RnB: Rate and Brightness Adaptation for Rate-Distortion-Energy Tradeoff in HTTP Adaptive Streaming over Mobile Devices

Zhisheng Yan, Chang Wen Chen
Department of Computer Science and Engineering
State University of New York at Buffalo
{zyan3,chencw}@buffalo.edu

ABSTRACT

Video streaming is a prevalent mobile service that drains a significant amount of battery power. While various efforts have been made toward saving both video transfer and display energy, they are independently designed in an ad-hoc way and thereby can cause some non-apparent yet critical performance issues. To fill in this gap, this paper presents a fundamentally new design by jointly considering the end-to-end pipeline from the initial video encoding to the final mobile display. In essence, we shift the classic R-D tradeoff that has governed streaming system designs for decades to a fresh rate-distortion-energy (R-D-E) tradeoff specifically tailored for mobile devices. We present RnB, a video bitrate and display brightness adaptation platform that is standard-compliant, backward compatible, and device-neutral in order to achieve the proposed R-D-E tradeoff. RnB is empowered by some new discovery about the inherent relationship among bitrate, display brightness, and video quality as well as by an control-theoretic formulation to dynamically adapt the bitrate and scale the display brightness. Experimental results based on real-time implementation show that RnB can achieve an average of 19% energy reduction with final video quality comparable to conventional R-D based schemes.

CCS Concepts

•Information systems → Mobile information processing systems; Multimedia streaming;

Keywords

Video streaming; mobile devices; Rate-Distortion-Energy; backlight scaling; rate adaptation

1. INTRODUCTION

Video streaming has been widely recognized as a killer mobile application in this media-rich era. According to a recent study [1], video data will account for an impressive 75% of global mobile traffic by 2019. However, video streaming has also been identified as one of the most power-consuming applications on mobile devices [2,3]. Not only does it generate substantial traffic, but it also needs to keep the mobile display always on.

Video transfer and display are two correlated components that impact both quality and energy of mobile video streaming. Transferring a higher bitrate video brings a better video quality, but will cause more transfer energy. On the other hand, mobile display adaptation, such as screen brightness scaling, can reduce display power, but may degrade video quality. To link these two elements, the display adaptation strategy is strongly dependent on video pixel values [4], which are in turn received and decoded in the mobile device.

Although energy reduction of mobile video transfer and display have been widely studied, these efforts have been made in an ad-hoc way without considering the inherent connection between transfer and display. After scrutinizing the workflow of mobile video streaming, we argue that designing these components separately could lead to some non-apparent yet critical pitfalls.

First, nearly all existing approaches [4–12] for reducing mobile video display energy choose to darken the screen or convert the pixel color under a distortion constraint. One serious issue is that since they have no access to the original video at the server, they will adapt the display by only considering the distortion between the decoded video frame received on the device and the adapted frame to be displayed on the screen. Thus, a received video that already contains encoding or transmission distortion may be overlaid with additional display distortion due to display energy saving. Eventually, the video playback may experience unacceptable double distortion compared against the original uncompressed video. Furthermore, energy-saving display algorithms typically require pixel-by-pixel analysis of decoded videos before applying the adaptation. This leads to significant local computation power, substantially diminishing the overall power reduction of the mobile device. Even more sadly, such computation overhead needs to be repeated by millions of mobile devices that request the same video content in the global streaming ecosystems.

Building on the above analysis, we conclude that the primary reason behind these problems is that mobile display has been independently designed as a complementary scheme on top of video transfer, guided by classic rate-distortion (R-D) theory. The objective of this paper is to rethink the

MobiCom'16, October 03-07, 2016, New York City, NY, USA

DOI: http://dx.doi.org/10.1145/2973750.2973780

fundamentals of mobile video streaming by jointly studying the entire end-to-end pipeline from video encoding to mobile display. The ultimate goal is to shift the R-D tradeoff that has governed video streaming for decades to a new rate-distortion-energy (R-D-E) tradeoff specifically designed for mobile devices. We focus on mobile display and data transfer energy due to their dominance in power dissipation [2,13]. We have also confirmed that they jointly take up 87% of the system power in mobile video streaming (Section 2).

To achieve this ultimate goal, one key challenge is whether it is feasible to integrate these seemingly separate components such as transfer and display for the proposed R-D-E tradeoff. By examining the basics of HTTP adaptive streaming (or generally called *DASH*), the leading video streaming technique, we answer this question affirmatively. In current practice, DASH divides a source video into video chunks, encodes them with multiple bitrate versions, and stores them in the server such that a client dynamically picks the best version based on its bandwidth status. Following this strategy, we can encode additional video versions that are suitable for power-saving display adaptation to be applied at mobile devices and accordingly prepare a source video in multiple versions with R-D-E tradeoff. The mobile client can then dynamically fetch the optimal video version in term of R-D-E tradeoff and perform the display adaptation allowed by this version. This strategy not only recognizes the total distortion and energy incurred across the streaming pipeline, but also alleviates the duplicated on-board display analysis of individual devices into one-time processing in the server. To exemplify the mobile display adaptation and energy reduction, we focus on the most commonly used technique, *screen brightness scaling*, and validate its feasibility via measurements (Section 2).

To realize this promising video adaptation, we are facing two difficult technical challenges. First, the relationship among video bitrate, screen brightness and video quality is unclear, which is the key to the end-to-end R-D-E tradeoff. Second, a dynamic adaptation algorithm that dictates the system performance will need to be deliberately designed.

To tackle these challenges, we present *RnB*, a **R**ate **and** **B**rightness adaptation framework for HTTP adaptive streaming over mobile devices. RnB is a client-driven video adaptation framework, where the communication parameters and protocols are compliant with MPEG-DASH standard [14]. To optimize the end-to-end R-D-E tradeoff in RnB, we start with exploring the relationship among bitrate, screen brightness, and video quality. We follow classic R-D analysis and use regression analysis for a dataset of 490 videos to obtain their content-dependent relationship. We then design a receding horizon control mechanism to enforce the R-D-E based adaptation, where we seek the video chunk version with optimal bitrate and brightness that will minimize device energy under the bandwidth and quality constraints.

We prototype RnB in commercial mobile devices. We validate RnB designs and algorithms using both trace-driven emulations and real-world system experiments. We not only evaluate each design component of RnB, but also carry out system-level experiments under a variety of practical settings including video content, display type, ambient light, and wireless access interface. These results show that RnB can save an average of 19% energy with final video quality comparable to conventional R-D based DASH.

To summarize, the contributions of this paper include:

- A mobile video adaptation framework for end-to-end R-D-E tradeoff that augments the traditional R-D tradeoff (Section 3).
- A mathematical model manifesting the inherent relationship among bitrate, screen brightness, and video quality (Section 4).
- A control-theoretic formulation for video adaptation to effectively minimize the streaming energy without sacrificing video quality (Section 5).
- A practical demonstration of the satisfactory performance achieved by the end-to-end R-D-E tradeoff (Sections 6-7).

2. BACKGROUND AND MOTIVATION

2.1 Background

Power breakdown of video streaming. Mobile display and network interface card are usually among the top energy-consuming elements in a wide range of applications [2,13]. We now study the power breakdown in mobile devices during a streaming session. The power dissipation primarily results from data transfer, video processing (from decoding the data until storing them in the frame buffer), video display and other background or idle tasks. To measure the power of each subfunction, we herein follow the methodology adopted in [5] by comparing two power readings with the target subfunction on and off.

Figure 1 shows the power breakdown for a 300-second streaming session (2 Mbps video) via WiFi and LTE on a LG Optimus G Pro smartphone. As expected, display and data transfer cost the most significant portion (87%) of system power. We also observe that the video processing energy mainly from decoding is relatively low, which is consistent with previous measurements in [3]. Furthermore, by varying the requested bitrate from 700 Kbps to 6 Mbps for 10 trials, we found the difference among their video processing power is negligible with respect to the system power. Therefore, we shall focus on reducing display and data transfer power.

DASH and transfer power. HTTP adaptive streaming (or DASH) has become the dominant video delivery method for future Internet. Both research designs [15–19], commercial products [20–22] and industry standards such as MPEG-DASH [14], have been actively developed. A DASH media source is structured as an adaptation set. Within an adaptation set, the content is split into chunks and encoded into multiple versions characterized by bitrate (most commonly), resolution or other content features. The server stores these versions and prepares a media presentation description (MPD) file that summarizes the metadata of different versions. After receiving the MPD, the client will dynamically request to stream a particular version for each chunk based on the network condition and device status.

The wireless interfaces of mobile devices typically exist four states: idle, promotion, data transfer and tail [23, 24]. To investigate the energy profiles of different bitrates, we run ten 300-second DASH sessions by fixing the selected bitrate from 700 Kbps to 6 Mbps. We identified the states through examining the variations in power trace. We found that the energy of idle, promotion and tail states are relatively stable among the 10 sessions whereas the energy of data transfer is highly dependent on the selected bitrate (i.e., data amount),

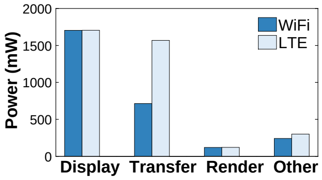


Figure 1: Display and transfer power dominate the system power.

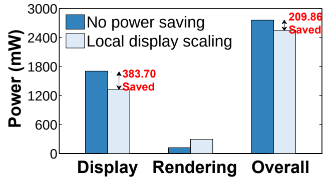


Figure 2: Local video processing reduces total power reduction.

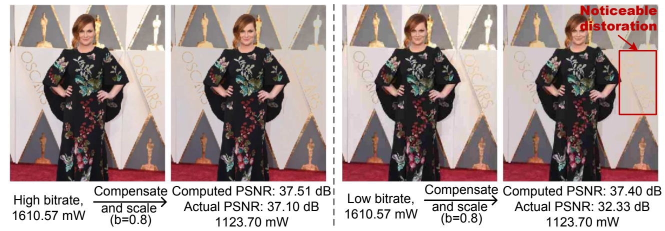


Figure 3: Brightness scaling is feasible, but needs to be jointly considered with received video quality.

ranging from to 14.8 to 122.6 J. This implies that there exists a large design space for balancing bandwidth, quality and transfer energy via rate adaptation.

Mobile display and brightness scaling. Liquid crystal display (LCD) and organic light-emitting diode (OLED) are two mainstream display technologies for mobile devices. Despite distinct display principle, one common approach that can save energy for both displays is to uniformly dim the screen brightness [4, 5]. This can be realized by dimming the backlight of LCD [12] or scaling the supply voltage of OLED [11]. In order not to limit our framework to either LCD or OLED devices, we choose this approach for power-saving display adaptation and collectively term it as *brightness scaling*. Since brightness scaling may degrade display quality, *luminance compensation* is normally needed to retain the perceptual quality. Luminance-compensated brightness scaling has a long history and has been guiding the designs of power-efficient displays [6–11]. We now summarize the foundations of this technique.

We first consider the YUV space of videos, where Y and U/V represent luminance and chrominance of a frame, respectively. If the screen brightness is dimmed by a factor b ($0 < b \leq 1$), the perceptual luminance of the content luminance Y reduces to $Y' = b \cdot Y$. To mitigate such distortion, luminance compensation is typically applied before the dimming by proportionally increasing the content luminance. As such, the perceptual luminance will be

$$Y' = b * \min(Y/b, 255) \approx Y \tag{1}$$

Note that the above process is not a lossless conversion since some pixel values may saturate to 255 after the compensation and thus the subsequent scaling would not perfectly recover the desired luminance. Figure 3 (left) illustrates the quality and display power for luminance-compensated brightness scaling. It indicates that this technique can simultaneously offer satisfactory perception and promising power reduction. By scaling the brightness for every group of video frames using average Y, this principle can be used in video playback.

2.2 Motivation

Pitfalls. From the above discussions, we have shown that video transfer and display are two design keys in energy-saving mobile streaming. However, display adaptation has been treated as an independent and post-processing task without joint considerations with the transfer. When saving display energy in mobile streaming, the display distortion can only be evaluated by comparing the brightness-scaled frame against the received frame after decoding rather than against the original uncompressed frame in the server. Such an evaluation ignores the possible encoding distortion in the backend. Note that we assume there is no transfer distortion since TCP-based DASH is loss-free. Thereby, if a video with relatively high encoding distortion is transferred, the screen should not be dimmed further. Figure 3 (right) shows that the actual PSNR between scaled frame and the original uncompressed frame (or the perceptual quality on the screen) could be bad even though the computed PSNR between the received and scaled frame is still reasonable.

Furthermore, the on-board brightness scaling would lead to non-optimal power reduction from a streaming system perspective. This is due to the fact that the pixel-by-pixel analysis and distortion computation in brightness scaling shall consume extra non-negligible power on mobile devices and will compromise the overall energy savings. We have tested a baseline implementation, in which scaling factor b is decreased from 1.0 in a step of 0.2 until the peak signal noise ratio (PSNR) between Y and Y' reaches 35 dB and then the optimal b is utilized for local compensation and scaling on the device. This procedure is operated once for a 2-second video chunk and thereby PSNR and luminance is averaged over the frames in a chunk. Figure 2 compares the power breakdown for streaming a 720p video with and without the local display adaptation. It shows that the energy saving is dropped significantly from 383.70 mW to 209.86 mW (45% less).

Insights. First, to avoid the aforementioned limitations, it is *desirable* to jointly design mobile display and transfer for saving video streaming energy. Second, it is also *feasible* to implement such an end-to-end R-D-E streaming on the basis of DASH. The bitrate version adaptability in DASH consolidates the feasibility. In particular, we can encode a video into additional versions that are well-suited to brightness scaling (i.e., luminance-compensated versions), and allow the client to dynamically pick the optimal version and to scale the screen brightness. Such a mechanism also eliminates the localized display analysis on a device and prevents the same repeated computation at massive number of devices that request the same video.

3. RnB ARCHITECTURE

In this section, we introduce the architecture of RnB as shown in Figure 4, which shall empower the system to accomplish end-to-end R-D-E optimization.

RnB servers. In the backend RnB servers, the distinct design is that we prepare a family of R-D-E based video versions for a given source video. To this end, we insert a new module, *Luminance Compensator* (LC), to pre-offset the potential distortion from brightness scaling in the mobile device. For a set of scaling factors $b \in \mathcal{B}$, LC compensates the original video luminance using (1) and creates a set of

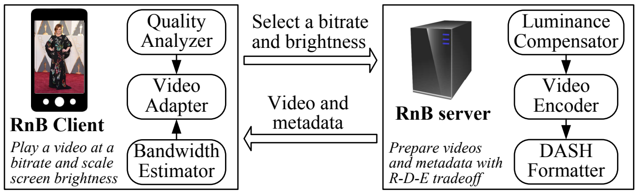


Figure 4: Modular architecture of RnB.

new videos. The U/V components of the videos remain unchanged. When scaling factor $b = 1.0$, the screen will be kept at full brightness and thereby no compensation is actually performed. Furthermore, LC also collects the intermediate frame statistics during the compensation and outputs certain content features for each video chunk. These motion or luminance features, e.g., average pixel luminance over all frames in a chunk, and the scaling factor set, will be sent along with the chunks and instruct the content-related R-D-E tradeoff in the client.

These luminance-compensated videos are then distributed to the *Video Encoder* (VE) for data compression. For a given luminance-compensated video, VE generates multiple versions with a set of different bitrates $r \in \mathcal{R}$. After looping through all the luminance-compensated videos, VE produces a family of DASH videos with R-D-E tradeoff.

Finally, the family of video versions flows to a *DASH Formatter* (DF) together with the scaling factor set $\mathcal{B}$, bitrate set $\mathcal{R}$, and the content features. DF then generates a DASH-compatible MPD file. In addition to the *video adaptation set* that expresses bitrate information, the MPD in RnB also records the information of brightness scaling factors and content features as the *metadata adaptation set*, which is supported by MPEG standard [25, 26]. These metadata are stored in text files and are associated with the available versions for each video chunk. This way, the client will be aware of the existence of such content features and can fetch them with the video chunks when needed.

RnB client. Similar to DASH, RnB client initially obtains the MPD and learns the version information of video chunks and the presence of metadata adaptation set. During the streaming process, it dynamically requests a version for each video chunk based on the bitrate, brightness scaling factor and content features of this chunk. The decision also relates to the predicted upcoming bandwidth from Bandwidth Estimator which leverages existing estimation scheme (Section 6). When the video chunk is received, the mobile device will scale the screen brightness accordingly using the scaling factor associated with this received version.

The principal intelligence of RnB client lies in two adaptation modules that optimize the dynamic decisions, i.e., *Quality Analyzer* (QA) and *Video Adaptor* (VA). First, QA models the relationship among video bitrate, screen brightness, video content, and video quality for encoded videos (Section 4). Given the content features, bitrate, and scaling brightness of a video chunk, QA will estimate the quality of this chunk considering both encoding and display distortion. Second, VA adaptively picks the optimal version for every chunk (Section 5). It will utilize QA as an internal function and minimize session energy. Thanks to such a joint bitrate and brightness adaptation, the double distortion and local overhead of existing schemes can be addressed.

Although RnB introduces some overhead such as extra video versions, the negative effects is limited. We will discuss practical issues in details in Section 9.

4. QUALITY ANALYZER

To allow end-to-end R-D-E optimization, RnB prepares multiple versions of videos by varying the bitrate and brightness scaling factor. Thus it is important to first understand and characterize the relationship among video quality and these two adaptation variables.

4.1 Methodology and Data Visualization

In classic rate-distortion analysis, there are generally two types of methods to model the correlation between video quality and bitrate [27, 28]. Analytic R-D analysis is to develop a mathematical formula for the R-D relationship by assuming known statistical models for the video source and individual components of the video encoder. The R-D formula is notoriously known for the mismatch between analytical bounds and practical performance. On the other hand, operational R-D analysis constructs R-D relationship by fitting the observed data from real video source and encoder. It can provide an effective guideline for streaming system designs [27]. For example, modern encoding algorithms will use this fitted model to conduct rate control, where a quantization step is assigned to each frame in order to achieve a target bitrate [29]. Hence, we proceed with the operational analysis for rate, screen brightness and quality.

It is well-known that motion features, e.g., the complexity of a scene, have strong effects on video bitrate and quality. Similarly, we envision that the luminance statistics of a video will also have substantial impacts on the performance of luminance compensation and subsequent encoding. Hence, we expect that the trained model is also impacted by content features.

Data collection. We gathered a dataset of 490 uncompressed video clips in a wide range of content categories including news, sports, movie, cartoon, music, etc. Since different chunks of a video may have different content features and the trained model should be able to estimate the quality of each chunk, we fix the content features for each training clips by manually ensuring that these clips have no scene change. The length of clips lasts from 2 to 10 seconds, which is consistent with the typical setting of DASH chunks [14]. As described in Section 3, we first compensate the luminance of source clips by a set of scaling factors $\mathcal{B} = \{0.1, 0.2, \cdots, 1.0\}$. By employing *FFmpeg* tool, we encode all clips identically at 1280x720 resolution, 25 frames/second, and H.264/AVC high profile with 10 bitrates $\mathcal{R} = \{700, 900, 1200, 1500, 2000, 2500, 3000, 4000, 5000, 6000\}$ (Kbps). We configure this setting since such encoding profiles could be broadly supported in most modern mid-end and high-end mobile devices.

We next assess the video quality, i.e., the perceptual distortion between the displayed video and original uncompressed video, by mimicking the operation in RnB client. We decode all encoded clips via *FFmpeg* and scale them by brightness factors to obtain the perceptual Y values. With perceptual and original Y values, we are ready to compute the video quality. We choose PSNR as the quality measure because it has been ubiquitously used. Although PSNR may not perfectly manifest the user experience in every case, it is still a general option to guide video streaming designs.

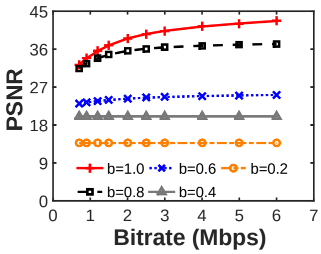


Figure 5: Example performance curve of clip #66.

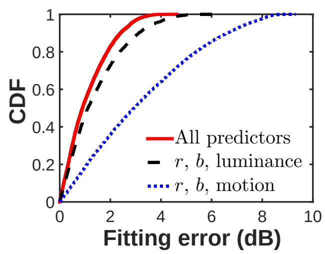


Figure 6: Adopting all features achieves the best fitting.

We eventually collect a total of 34000 PSNR values (340 contents x 10 brightness x 10 bitrates) for model training.

Example result. We can identify the general relationship among video quality, bitrate, and brightness scaling factor because these source clips exhibit similar behaviors (Figure 5), i.e., larger bitrate and scaling factor generate better quality. However, the exact positions and slopes of the curves are somewhat different when their contents differ. This is expected since a complex scene with large motion will demand a high bitrate to guarantee the quality. Similarly, if a content has a brighter scene, more pixels may saturate after luminance compensation, in which case their original values cannot be completely recovered and thus will increase the overall distortion.

4.2 Regression Analysis

Based on the asymptotic trend of the curves, we propose to harness a logistic function, an effective choice for video quality modeling suggested by ITU-T [30], in order to model PSNR versus bitrate, brightness factor and content features. A general logistic function can be written as:

$$F(\vec{x}) = \alpha_1 + \frac{\alpha_2}{1 + \exp^{-(\alpha_3 + \alpha_4 x_1 + \alpha_5 x_2 + \alpha_6 x_3 + \cdots)}} \quad (2)$$

where $\vec{\alpha} = (\alpha_1, \alpha_2, \cdots)$ are the coefficients and $\vec{x} = (x_1, x_2, \cdots)$ are the predictors. In this research, the predictors are bitrate, brightness scaling factor and content features. To signify the content complexity and motion, we adopt the ITU-T standardized spatial information (SI) and temporal information (TI) as *motion features* [31]. These features manifest the spatial difference of an average video frame and the temporal difference between successive frames. Furthermore, we test multiple luminance statistics to identify how bright or dark a content is as no standard metric is available. We calculate the mean, 25 percentile (25pY), median and 75 percentile (75pY) of the Y component in every frame and average them over all frames in the clip to acquire the 4 *luminance features*. These features can capture the luminance distribution of a video and thus impact the results of luminance compensation directly.

We use least-squares nonlinear regression to fit the data. We have tested a sequence of 8 potential models by adding each predictor one by one. The modeling results show that all predictors are statistically significant by passing the *t test* in each model. However, the error between fitted data and ground truth is diverse among these models. Figure 6 plots the errors of three example models that use bitrate and scaling factor, along with all content features, only motion features or only luminance features. It turns out that the model with all features achieves the least fitting errors among all 8 models with > 90% samples having < 2dB error, which is marginal in terms of user experience [32]. We show the regression results in Table 1.

Table 1: Model Coefficients and Evaluations

α_1	α_2	α_3	*bitrate*	*b*	*SI*
6.472	40.937	-0.816	0.023	5.455	-0.008
TI	***mean Y***	***median Y***	***75pY***	***25pY***	
-0.045	-0.094	0.016	-0.015	0.033	
RMSE: 1.47, PCC: 0.9846, *rho*: 0.9840					

After model selection, we evaluate the goodness of model fit in Table 1. The proposed model obtains a small root-mean-square error (RMSE) with respect to the range of PSNR and close-to-one values for Pearson correlation (PCC) and Spearman rank correlation (*rho*). This implies that the model captures the training data accurately and reasonably. We will validate the model against new testing data in Section 7.

Finally, we can obtain the Quality Analyzer in RnB by replacing $\vec{x}$ and $\vec{\alpha}$ in (2) with the terms and coefficients in Table 1. Note that the model training is not needed separately for every video content because content features are already included as predictors in the model. Besides, the model is not client specific because no energy term is included. Hence, the training only needs to be done offline once on the server. This model can be embedded in the client software and the smartphone can obtain it after installation. Then given the content features, scaling factor, and bitrate, RnB would be able to estimate the PSNR of any arbitrary chunk.

5. VIDEO ADAPTER

In this section, we formulate and solve the end-to-end R-D-E adaptation executed in Video Adapter (VA) by utilizing Quality Analyzer.

5.1 Formulation of R-D-E Optimization

We first develop a mathematical model for RnB. A stored video sequence is chopped into N chunks, each of which is indexed by $n (n = 1, 2, \cdots)$. Each chunk lasts T seconds and is pre-encoded into multiple versions using a set of brightness scaling factors $\mathcal{B}$ and bitrates $\mathcal{R}$. Upon downloading each chunk n, RnB client selects a video version tuple $(r(n), b(n)) \in \mathcal{R} \times \mathcal{B}$.

Constraints. Given the content features of nth chunk embedded in metadata adaptation set, the PSNR of the chunk can be estimated by Quality Analyzer, i.e., $Q(r(n), b(n))$. To provide high-quality video chunks and playback, the *first constraint* of VA is to ensure $Q(r(n), b(n))$ for each chunk to be reasonably high.

The video chunks are then fetched into a video buffer in RnB client before playback. The buffer size, i.e., the duration of the downloaded-yet-unplayed video, evolves dynamically as video chunks being downloaded and played. At the nth step, the client downloads the whole chunk n and the downloading time depends on the chunk size and network bandwidth. Simultaneously, the video is being played and the buffer size is being reduced. After downloading chunk

n, the buffer size $B(n)$ becomes

$$B(n) = B(n-1) + T - \frac{T \cdot r(n)}{W(n)} \tag{3}$$

where $T \cdot r(n)$ is the size of chunk n and $W(n)$ is the average bandwidth when downloading chunk n.

To derive $W(n)$, we assume the client finishes the downloading of chunk $n-1$ at time $t(n-1)$. Once chunk $n-1$ is completely downloaded, the client immediately starts to load chunk n at time $t(n)$. Note that the scheduling of chunk requests can impact the performance of a video client and is an active research topic in networking and multimedia communities [17, 18]. In this paper, we focus on exploring the fundamental R-D-E tradeoff without auxiliary designs and thereby consider the most general case of chunk scheduling, i.e., continuously loading the chunks without time gaps. Accordingly, average bandwidth $W(n)$ can be derived as,

$$W(n) = \frac{\int_{t(n-1)}^{t(n)} W_t \, dt}{t(n) - t(n-1)} \tag{4}$$

where W_t is the bandwidth at time moment t.

In addition to fetch high-quality chunks, another essential goal for video streaming is to avoid video rebuffer, where the display freezes at a still image when the buffer is empty. To prevent this annoying effect, the *second constraint* of VA is to bound the buffer size at every step as follows,

$$0 < B(n) \le B_{max} \tag{5}$$

where B_{max} is the maximum buffer size determined by the storage capacity of the player. In a rare case that the buffer becomes full, the client will wait for a while before fetching new chunks.

Objective. We aim to minimize the sum of display energy and transfer energy in streaming. The data transfer energy for chunk n can be written as,

$$E_{tran}(n) = P_{tran} \frac{T \cdot r(n)}{W(n)} \tag{6}$$

where P_{tran} is the constant data transfer power of the mobile device that is differed by devices and wireless access modes [23,33]. To download different bitrate $r(n)$, E_{tran} may differ as measured in Section 2.1.

Similarly, the display energy of chunk n becomes,

$$E_{disp}(n) = P_{disp}(b(n)) \cdot T \tag{7}$$

where $P_{disp}(b(n))$ is the display power when playing chunk n and is a non-decreasing concave function of screen brightness factor $b(n)$. Note that this is a simplified model for OLED since OLED energy is decided by both the screen brightness (as controlled by Android Settings) and the content color.

We can now summarize the objective of the R-D-E optimization as minimizing the total energy during the entire DASH session (totally N chunks), i.e.,

$$\min \sum_{n=1}^{N} E_{tran}(n) + E_{disp}(n) \tag{8}$$

Optimization. Hence, we can formally define the R-D-E optimization problem as follows,

$$\begin{array}{ll} \min_{\{\vec{r},\vec{b}\}} & \sum_{n=1}^{N} E_{tran}(n) + E_{disp}(n) \\ \text{s. t.} & Q(r(n), b(n)) \ge \theta \\ & 0 < B(n) \le B_{max} \\ & B(0) = B_{start} \\ & r(n) \in \mathcal{R},\ b(n) \in \mathcal{B} \\ & Eq.\ (3), (4), (6), (7),\ \forall n = 1, \cdots N \end{array} \tag{9}$$

where θ is the threshold of chunk PSNR and B_{start} is the startup buffer size after which the playback starts.

The intuition of (9) is to optimize the resource allocation across all N video chunks such that minimal energy can be achieved while still satisfying requirements of video quality. Note that RnB is generically designed and it is straightforward to formulate other types of R-D-E tradeoff. We revisit this issue in Section 9.

5.2 A Receding Horizon Control Solution

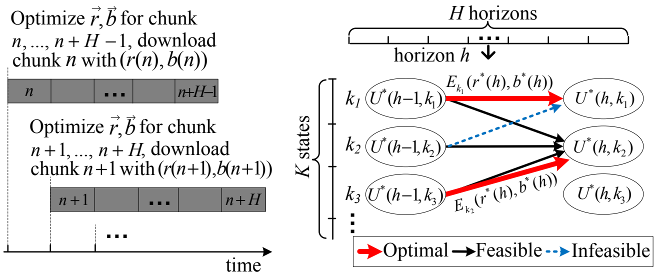
Ideally, if the future bandwidth for downloading each chunks $W(1), \cdots, W(N)$ is known, one can solve the problem in (9) by a one-time offline computation and obtain the optimal version trace $(\vec{r}, \vec{b})$ for chunk 1 to n. However, it is impossible in practice to acquire such perfect knowledge. Instead, an online decision-making strategy needs to be developed for the adaptation.

Online solution framework. In essence, the combinational optimization problem in (9) can be recognized as a dynamic stochastic control process. There are widely accepted treatments for this type of non-trivial problem: (a) Proportional-integral-derivative [34] control is a feedback control system that adjusts the adaptation variables based on the departure of observed utility from the desired utility. It is computationally simple, but it does not explicitly produce the optimal decision. (b) Various types of heuristic algorithms have been designed [19, 33]. For example, the most common heuristic is greedy algorithms, where the video version optimizing current utility would be selected. Heuristic algorithms are generally intuitive, but have no guarantee on the performance. (c) Markov Decision Process (MDP) [35, 36] assumes that the system states, e.g., bandwidth, evolves as a Markov process. It then probabilistically derive the future bandwidths and compute the expected utilities for all possible adaptation choices at every chunk offline. A table-lookup will be used to adapt the video online for maximum utility. Nevertheless, MDP relies on the strong assumption of Markov model for bandwidth dynamics. This has not been proved in real-world environment and how to obtain a general model for different wireless networks is not known (e.g., number of states, transition parameters).

Finally, as network conditions are generally stable and do not change dramatically over a short period, it is possible to estimate the future bandwidth within a short horizons with reasonable accuracy. Receding horizon control (RHC) [37, 38] that optimizes utility over a finite horizons H becomes a promising option. Therefore, we choose RHC as the basis of our online solution. Figure 7a shows the following steps for each chunk n.

1. *Obtain* current buffer $B(n-1)$ and the metadata for chunk n to $n+H-1$

2. *Predict* bandwidth $W(n), \cdots, W(n+H-1)$



(a) Receding horizon control (b) Dynamic programming

Figure 7: Illustrating the proposed algorithms.

3. *Solve* problem (9) over H horizons and obtain $\vec{r} = r(n), \cdots, r(n+H-1)$, $\vec{b} = b(n), \cdots, b(n+H-1)$

4. *Download* video chunk n with $(r(n), b(n))$ and the metadata for chunk $n+H$

5. *Move* horizon forward $(n = n+1)$, go to step 1

For initialization, the metadata for first H chunks is fetched during buffer startup such that the client would always know the content features of H chunks in advance. One attractive advantage of the proposed algorithm is its capacity to smooth out prediction error by optimizing the video adaptation over several chunks period. Thus large estimation error of one specific chunk will have reduced impacts on the overall performance. Furthermore, we can formally and flexibly optimize the system rather than relying on heuristic rules or bandwidth model assumptions.

Solving (9) over H horizons. To efficiently solve the R-D-E optimization for every chunk n (step 3 above), we propose a dynamic programming (DP) algorithm. Figure 7b shows the intuition. The algorithm simulates the download of future H chunks, where each horizon h corresponds to a chunk index within $[n, n+H-1]$. At horizon h, the buffer size before download can be in several states $B(h-1) = k_1, k_2, \cdots, k_K$. For each buffer state $B(h-1) \in [k_1, k_K]$, we denote the minimum cumulative energy leading to this buffer from the starting buffer by $U^*(h-1, B(h-1))$. By selecting a video version with $(r(h), b(h))$ at horizon h, an energy utility of $E(r(h), b(h))$ is consumed. The buffer will also evolve to a new state $B(h) \in [k_1, k_K]$ (the arrows in the Figure). Due to multiple video versions, there are multiple paths between two buffer states, forming an arbitrary bipartite graph. For each new buffer state $B(h)$, there is at most one path achieving minimum energy $U^*(h, B(h)) = U^*(h-1, B(h-1)) + E(r^*(h), b^*(h))$ by picking version $(r^*(h), b^*(h))$. If we iterate this process for H rounds, we will obtain the optimal utility for each possible buffer state after H horizons, $B(H)$. Then we can find the minimum cumulative energy among possible buffer, $U^*(H, B^*(H))$, and backtrace a sequence of optimal versions $\{(r^*(H), b^*(H)), \cdots, r^*(1), b^*(1))\}$, where $(r^*(1), b^*(1))$ is used toward chunk n. The DP algorithm is summarized in Algorithm 1.

The complexity of DP algorithm is $\mathcal{O}(HK|\mathcal{R}|)$. By tuning the horizon length H and buffer discretization level K, we can carefully balance the performance and complexity. We will show that $H = 8$ and $K = B_{max}$ can achieve a desired performance with negligible computation overhead (Section 7).

Algorithm 1 Dynamic Programming Algorithm for (9)

```
For 2-D table U, U*(0, k) ← 0
for horizon h = 1 to H do
    for buffer B(h − 1) = k_1 to k_K do
        for bitrate r ∈ R do
            · Find minimum feasible b for r that
              satisfies (9). If no feasible b, break.
            · Compute U = U*(h − 1, B(h − 1)) + E(r, b)
              and the new buffer B(h)
            · If U*(h, B(h)) = ∅ or U*(h, B(h)) > U,
              U*(h, B(h)) ← U, store side info
        end for
    end for
end for
Backtrace to obtain optimal version (r*(h), b*(h)),
where h = 1, ···, H that yields U*(H, B(H))
```

6. IMPLEMENTATION

Server. We implement RnB server in a Ubuntu 12.04 machine using Apache 2. We realize Luminance Compensator via MATLAB in order to easily manipulate high-resolution matrix by calling standard functions, e.g., `prctile`, to obtain content features. We have tested different number of brightness scaling versions to strike a tradeoff between performance and storage. We have found that $\mathcal{B} = 0.4, 0.6, 0.8, 1$ is sufficient to secure the performance gains and herein adopt this setting. We employ *FFmpeg* to encode the luminance-compensated videos into bitrates $\mathcal{R}$ described in Section 4. For DASH Formatter, we use another popular tool MP4Box from GPAC [39]. One minor modification is to enable MP4Box to index video metadata within the output MPD file. We associate metadata with each video chunk by `@associationId` and `@associationType` in MPD file, as suggested by MPEG [25,26]. We eventually batch these steps and prepare source contents with typical 2-second chunks.

Client. Further, we have implemented RnB client by extending ExoPlayer from Google [40]. We replace the adaptation class `AdaptiveEvaluator` by our implementation of R-D-E optimization. As our goal is not a specific bandwidth estimation design, we inherit ExoPlayer's method, where predicted bandwidth is the median over a sliding window of weighted bandwidths of past chunks. We will evaluate the impacts of bandwidth prediction in Section 7.1.2. The default startup buffer $B_{start} = 1$ and maximum buffer $B_{max} = 30$ are also kept. The power profile P_{tran} and $P_{disp}(\cdot)$ is obtained by measurements similar in Section 2.1. We achieve brightness scaling through Android's `WindowManager` API. A `Handler` is created to control brightness change and guarantee that it is synchronized with the video and audio codec to support pause, fast forward, etc.

Handling ambient light. Ambient light is an important issue related to screen brightness control. In general, it is visually desired to decrease the brightness in a dark room while increase the brightness in a bright environment. To handle this issue, RnB utilizes the light sensor in mobile devices to detect the lighting condition. We categorized ambient light into dark (<400 lux), normal ($\geq 401, \leq 1000$ lux) and bright (>1000 lux) for indoor conditions based on sensor reading [41]. Upon a session starts, RnB decides the light category and change the brightness set to stream accordingly. The brightness in dark and bright condition is set to $\{0.4, 0.6, 0.8\}$ and $\{0.6, 0.8, 1.0\}$, respectively.

7. EVALUATION

In this section, we start with component-wise evaluation of RnB for Quality Analyzer and Video Adapter (VA). We then present extensive system-level experiments and user studies under various practical situations. All experiments are carried out in real-world environment, except the sensitivity analysis of Video Adapter by controllable emulation.

Since there is no existing DASH algorithms with end-to-end R-D-E tradeoff, we compare RnB with common R-D based DASH solutions to show the promising energy reduction with comparable video quality. We also evaluate the myopic R-D-E optimization to verify the advantages of proposed control algorithms. These benchmark implementations are only differentiated by their video adaptation logic and are summarized as follows: (a) *RBA*: Rate based algorithm represents a large body of works that pick the maximum bitrate bounded by the estimated upcoming bandwidth, e.g., default implementations in [14, 39]. (b) *BBA*: Buffer based algorithm is another typical type of algorithms that maps buffer status to the selected bitrate. We adopt the piecewise linear function suggested in [42] with reservoir 5s and cushion 20s. (c) *Exo*: Google's ExoPlayer requires not only the bitrate to be lower than the estimated bandwidth, but also the buffer should be large/small enough for bitrate increase/decrease. (d) *Myopic*: The R-D-E optimization in (9) is solved by greedily considering only the upcoming chunk, i.e., horizon $H = 1$.

We evaluate RnB using three performance metrics indicating both energy and video quality. (a) *Session energy* measures total energy consumption of the device to stream and play a video session. All the sessions lasts 300 seconds. (b) *Average PSNR* represents the video clarity/details of the played video. It is the average PSNR over all the streamed video frames. (3) *Rebuffer rate* manifests playback consistency and stall which are defined by the total duration of rebuffer over the session duration. We use Monsoon power meter to obtain metric (1) and analyze the chunk statistics dumped by Android *logcat* to derive metric (2) and (3).

By default, all the evaluations is repeated for 10 runs on LG G Pro phone with a new test content (*Tear of Steel*) that has both stable scenes, e.g., talking, and moving scenes, e.g., fighting. Thereby, the run-averaged result is reported in this section.

7.1 Component-wise Evaluations

7.1.1 Quality Analyzer (QA)

Recall that in (9), our goal of using QA is to estimate whether a given video version is unsatisfactory (PSNR$< \theta$) or satisfactory (PSNR$\geq \theta$). Hence, QA can be virtually viewed as a binary classifier. We essentially need to evaluate the accuracy of classification. We pull a set of 150 test contents not used in training from the dataset described in Section 4. We calculate the ground truth PSNRs (totally 15000 values) after compensation and encoding and compare them against different thresholds θ to decide whether each video is unsatisfactory. By passing the bitrate, brightness, and content features of each video to QA, we also derive the estimated PSNR and satisfiability.

Figure 8 shows the good classification performance of QA

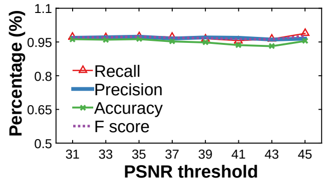


Figure 8: QA can accurately estimate if video quality is good.

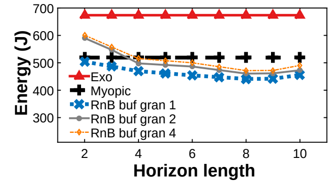


Figure 9: Session energy versus horizon

versus varying θ. With $\theta = 35$ in RnB implementation, QA achieves 96.25% accuracy in estimating whether or not a video has satisfactory quality. More importantly, the performance is stable under a set of realistic PSNR thresholds. This demonstrates that QA can be generally applied for different contents and user requirements. Furthermore, we compute multiple metrics by comparing the estimated and real PSNR. The RMSE, Pearson correlation and Spearman correlation are 2.1187, 0.9845, 0.9851, respectively. This indicates a close match between the estimated and true PSNR.

7.1.2 Video Adapter (VA)

We now present the sensitivity analysis for the proposed algorithms in VA. To guarantee the same network environment under different algorithm parameter settings, we use controlled emulation. For each emulation, we verify the last-hop wireless link is the bandwidth bottleneck by *iPerf*. We collect 10 traces of per-second available bandwidth in a reasonable WiFi network via *iPerf*, i.e., we avoid the case where the maximum bitrate is always selected due to the extremely high bandwidth. We then move to a fast WiFi network and use Linux *tc* tool to throttle the bandwidth between the server and the phone according to the traces.

Horizon length and buffer granularity. These are two critical parameters in the dynamic programming algorithm. Horizon length H denotes how far we look ahead into the future chunks and bandwidths. The buffer granularity K decides how many possible buffer levels we can search. A large K will reduce computation but may degrade performance. Figure 9 shows that session energy decreases and becomes stable as H increases since more future chunks and bandwidth information is considered. Nevertheless, bandwidth estimation error will propagate as a new estimation is based on previous estimations. Thus session energy can even increase when H keeps growing.

When the buffer granularity is coarse, less buffer levels are searched. This allows less space for RnB to allocate the resources across chunks. Therefore, performance of energy minimization degrades. We also mark the energy of other two algorithms for benchmark purpose. When $H = 1$, RnB reduces to Myopic algorithm. RnB can achieve 11% - 35% less energy than Exo.

Regarding video quality (not plotted), we notice that average PSNR of RnB is stable across all cases (mean:41.14, max:41.77, min:40.72). This result is comparable with other solutions, where Myopic reaches a mean of 41.21 and Exo obtains 43.07. Furthermore, no rebuffer is found for all algorithms using all parameters.

Overhead. Due to the real-time requirement of video adaptation, we need to investigate the overhead of RnB. Table 2 lists the execution time of the adaptation module in

Table 2: Adaptation Execution Time (ms)

ExoPlayer	RnB buf. gran.	RnB horizon $= 2, 4, 6, 8$			
1.13	1 second	2.96	6.89	10.16	12.54
	2 seconds	3.46	6.07	8.52	10.60
	4 seconds	2.13	5.26	7.82	9.04

several parameter settings. With a finer granularity and a larger horizon, the computation time of RnB becomes larger. However, the overhead is minimal and should be acceptable for real-time adaptation even when $H = 8$ and buffer granularity is 1s. Moreover, we detect that the memory and CPU load of all algorithms are almost identical at 30 MB and 5%, respectively. One reason for the relative low overhead may be because we use `HashMap` instead of regular `Array` to implement the dynamic programming search. This overhead is expected to be further reduced if implemented on newer smartphone models. In summary, considering the acceptable overhead and significant energy saving, we choose horizon $H = 8$ and buffer granularity 1s in the subsequent evaluations.

Bandwidth estimation. Since our focus is to improve energy efficiency for various practical scenarios rather than designing a bandwidth estimator with maximal accuracy, we choose not to validate a particular estimation algorithm. Instead, we evaluate the impacts of estimation errors from a black box of general estimators. We added a random noise to the real bandwidth as the estimated bandwidth (at every horizon). Figure 10 shows that as prediction error grows from negative (underestimation) to positive (overestimation), session energy of RnB drops whereas rebuffer rate increases. Overestimation allows RnB to pick a version of higher bitrate and lower brightness in some chunks, compared to underestimation. As the energy reduction from low brightness can compensate the extra energy from downloading high bitrate, the total energy decreases. Average PSNR remains stable since RnB aims to just secures the threshold even though the estimated bandwidth is high. But fetching a higher bitrate than the true bandwidth may cause annoying rebuffer. In contrast, RBA makes aggressive bitrate choices, which brings high-quality chunks but unsmooth playback.

7.2 System-level Evaluations

We first perform the system-level evaluation in an office WiFi network with multiple competing users and refer it as the *baseline* case. Through *iPerf* tests, we confirm that there is no bottleneck between the server to the WiFi access point and that the WiFi bandwidth is not trivially large than the maximum bitrate option. Figure 11a shows that RnB substantially reduces streaming energy while achieving comparable video quality to existing DASH schemes. On average, RnB consumes 20% less energy than *Exo*, 21% than RBA, 24% than BBA, and 11% than Myopic. The performance gain of RnB can be attributed to R-D-E streaming framework, where a luminance-compensated version can be requested and the screen brightness can be dimmed. RnB also avoids pushing the bitrate to the bandwidth limit, but rather choosing a sufficiently good version to meet quality constraint. Finally, these bitrate and scaled brightness selections are optimized over a future horizon to accommodate the bandwidth dynamics.

Myopic algorithm can save some energy due to the same R-D-E framework as RnB. As both bitrate and brightness

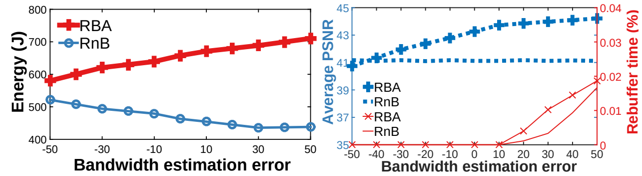


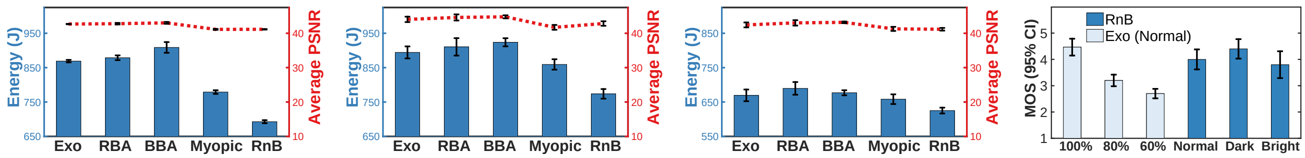
Figure 10: Energy (left) and video quality (right) versus bandwidth estimation error

contribute to energy and quality, there is a delicate trade-off between these two variables. Without considering future bandwidth, however, greedy choice of Myopic will lead to a degraded resource allocations among video chunks. On the other hand, existing approaches cause more energy because they fail to include energy into the request decisions. They all attempt to maximize the bitrate. However, user experience may not improve after video quality reaches a threshold, which is referred as *just noticeable distortion* theory. Therefore, the 1$\sim$2 dB PSNR improvement over RnB is negligible and we will confirm this via user studies. Among three conventional R-D schemes, ExoPlayer spends the least energy due to its conservative bitrate choice. Only when buffer and estimated bandwidth are both large enough, a higher bitrate will be picked.

Impacts of content. We then evaluate RnB by replacing the source video in baseline case to a new content, *Big Buck Bunny*. Figure 11b shows that RnB again saves a large amount of energy, i.e., 13%, 15%, 16% and 10% over Exo, RBA, BBA, and Myopic, respectively, without sacrificing video quality. By looking into the content features of both videos, we observe the energy difference of RnB is resulted from the general luminance level of the content. As a cartoon video, Bunny has extremely bright scenes and background while the Sci-Fi movie Steel mostly presents a normal or dark frame. Hence, more pixels in Bunny saturate after luminance compensation and cannot be recovered, which increases the overall distortion of encoded versions. To guarantee PSNR, a higher bitrate or brightness scaling version is needed and thereby cost more energy.

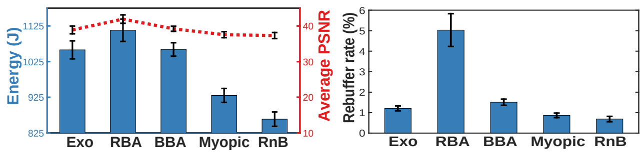
Impacts of display type. Instead of the LCD-based LG phone in baseline case, we further use a Google Nexus 6 with OLED display to run the evaluation. Figure 11c demonstrates the results. First, all the schemes achieve a lower energy compared to Figure 11a. This is due to the different efficiency between the devices in hardware and system kernel. It is also interesting to observe that the energy saving of RnB is somewhat reduced (8% less than R-D schemes on average). The energy cost of OLED is not only decided by the circuit supply voltage that regulates the overall display illuminance and screen brightness, but also by the luminance efficacy of individual pixel cell. Even though brightness scaling reduces the voltage, the compensated pixels are typically less energy-efficient and need higher current to drive, which reduces the advantage of RnB. However, for tablets with larger screen, RnB is expected to save more energy for both LCD and OLED devices.

Impacts of ambient light. We repeat the baseline case in both dark and bright environment and verify the ambient light sensor reading before evaluations. We observe that RnB consumes $> 4\%$ less energy than that under normal light since it excludes the versions with $b = 1.0$. Conversely,



(a) Baseline case (b) Bunny video (c) OLED device (d) User study

Figure 11: RnB achieves promising energy saving and comparable video quality in various settings.



(a) Energy and PSNR (b) Rebuffer rate

Figure 12: RnB in LTE.

RnB's energy is comparable ($< 1\%$ difference) in bright and normal conditions. This results from the fact that video chunks with low brightness (e.g., 0.4) can hardly meet the PSNR threshold and were seldom picked in the normal case. Even though they are not allowed under bright light, this fact does not make noticeable contribution.

User study. We assess the subjective perception of RnB by a user study. We have recruited 15 participants and asked them to rate the video streaming experience in terms of clarity/bitrate and brightness. We follow single-stimulus protocol by ITU [43] to compare the user experience of RnB under different ambient light conditions. We also evaluate traditional R-D based Exo when the screen brightness is simply scaled by different levels in normal ambient light. Figure 11d plots the mean opinion score (MOS) with 95% confidence interval (CI). In general, RnB provides a satisfactory experience and is comparable with Exo using 100% brightness because RnB compensates the video luminance before scaling the screen. The major reason of the MOS drop (4.4 to 4.0) is that the inter-chunk brightness variation can be occasionally unsmooth (or flickering effect). This limitation can be overcome by adding a new constraint of inter-chunk brightness variation into the optimization formulation and can be considered as an important future work. On the other hand, the user experience of Exo quickly degrades when using 80% and 60% screen brightness because the perceptual luminance of videos is simply dark. In dark condition, even though the average selected brightness of RnB drops, the user experience still improves. This could be explained by the reduced variation range of brightness under dark case, which brings less video quality fluctuation.

Handling LTE. We move the baseline evaluation to a commercial LTE network in a department building with many coexistent users. Figure 12 shows that RnB can satisfactorily adapt to LTE networks by achieving an average 20% energy saving and comparable video quality to R-D based DASH. Since the LTE interface of mobile devices is more power-consuming than that of WiFi, the energy of all schemes are increased. Due to the higher percentage of transfer energy, the contribution of display energy saving also diminishes. Besides, the PSNR level are generally decreased and rebuffer has been observed for all the approaches. The quality degradation is due to the frequently varying LTE channel that is difficult to track. Such fluctuated bandwidth increases the bandwidth prediction error. RBA suffers the most in LTE since it is purely dependent on bandwidth estimation whereas other algorithms consider buffer occupancy and thereby can enjoy one more signal for more appropriate adaptation.

8. RELATED WORK

Mobile energy dilemma. Many energy measurement and profiling studies have been done on mobile devices [44–46]. Unfortunately, the most power-hungry mobile applications are usually those popular applications with multimedia data, e.g., video streaming, web loading, and gaming. Addressing the energy issue of these applications become extremely urgent. Bui *et al.* [47] reduced web loading energy by revising the current browser design with only a small increase of loading time. He *et al.* [48] dynamically scale display resolution for high-end smartphones to save the GPU power in gaming. However, there is a lack of systematic treatment for mobile video streaming energy. In this paper, we fill this gap by designing and implementing a end-to-end R-D-E optimized video adaptation framework.

Display energy reduction. Display has been an active target in mobile energy research. Many schemes [6–10, 12] aim to dim LCD backlight, e.g., Lin *et al.* [9] optimize dynamic video backlight scaling under user experience constraints. For OLED, dynamic voltage scaling [11] and color transform [4] attract the most attention. Chameleon [4] is the first per-pixel color transform scheme to reduce OLED power. All these schemes may work well for offline video playback. For streaming video, however, they lack the original video as the benchmark for adaptation, which makes the distortion intractable. In contrast, RnB fundamentally solves this problem based on an adaptation framework with end-to-end joint design, which offers a promising alternative to the prevailing mobile streaming applications.

Video transfer energy reduction. Several network protocols [24, 33, 49, 50] have been proposed to minimize the radio activity or duration of tail state in mobile video streaming. For example, eSchedule [33] uses crowd-sourced statistics to balance energy waste of user leaving session and tail state. However, these complementary schemes do not explicitly consider display energy. In fact, they are orthogonal to RnB framework and can be used to further improve RnB performance.

9. DISCUSSION

Incentives. There are strong incentives for different parties involved in RnB to adopt the R-D-E tradeoff in DASH. Mobile users have a natural incentive as they can maximize the lifetime of their devices and view the videos with satisfactory quality. Content providers can enhance customer experience by extending their mobile viewing time, which can be marketed as an advantage over their competitors. Content providers can also obtain more video consumption since users are allowed to spend more time with the same battery, which could bring more advertising income.

Overhead. Although R-D-E tradeoff introduces additional video versions in RnB server, we believe the overhead is tolerable. First, only a small number of versions are practically needed to ensure the performance gains, e.g., three extra brightness scaling versions in our implementation ($\mathcal{B} = \{0.4, 0.6, 0.8, 1.0\}$). Overly aggressive brightness scaling would lead to unrecoverable display distortion. Second, storage cost is becoming more affordable in this cloud-computing age. That is probably why traditional DASH with multiple bitrate versions has been and is still popular.

Furthermore, although RnB necessitates the transmission of additional metadata to aid the video adaptation, each video chunk only adds several floating-point numbers for metadata and thereby the extra cost for bandwidth and energy is negligible.

Compatibility and user preference. RnB is a standard-compliant framework. The media adaptation set, metadata adaptation set, MPD file all strictly follow MPEG standard [14, 25]. Furthermore, RnB is backward compatible in a sense that if energy saving is not preferred by users, the framework can be reduced to conventional DASH. Upon user input, RnB client simply requests video versions with brightness factor $b = 1.0$ via the regular R-D tradeoff.

More importantly, RnB provide the user-defined PSNR threshold such that users can flexibly strike a tradeoff between user experience and energy saving. By increasing the threshold via a simple UI, users can have a better experience if energy saving is an inferior factor.

10. CONCLUSION

In this paper, we address the fundamental limitation of conventional R-D DASH systems in mobile energy reduction. The key observation is that independently designed video transfer and on-board display energy adaptation cause intractable distortion and repeated computation overhead. We present RnB, a radically new designed video bitrate and screen brightness adaptation to jointly explore transfer and display in order to balance mobile R-D-E tradeoff. RnB is characterized by encoding multiple versions of R-D-E based video versions, as well as by leveraging a content-dependent rate-brightness-quality relationship we have discovered and a receding horizon control based algorithm we have proposed. Both system-level and component-wise evaluations shows that RnB shall achieve substantial (an average of 19%) energy reduction with comparable quality to existing R-D schemes. We believe RnB can enable a suite of future R-D-E based explorations in mobile video streaming and display adaptation, including video requests scheduling and metadata assisted display adaptation.

11. ACKNOWLEDGEMENTS

This work is supported by NSF Grant ECCS-1405594.

12. REFERENCES

[1] Cisco. Cisco visual networking index, 2014-2019. http://goo.gl/v8pMqV.

[2] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *USENIX annual technical conference (USENIXATC)*, June 2010.

[3] X. Chen, Y. Chen, Z. Ma, and F. C. Fernandes. How is energy consumed in smartphone display applications? In *ACM Workshop on Mobile Computing Systems and Applications (HotMobile)*, February 2013.

[4] M. Dong and L. Zhong. Chameleon: a color-adaptive web browser for mobile oled displays. In *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2013.

[5] M. Dong and L. Zhong. Power modeling and optimization for oled displays. *IEEE Trans. Mobile Comput.*, 11:1587–1599, September 2012.

[6] N. Chang, I. Choi, and H. Shim. Dls: dynamic backlight luminance scaling of liquid crystal display. *IEEE Trans. VLSI Syst.*, 12:837–846, August 2004.

[7] A. Iranli, H. Fatemi, and M. Pedram. Hebs: histogram equalization for backlight scaling. In *Proc. of Design, Automation and Test in Europe (DATE)*, March 2005.

[8] 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*, 2007(1):4–4, 2007.

[9] 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 Trans. Comput.*, 63:335–348, February 2014.

[10] 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 *International World Wide Web Conference (WWW)*, April 2016.

[11] D. Shin, Y. Kim, N. Chang, and M. Pedram. Dynamic driver supply voltage scaling for organic light emitting diode displays. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 32:1017–1030, July 2013.

[12] Z. Yan, Q. Liu, T. Zhang, and C. W. Chen. Exploring qoe for power efficiency: A field study on mobile videos with lcd displays. In *ACM International Conference on Multimedia (MM)*, October 2015.

[13] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. Morley Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *IEEE/ACM/IFIP International Conference on Hardware/software Codesign and System Synthesis (CODES+ISSS)*, October 2010.

[14] I. Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE Multimedia*, 18:62–67, April 2011.

[15] X. Xie, X. Zhang, S. Kumar, and L. E. Li. pistream: Physical layer informed adaptive video streaming over lte. In *ACM International Conference on Mobile Computing & Networking (MobiCom)*, pages 413–425, Paris, France, September 2015.

[16] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang. A scheduling framework for adaptive video delivery over cellular networks. In *ACM International Conference on Mobile Computing & Networking (MobiCom)*, pages 389–400,

Miami, USA, September 2013.

[17] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, AC. Begen, and D. Oran. Probe and adapt: rate adaptation for HTTP video streaming at scale. *IEEE J. Sel. Areas Commun.*, 32(4):719–733, April 2014.

[18] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In *ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 97–108, Nice, France, September 2012.

[19] Z. Yan, J. Xue, and C. W. Chen. QoE continuum driven HTTP adaptive streaming over multi-client wireless networks. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, Chengdu, China, July 2014.

[20] Microsoft. Smooth streaming. http://www.iis.net/downloads/microsoft/smooth-streaming.

[21] Adobe. Http dynamic streaming. http://www.adobe.com/products/hds-dynamic-streaming.html.

[22] Apple. Http live streaming. https://developer.apple.com/streaming.

[23] J. Huang, F. Qian, A. Gerber, Z. Morley Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2012.

[24] W. Hu and G. Cao. Energy-aware video streaming on smartphones. In *IEEE Conference on Computer Communications (INFOCOM)*, April 2015.

[25] MPEG systems technologies - part 11: Energy-efficient media consumption (green metadata), ISO/IEC 23001-11:2015, 2015.

[26] F. C. Fernandes, X. Ducloux, Z. Ma, E. Faramarzi, P. Gendron, and J. Wen. The green metadata standard for energy-efficient video consumption. *IEEE Multimedia*, 22:80–87, March 2015.

[27] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Process. Mag.*, 15:23–50, November 1998.

[28] Z. He and S. K. Mitra. A unified rate-distortion analysis framework for transform coding. *IEEE Trans. Circuits Syst. Video Technol.*, 11:1221–1236, December 2002.

[29] X. Jing, L. P. Chau, and W. C. Siu. Frame complexity-based rate-quantization model for h.264/avc intraframe rate control. *IEEE Signal Process. Lett.*, 15:373–376, March 2008.

[30] ITU opinion model for video-telephony applications. ITU-T Recommendation G.1070. 2007.

[31] ITU subjective video quality assessment methods for multimedia applications. ITU-T Recommendation P.910. 1999.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13:600–612, April 2004.

[33] M.A. Hoque, M. Siekkinen, and J.K. Nurminen. Using crowd-sourced viewing statistics to save energy in wireless video streaming. In *ACM International Conference on Mobile Computing & Networking (MobiCom)*, September 2013.

[34] G. Tian and Y. Liu. Towards agile and smooth video adaptation in dynamic http streaming. In *ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 109–120, Nice, France, December 2012.

[35] J. Chen, A. Ghosh, J. Magutt, and M. Chiang. Qava: quota aware video adaptation. In *ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 121–132, Nice, France, December 2012.

[36] S. Xiang, L. Cai, and J. Pan. Adaptive scalable video streaming in wireless networks. In *ACM conference on Multimedia systems (MMSys)*, pages 167–172, Chapel Hill, USA, February 2012.

[37] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control. *IEEE Control Syst.*, 31:52–65, June 2011.

[38] 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, London, UK, August 2015.

[39] Gpac. https://gpac.wp.mines-telecom.fr.

[40] Exoplayer. https://google.github.io/ExoPlayer.

[41] Microsoft. Understanding and interpreting lux values. http://goo.gl/sJ7nXg.

[42] 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, Chicago, USA, August 2014.

[43] ITU methodology for the subjective assessment of the quality of television pictures. ITU-R BT.500-13. 2012.

[44] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *ACM Internet Measurement Conference(IMC)*, November 2009.

[45] 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)*, June 2012.

[46] R. Mittal, A. Kansal, and R. Chandra. Empowering developers to estimate app energy consumption. In *ACM (MobiCom)*, August 2012.

[47] D. H. Bui, Y. Liu, H. Kim, I. Shin, and F. Zhao. Rethinking energy-performance trade-off in mobile web page loading. In *ACM MobiCom*, September 2015.

[48] S. He, Y. Liu, and H. Zhou. Optimizing smartphone power consumption through dynamic resolution scaling. In *ACM MobiCom*, September 2015.

[49] X. Li, M. Dong, Z. Ma, and F.C. Fernandes. Greentube: power optimization for mobile videostreaming via dynamic cache management. In *ACM International Conference on Multimedia (MM)*, October 2012.

[50] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *ACM MobiSys*, June 2006.