

# Battery Aware Video Delivery Techniques Using Rate Adaptation and Base Station Reconfiguration

Ranjini Guruprasad and Sujit Dey, *Fellow, IEEE*

**Abstract**—With mobile video increasingly becoming an important driver of mobile device usage, the battery consumption of mobile devices will be dominated by video delivery and playback. In this paper, we develop battery efficient video download techniques that vary video download rate dynamically, including stopping video download at times, depending on mobile device buffer levels and the channel conditions experienced, to maximize battery life while ensuring no degradation in user experience. The proposed dynamic download rate adaptation techniques enable the base station to adapt the MIMO transceiver configurations to reduce battery load required by MIMO components on the mobile device. In order to further enhance battery life, we propose to utilize video bit rate adaptation, in addition to download rate adaptation and MIMO reconfiguration. The proposed battery aware bit rate adaptation techniques take into account the mobile device battery and buffer levels, and network load and channel conditions experienced, to maximize battery lifetime (hence video viewing time) while ensuring desired level of video experience (measured in terms of video quality and stalls experienced). We propose a new metric termed “video experience longevity (VEL)” which quantifies the performance of the proposed bit rate adaptation techniques in terms of video viewing time and video experience. Extensive experiments conducted under variable channel conditions and network load demonstrate that the proposed battery aware video delivery techniques can significantly outperform other video delivery techniques in terms of battery lifetime and VEL metric (for bit rate adaptation techniques) while ensuring desired level of video experience.

**Index Terms**—Base station reconfiguration, battery life, mobile video, power consumption, rate adaptation, user experience, video viewing time.

## I. INTRODUCTION

**B**Y 2018, mobile video is expected to contribute to about two thirds of the total mobile data traffic [1], making it the leading multimedia application on mobile devices. As mobile

video is a data and compute intensive application, it places significant demands on processing and battery capabilities of mobile devices. While the processing capabilities of mobile devices continue to increase significantly, the incremental improvements in battery technologies will lead to frustratingly lower battery lifetime. Consequently, it is critical to develop techniques that can lower mobile video battery consumption. It has been shown that RF and baseband components used for video download are major contributors to battery consumption in addition to decoder and display used for playback [2]. With the adoption of MIMO technologies that use multiple antennas with power consuming baseband processing, power due to radio frequency (RF) and baseband components will dominate the power consumption for high bit rate mobile video applications. Hence, this paper focuses on reducing battery demand imposed by MIMO RF and baseband components while downloading video.

We first consider the widely adopted Progressive Download video delivery approach, which attempts to download video at a rate higher than the video bit rate and hence the video playback rate, thereby buffering video at the mobile device while it is simultaneously being played back [3]. The higher download rate and hence buffering is done to avoid buffer underflow (stalling) in case of bad network conditions during the video session, but there is no consideration about the effect of video download on the mobile device battery. In contrast, we propose a new *battery efficient video download approach* that utilizes elasticity of the video buffer to dynamically adapt the video download rate, sometimes even stopping video download, enabling reconfiguration or idling of the base station RF and baseband components in a manner that reduces or eliminates battery demand of the mobile device RF and baseband components. While adapting the download rate, the proposed approach also tries to avoid buffer underflow, and since the video bit rate is never adapted, user experience is not compromised while enhancing battery lifetime.

To further enhance battery lifetime, we next consider adapting the video bit rate in addition to adapting the video download rate as the former can further reduce the amount of data to be downloaded and hence the battery load. However, adapting the video bit rate will compromise video quality, leading to a possible tradeoff between enhanced longevity of video experience and video quality. Adaptive Bit Rate (ABR) streaming techniques [4] are gaining popularity, but they primarily address minimizing stalling of video under challenging network conditions. In contrast, we propose *battery aware adaptive bit rate streaming techniques* which adapt

Manuscript received May 03, 2014; revised January 18, 2015; accepted May 04, 2015. Date of publication May 22, 2015; date of current version August 10, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yiannis Andreopoulos.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: rgurupra@ucsd.edu; dey@ucsd.edu).

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org> provided by the authors. This includes a supplementary file that contains additional material not included within the paper itself. This material is 714 KB in size.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMM.2015.2436821

video bit rate, download rate and MIMO RF and baseband configurations, depending on the battery and buffer levels, and network load and channel conditions experienced during video streaming to maximize the longevity of video experience while ensuring a desired level of quality of video experience. We extend the conventional notion of video user experience to include the longevity of video watching (which can be limited by battery lifetime) by introducing the *Video Experience Longevity (VEL)* metric. We use the VEL metric to quantify and compare the performance of the proposed battery aware ABR techniques with other ABR techniques. As dynamic streaming over HTTP (DASH) is a widely accepted standard for ABR streaming, we will henceforth refer to ABR as DASH.

#### A. Related Work

In this section we will briefly describe past work related to base station and mobile device MIMO reconfiguration, video bit rate adaptation and battery efficient video delivery. As we will discuss, either these techniques do not address maximizing battery lifetime, or the ones that address do not consider using rate adaptation and transceiver reconfiguration whose effectiveness we will demonstrate in this paper.

Base station reconfiguration techniques have been developed for cognitive radios for dynamic spectrum management [5], which is not the focus of this paper. The focus in [6] was on choosing optimal MIMO parameter set to minimize overall link energy while satisfying bit error rate and throughput. While the above technique does not consider video delivery, [7] proposed to use Space Time Multiplexing (STM) and Space Time Block Coding (STBC) to reduce video distortion due to wireless video delivery; however, the latter does not address energy consumption. In [8], rate adaptation and corresponding switching between Single Input Multi Output (SIMO) and MIMO is proposed to save uplink RF transmission energy when mobile device is transmitting files. However [8], does not aim to reduce downlink RF and baseband processing battery consumption when mobile device is downloading video, which is the objective of this work. The energy efficient rate adaptation (EERA) technique proposed in [9] achieves energy efficiency at the client by selecting RF and MIMO baseband components at the access point and client Wireless Network Interface Card (WNIC) in a manner that reduces per bit energy while maintaining the minimum required goodput determined by the video bit rate and channel condition. However the energy efficient rate adaptation technique proposed in [9] does not utilize the elasticity of the video buffer to dynamically adapt the video download rate, including stopping transmission, to avoid stalling and reduce battery load, which constitutes an important part of our proposed approach. Also, mode selection in [9] requires base station to allocate maximum number of antennas to each user which places high demand on base station resources whereas our techniques have no such requirement.

Recently, there has been significant research done on developing video bit rate adaptation techniques [10], [11], including several commercial HTTP based Adaptive Bit Rate video streaming solutions like Apple HTTP Live Streaming [12], Microsoft Smooth Streaming [13] and Adobe Open Source

Media Framework (OSMF).<sup>1</sup> Unlike the above adaptive HTTP streaming clients and techniques which to the best of our knowledge (based on available public information at the time of writing this manuscript, including the Adobe OSMF source files) focus on ensuring user experience in a non-battery aware manner, our proposed techniques focus on maximizing battery lifetime while also ensuring desired level of video experience.

Techniques have also been developed to address energy and battery life of mobile devices during video delivery. In [14], a base station scheduling technique is proposed which utilizes the Variable Bit Rate (VBR) encoding of multiple broadcast streams in a manner that does not under/overflow the client buffers and allows transmission of video streams in bursts, the latter allowing switching off the client WNIC in between bursts to reduce energy consumption on mobile devices. However, the above approach cannot be applied to on-demand unicast video delivery (like YouTube) which is the target of this paper. Authors in [15] propose battery aware video streaming by changing video encoding parameters such as bit rate, frames/second in real time using a proxy server and switching off the client WNIC after bulk download. Our proposed approach does not require computationally intensive real time transcoding and utilizes different bit rate representations of a given video available on the server for video bit rate adaptation. Battery and stream aware adaptive multimedia (BaSe-AMy) streaming techniques proposed in [16] adapt video bit rate depending on battery level, packet loss and remaining video stream duration. However, these techniques do not adapt download rate and transceiver configuration which increases the battery savings achieved by our proposed techniques.

To the best of our knowledge, this is the first work which proposes to: (a) jointly adapt video download rate and MIMO transceiver components to maximize battery lifetime and ensure user experience during video download; (b) additionally adapt video bit rate to maximize video experience longevity while maintaining desired level of video experience during adaptive bit rate streaming; and (c) quantify the performance of adaptive bit rate streaming techniques in terms of both video viewing time and user experience. In Section II, we provide an overview of our battery aware video delivery approach. In Section III, we formulate the download rate and transceiver configuration selection as an optimization problem and provide a solution. In Section IV, we present the simulation framework developed for video download and experimental results obtained using different video download techniques. In Section V, we formulate bit rate, download rate and transceiver configuration selection as an optimization problem and offer a solution which guarantees minimum desired video quality, and subsequently extend the solution with a heuristic to achieve higher video quality when possible. We conclude the section with formulation of the “Video Experience Longevity” metric. In Section VI, we present the simulation framework developed for DASH streaming and experimental results obtained using different DASH based streaming techniques. We conclude in Section VII.

<sup>1</sup>“Open source media framework,” [Online]. Available: <http://www.osmf.org>

## II. BATTERY AWARE VIDEO DELIVERY-OVERVIEW

In this section, we will describe our overall approach towards battery aware video delivery. We will then discuss in detail different ways video bit rate and download rate can be selected and base station and mobile device can be reconfigured, to reduce battery load and the effect on user experience.

### A. Overall Approach

Our overall approach towards video bit rate and download rate adaptation and corresponding transceiver reconfiguration for battery aware video delivery consists of two main objectives namely, maximization of battery lifetime and ensuring user experience.

Our approach towards prolonging battery life [17] is based on the following factors: (1) minimizing battery load (current drawn from battery), and duration of load, and (2) idling the battery allowing it to recover charge, and increasing the duration of idling. Our proposed approach affects the above two factors in the following three ways. (a) Varying video download rate: A required video download rate is determined by the video bit rate (rate at which video is encoded by the encoder and decoded by the decoder), amount of data buffered at the mobile, and channel conditions. The required download rate is achieved by the base station with suitable configuration of its RF and baseband components, with corresponding mobile device configurations, the latter affecting battery load. Hence, for a given video bit rate, by utilizing the elasticity that the video buffer offers, the download rate can be varied and the base station reconfigured in a way that reduces the battery load imposed by the mobile device RF and baseband processing. (b) Stopping download: If for certain periods of time, video download and hence related processing on mobile device can be stopped, the battery load can be reduced to just playback load which is much lower than load due to downloading. Due to significant difference in consecutive loads (download + playback followed by playback only load), effect on battery is similar to that of idling thereby enabling battery to recover charge [17], [18] (we show this later in Section IV-E). We term this as “download idle”. Note that extensive analysis of charge recovery due to idling is presented in [17] using the analytical Rakhmatov Vruthula (RV) rechargeable lithium ion battery model and authors in [18] have shown the ability of battery to recover charge due to idling using measurements on a commercially available lithium ion battery. (c) Varying video bit rate: As bit rate determines the amount of data that needs to be downloaded, bit rate can be varied in a manner that minimizes amount of data to be downloaded. This offers the opportunity to either further reduce the duration of download and hence introduce download idle periods, or choose lower download rates and less power intensive modes reducing the load imposed on the battery.

While maximizing battery lifetime, we need to also ensure user experience. Consequently, our approach needs to ensure that (1) the video download rate variation, including periods of idling, is done in a way that does not lead to buffer overflow or underflow (stalling of video playback), so that user experience is not affected; (2) the base station reconfiguration is done taking into account the wireless channel condition (estimated using Signal to Noise Ratio - SNR) so that a desired bit error

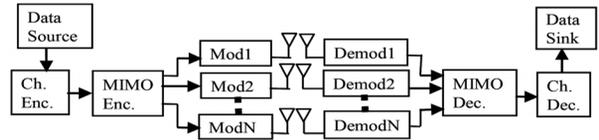


Fig. 1. MIMO transmitter and receiver.

TABLE I  
MIMO TRANSMITTER PARAMETERS

Channel Coding Rate ( $CR$ )	1, 2/3, 1/2, 1/3
MIMO Encoding ( $MIMO_{Enc}$ )	STM, STBC
Modulation Schemes ( $Mod$ )	Binary Phase Shift Keying (BPSK), Quadrature Amplitude Modulation (QAM) - 4QAM, 16QAM, 64QAM
Number of Antennas ( $N_T$ )	1, 2, 3, 4

TABLE II  
MIMO RECEIVER PARAMETERS

Number of Antennas ( $N_R$ )	1, 2, 3, 4
MIMO Decoding ( $MIMO_{Dec}$ )	Zero Forcing (ZF), K-Best
Channel Decoding ( $Ch_{Dec}$ )	Viterbi Decoding, Turbo Decoding

rate (BER) (hence PSNR [19], [20], and video quality) is satisfied; and (3) when additional video bit rate adaptation is done, a minimum video quality is satisfied in a way that increases the overall Video Experience Longevity.

### B. Download Rate Adaptation and Base Station Reconfiguration

In this section, we will first describe RF and baseband processing components of base station and mobile device, and their effects on power consumed. Subsequently we will discuss ways download rate can be varied and transceiver be reconfigured to reduce battery load. Note we sometimes refer as baseband components both RF antenna chains and baseband components.

Fig. 1 shows a MIMO transmitter and receiver. The transmitter consists of channel encoder, MIMO encoder, and set of antennas each with an associated modulator. The receiver consists of antennas, demodulator, MIMO decoder and channel decoder. Tables I and II list some of the possible configuration choices that can be used for MIMO transmitter and receiver. The set of all possible combinations of transmitter and receiver baseband components constitutes the configuration spaces of base station and mobile device respectively. Henceforth we will refer to the combination of transmitter - receiver antennas, channel encoding rate, MIMO encoding, modulation, MIMO and channel decoding algorithms as the *transceiver mode* selected.

Among all the MIMO receiver baseband components, the antenna RF chain is the most power intensive, and the battery load can increase significantly with increase in number of antennas. We consider two MIMO decoding algorithms, Zero Forcing (ZF) and K-Best, both of whose power consumption depends on the number of antennas and modulation scheme used; however, ZF is more power efficient but provides less BER performance than K-Best. Note the power consumed by demodulation is included in MIMO decoding, as demodulation is performed as part of MIMO decoding. Finally, power consumed by channel

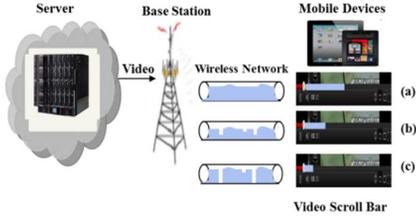


Fig. 2. Video download using different rates.

TABLE III  
EXAMPLES OF MODES WITH DIFFERENT DOWNLOAD RATES

Mode A	CR: 1/2, STM, BPSK, 2X2, ZF, Viterbi
Mode B	CR: 1/2, STBC, 4QAM, 2X1, ZF, Viterbi
Mode C	CR: 1/2, STM, 4QAM, 4X4, K-Best, Viterbi
Mode D	CR: 1/2, STM, 4QAM, 2X2, ZF, Viterbi

decoding depends on the algorithm used. Viterbi decoding consumes less power than Turbo decoding, but also has a lower BER performance than Turbo [21]. The battery load of a receiver configuration can be estimated by adding the power consumptions of the individual receiver components as elaborated in Section III-B.

Fig. 2 shows typical video download scenarios from the video server through the base station to the mobile device over the wireless network. The pipes are representative of the wireless network. The height and shape of the contents of the pipe depict the amount and flow of video data. The red portion on the scroll bar indicates the portion of downloaded video that has been viewed and the blue portion indicates the buffered portion. Fig. 2(a) depicts the scenario wherein the video is downloaded as fast as possible (as is the case with HTTP Progressive Download) depicted by the near fullness of the pipe and buffer. This may require the highest download rate possible under the given channel condition and BER value. Multiple transceiver modes may satisfy the required download rate under a given channel condition (SNR) and BER value. Some of these modes may actually increase the power consumption in the base station, but will reduce the mobile battery load. For example, the two modes A and B listed in Table III result in the same download rate. For the given SNR, mode B increases the power consumed by the base station as it uses 4QAM modulation scheme which consumes more power than BPSK used in mode A. However, mode B will reduce battery load, as only one receiver antenna is used as opposed to two antennas used in mode A. Note that the reduction in battery load due to reduction in receiver antennas far outweighs any increase in battery load due to higher order demodulation. There may also exist certain modes that reduce mobile battery load without increasing power consumption at the base station. For example, if channel condition improves, for the same download rate, it may be possible to reconfigure receiver to use ZF decoding instead of K-Best if BER requirement is met. Hence even when high download rate is required, it may be possible to choose a transceiver configuration which reduces battery load.

The opportunities for finding battery efficient modes can be increased if the required download rate can be reduced. As shown in Fig. 2(b), using the elasticity of the buffer, it is possible to reduce the download rate (depicted by dips in

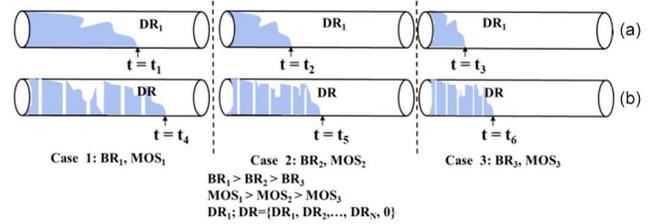


Fig. 3. Adaptive bit rate streaming with different download rates.

the pipe) which results in lesser buffered data (smaller blue portion), as long as there is no buffer underflow. For instance, consider modes C and D in Table III. If the download rate needed is reduced by half, given the same channel condition and BER requirement, mode D can be used instead of mode C. Reconfiguring to mode D will significantly reduce the battery load, as it uses less number of antennas and less power intensive ZF MIMO decoding.

When buffer levels permit, download rate can be reduced to zero. Download idling reduces the battery load to just the playback load, thereby enabling battery to recover charge. Note that the idling will deplete the buffer (shown in Fig. 2(c) as gaps in the pipe and smallest blue portion on the scroll bar), and hence can be done if no buffer underflow can be ensured.

### C. Video Bit Rate Adaptation

In this section, we will elaborate on how video bit rate adaptation affects battery lifetime and video quality. We pictorially represent adaptive bit rate video streaming in Fig. 3. As in Fig. 2, the pipes are representation of wireless network; height and shape of contents indicate the amount and flow of video data across time; to conserve space, we have omitted the server, base station and mobile device. Cases 1, 2 and 3 in Fig. 3 illustrate the effect of using bit rates  $BR_1$ ,  $BR_2$  and  $BR_3$  with associated Mean Opinion Score (MOS) values  $MOS_1$ ,  $MOS_2$  and  $MOS_3$ , while Figs. 3(a) and (b) show the effect of using single download rate,  $DR_1$ , and a set of download rates  $DR$ , on the amount and flow of data in the pipe. Note that the download rates in the set are listed in descending order.

From Fig. 3(a), we draw the following observations. (1) When highest download rate  $DR_1$  and bit rate  $BR_1$  are used, as in case 1, the battery load is highest because the download duration,  $t_1$  is longer than  $t_2$  and  $t_3$ . Using lower bit rate (cases 2 and 3) reduces the amount of data to be downloaded, and hence duration of download ( $t_3 < t_2 < t_1$ ) and battery load. Case 1 of Fig. 3(b) illustrates the proposed video download rate (and mode) adaptation techniques which use a combination of high and low download rates including download idle from the set  $DR$ . As elaborated in the previous subsection the combination of higher, lower download rates and idling offers the potential to reduce battery load. Additionally, using lower bit rates as in cases 2 and 3 of Fig. 3(b) reduces battery load and the reduced download duration ( $t_6 < t_5 < t_4$ ) may allow choosing a more battery efficient combination of download rates (and modes), for instance, introducing more periods of idling. Therefore, we can infer that bit rate adaptation potentially furthers the battery savings due to download rate and mode adaptation but at the expense of video quality.

### III. BATTERY EFFICIENT DOWNLOAD RATE AND MODE SELECTION

In this section, we will assume fixed video bit rate, and formulate the optimization problem of adapting video download rate and corresponding transceiver configuration to maximize battery life. We then present an algorithm, MoDS that solves the problem using an optimization solver.

#### A. Download Rate and Mode Selection Problem Definition

The objective of download rate and mode selection is maximization of battery lifetime during video download subject to download rate and application BER constraints. Video download session consists of several download epochs requiring download rate and mode selection in every download epoch. As battery lifetime is a cumulative result of several such selections and their effect on battery level, we split the optimization problem in to sub-problems and solve it in each download epoch in order to make it tractable. Each sub-problem defined in (1) below consists of selecting an optimal mode  $\mathbf{M}$  for the download epoch under consideration such that battery level  $Bat_{Lev}$  (function of mode parameters listed in Tables I and II) is maximized while download rate  $DR$  constraint upper bounded by  $DR^{Max}$  and lower bounded by  $DR^{Min}$ , and BER constraint upper bounded by  $BER_{App}$ , are satisfied. The sub-problems though seem independent, are connected with each other as the download rate selected in current epoch changes the buffer level which in turn affects the download rate selection in the subsequent epoch.

$$\begin{aligned} & \max Bat_{Lev}(\mathbf{M}) \\ & \text{s.t. } DR^{Min} \leq DR(\mathbf{M}) \leq DR^{Max} \\ & \quad BER(SNR, \mathbf{M}) \leq BER_{App} \end{aligned} \quad (1)$$

The  $DR^{Max}$  and  $DR^{Min}$  values, which will be defined later in this section, ensure that buffer does not overflow or underflow respectively. The application BER value  $BER_{App}$  ensures that video quality (PSNR) is maintained at desired level. Note that it has been shown in [19] and [20] that BER below  $3 \cdot 10^{-5}$  results in PSNR levels greater than 37 dB (corresponding to MOS value of 5 [19]) thereby ensuring high video quality for videos with different space-time characteristics. Hence, choosing  $BER_{App}$  value lesser than  $3 \cdot 10^{-5}$  will ensure that PSNR of the received videos will be greater than 37 dB.

It should be noted that (1) may not have a feasible solution always. When no mode satisfies  $BER_{App}$ , then download idle ( $DR(\mathbf{M}) = 0$ ) is chosen. This may be at the expense of buffer underflow if  $DR^{Min}$  is greater than zero. In case the  $DR^{Min}$  constraint is violated, mode which gives highest download rate (lower than  $DR^{Min}$ ) and satisfies  $BER_{App}$  is chosen leading to buffer underflow. On the other hand when  $DR^{Max}$  is violated, download idle is chosen to avoid buffer overflow. Having defined the download rate and mode selection problem, we will next discuss the objective and constraint functions.

#### B. Modeling of Objective and Constraint Functions

Each download epoch involves video download and simultaneous playback. The RV lithium ion battery model [17], [22] used to estimate the battery level given in (2) is characterized

by two parameters, namely,  $\alpha$  which is the battery capacity and  $\beta$ , a function of ion diffusion coefficient, is the measure of battery nonlinearity. The second term in (2) represents the ratio of total charge consumed in time  $T$  or equivalently in  $E$  download epochs due to variable load  $I$  and the total charge present in the fully charged battery. The charge consumed in each download epoch  $i$  is the sum of the linear term (first term in summation over  $E$ ) and the summation of nonlinear terms (second term in summation over  $E$ ) with summation index  $m$ . The summation of nonlinear terms is a function of  $\beta$  and accounts for the nonlinearity in diffusion and hence charge recovery when  $I_i < I_{i-1}$ . Note that our proposed techniques are not battery model specific and can be used with any model that gives an estimate of battery level in response to battery load.

$$\begin{aligned} Bat_{Lev} &= 1 - \frac{1}{\alpha} \sum_{i=1}^E I_{i-1} \\ & \quad \times \left[ (t_i - t_{i-1}) + 2 \sum_{m=1}^{m=10} \frac{e^{-\beta^2 m^2 (T-t_i)} - e^{-\beta^2 m^2 (T-t_{i-1})}}{\beta^2 m^2} \right] \end{aligned} \quad (2)$$

Maximization of  $Bat_{Lev}$  is equivalent to minimizing numerator of second term in (2) which represents the battery charge consumed due to battery load  $I$  in time  $T$ . Further, as charge consumed is estimated in each download epoch of duration  $D_{Period}$ , which we assume is a constant, maximization of  $Bat_{Lev}$  is equivalent to minimizing battery load  $I$  in each download epoch. As each download epoch involves simultaneous download and playback,  $I$  is given by

$$I = I_{Download} + I_{Playback} \quad (3)$$

where  $I_{Playback}$  is the battery load due to video decoder and display used for playback. While the playback load may vary depending on the resolution of the video, for download epochs of the same video session, it is fair to treat it as constant. Hence maximization of  $Bat_{Lev}$  is equivalent to minimizing battery load  $I_{Download}$  imposed by the mode  $\mathbf{M}$  during download subject to the download rate and BER constraints in (4).  $I_{Download}$  is given by (5)

$$\begin{aligned} & \min I_{Download}(\mathbf{M}) \\ & \text{s.t. } DR^{Min} \leq DR(\mathbf{M}) \leq DR^{Max} \\ & \quad BER(SNR, \mathbf{M}) \leq BER_{App} \\ & \quad I_{Download} = P_{Download}/V_{Bat} \end{aligned} \quad (4)$$

where  $V_{Bat}$  is the battery voltage; we assume that it is constant during discharge. Download power  $P_{Download}$  given by (6) consists of four components, namely power due to RF chain ( $P_{RF-Chain}$ ), MIMO decoding ( $P_{MIMO-Dec}$ ), channel decoding ( $P_{Ch-Dec}$ ) and baseband processing ( $P_{Baseband}$ ).

$$\begin{aligned} P_{Download} &= P_{RF-Chain} + P_{MIMO-Dec} \\ & \quad + P_{Ch-Dec} + P_{Baseband} \end{aligned} \quad (6)$$

$P_{RF-Chain}$  depends on  $N_R$  and system bandwidth  $BW$ . It is determined using (7) obtained from relations in [6] [23].

$$P_{RF-Chain} = (1.8 \cdot 10^{-8} BW + 0.061) N_R + 0.1 \quad (7)$$

$P_{MIMO-Dec}$  depends on MIMO encoding rate  $MIMO_{Enc}$ , number of antennas, algorithm chosen (ZF or K-Best) and modulation scheme used.  $MIMO_{Enc}$  given by (8) and (9), at the bottom of the page, is dependent on the type of MIMO encoding (STM or STBC) used.  $P_{MIMO-Dec}$  is estimated using (10)–(13), at the bottom of the page, by calculating number of search steps [6] required to decode a symbol and determining number of parallel search engines [24] required to execute the steps. We consider only Viterbi channel decoding algorithm in this work;  $P_{Ch-Dec}$  estimate is obtained from [25].  $P_{Baseband}$  is given by (14) [6].

$$P_{Baseband} = 1.62 \cdot 10^{-9} N_R BW \quad (14)$$

The download rate  $DR$  given by (15) forms the first constraint function and is calculated using the specifications in 3GPP LTE standard [26]

$$DR = RB \cdot SUB_C \cdot TS \cdot OFDM_{Sym} \cdot Mod \cdot CR \cdot MIMO_{Enc} \cdot T_{Frame}^{-1} \quad (15)$$

where  $RB$  represents the number of resource blocks associated with  $BW$ .  $SUB_C$  is the number of subcarriers used in each resource block.  $TS$  is the number of slots used to transmit  $OFDM_{Sym}$  number of Orthogonal Frequency Division Multiplexing (OFDM) symbols.  $T_{Frame}$  is the duration of 3GPP LTE frame.

The upper bound  $DR^{Max}$  given by (16) is calculated using video buffer size  $Bu f_{Size}$ , amount of data buffered  $Bu f_{Avail}$  and duration of download epoch  $D_{Period}$ .

$$DR^{Max} = (Bu f_{Size} - Bu f_{Avail}) / D_{Period} \quad (16)$$

Playback time  $PBT$  available is calculated using  $Bu f_{Avail}$  and video bit rate  $V_{BR}$  as shown in (17).

$$PBT = Bu f_{Avail} / V_{BR} \quad (17)$$

The lower bound on  $DR$ ,  $DR^{Min}$  given by (18) is calculated using  $PBT$ ,  $V_{BR}$  and minimum buffer value,  $Bu f_{Min}$  chosen to avoid stalling. It should be noted that the lower bound for  $Bu f_{Min}$  is  $D_{Period}$  in which case the  $PBT$  will at least be  $D_{Period}$ . However, this might stall video when channel conditions do not permit minimum download rate  $DR^{Min}$ , hence

$Bu f_{Min}$  greater than  $D_{Period}$  will increase  $PBT$  and allow idling while avoiding stalling.

$$DR^{Min} = \begin{cases} 0, & PBT > Bu f_{Min} \\ V_{BR} + \frac{V_{BR}(|PBT| - PBT + Bu f_{Min})}{D_{Period}}, & PBT \leq Bu f_{Min} \end{cases} \quad (18)$$

The second constraint in terms of  $BER_{App}$  ensures that mode selected does not lead to unacceptable BER and hence adversely impact video quality. We use a BER–SNR look up table (LUT) (Section IV-A) in lieu of the BER constraint function in the optimization framework. The BER–SNR LUT lists the BER values for different transceiver configurations under different channel (SNR) conditions.

From (2) to (18), it is evident that the objective and constraint functions are nonlinear making mode and download rate selection a nonlinear constrained optimization (minimization) problem. In the next subsection we will present a solution to this problem.

### C. Mode and Download Rate Selection (MoDS) Algorithm

In this section we will describe in detail the MoDS algorithm developed to search the transceiver configuration space (Tables I and II) for the mode that minimizes the battery load  $I$  subject to download rate and BER constraints.

As power calculation functions for MIMO decoding given by (10) to (13) are different for different MIMO encoding schemes and MIMO decoding algorithms, mode selection in each download epoch needs to be carried out separately for each MIMO encoding scheme and decoding algorithm. This implies that  $MIMO_{Enc}$  and  $MIMO_{Dec}$  parameters cannot be part of the transceiver mode search space. On the same line of reasoning,  $Ch_{Dec}$  cannot be used as an optimization parameter. Hence, we split the transceiver configuration space CS in to two spaces as shown in Fig. 4: the outer space  $OS$  consisting of parameters  $MIMO_{Enc}$ ,  $MIMO_{Dec}$  and  $Ch_{Dec}$ , and inner space  $IS$  consisting of parameters  $CR$ ,  $Mod$ ,  $N_T$  and  $N_R$ . The BER–SNR LUT used instead of BER constraint function requires the BER constraint to be evaluated for each mode outside the optimization framework. Having made the above two modifications to the problem stated in (4), the basic working principle of MoDS algorithm is pictorially shown in

$$MIMO_{Enc-STM} = N_R \quad (8)$$

$$MIMO_{Enc-STBC} = \begin{cases} N_R, & (N_T/N_R) \geq N_T \\ (N_R - 1) N_T/N_R, & (N_T/N_R) < N_T \end{cases} \quad (9)$$

$$P_{MIMO-Dec-K-Best}^{STM} = 10^{-4} [MIMO_{Enc-STM} (0.5N_T^2 + 1.5N_T) + 3.1N_T^2 Mod^{2.5} + 0.8N_T Mod^{3.5} + 1.5N_T Mod] \quad (10)$$

$$P_{MIMO-Dec-ZF}^{STM} = 10^{-4} [MIMO_{Enc-STM} (N_T + 0.3N_T^2) + 0.13N_T^2 + 0.06N_T^3] \quad (11)$$

$$P_{MIMO-Dec-K-Best}^{STBC} = 10^{-4} [3.1N_T N_R + 4.1N_R MIMO_{Enc-STBC}^2 + N_T Mod (1.5 + 0.8Mod^{2.5} + 6.2Mod^{1.5} MIMO_{Enc-STBC})] \quad (12)$$

$$P_{MIMO-Dec-ZF}^{STBC} = 10^{-4} [1.9N_T N_R + 0.25N_R MIMO_{Enc-STBC} + MIMO_{Enc-STBC}^2 (0.5 + 2.3N_R + 0.5MIMO_{Enc-STBC})] \quad (13)$$

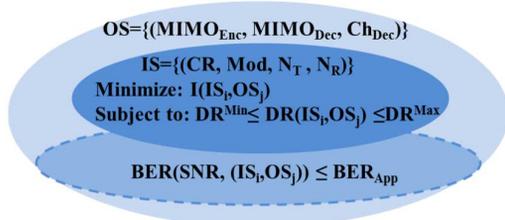


Fig. 4. Splitting of configuration space and optimization problem.

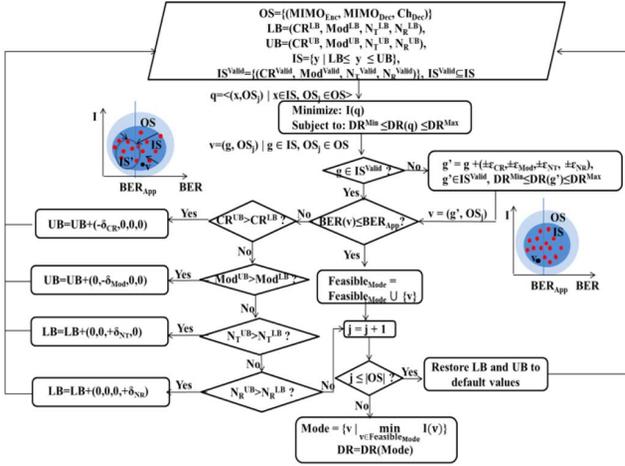


Fig. 5. Mode and download rate selection (MoDS) algorithm.

Fig. 4. For a given point in outer space,  $OS_j$ , MoDS searches the inner space for the mode  $(IS_i, OS_j)$  that minimizes battery load and satisfies download rate. Subsequently the BER constraint is evaluated as shown in Fig. 4. This process is repeated till the entire  $OS$  is explored resulting in battery efficient mode that satisfies the constraints in (4).

The outer space  $OS$ , the inner space  $IS$ , upper bound  $UB$  and lower bound  $LB$  representing the maximum and minimum values possible for the elements of inner space, and set of valid inner space points  $IS^{Valid}$  form the inputs to the MoDS algorithm shown in Fig. 5. Given an outer space point, the nonlinear optimization solver, 'nlopt'<sup>2</sup> is used to determine the mode that minimizes the battery load  $I$  and satisfies the download rate constraints. It should be noted that when  $DR^{Min}$  is zero, download idling ( $DR(\mathbf{M}) = 0$ ) is chosen as this minimizes the battery load  $I$ . If the mode does not belong to  $IS^{Valid}$ , it is rounded off to the nearest valid mode  $v$  by adding  $\varepsilon$  such that the resulting mode does not violate the download constraint. The BER value of mode  $v$  is obtained from the BER-SNR LUT. As pictorially shown, if the BER value of mode  $v$  lies to the left of  $BER_{App}$ , then mode  $v$  is added to the set  $FeasibleMode$  as the battery efficient mode for the chosen point  $OS_j$  of outer configuration space. When BER value lies to the right of  $BER_{App}$ , the inner space is constrained to  $IS'$  by lowering and increasing upper and lower bounds respectively; thereby eliminating modes that do not meet the BER requirement. The upper bound is shifted to lower points by first gradually reducing  $CR$  and then  $Mod$  to lower values. Lowering  $CR$

and  $Mod$  values constricts the configuration space to modes with lower  $CR$ ,  $Mod$  and BER values, thereby increasing the chances of finding mode that satisfies the BER requirement. If BER requirement is not met even at the lowest value of  $CR$  and  $Mod$ , in the final iteration, the lower bound is shifted to higher points by gradually increasing the number of antennas,  $N_T$  and  $N_R$ . As increasing  $N_T$  and  $N_R$  values will lead to selection of power intensive modes, it is done in the final iteration. The mode selected in the final iteration is the battery efficient mode corresponding to the chosen  $OS$  point  $OS_j$  and is added to the set of feasible modes,  $FeasibleMode$ . This process is repeated till the outer space is completely explored and then the most battery efficient mode  $\mathbf{M}$  is chosen from  $FeasibleMode$ . The corresponding download rate  $DR(\mathbf{M})$  is the chosen rate for the ensuing download epoch. The computational complexity of MoDS algorithm which iteratively searches the  $IS$  and  $OS$  for battery efficient mode is presented in online Appendix A.

The overall framework for information and control data exchange between base station and mobile device, mode selection and reconfiguration during battery efficient video download is described in detail in the online Appendix B. As elaborated in online Appendix B, additional data transmitted for conveying buffer levels to base station is nominal—a byte resulting in 1.14 mW of power consumption [27]. On the other hand, receiving information from the BS about mode selected requires 8 bytes, and results in about 2.22uW of power consumption when the mode  $(1 \times 1, \text{BPSK}, CR = 1, \text{ZF})$  is used. In addition to the power consumed due to information exchange, during mode reconfiguration at the mobile device, a change in the number of antennas used in the previous mode to the current mode results in RF component switching power of 100mW per antenna and switching time of 5us [23].

#### IV. SIMULATION FRAMEWORK AND RESULTS

In this section, we describe the simulation framework developed and experimental results obtained by using our proposed battery aware video download technique MoDS, and compare with results obtained using conventional HTTP Progressive Download (HTTP-PD) as well as the EERA technique [9] discussed earlier in Section I-A.

We have developed a very modular and flexible MATLAB based simulation framework to estimate battery consumption and assess user experience during video download and playback. The simulation framework consists of power, battery, BER and user experience models, and allows us to implement and assess different video download techniques to download video sequences under varying channel conditions and video quality requirements. We briefly describe the models followed by discussion of the framework integrated with the models and various video download techniques.

##### A. Power and Battery Models

The power model is used to estimate the power consumed in the mobile device due to video download and playback. As elaborated in Section III-B, download power consists of four components namely,  $P_{RF-Chain}$ ,  $P_{MIMO-Dec}$ ,  $P_{Ch-Dec}$ , and  $P_{Baseband}$  and is modeled using (6) to (14) which are in turn based on measurements made on ASIC implementations of the respective blocks. Similarly, playback power is estimated using

<sup>2</sup>"The NLOpt nonlinear-optimization package," [Online]. Available: <http://ab-initio.mit.edu/wiki/index.php/NLOpt>

TABLE IV  
BER MODEL SIMULATION PARAMETERS

Channel Model	Spatial Channel Model (SCM)–Case II, Vehicular A
Carrier Bandwidth	5MHz
SNR (dB)	0 - 40
FFT Size	512 points
Channel Coding	1, 2/3, 1/2, 1/3
Modulation Schemes	BPSK, 4QAM, 16QAM, 64QAM
Antenna Configurations	STM: 1x1, 2x2, 3x3, 4x4; STBC: 2x1, 2x2
MIMO Decoding	Zero Forcing, K-Best
Channel Decoding	Viterbi Decoding

measurements from video decoder<sup>3</sup> and mobile device display [28]. Note that since the overall device power is the sum of the power consumed by the different components of download and playback power, the power model can be adapted to a different device by modeling and substituting for the components that are different. For example, if the new device uses a different implementation of say, the baseband, then (14) will need to be updated with the appropriate model for the new baseband implementation.

Next we will discuss the RV rechargeable lithium ion battery model [17] which takes the output of power model to estimate the battery level. As elaborated in Section III-B, (2) is used to estimate the battery level given the magnitude and duration of battery load which is obtained using (5). It should be noted that the RV model can be used to estimate battery levels of rechargeable lithium ion batteries with different battery voltage and capacities (battery specific parameters required by the battery model are obtained by running discharge tests with constant battery load [17], [22]). This implies that the proposed video download techniques can be evaluated on mobile devices of varying form factors and battery capacities. Moreover, the proposed techniques are not battery model specific and can be used with any model that gives the battery level in response to the battery load.

### B. BER Model

As elaborated in Section III-A, given the channel condition, BER values of transceiver modes are required by MoDS to ensure that desired BER is maintained. We have developed a BER model by using MATLAB to simulate different modes under different channel (SNR) conditions and obtain BER values which are stored in the BER-SNR look up table (LUT). The simulation parameters used to generate the BER-SNR LUT are listed in Table IV, including the modulation schemes, antenna configurations, MIMO decoding and channel decoding algorithms. Using the specified channel model,<sup>4</sup> carrier bandwidth and FFT size, the SNR is varied to obtain the BER values of all the modes constituting the reconfiguration space.

### C. User Experience Model

User experience for video download is primarily determined by the video quality and any stalling in video playback. Video quality of received video is affected by adaptation of video char-

acteristics such as video resolution, bit rate, and frame rate, and any packet losses that may occur due to BER during transmission. Since the original video resolution, bit rate, and frame rate are not changed by HTTP-PD, EERA or MoDS, the video quality is not affected. Further, by choosing a very low application BER,  $BER_{App} (< 3 \cdot 10^{-5}, [19], [20])$  and carrying out mode selection so as to meet the desired application BER requirements ( $10^{-6}$  in our experiments), no loss in PSNR and thereby video quality due to packet loss is ensured. Hence, the only user experience impairment in the case of video download techniques to be compared here is stalling. Consequently, the user experience model uses the stalling–MOS relationship developed in [29] to map the number and duration of stalling events recorded (by the simulation framework developed) to MOS scores.

### D. Simulation Framework

The simulation framework for video download techniques consists of power, battery and BER models along with the video download algorithm/technique and simulation time counter. When video download is initiated, the simulation time counter is started. The simulation step is equal to the download epoch duration,  $D_{Period}$  and in our experiments it is fixed at 2 seconds, though it can be made longer or shorter. In case of the proposed battery aware video download technique, the MoDS algorithm determines the battery efficient mode and download rate depending on the current buffer level and channel condition (SNR) for each simulation step.

For the energy efficient rate adaptation technique [9], the EERA algorithm determines the energy efficient mode and the download rate depending on the video bit rate and channel condition. While simulating the conventional HTTP-PD technique implemented using the download mechanism (consisting of initial phase and throttle phase) in [30], we fix the desired download rate to maximum value determined using (16) in the initial phase and to that which will allow a constant average rate of 1.25 times the video encoding rate when data is sent in bursts of 64KB in the throttle phase. We select the mode that satisfies the download rate and BER requirement and if no such mode exists, then the mode that gives highest download rate (lower than the desired rate) at the given SNR and BER value is chosen. For all the aforementioned techniques, the BER-SNR LUT (Section IV-B) is used to ascertain whether the BER of the selected mode satisfies the  $BER_{App}$ . The download power and playback power are calculated using the power model (Section IV-A) and the resulting battery load is input to the battery model (Section IV-A) to estimate the battery level. It should be noted that for MoDS, the power consumed due to information (1.14mW, see Section III-C) and control data exchange (2.2uW, see Section III-C) and RF component switching power (100mW for 5us, see Section III-C) is also added to the download power and playback power before determining battery load. The simulation framework also records the number and duration of stalls (buffer underflow/overflow) if any and uses the user experience model (Section IV-C) to quantify the user experience in terms of MOS value.

If the viewer switches to a new video or current video is completely downloaded, new video download begins. This continues till battery is completely drained. The simulation counter

<sup>3</sup>[Online]. Available: [http://www.privateline.com/imode/MPEG\\_4\\_CODEC.pdf](http://www.privateline.com/imode/MPEG_4_CODEC.pdf)

<sup>4</sup>[Online]. Available: [http://www.etsi.org/deliver/etsi\\_tr/125900\\_125999/125996/11.00.00\\_60/](http://www.etsi.org/deliver/etsi_tr/125900_125999/125996/11.00.00_60/)

TABLE V  
VIDEO DOWNLOAD SIMULATION PARAMETERS

Video Characteristics	Video Bit Rate $V_{BR} = 4.12\text{Mb/s}$
	Video Sequence 1: {184s, 226s, 195s, 197s, 226s, 257s, 274s, 231s, 200s, 224s, 298s, 235s, 285s, 198s, 233s, 291s, 298s, 236s, 221s, 205s}
Client Characteristics	Video Buffer Size $Bu_{fsize} = 300\text{s}$
	Playback Load (Decoder + Display) $I_{playback} = 34\text{mA}$
	RF Component Switching Power = 100mW [23] RF Component Switching Time = 5us [23]
User Characteristics	Constant SR=1 Variable SR: {0.5, 0.1, 0.97, 0.43, 0.27, 0.93, 0.22, 0.19, 0.28, 0.67, 0.6, 0.39, 0.93, 0.82, 0.05, 0.82, 0.38, 0.45, 0.01, 0.28}
Algorithm Parameters	Minimum Buffer Level $Bu_{fmin} = 10\text{s}$
SNR (dB)	High: 40, Low: 9, Variable: In the range 0 – 40
BER	Application BER $BER_{App} = 10^{-6}$

at this instant gives the battery lifetime for downloading and watching the chosen video sequence under simulated channel conditions and quality requirements. It should be noted that while battery lifetime is a cumulative result of multiple video download and viewing sessions, user experience is assessed for each session.

Table V lists the simulation parameters used in our experiments. Video characteristics specify the video bit rate used to encode the video and the sequence of videos watched. Client characteristics enumerate buffer size, playback current, switching power and switching time specifications of RF components (antennas). In our experiments, we also consider the increasingly prevalent “video snacking” user viewing pattern wherein the user begins to watch a video and then switches to a new video without finishing the current video. This pattern is modeled by randomly generated values of snacking ratio (SR) which is the ratio of the duration of the video viewed by the user to the actual duration of the video. In other words, each snacking ratio value specified in user characteristics indicates how much of the corresponding video in the video sequence the user will watch. The value of the algorithm parameter, minimum buffer level,  $Bu_{fmin}$  used to determine  $DR^{Min}$  in (18) is also listed in Table V. Table V lists the channel conditions based on measurements of cellular network (high, low and variable) and the application BER requirement that ensures high quality (Section III-A, [19], [20]). Note the resolution of temporal variation in channel condition is assumed to be comparable to the simulation step.

### E. Experimental Results

Next, we present results obtained by simulating video download under different channel conditions and snacking ratios (and low BER/high video quality requirement) shown in Table V. Figs. 6 and 7 show the effects on download rate, battery load, level, and lifetime while using HTTP Progressive Download (HTTP-PD, shown as red dot-dash line/red bar) [30], the energy efficient rate adaptation technique (EERA, shown as blue dashed line/blue bar) [9] and our proposed battery aware download technique (MoDS, shown as green solid line/green bar).

We will first describe the download characteristics of each of the techniques and then discuss their impact on battery consumption under different snacking ratio values and SNR con-

ditions. HTTP-PD delivers video at maximum download rate in the initial phase followed by constant average download rate in the throttle phase [30] without attempting to choose battery efficient modes in both phases resulting in maximum battery drain during video download. However, the above factors contribute to reduced download duration and extended playback only period after download during which significant charge recovery takes place as battery load is reduced to only playback load. EERA reduces battery drain by selecting energy efficient modes; however it downloads at the video encoding bit rate (4.12 Mb/s) which not only extends the download duration but also does not fill the buffer and thereby can neither vary download rate nor download idle to achieve additional battery savings. On the other hand, MoDS selects modes that maximize battery level and battery load is further reduced by selecting download idling whenever playback time available is greater than  $Bu_{fmin}$  [as in (18)].

We will first examine the scenario when the mobile device is experiencing good network condition (high SNR). The download rates selected while downloading and viewing a single 184s video with  $SR = 1$  (video viewed completely), and the resulting battery load, are shown in Figs. 6(a) and 6(b) respectively. The green solid line shows the effect of MoDS performing download idling. Fig. 6(c) shows the effect on battery level, when the simulation is started with battery level of 0.2. Note that for MoDS, download idle followed by transmission results in alternate fall and rise in load with corresponding rise and fall in battery level clearly indicating that battery recovers charge as a result of idling. HTTP-PD results in maximum battery drain (as explained above) till video download is complete at  $t = 125\text{s}$ . Subsequently, it recovers significant charge during the playback only period lasting about 59s as shown in Fig. 6(c). This explains how HTTP-PD reduces most of the gains achieved by MoDS through selection of battery efficient modes and idling. At the end of video duration (184s), we can see that EERA causes maximum decrease in battery level, followed by HTTP-PD and finally by MoDS, the latter reducing degradation in battery level significantly compared to EERA but only marginally compared to HTTP-PD. On the other hand, if we consider  $SR = 0.5$ , then video download and playback will stop at  $t = 92\text{s}$  indicated by vertical line in Fig. 6(c). In this case, HTTP-PD cannot utilize the playback only period to recover charge, hence it causes maximum battery drain (about 3.6%) followed by EERA and then MoDS.

Figs. 7(a) and 7(b) show the impact on battery lifetime when video sequence 1 is seen with  $SR = 1$  and variable SR respectively and with a starting battery level of 0.2. For  $SR = 1$  [Fig. 7(a)], even though HTTP-PD recovers charge at the end of single video as elaborated above, subsequently, as download progresses, the high download load depletes the battery more during download than that can be recovered during playback period. This widens the gap in battery levels between HTTP-PD and MoDS with the lower download load and idling for MoDS further extending the battery lifetime to result in overall gain of 16%. For EERA, the maximum battery drain for single video continues for subsequent videos resulting in 46% lower battery lifetime compared to MoDS. For variable SR [Fig. 7(b)], HTTP-PD cannot recover charge in the playback only period un-

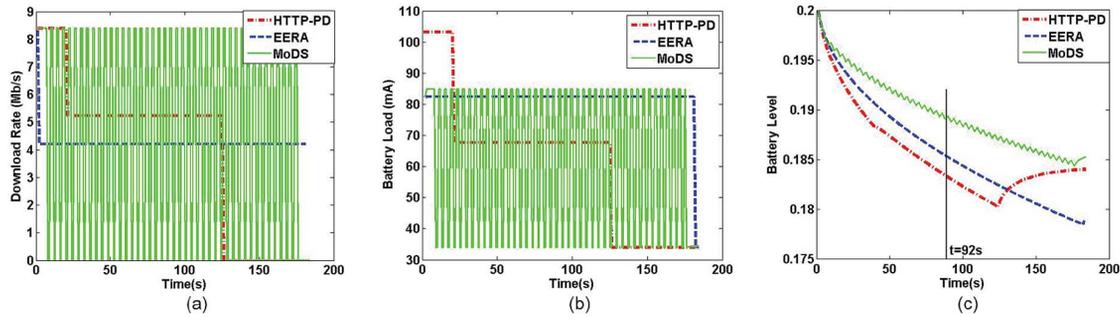


Fig. 6. Effect of downloading and viewing a single 184s video under high SNR conditions on (a) download rate, (b) battery load, and (c) battery level.

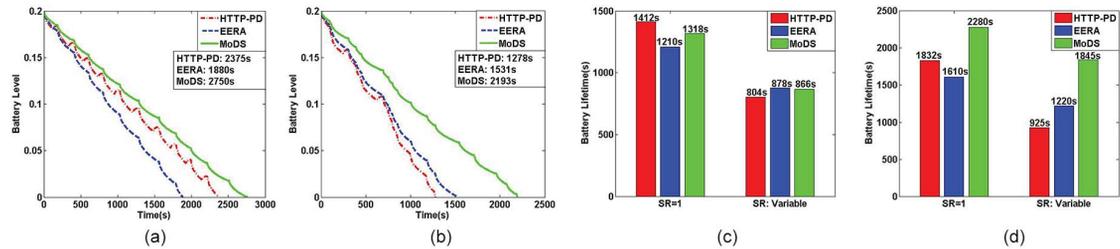


Fig. 7. Effect of downloading and viewing video sequence 1 on battery lifetime under (a) high SNR with SR = 1, (b) high SNR with variable SR, (c) low SNR with SR = 1 and variable SR and (d) variable SNR with SR = 1 and variable SR.

less SR is comparable to 1. On the other hand, as EERA extends download time and does not vary download rate or idle, variable SR has negligible effect on its performance. MoDS which reduces battery load right from the outset, gains about 71.5% and 43% in battery lifetime over HTTP-PD and EERA respectively. As no stalling is recorded for either of the techniques with either SR = 1 or SR = 0.5 or variable SR, user experience is same for HTTP-PD, EERA and MoDS and MOS is 5 [29]. It should be noted that the MOS values are the average of the MOS values of video downloaded and viewed.

Next we will discuss the scenario when mobile device experiences bad channel condition (low SNR). Low SNR condition does not allow filling up the buffer as fast resulting in shorter playback period for HTTP-PD. On the other hand, the selection of modes that minimize battery load by MoDS under low SNR conditions results in reduced download rates (higher download rates require power intensive modes to maintain BER) that not only extend duration of download but also do not allow idling. As can be seen in Fig. 7(c), for MoDS, this results in loss of about 6.6% over HTTP-PD when SR = 1. However, it gains by about 9% in battery lifetime compared to EERA which stalls for 380s (31%) as it attempts to download at video bit rate by selecting battery efficient modes which under low SNR conditions does not allow buffer to fill and avoid stalling. HTTP-PD and MoDS do not result in stalling, hence result in the same user experience (average MOS = 5 [29]). On the other hand, EERA on an average (1210s battery lifetime corresponds to approximately 6 videos of video sequence 1 and 380s stalling corresponds to about 63s of stalling per video) results in MOS score below 2 [29]. From Fig. 7(c), one can see that for variable SR, MoDS gains by 7.7% over HTTP-PD. With no video stalling, MOS = 5 for both HTTP-PD and MoDS. Though EERA gains by 1.4% over MoDS, it stalls for 258s (878s, 9 videos, 29s of stalling per video) resulting in MOS below 2.

Under variable SNR conditions, gains under high SNR offset the loss under low SNR to result in net gain in battery lifetime for MoDS. In this case, the combination of power intensive modes under low SNR and battery inefficient modes under high SNR reduces the gain due to charge recovery for HTTP-PD. It can be seen from Fig. 7(d) that when SR = 1, a gain of 24% over HTTP-PD and 41% over EERA is possible when MoDS is used. HTTP-PD and MoDS achieve MOS = 5 with no stalling whereas EERA results in an average of 6s of stalling per video (1610s, 7 videos, 45s stalling) resulting in a MOS score of about 2 [29]. For variable SR [Fig. 7(d)], the above gains for MoDS are extended to 99% and 51% over HTTP-PD and EERA respectively. As with SR = 1, no stalling results in MOS = 5 for both HTTP-PD and MoDS whereas EERA results in an average of 2.1s stalling per video (1220s, 11 videos, 24s of stalling) resulting in MOS value of 3.5.

Studies conducted in [31] and [32] show that the average video completion rate is as low as 15% on smartphones and slightly higher on Tablets and that 80% of YouTube sessions are less than half of the video duration indicating that video snacking is highly prevalent among users. With reference to these statistics, variable SR values less than 1 is more realistic than constant SR equal to 1. From the above results, it can be seen that MoDS significantly increases battery lifetime in the realistic scenario of variable SR. With respect to SNR, the statistics presented in [33] for signal strength (SNR) experienced by users shows that variable SNR conditions are most prevalent and also that low SNR conditions throughout video download are less likely to occur. Again, the above results show that MoDS performs best under the most prevalent case of variable SNR conditions while the loss or nominal gains under low SNR conditions are less likely to occur.

In the next section, we will present battery aware techniques for DASH video that add to the battery savings achieved by download rate and mode reconfiguration while ensuring minimum desired video quality.

## V. BIT RATE, DOWNLOAD RATE, AND MODE SELECTION

As explained in Section II-C, adapting the video bit rate may offer the opportunity for further battery savings beyond download rate and transceiver mode adaptation. However, bit rate adaptation may also affect video quality. When the mobile device is battery constrained limiting the longevity of watching video, the overall user experience may be enhanced by considering bit rate adaptation to elongate the battery lifetime and hence the viewing experience even with some acceptable degradation in video quality. In this section, we explore the potential additional benefit of video bit rate adaptation, along with download rate and mode adaptation, to increase the battery lifetime and thereby the video viewing experience, while ensuring an acceptable video quality. We first formulate the optimization problem formally, and then present algorithm developed namely BR-MoDS which uses optimization solver to solve the optimization problem. We then extend the formulation to consider battery level while selecting bit rate and present the B<sup>2</sup>R-MoDS algorithm that solves the extended optimization problem. We conclude the section by defining the new Video Experience Longevity metric which quantifies the performance of DASH based techniques in terms of battery lifetime (longevity of video experience) and quality of video experience.

### A. Maximization of Battery Lifetime With Acceptable Quality

The objective of video bit rate, download rate and mode selection is maximization of battery lifetime during adaptive video streaming subject to bit rate, download rate and user experience constraints. In adaptive bit rate streaming, the video is fragmented in to equally sized segments, each segment encoded using the set of discrete bit rates available [4]. A segment download can be viewed as a two tiered process wherein first the bit rate for the segment and subsequently, download rate and mode is selected. It should be noted that the download rate and mode selection may need to be done multiple times during a segment download; in other words each segment may consist of one or more ‘download epochs’ during which download rate and mode selection is carried out. This implies that battery lifetime maximization is achieved at two levels, namely at the segment level and at the download epoch level and hence, we will adopt a two tiered approach towards selecting a battery efficient combination of bit rate, download rate and mode. Selections made at either segment or download epoch level maximize battery level and the cumulative result of these selections maximizes battery lifetime. Therefore, henceforth we will refer to maximization of battery level instead of battery lifetime as the objective of bit rate, download rate and mode selection.

First we will formulate the sub-problem that maximizes battery level by choosing bit rate for each segment subject to bit rate and user experience constraints. We consider segments of duration  $SegTime$  encoded using bit rates belonging to set  $V_{BR-Set}^{Valid}$  lower bounded by  $V_{BR-Min}$  and upper bounded by  $V_{BR-Max}$ . The amount of data downloaded  $SegData$ , for a segment is given by the product of  $SegTime$  and bit rate chosen  $V_{BR}$ . Choosing lower bit rates reduces the amount of data downloaded which in turn reduces battery load and/or duration of load thereby maximizing battery level (as elabo-

rated in Section II-C). However, as bit rate selection affects video quality  $VQ$ , it has to be done in a manner that the  $VQ$  exceeds a certain threshold  $VQ_{Thr}$  in order to ensure user experience. In addition to video quality, maintaining user experience also requires that  $V_{BR}$  does not exceed the network throughput  $NW_{TPut}$  in order to avoid video stalling. Hence bit rate selection to maximize battery level can be viewed as minimizing  $SegData$  subject to bit rate, video quality and network throughput constraints as shown in (19).

$$\begin{aligned} & \min SegTime V_{BR} \\ & s.t. V_{BR-Min} \leq V_{BR} \leq V_{BR-Max} \\ & VQ_{Thr} \leq VQ \\ & V_{BR} \leq NW_{TPut} \end{aligned} \quad (19)$$

Video quality  $VQ$  is measured in terms of average MOS value,  $MOS_{Avg}^{Video}$ .  $MOS_{Avg}^{Video}$  defined in (20) is the average of MOS values corresponding to bit rates of previously downloaded  $N$  segments and the bit rate to be selected using (19) for the current  $N + 1^{th}$  segment. As MOS value corresponding to  $V_{BR-Max}$ ,  $MOS(V_{BR-Max})$  represents maximum video quality, we define the lower bound on video quality,  $VQ_{Thr}$  as  $MOS(V_{BR-Max})$  reduced by the factor  $VQ_{Red}$  which specifies the acceptable loss in video quality due to battery aware DASH techniques.  $VQ_{Thr}$  is given by (21).

$$\begin{aligned} MOS_{Avg}^{Video} &= (MOS_{Seg_1} + \dots + MOS_{Seg_N} \\ &+ MOS_{Seg_{N+1}}(V_{BR}))/N + 1 \end{aligned} \quad (20)$$

$$VQ_{Thr} = VQ_{Red} MOS(V_{BR-Max}), 0 < VQ_{Red} \leq 1 \quad (21)$$

Network throughput  $NW_{TPut}$  given by (22) is the ratio of  $SegData$  and segment download duration  $SegDT$ .  $SegData$  and  $SegDT$  used to estimate  $NW_{TPut}$  corresponds to the  $N^{th}$  segment, that is the network load conditions experienced during the download of the previous segment influences the selection of bit rate of the current segment. It should be noted that  $SegDT$  may be lesser than, equal to or greater than  $SegTime$  depending on network load and channel conditions.

$$NW_{TPut} = SegData/SegDT \quad (22)$$

A feasible solution to (19) may not always exist. In case  $NW_{TPut}$  is lesser than  $V_{BR-Min}$ , then  $V_{BR-Min}$  is selected which may lead to video stalling and violation of  $VQ_{Thr}$ . When both  $NW_{TPut}$  and  $VQ_{Thr}$  constraints cannot be satisfied, bit rate which satisfies the  $NW_{TPut}$  is selected to avoid video stalling and leads to violation of  $VQ_{Thr}$ .

Subsequent to bit rate selection, we will now formulate the download rate and mode selection sub-problem for all the download epochs that constitute the segment download. We use the problem formulation given by (4) and elaborated in Section III-B, except that upper bound on  $DR$  is the amount of segment data that needs to be downloaded and not amount of data needed to fill the buffer [as in (16)] and lower bound ensures that playback time is at least equal to segment time and not minimum buffer level,  $Buf_{Min}$  [as in (18)].  $DR^{Max}$  corresponding to any download epoch in a segment cannot exceed the difference of total segment data,  $SegData$  and

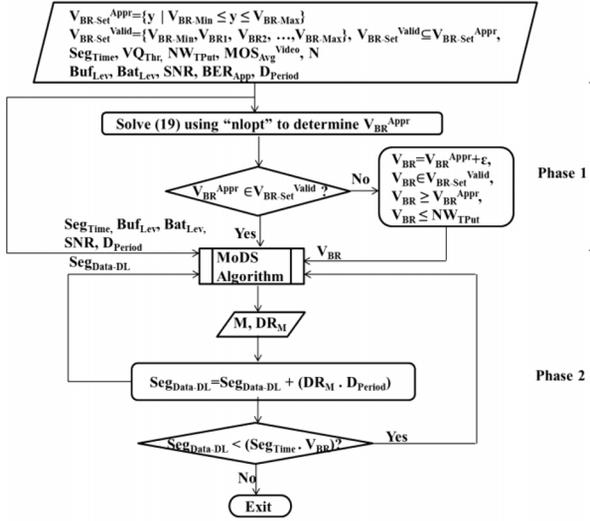


Fig. 8. Bit rate, mode, and download rate selection (BR-MoDS) algorithm.

segment data downloaded so far, the latter being the sum of the products of duration of each download epoch  $D_{Period}$  and download rate  $DR$  chosen for the epoch. On the other hand,  $DR^{Min}$  is zero when the playback time available  $PBT$  exceeds  $SegTime$ . When  $PBT$  available is less than  $SegTime$ ,  $DR^{Min}$  corresponds to the deficit required to increase  $PBT$  to at least  $SegTime$  to ensure that buffer contains enough data to playback the segment without stalling. Hence the bounds on download rate are now defined as shown in (23) and (24).

$$DR^{Max} = (SegTime V_{BR} - \sum_{i=1}^N D_{Period}^i DR^i) / D_{Period}^{N+1} \quad (23)$$

$$DR^{Min} = \begin{cases} 0, & PBT \geq SegTime \\ (SegTime - PBT) V_{BR}, & PBT < SegTime \end{cases} \quad (24)$$

### B. Bit Rate, Mode, and Download Rate Selection (BR-MoDS) Algorithm

In this section we will describe BR-MoDS algorithm that adopts the two tiered problem formulation elaborated in the previous subsection to search the bit rate space and transceiver configuration space (Tables I and II). Fig. 8 shows the inputs, two phases and outputs of BR-MoDS algorithm. As shown in Fig. 8, phase 1 involves selecting bit rate  $V_{BR}^{Appr}$  that minimizes the  $SegData$  given the bit rate, video quality and network throughput constraints. If  $V_{BR}^{Appr}$  does not belong to  $V_{BR}^{Valid}$ , it is rounded off to the nearest higher valid bit rate  $V_{BR}$  by adding  $\epsilon$  such that the resulting bit rate does not violate the network throughput constraints. It should be noted that the rounding off of bit rate does not violate the video quality threshold as a higher bit rate is chosen. The output of phase 1,  $V_{BR}$  along with  $Buf_{lev}$ ,  $Bat_{lev}$ ,  $SNR$  and  $D_{Period}$  form the inputs to MoDS algorithm (Section III-C, Fig. 5). As elaborated in the previous subsection, the bounds on download rate constraint used by the MoDS algorithm are defined by (23) to (24) instead of (16) to (18). The MoDS algorithm is iteratively called, with iterations corresponding to download epochs, till the aggregate of the segment data downloaded is equal to  $SegData$  as shown

in phase 2, Fig. 8. The output of MoDS is the mode  $M$  and download rate  $DR_M$  used in that epoch.

Having discussed in detail the framework and algorithm developed to maximize battery lifetime during DASH streaming, we next discuss an approach to jointly maximize both battery lifetime and video quality.

### C. Joint Maximization of Battery Lifetime and Video Quality

The BR-MoDS algorithm described above selects the minimum (optimal) bit rate that satisfies the video quality and network throughput constraints even though battery level and network conditions may allow selection of higher bit rate as it aims to maximize only battery lifetime and not aggregate video quality. On the other hand, aggregate video quality can be potentially enhanced by choosing a higher video quality threshold  $VQ_{Thr}$  which will result in choosing higher bit rates, but will decrease battery savings. This implies that joint maximization of battery savings and aggregate video quality is required to balance the battery lifetime–video quality tradeoff achieved by bit rate adaptation. However, while bit rate impacts video quality directly, it has an indirect relationship with battery lifetime. Bit rate determines the amount of data to be downloaded, which in turn (along with battery and buffer levels, channel and network load) determines the mode and download rate and hence battery lifetime. This indirect relationship does not lend itself naturally to a joint battery lifetime–aggregate video quality maximization formulation. Hence in this section, we propose a heuristic approach which uses information about battery level and network conditions during bit rate selection to opportunistically maximize both battery lifetime and aggregate video quality.

One possible way of utilizing battery level information during bit rate selection is to scale bit rate with battery level. The basis for this approach is that when battery level is high, battery can support higher drain due to higher bit rates whereas when battery level is low, lower bit rates have to be chosen because higher bit rates will deplete the battery to a greater extent than when battery level is high. However, though the choice of low bit rates when battery level is low will conserve battery and extend video viewing time, it will also result in consistently low video quality and may not meet the video quality constraint. A better approach will be scaling bit rate with the ratio of battery level  $Bat_{lev}$  to the starting battery level  $Bat_{lev-Init}$ . Using the ratio ensures that scaling of bit rate and rate of increase in scaling during a session is lesser when  $Bat_{lev}$  is higher, and much more when  $Bat_{lev}$  is lower. For instance, consider the two cases when battery level reduces by 0.05 and 0.1, the ratio values are 0.95 and 0.9 respectively when  $Bat_{lev-Init}$  is 1 and 0.75 and 0.5 when  $Bat_{lev-Init}$  is 0.2. This results in wider range of bit rates selected during a session when  $Bat_{lev}$  is low with higher bit rates boosting quality and lower bit rates offsetting the drain due to higher bit rates. It should be noted that whenever the bit rate selected exceeds  $NW_{TPut}$ , it is set to  $NW_{TPut}$  in order to avoid buffer underflow. As the bit rate selection stated in (19) selects the minimum bit rate that satisfies the constraints, based on the above observations, we modify the lower bound on bit rate,  $V_{BR-Min}$  to a battery level dependent bit rate  $V_{BR}^{BA}$  given by (25)

$$V_{BR}^{BA} = V_{BR-Min} + Bat_{lev} Bat_{lev-Init}^{-1} (V_{BR-Max} - V_{BR-Min}) \quad (25)$$

This implies that the lower bound on bit rate shifts higher or lower depending on  $Bat_{Lev}$  thereby using battery level information for bit rate selection. The modified bit rate selection problem is same as that stated in (19) except that  $V_{BR-Min}$  is replaced by  $V_{BR}^{BA}$ . The new algorithm termed **Battery Level Aware BR-MoDS**,  $B^2R$ -MoDS is same as BR-MoDS except that phase 1 is modified to reflect the above change. The computational complexity of BR-MoDS and  $B^2R$ -MoDS which use nonlinear optimization solver “*nlopt*” to determine the minimum bit rate  $V_{BR}^{Appr}$  is presented in the online Appendix A.

#### D. Battery Aware Video Streaming—Framework

In our proposed framework, the execution of BR-MoDS/ $B^2R$ -MoDS algorithms is distributed as the bit rate selection is mobile device driven (like any DASH based technique) and download rate and mode selection carried out by MoDS is base station driven. The framework is the same as that elaborated in Online Appendix B except that the bit rate is sent by the mobile device prior to each segment download. Also, the initial information conveyed by mobile device at the beginning of video session consists of  $V_{BR-Max}$ ,  $V_{BR-Min}$ , maximum  $PBT$  possible and also the segment time  $SegTime$ . Subsequently, for each of the download epoch that constitutes the segment download, the information exchange between base station and mobile device is as explained in online Appendix B. However, the buffer status update is used to calculate  $DR^{Max}$  and  $DR^{Min}$  defined in (23) and (24).

#### E. Video Experience Longevity (VEL) Metric

In this section, we develop the Video Experience Longevity (VEL) metric to quantify the performance of the proposed battery aware bit rate adaptation techniques in terms of both the longevity of video experience and the quality of video experience as compared to alternative DASH based techniques. In this paper, for comparison we consider the non-battery aware rate adaptation algorithm proposed in [11] for DASH [34] (termed RA-DASH) and the battery aware rate adaptation technique (termed BaSe-AMy) proposed in [16]. The VEL metric is developed to compare performances of the different techniques for the most demanding scenario when the mobile device continuously downloads and watches videos till the battery gets exhausted. In this scenario, note that the longevity of video experience  $ExpTime$  is the same as battery lifetime  $Bat_{Lifetime}$  minus any stalling time  $StallTime$  during the video sessions, as given by (26) below. However, even in other user scenarios, a DASH technique with higher VEL score than another technique can be considered more efficient in terms of battery lifetime and/or video experience. While modeling of the quality of video experience  $VE$  continues to be an active area of research, in this paper we model  $VE$  as shown in (29) as a weighted sum of video spatial quality measured by the  $MOS_{Avg}^{Total}$  defined in (27) as the average of  $MOS_{Avg}^{Video}$  [defined in (20)] of all the  $K$  videos streamed till the battery dies, and video temporal quality reflected by a term  $NStall_{Norm}$  defined in (28), which measures how free the video experience is from stalls/jitter. The weights  $w_{MOS}$  and  $w_{NStall}$  in (29) reflect relative priority for spatial quality versus stall-free video in determining user experience. Note that we normalize  $NStall_{Norm}$  to 5 in line with MOS score so

we can consider both of them in  $VE$ ; when there is no stalling, the value is 5, while in the extreme case that no video playback is possible at all due to stalling, the value is 0

$$ExpTime = Bat_{Lifetime} - StallTime \quad (26)$$

$$MOS_{Avg}^{Total} = \left( MOS_{Avg}^{Video1} + MOS_{Avg}^{Video2} + \dots + MOS_{Avg}^{VideoK} \right) / K \quad (27)$$

$$NStall_{Norm} = 5ExpTime / Bat_{Lifetime} \quad (28)$$

$$VE = w_{MOS}MOS_{Avg}^{Total} + w_{NStall}NStall_{Norm}, \quad 0 < w_{MOS}, w_{NStall} \leq 1. \quad (29)$$

Next we define the VEL metric in (30) to quantify the joint gain/loss in experience longevity and quality of video experience achieved by the proposed battery aware DASH techniques over DASH techniques used for comparison. The ratio increases (decreases) when there is gain (loss) in experience longevity relative to gain (loss) in video experience.

$$VEL = (1 + \Delta ExpTime) / (1 - \Delta VE) \quad (30)$$

$\Delta ExpTime$  defined in (31) and  $\Delta VE$  defined in (32) represent the gain/loss in experience longevity and video experience respectively achieved by the proposed battery aware DASH (BA-DASH) techniques (BR-MoDS and  $B^2R$ -MoDS) over other DASH techniques (non-battery aware [11] and battery aware [16])

$$\Delta ExpTime = (ExpTime_{BA-DASH} - ExpTime_{DASH}) / ExpTime_{DASH} \quad (31)$$

$$\Delta VE = (VE_{BA-DASH} - VE_{DASH}) / VE_{DASH}. \quad (32)$$

Note that  $\Delta ExpTime$  and  $\Delta VE$  for technique used for comparison (RA-DASH or BaSe-AMy) are zero, implying VEL value of 1. VEL for the proposed techniques can be greater than or lesser than 1 depending on values of  $\Delta ExpTime$  and  $\Delta VE$ . If a proposed technique has VEL greater than 1, it is more efficient than the DASH technique used for comparison in terms of experience longevity and/or video experience.

## VI. SIMULATION FRAMEWORK AND RESULTS

In this section, we describe the simulation framework developed to adaptively stream different video sequences under varying channel and network load conditions and video quality requirements. We will then discuss the experimental results obtained using the proposed battery aware DASH techniques, as compared to using the conventional RA-DASH technique and battery aware rate adaptation technique, BaSe-AMy.

#### A. Simulation Framework

In this section, we will elaborate on the modifications made to the simulation framework developed in Section IV to estimate battery consumption during adaptive bit rate streaming and playback. As rate adaptation techniques for DASH adapt video bit rate under challenging channel conditions and network load, we extend the simulation framework developed in Section IV to simulate varying network load (equivalent to varying number

TABLE VI  
SIMULATION PARAMETERS FOR DASH STREAMING

Video Characteristics	Video Bit Rate $V_{BR-Set}^{Valid}$ (Mb/s): {4.5, 3.75, 3.125, 2.6, 2.17, 1.81, 1.51}
	MOS Values: {4.8, 4.6, 4.49, 4.35, 4.2, 4.02, 3.9}
	Segment time $SegTime = 10s$
	Video Sequence 2: {404s, 1942s, 124s, 360s, 526s, 190s, 757s, 738s, 360s, 255s, 232s, 396s, 181s, 219s, 3198s, 139s, 348s, 408s}
Client Characteristics	Video Buffer Size $Buf_{Size} = 50s$
	Playback Load (Decoder + Display) $I_{Playback} = 34mA$
Video Quality requirements	Quality Threshold $VQ_{Thr} = 4.32$ ( $VQ_{Red} = 0.9$ , 10% degradation from highest MOS value of 4.8)
Network Load	Variable: Peak Throughput = 2.52 - 8.4Mb/s

of users) by modulating the peak throughput available to a particular user while downloading video. Also in the framework, MoDS algorithm is replaced by BR-MoDS and B<sup>2</sup>R-MoDS algorithms. When video download is initiated, the simulation time counter is started. As before, in our experiments, simulation step is fixed at 2 seconds. In the simulation step that marks the beginning of segment download, BR-MoDS/B<sup>2</sup>R-MoDS determines the bit rate of the segment. For all the subsequent simulation steps that download this segment data, MoDS algorithm (Sections IIIC and IVD) determines the mode and corresponding download rate. The simulation counter when the battery is fully drained gives battery lifetime when user downloads and watches chosen video sequence under simulated channel and network load conditions and quality requirements. In order to capture the effect of bit rate adaptation on user experience, we modify the User Experience Model (Section IVC) to include the MOS corresponding to the bit rate selected. We use the bit rate–MOS model [35] to map the bit rate of each segment to a MOS value and calculate the average MOS value for the video streamed using (20). Given these MOS values and stalling measurements, the video experience  $VE$  of the user is measured using (29).

To allow comparison, we use the same framework to simulate the RA-DASH and BaSe-AMy, except that, instead of using BR-MoDS/B<sup>2</sup>R-MoDS, we use the algorithm implemented in [11] and [16] respectively to determine bit rate. For RA-DASH and BaSe-AMy, the download rate is determined by (23), and mode that satisfies the download rate and BER requirement is selected. It should be noted that if download rate determined by BR-MoDS/B<sup>2</sup>R-MoDS or for RA-DASH/BaSe-AMy exceeds the peak throughput, then the base station limits download rate to the peak throughput rate. The user characteristics, channel conditions and application BER requirements are identical to those in Table V.

Table VI lists the other required simulation parameters used in our DASH streaming experiments. The video characteristics consist of the set of video bit rates available for selection, the corresponding MOS values (derived from the video bit rate–MOS mapping for VGA screen resolution presented in [35]), duration of segments and the video sequence viewed. Each of the videos in video sequence 2 are available in segments of duration  $SegTime$  and each of these segments are encoded using  $V_{BR-Set}^{Valid}$ . Client characteristics enumerate buffer size and playback current requirements. The video quality requirements specify the maximum quality reduction acceptable  $VQ_{Red}$  and the quality threshold  $VQ_{Thr}$  that must be satisfied. Table VI also lists the peak throughput under variable network load conditions.

## B. Experimental Results

In this section, we will present the experimental results obtained by simulating adaptive bit rate streaming of video under variable network load conditions and different channel conditions. In all the experiments reported below, we set the weights  $w_{MOS}$  and  $w_{NStall}$  in (29) to 0.5, giving equal priority to spatial quality and stall-free video.

Fig. 9 shows the selection of bit rate (shown as green solid line) and download rate (shown as blue pluses) while streaming a video of 200s duration using RA-DASH, and our proposed BR-MoDS and B<sup>2</sup>R-MoDS techniques, under variable network load (shown as red dashed line representing the variation in peak throughput) and variable channel conditions. For lack of space, we do not illustrate the same for BaSe-AMy technique. The 200s video has the same bit rate/MOS characteristics shown in Table VI. Fig. 9(a) shows that RA-DASH attempts to track the network throughput while selecting bit rates, and downloads at the highest rate possible during each download epoch. From Fig. 9(b), it can be seen that BR-MoDS chooses the lowest bit rate possible initially, followed by higher bit rates (in order to boost  $MOS_{Avg}$  and satisfy the video quality constraint) and also lowest download rates possible. Fig. 9(c) shows that B<sup>2</sup>R-MoDS, as designed, chooses bit rates higher than that selected by BR-MoDS (except when BR-MoDS selects higher bit rates to boost  $MOS_{Avg}$ ), with the bit rate selected going down as it tracks battery level ratio which decreases as download progresses. However, like BR-MoDS, it also selects the lowest download rate possible. Though we do not illustrate the bit rate selection carried out by BaSe-AMy, it should be noted that BaSe-AMy always selects the highest bit rate possible. BaSe-AMy lowers the bit rate only when battery lifetime remaining is lesser than that required to completely stream the video and the battery level is below a certain threshold.

Next we report on the effect of the DASH based techniques on battery level and quality of video experience. Assuming the battery level is 0.2 at the start of the 200s video download, the battery level reduces by 16.1%, 17.34%, 10.45%, and 12% for RA-DASH, BaSe-AMy, BR-MoDS and B<sup>2</sup>R-MoDS respectively while achieving a video experience of 4.83, 4.76, 4.66, and 4.793. This shows that the proposed battery aware DASH techniques result in more battery efficient video streaming than the conventional RA-DASH and BaSe-AMy techniques. We also see that BR-MoDS can be more battery efficient than B<sup>2</sup>R-MoDS as it uses lower bit rates, while B<sup>2</sup>R-MoDS can achieve higher video experience.

In the next set of experiments, we simulate the video snacking behavior (variable snacking ratio, Table V) by the mobile device downloading video sequence 2 (Table VI), starting with battery level 0.2 till the battery gets exhausted, giving the battery lifetime. We report in Table VII values for Experience Longevity  $ExpTime$ , quality of Video Experience  $VE$ , and VEL metric obtained by RA-DASH and BaSe-AMy when streaming video sequence 2 under variable network load (red dashed line in Fig. 9) and variable, high, and low SNR conditions. Also reported are the percentage gains (loss) over RA-DASH and BaSe-AMy in Experience Longevity  $\% \Delta ExpTime$  and Video Experience  $\% \Delta VE$ , as well as VEL values, when using BR-MoDS and B<sup>2</sup>R-MoDS. From

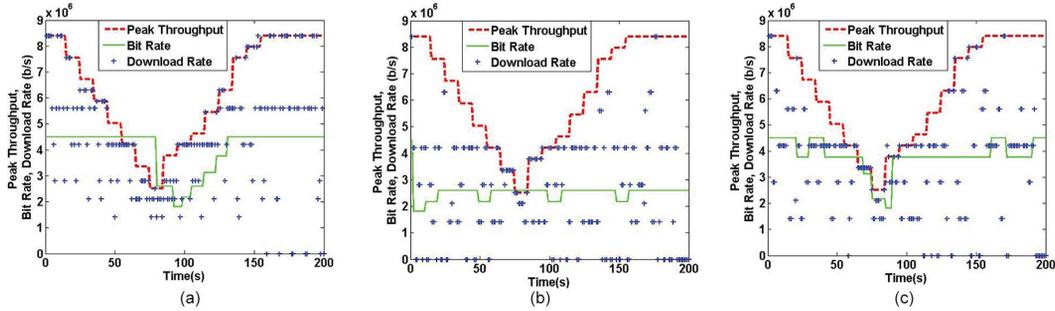


Fig. 9. Effect of downloading a single video of 200s on bit rate and download rate under variable network load and channel conditions while using (a) RA-DASH, (b) BR-MoDS, and (c) B<sup>2</sup>R-MoDS.

TABLE VII  
 $Exp_{Time}$ ,  $VE$ , VEL METRIC VALUES FOR RA-DASH AND BaSe-AMy,  $\% \Delta Exp_{Time}$ ,  $\% \Delta VE$ ,  
 AND VEL METRIC VALUES FOR BR-MoDS AND B<sup>2</sup>R-MoDS

	$Exp_{Time}$ (s)			$\% \Delta Exp_{Time}$			$VE$			$\% \Delta VE$			VEL					
	RA-DASH	BR-MoDS	B <sup>2</sup> R-MoDS	RA-DASH	BR-MoDS	B <sup>2</sup> R-MoDS	RA-DASH	BR-MoDS	B <sup>2</sup> R-MoDS	RA-DASH	BR-MoDS	B <sup>2</sup> R-MoDS	RA-DASH	BR-MoDS	B <sup>2</sup> R-MoDS			
SNR: Variable	1134	46.2	34.8	4.825	-3	-1.9	1	1.42	1.33									
SNR: High	1286	61.1	41.4	4.821	-3.2	-0.4	1	1.56	1.40									
SNR: Low	1013	32	20.2	4.78	-2.4	-0.78	1	1.29	1.19									
	BaSe-AMy			BR-MoDS			B <sup>2</sup> R-MoDS			BaSe-AMy			BR-MoDS			B <sup>2</sup> R-MoDS		
SNR: Variable	1082	53.2	41.3	4.75	-1.7	0.29	1	1.5	1.41									
SNR: High	1313	57.8	38.5	4.76	-2	0.733	1	1.54	1.39									
SNR: Low	977	36.9	24.6	4.54	-2.35	-0.66	1	1.4	1.3									

Table VII we observe that for variable SNR conditions (row 1), the experience longevity is significantly increased by using BR-MoDS and B<sup>2</sup>R-MoDS; 46.2% and 34.8% compared to RA-DASH and 53.2% and 41.3% compared to BaSe-AMy. In terms of video experience, BR-MoDS loses 3% and 1.7% compared to RA-DASH and BaSe-AMy while B<sup>2</sup>R-MoDS loses 1.9% and gains 0.29% compared to RA-DASH and BaSe-AMy respectively. As can be expected from the  $\% \Delta Exp_{Time}$  and  $\% \Delta VE$  results, BR-MoDS and B<sup>2</sup>R-MoDS show significant gains in VEL compared to both RA-DASH and BaSe-AMy.

Under high SNR conditions, the longevity of video experience is higher than under variable SNR conditions for all the techniques, including RA-DASH and BaSe-AMy, as less power consuming modes can be used to achieve the required BER. It can be seen from Table VII that by using BR-MoDS and B<sup>2</sup>R-MoDS, experience longevity increases by 61.1% and 41.4% compared to RA-DASH and by 57.8% and 38.5% compared to BaSe-AMy. In terms of video experience, BR-MoDS loses 3.2% and 2% compared to RA-DASH and BaSe-AMy while B<sup>2</sup>R-MoDS loses 0.4% and gains 0.7% compared to RA-DASH and BaSe-AMy respectively. As expected, BR-MoDS and B<sup>2</sup>R-MoDS outperform RA-DASH and BaSe-AMy in terms of VEL values.

Lastly, when channel conditions are bad (low SNR), all the DASH techniques achieve lower battery lifetime compared to high and variable SNR conditions as more power intensive modes have to be used to meet BER requirements resulting in lower battery lifetime. BR-MoDS and B<sup>2</sup>R-MoDS extend experience longevity by 32% and 20.2% compared to RA-DASH and 36.9% and 24.6% compared to BaSe-AMy. In terms of video experience, BR-MoDS and B<sup>2</sup>R-MoDS lose 2.4% and 0.78% compared to RA-DASH and 2.35% and 0.66% compared

to BaSe-AMy. As before, both BR-MoDS and B<sup>2</sup>R-MoDS outperform RA-DASH and BaSe-AMy in terms of VEL metric.

## VII. CONCLUSION

In this paper, we presented novel battery aware HTTP video delivery schemes. First, we proposed battery aware video progressive download techniques that dynamically adapt video download rate and transceiver configurations to reduce battery consumption while ensuring user experience. Next, we presented battery aware DASH streaming techniques that aim to maximize both battery lifetime and video quality while ensuring minimum desired video quality by adapting video bit rate in addition to download rate and transceiver configuration. Lastly, we proposed the Video Experience Longevity metric that quantifies the performance of the proposed battery aware DASH techniques in terms of experience longevity and video experience. Our simulation results demonstrated the ability of the proposed techniques to achieve significant increase in battery lifetime, no more than the desired (video quality threshold) loss in video experience and high VEL values as compared to conventional non-battery aware techniques and other battery aware techniques.

While the proposed battery aware video delivery techniques focus on increasing battery lifetime, in future, we aim to investigate techniques that jointly reduce the power consumption at the base station and battery consumption of mobile device while downloading mobile video. We would also like to extend our techniques to explore battery savings when video is streamed and uploaded from mobile devices.

## REFERENCES

- [1] "Cisco visual networking index: Global mobile data forecast update, 2013-2018," Cisco, San Jose, CA, USA, Tech. Rep., 2014.

- [2] X. Yu, R. S. Kalyanaraman, and A. Yla-Jaaski, "Energy consumption of mobile YouTube: Quantitative measurement and analysis," in *Proc. 2nd Int. Conf. Next Generation Mobile Appl. Services Technol.*, Sep. 2008, pp. 61–69.
- [3] K. J. Ma, R. Bartos, S. Bhatia, and R. Nair, "Mobile video delivery with HTTP," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 166–175, Apr. 2011.
- [4] A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Trans. Internet Comput.*, vol. 15, no. 2, pp. 54–63, Mar./Apr. 2011.
- [5] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [6] H. Kim and B. Daneshrad, "Energy-constrained link adaptation for MIMO OFDM wireless communication systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, pp. 2820–2832, Sep. 2010.
- [7] R. Hormis, E. Linzer, and W. Xiaodong, "Adaptive mode- and diversity-control for video transmission on MIMO wireless channels," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3624–3637, Sep. 2009.
- [8] K. Hongseok, C. Chan-Byoung, G. de Veciana, and R. W. Heath, "A cross-layer approach to energy efficiency for adaptive MIMO systems exploiting spare capacity," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4264–4275, Aug. 2009.
- [9] C.-Y. Li, C. Peng, S. Lu, and X. Wang, "Energy-based rate adaptation for 802.11n," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2012, pp. 341–352.
- [10] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proc. 8th Int. Conf. Emerging Netw. Experiments Technol.*, 2012, pp. 109–120.
- [11] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 169–174.
- [12] R. Pantos and W. May, "HTTP live streaming, draft-pantos-http-live-streaming-16," Apple Inc., Cupertino, CA, USA, Apr. 2010 [Online]. Available: <https://datatracker.ietf.org/doc/draft-pantos-http-live-streaming/>
- [13] *IIS Smooth Streaming*. Microsoft, Redmond, WA, USA, 2009 [Online]. Available: <http://www.iis.net/downloads/microsoft/smooth-streaming>
- [14] F. M. Tabrizi, J. Peters, and M. Hefeeda, "Dynamic control of receiver buffers in mobile video streaming systems," *IEEE Trans. Mobile Comput.*, vol. 12, no. 5, pp. 995–1008, May 2013.
- [15] M. Tamai, N. Shibata, K. Yasumoto, and M. Ito, "An energy-aware video streaming system for portable computing devices," in *Proc. 7th Int. Conf. Mobile Data Manage.*, May 2006, pp. 58–61.
- [16] M. Kennedy, H. Venkataraman, and G. Muntean, "Battery and stream-aware adaptive multimedia delivery for wireless devices," in *Proc. IEEE Conf. Local Comput. Netw.*, Oct. 2010, pp. 843–846.
- [17] D. N. Rakhmatov and S. B. K. Vrudhula, "An analytical high-level battery model for use in energy management of portable electronic systems," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Nov. 2001, pp. 488–493.
- [18] C. Min and G. A. Rincon-Mora, "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504–511, Jun. 2006.
- [19] A. A. Atayero, O. I. Sheluhin, and Y. A. Ivanov, "Modeling, simulation and analysis of video streaming errors in wireless wideband access networks," in *IAENG Trans. Engineering Technologies, Lecture Notes in Electrical Engineering*. New York, NY, USA: Springer, 2011, vol. 170, pp. 15–29.
- [20] O. I. Sheluhin, A. A. Atayero, Y. A. Ivanov, and J. O. Iruemi, "Effect of video streaming space-time characteristics on quality of transmission over wireless telecommunication networks," in *Proc. World Congr. Eng. Comput. Sci.*, 2011, pp. 572–577.
- [21] S. Shah and V. Sinha, "Iterative decoding vs. viterbi decoding: A comparison," in *Proc. Nat. Conf. Commun.*, Feb. 2002, pp. 494–497.
- [22] H. Wu, S. Nabar, and R. Poovendran, "An energy framework for the network simulator 3 (ns-3)," in *Proc. 4th Int. ICST Conf. Simulation Tools Techniques*, 2011, pp. 222–230.
- [23] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2349–2360, Sep. 2005.
- [24] D. Garrett, L. Davis, S. T. Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1544–1552, Sep. 2004.
- [25] C.-C. Lin, Y.-H. Shih, H.-C. Chang, and C.-Y. Lee, "Design of a power-reduction viterbi decoder for WLAN applications," *IEEE Trans. Circuits Syst.I: Reg. Papers*, vol. 52, no. 6, pp. 1148–1156, Jun. 2005.
- [26] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation*, TS 36.211, 2010.
- [27] L. Wang, A. Ukhanova, and E. Belyaev, "Power consumption analysis of constant bit rate data transmission over 3G mobile wireless networks," in *Proc. Int. Conf. ITS Telecommun.*, Aug. 2011, pp. 217–223.
- [28] A. Carroll and G. Heiser, "An analysis of power consumption in a smart-phone," in *Proc. USENIX*, 2010, pp. 271–284.
- [29] T. Hobbeld *et al.*, "Quantification of YouTube QoE via crowdsourcing," in *Proc. IEEE Int. Workshop Multimedia Quality Experience—Modeling, Eval., Directions*, Dec. 2011, pp. 494–499.
- [30] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. Lopez-Soler, "Analysis and modelling of YouTube traffic," *Trans. Emerging Tel. Technol.*, vol. 23, no. 4, pp. 360–377, Jun. 2012.
- [31] "The U.S. digital video benchmark—2012 review," Adobe Systems, Inc., New York, NY, USA, 2012.
- [32] X. Li, M. Dong, Z. Ma, and F. C. A. Fernandes, "Greentube: Power optimization for mobile video streaming via dynamic cache management," in *Proc. ACM Multimedia*, Oct. 2012, pp. 279–288.
- [33] N. Ding *et al.*, "Characterizing and modeling the impact of wireless signal strength on smartphone battery drain," in *Proc. ACM SIGMETRICS/RICS*, 2013, pp. 29–40.
- [34] *Transparent End-to-End Packet Switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming Over HTTP (3GP-DASH)*, 3GPP TS 26.247, 2011.
- [35] G. Cermak, M. Pinson, and S. Wolf, "The relationship among video quality, screen resolution, and bit rate," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 258–262, Jun. 2011.



**Ranjini Guruprasad** is currently working toward the Ph.D. degree in electrical and computer engineering at the University of California at San Diego, La Jolla, CA, USA.

Her research interests include the design and analysis of algorithms for green networks, specifically with a focus on energy efficient operation of mobile devices and renewable energy powered base stations.



**Sujit Dey** (S'03–M'91–SM'03–F'14) received the Ph.D. degree in computer science from Duke University, Durham, NC, USA, in 1991.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA, USA, where he heads the Mobile Systems Design Laboratory, which is engaged in developing innovative mobile cloud computing architectures and algorithms, adaptive multimedia and networking techniques, green and sustainable computing and communication, and reliable system-on-chips, for the next-generation of mobile multimedia applications. He is also the Director with the UCSD Center for Wireless Communications, and is affiliated with the Qualcomm Institute, La Jolla, CA, USA. He served as the Chief Scientist, Mobile Networks, with Allot Communications, Hod HaSharon, Israel, from 2012 to 2013. He founded Ortiva Wireless, La Jolla, CA, USA, in 2004, where he served as its founding CEO and later as CTO until its acquisition by Allot Communications in 2012. Prior to Ortiva, he served as the Chair of the Advisory Board of Zyray Wireless until its acquisition by Broadcom, Irvine, CA, USA, in 2004. Prior to joining UCSD in 1997, he was a Senior Research Staff Member with NEC Research Laboratories, Princeton, NJ, USA. He is the co-inventor of 18 U.S. patents, resulting in multiple technology licensing and commercialization. He has authored or coauthored over 200 publications, including journal and conference papers and a book on low-power design.

Dr. Dey has chaired multiple IEEE conferences and workshops. He was the recipient of six IEEE/ACM Best Paper Awards.