www.prnewswire.com/news-releases/inspur-releases-otii-server-for-edge-computing-at-mwc-2019-300802977.html, Feb. 2019.
[34] CPLEX Optimizer, https://www.ibm.com/analytics/cplex-optimizer.
[35] Gurobi Optimization, http://www.gurobi.com/.
[36] CVX Research, http://cvxr.com/cvx/.
[37] Twitch Statistics, https://twitchtracker.com/statistics.

APPENDIX: SURVEY PARTICIPANTS INFORMATION AND ETHICS

Table II shows the information of participants in our low-battery anxiety survey. Each participant was informed of the intention of this study, the course of data collection and processing, and how the data would be used. The survey does not raise any ethical issues.

TABLE II
SURVEY SUBJECTS AND CORRESPONDING FREQUENCIES (PARTICIPANTS AND MOBILE PHONES, $N = 2,032$).

Survey Subjects	Frequency (%)
Meta Info.	
# Cities	150
# Provinces*	31
# Countries	11
Gender	
Male	1095 (53.89)
Female	937 (46.11)
Age	
Under 18	9 (0.52)
$18 \sim 25$	888 (51.45)
$25 \sim 35$	460 (26.65)
$35 \sim 45$	250 (14.48)
$45 \sim 65$	119 (6.89)
Occupation	
Student	1024 (50.39)
Gov/Inst	271 (13.34)
Company	434 (21.36)
Freelance	144 (7.09)
Others	159 (7.82)
Smartphone Brand	
iPhone	737 (36.27)
Huawei	682 (33.56)
Xiaomi	228 (11.22)
Others	385 (18.95)

**Provinces refer to provincial-level administrative units of China.*