Multimed Tools Appl
https://doi.org/10.1007/s11042-017-5603-z

Battery-aware rate adaptation for extending video streaming playback time

Hasnah Ahmad[1] · **Navrati Saxena**[1] · **Abhishek Roy**[2] · **Pradipta De**[3]

Received: 15 May 2017 / Revised: 14 December 2017 / Accepted: 29 December 2017

Abstract Multimedia streaming applications are computation and network intensive that put a high demand on battery usage of mobile devices. Battery usage forms an important metric in user satisfaction, as increased battery consumption results in faster battery depletion and eventually leads to battery outage. In this paper, we propose an adaptation technique, referred as Battery-Aware Rate Adaptation (BARA) scheme, which adapts to the appropriate bit rate to prolong the battery lifetime. BARA considers both the wireless channel conditions, as well as the device's battery level, to determine the best transmission rate for optimizing the mobile battery consumption. Actual experiment and simulation results corroborate that compared to the conventional techniques, BARA can save more than 40% of battery power, while extending the video playback time by 20%.

Keywords Rate adaptation · Battery consumption · Mobile devices · Video streaming · Wireless networks · Quality of experience

1 Introduction

The progress of mobile broadband technology is spurring the growth of mobile video streaming applications and devices. According to Cisco Visual Networking Index, Internet video streaming and downloads will consume more than 62% of total Internet traffic

✉ Navrati Saxena
navrati@skku.edu

1 Department of Software, Sungkyunkwan University, Suwon 440-746, South Korea

2 System Design Laboratory, Networks Division, Samsung Electronics, Suwon 440-600, South Korea

3 Department of Computer Science, Georgia Southern University, Statesboro, GA, 30458, USA

Published online: 03 February 2018

by 2020 [14]. However, streaming on mobile devices requires significant power. Although the mobile communication technology has evolved rapidly over the last couple of decades, the battery technologies have almost remained the same. Naturally this has created a significant gap between the power-hungry mobile applications' demand and the mobile battery power supply. As a consequence, streaming experience on mobile devices is constrained by the limited battery power supply. Therefore, more effort is required to devise a power-saving solution that can optimize or prolong battery lifetime in mobile devices.

Over the last decade, energy conservation in battery powered mobile devices topic has attracted many research works and a wide range of solutions for optimizing energy consumption for video streaming on mobile devices have been proposed. Most of the energy-saving solutions fall under two major categories; traffic scheduling strategies and video content adaptation schemes. The former contains solutions that optimize the receiving energy without modifying the actual video content [22]. Alternatively, the latter mechanism specifically adjusts and changes the contents to reduce energy consumption during receiving, decoding and displaying. The traffic scheduling mechanism, such as sleep scheduling (i.e., periodically turns off mobile devices' radio interfaces), might not be suitable for real time communication. The sleeping schedules, for example, might cause conflict with highly delay-sensitive video services, such as Skype, Hangouts or Voice over Internet Protocol (VoIP), and might affect the user's Quality of Experience (QoE) [33]. A more in depth discussion on the energy-saving techniques developed for multimedia streaming is provided in Section 2.

The widely adopted HTTP based video download techniques, such as Progressive Download and Live Streaming, provide flexibility in varying the download rate [20]. These techniques can be used to adapt the video contents streamed to mobile devices, based on changing network conditions to avoid stalling. However, these techniques do not consider the effect of video download on mobile device's battery. In this paper, we design a solution, referred as Battery-Aware Rate Adaptation (BARA), that takes into account two major components: (1) the Channel Quality Indicator (CQI) between a mobile device and the base station (BS) and (2) the residual battery level in a mobile device while receiving and playing a video stream. Based on these two parameters, a suitable bit rate for downloading is determined such that it can reduce the battery consumption.

To verify the effectiveness of BARA scheme, the results are compared with the conventional streaming method, with adaptation process based only on the network conditions. For an accurate performance analysis, real-time mobility trace is used in the simulations. In addition, to ensure user experience during streaming, the effect of reducing power consumption over video quality is also analyzed. To summarize, the major contributions of the paper are as follows:

1. We analyze the impact of continuous streaming with different video resolutions on mobile device's power consumption. The results gathered substantiate that streaming in higher resolutions consumes higher battery power (more than 50%). Thus, this serves as a major motivation for us to perform battery-aware rate adaptation, as to avoid video playing disruption caused by battery outage.
2. We formulate the rate adaptation for BARA using the Markov decision process approach. The adaptation process is characterized by incorporating the mobile device's battery information as well as the time-varying channel conditions.

3. Based on the results generated through exhaustive simulations using MATLAB and experiments conducted over Android based mobile device, we propose a feasible alternative to the existing streaming method which can prolong battery lifetime and extend the video playback time up to 20%.
4. We provide the trade-off analysis between video quality and battery power consumption.

By extending our previous work in [1] published in IEEE Communications Letters, this paper formulates the rate adaptation process as a Markov Decision Process (MDP) based optimization problem. In [1], we use an aggressive adaptation approach by forcing the algorithm to always choose the minimum rate based on the channel conditions and the battery level. This method could cause an abrupt quality switching during streaming. On the other hand, the newly formulated MDP-based approach can provide a more gradual and smoother rate-switching decision in our algorithm. MDP is selected since it can be used to effectively model the dynamics of a video streaming system under time-varying network conditions [47].

The remainder of the paper is organized as follows: Section 2 provides discussion on the existing energy saving techniques in multimedia streaming. The proposed system model is discussed in Section 3. We formulate the rate adaptation as an MDP problem in Section 4. Section 5 evaluates the performance of the proposed scheme and presents the results to validate its ability to prolong battery lifetime. In Section 6, we investigate the battery power consumption and video quality relationship by performing the power-video quality trade-off analysis. Finally, Section 7 presents the conclusion.

2 Existing adaptive streaming solutions

The delivery and transmission of large amounts of video data on mobile devices generate critical challenges. Issues such as resource allocation, battery power consumption, CPU, memory and bandwidth constraints must be taken into consideration when designing streaming solutions. The heterogeneity of the multimedia streaming protocols and the variation of media formats and codecs supported by different mobile devices also add to the already existing challenges. However, these challenges open vast opportunity for researchers to propose different effective streaming solutions. We divide relevant solutions proposed into several categories as follows:

2.1 Existing solutions based on traffic scheduling

Some of the relevant works that fall under this category are published in [11, 23, 27, 31]. In [11] and [23], the authors propose techniques to save energy during streaming by aggregating data traffic to multiple interfaces (LTE and WiFi) and by setting the wireless interface on mobile device to sleep most of the time respectively. However, running multiple interfaces could increase the energy and bandwidth consumption especially when the user chooses to skip or quit the video. The work in [31] proposes a GreenTube scheme that could dynamically select buffer size based on the remaining time a user would watch a video. This scheme, however, relies on accurate prediction of user watching behaviors. Khan et al. in [27] make use of the Discontinues Reception (DRX) mechanisms in LTE. The authors propose an adaptive DRX parameters in which the DRX cycles can be adjusted according to the ongoing traffic pattern or packet arrivals. The DRX configuration can save energy but at the cost of delays addition, resulting in throughput decrements as well as video quality degradation, if it is not properly configured.

2.2 Existing solutions based on content adaptation

Works published in [16, 21, 24, 29, 45] propose solutions based on content adaptation. Authors in [21] develop an energy-efficient adaptive streaming technique depending on the mobile device buffer levels and the channel conditions, to maximize battery life. A recent work in [45] introduces a scheme refers to as RnB which analyzes the rate distortion and energy trade-off across the entire end-to-end pipeline from the initial video encoding to the final mobile display. RnB encodes multiple video versions that are suitable for brightness and scaling. The client can download the optimal video version considering the trade-off and perform the brightness scaling allowed by the version to reduce energy consumption. Kennedy et al. in [24] design a cross-layer smartphone battery and stream-aware adaptive multimedia delivery mechanism (BaSe-AMy). The proposed solution adapts the video quality based on mobile device remaining battery, remaining video stream duration and packet loss rate. Similarly, authors in [16] propose a device characteristics-based differentiated energy-efficient adaptive solution (DEAS). DEAS performs energy-efficient quality adaptation based on the constructed energy-oriented system profile including a power signature of the various device components for each running application. However, energy efficiency improvement is relatively small under stable network conditions. In [29], a video streaming system that utilized the Adaptive Spatial Resolution Control for mobile energy saving is proposed. The scheme, first calculates the energy required for decoding, before the video data is transmitted. Then it encodes the video at low spatial resolution after down-sampling and scales the video up after decoding at the mobile device.

2.3 Existing solutions based on different technologies

Undeniably, media streaming over mobile network is a well researched topic. Various other streaming and adaptation solutions have been proposed to optimize the streaming experience. While most of the solutions mentioned in Sections 2.1 and 2.2 are mainly focused on energy efficiency, there are other solutions that focus on enhancing user experience during streaming. Works in [48] and [12] exploit the Device-to-Device (D2D) communications for video streaming. Zhu et al. [48] aim to maximize the video playback quality while maintaining the long-term queue stability. This is achieved by allocating more resources to users with smaller queue lengths to avoid stalling events from happening. Authors in [12] utilize the content similarity to identify suitable peers and form D2D clusters for video multicasting. Through the proposed method in [12], user can obtain missing packets from other users and restores incomplete video frames, thus improving the user perception of the video quality. Authors in [43] and [17] integrate the cloud computing concept into the mobile environment to enhance video streaming services. Wang et al. [43] propose a novel mobile streaming framework with two main parts: adaptive mobile video streaming and efficient social video sharing. Cloud computing is considered a promising solution in terms of computation offloading, data storage capacity, scalability and improved reliability [17]. However, there are many factors to consider for extending the cloud computing-based services to mobile environments, such as, wireless link dynamics, user mobility and the limited capability of mobile devices [17]. Other related works are published in [46] and [41].

Both [46] and [41] perform adaptation based on throughput prediction. The method introduced in [46], makes use of the historical TCP throughput of last few video segments to estimate the current bandwidth and instantaneously adapt the segment quality. The drawback of this approach lies in finding and providing an accurate throughput prediction in a highly variable environment. Recent study in [41], tries to overcome this limitation by using a large-scale dataset to train its Hidden-Markov-Model-based prediction and adaptation algorithms.

However, due to the mobility and heterogeneity of wireless network, designing good bit rate selection and adaptation algorithms is still an open research area. Within the literature, the closest to our work are presented in [47] and [30]. Both solutions adopt the Markov decision process framework in their rate adaptation schemes and also combine several system parameters for the rate switching decision. For example, work in [47] combines the video rate switching frequency and amplitude as well as the buffer information, where as, authors in [30], jointly combine packet scheduling and playout control in their algorithm. However, there are two differences. The first difference, we introduce the mobile device battery level element in the decision making process. The second difference, in contrast to our work, these two aforementioned works only focus on delivering good video QoE without addressing the energy efficiency element.

3 Framework and system model

The proposed streaming system is based on an adaptive video streaming method (stream-switching technique), which is commonly employed by Apple HTTP live streaming, Microsoft IIS server, Adobe Dynamic Streaming, Akamai HD Video Streaming and Move Networks [13]. With adaptive streaming, the video source can be adapted on-the-fly, such that the user is able to watch videos at the maximum bit rate allowable depending on the channel condition. The server encodes the video content at different bit rates and it switches from one video version to another based on client feedback such as the measured available bandwidth.

3.1 Overall framework

BARA adopts the Content Distribution Network (CDN) as a platform for its overall framework, as illustrated in Fig. 1. BARA framework consists of a Scalable Video Coding (SVC)-based video server, a base station equipped with an edge server and multiple UEs having different residual battery levels. The system adopts the multi-rate, layer-based SVC media streaming scheme to enable dynamic adaptation based on network conditions and device capabilities. SVC is an extension of H.264/Advanced Video Coding scheme and supports video streaming in a more heterogeneous (large-scale multi-user mobile video-streaming) scenario [29].

CDN is deemed as the most suitable platform to be adopted in BARA, as it could provide an efficient and fast media delivery services [4]. In CDN, the edge servers are distributed and dispersed geographically close or in close network proximity to the end users. By exploiting the location information to bypass the content providers or the original servers in the video distribution path during streaming, the edge server can provide fast video delivery. In BARA, the edge server is not only responsible to pull and cache the video content but also is responsible to perform our MDP-based BARA. This strategy, however, incurs additional IP lookup overhead at the BS [37]. However, the BSs in LTE

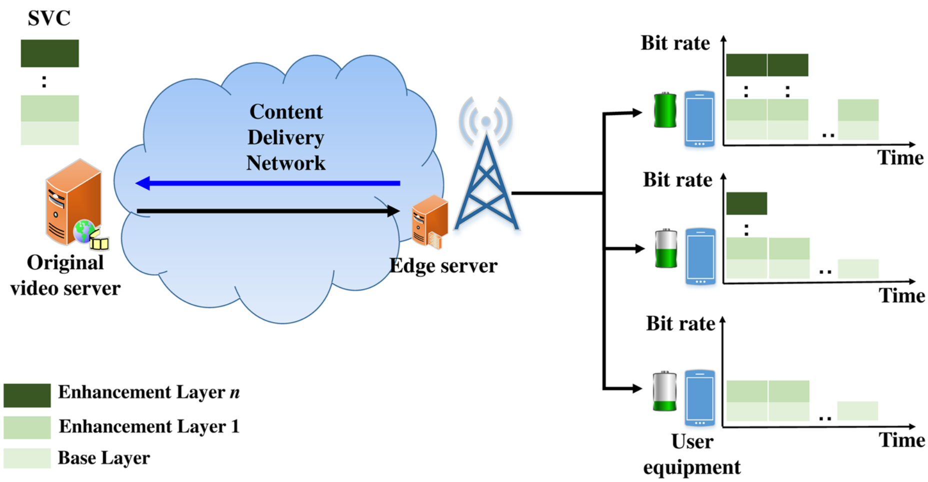


Fig. 1 BARA overall framework with a SVC (multi-layered) media streaming scheme, an edge server and UEs with different battery levels

networks are equipped with high performance processors (e.g. Octeon 8-core processors). This feature makes the BS capable of processing high traffic rates with negligible additional latency for IP lookup and performing fast MDP's computation to support our BARA implementation [37].

Figure 1 also illustrates a video download scenario from the video server through the BS to the UEs over the network. The bit rate versus time graphs in the figure represent the amount and flow of video data received on each UE depending on the UE's battery level at the time of downloading. For example, UE with a low battery level would receive only the base layer or bit stream while a UE with high battery level would receive both the base layer and multiple enhancement layers.

3.2 LTE channel state

In LTE, the channel state is estimated based on the reports sent by the UE to the network via the UE Channel State Information (CSI). CQI is considered as the most intuitive channel or CSI feedback and is used to deliver information to the BS about the downlink channel state [34]. The CQI index indicates the Modulation and Coding Scheme, which is then used to select the suitable rate for video transmission. The CQI index is determined based on the Signal-to-Noise-Ratio (SINR). We calculate SINR in terms of distance, d between UE to the BS as [39]:

$$\mathrm{SINR}(d) = \frac{E_{ta}/\varphi_0}{\sum_{i=1}^{n} \delta_i \left(\frac{E_{nb_i}}{\varphi_i}\right) + \mathfrak{N}}, \tag{1}$$

where E_{ta} is the target BS's downlink power and E_{nb} indicates the neighboring BS's power. $\varphi = d^{\eta} 10^{\psi/10}$ is the downlink path loss, where a constant, $\eta \in [4, 8]$ and ψ is the shadowing variable. The φ_0 and φ_i refer to the path loss between UE and target BS and the path loss between UE and other neighboring BSs respectively. The δ and $\mathfrak{N}$ represent the corresponding interference and noise elements respectively.

3.3 Battery model

In this paper, we adopt the battery model proposed in [38] to model the Lithium-ion battery discharging behavior in the mobile device. Given that the temperature effect on the battery model behavior is neglected, the mathematical model parameters for discharging at time, t, is defined as [38]:

$$V_{batt} = V_0 - \mu \cdot i - K \cdot \left(\frac{C}{C - it}\right) \cdot i^* - K \cdot \left(\frac{C}{C - it}\right) \cdot it + A \cdot \exp(-B \cdot it) \quad (2)$$

The detailed description of each parameter in (2) is provided in Table 1.

3.4 LTE power consumption model

We use LTE UE power consumption model proposed by Lauridsen et al. in [28]. The model covers the LTE cellular subsystem and the overall power consumption E_{cell}, is defined as:

$$E_{cell} = \sigma_{con} \times E_{con} + \sigma_{idle} \times E_{idle} + \sigma_{DRX} \times E_{DRX}, \quad (3)$$

where σ is a binary variable describing whether the UE is in Radio Resource Control (RRC)_connected (con), RRC_idle (idle), or Discontinuous Reception (DRX) mode. The E value describes the power consumption in the given mode as a function of mode specific parameters. The power consumption model of RRC_connected mode E_{con}, is divided into transmitting (Tx) and receiving (Rx) Base Band (BB) and Radio Frequency (RF) functional blocks. Each block defines the power consumption as a function of either Tx or Rx power levels (ε) and data rates (R) as shown in (4).

$$\begin{aligned} E_{con} = E_{on} + [\sigma_{Rx} \times (E_{Rx} + E_{Rx_{BB}}(R_{Rx}) + E_{Rx_{RF}} \\ (\varepsilon_{Rx}))] + [\sigma_{Tx} \times (E_{Tx} + E_{Tx_{BB}}(R_{Tx}) + E_{Tx_{RF}}(\varepsilon_{Tx}))] \end{aligned} \quad (4)$$

The constants E_{on}, E_{Rx} and E_{Tx} describe the power consumed when the cellular subsystem is ON, the receiver is actively receiving and the transmitter is actively transmitting,

Table 1 Mobile device and battery parameters

Parameter	Value
DEVICE	
Model	SAMSUNG Galaxy S4, GT-I9505 (with LTE data plan)
Operating system	Android 4.2.2
Processor	Quad-core 1.9 GHz Krait 300 with 2 GB RAM
BATTERY	
Fully charged battery voltage	4.42 V
Nominal battery voltage	3.80 V
Constant voltage, V_0	4.12 V
Internal resistance, μ	0.014615 Ω
Polarization constant, K	0.01095 Ah^{-1}
Max. battery capacity, C	2.60 Ah
Battery current, i	1.1304 A
Low frequency current dynamics, i^*	1.51 A
Exponential voltage, A	0.31911 V
Exponential capacity, B	23.4854 V

Table 2 Parameters for power consumption in connected mode

Parameter	Value
σ	$\{0, 1\}$
ON power, E_{on}	853 mW
Receiving power, E_{Rx}	25.10 mW
Receiving power (Base Band), $E_{Rx_{BB}}$	$= 0.97 \times R_{Rx} + 8.16$ (mW)

Receiving power (Radio Frequency), $E_{Rx_{RF}}$	if $\varepsilon_{Rx} \leq -52.5$ dBm:
	$= -0.04 \times \varepsilon_{Rx} + 24.8$ (mW)
	if $\varepsilon_{Rx} > -52.5$ dBm:
	$= -0.11 \times \varepsilon_{Rx} + 7.86$ (mW)

Transmitting power, E_{Tx}	29.90 mW
Transmitting power (Base Band), $E_{Tx_{BB}}$	0.62 mW

Transmitting power (Radio Frequency), $E_{Tx_{RF}}$	if $\varepsilon_{Tx} \leq 0.2$ dBm:
	$= 0.78 \times \varepsilon_{Tx} + 23.6$ (mW)
	if 0.2 dBm $< \varepsilon_{Tx} \leq 11.4$ dBm:
	$= 17.0 \times \varepsilon_{Tx} + 45.4$ (mW)
	if $\varepsilon_{Tx} > 11.4$ dBm:
	$= 5.90 \times {\varepsilon_{Tx}}^2 - 118 \times \varepsilon_{Tx} + 1195$ (mW)

respectively. The parameters and values for power consumption in connected mode are provided in Table 2 [28].

3.5 Video model

As highlighted before, we adopt the multi-rate, layer-based SVC video compression standard, for video encoding. We consider a video as a set of consecutive video segments, $m = \{1, 2, ..., M\}$, each of which contains k seconds of video. Thus, the total length of the video is $M \times k$ seconds. On the server, each video segment is encoded with n different bit rate levels. The video level bit rate l_m can assume values in the discrete set of available video levels $L = \{l1, ..., ln\}$. In this paper, we encode the video at eight different bit rate levels as shown in Table 3. The higher bit rate is chosen, the more enhancement layers are added to the video segment to be downloaded. Distinctly, a segment encoded at a higher bit rate has a higher video quality perceived by user. To ensure the video quality is maintained above a minimum acceptable bit rate level, the bit rate needs to be above a minimum threshold l_{min} = 200 kbit/s. Akamai [2] recommends that video for wireless networks should be encoded

Table 3 Set of available video levels L

Video level	Bit rate (Mbit/s)	Resolution (width x height)
$l1$	0.2	256 × 144
$l2$	0.4	426 × 240
$l3$	0.7	640 × 360
$l4$	1.0	854 × 480
$l5$	2.5	1280 × 720
$l6$	3.5	1280 × 720
$l7$	5.0	1920 × 1080
$l8$	8.0	1920 × 1080

at 110 kbit/s for low quality streaming, whereas, Ejembi et al. in [18] propose 128 kbit/s as the minimum acceptable target bit rate for video encoding, suitable for small hand-held devices. Hence, based on these two proposed encoding rates, as a way of preserving better video quality in BARA, the minimum acceptable bit rate is set to 200 kbit/s, an acceptably higher value as compared to the proposed minimum bit rates in both references.

In this section, we have modeled the channel state, the battery level, and the video bit rate representation. In the following section, we formulate the MDP-based rate adaptation algorithm using these three parameters (i.e., channel state, battery level and video bit rate information) as input parameters, for an improved rate-switching decision process in BARA.

4 HTTP-based adaptive streaming using MDP

A Markov Decision Process is an optimization model for decision making under uncertainty [5]. The MDP describes a stochastic decision process of an agent interacting with an environment or system. At each decision time instance, the system stays in a certain state s and the agent chooses an action a available at this state. After the action is performed, the agent receives an immediate reward $\mathbb{R}$ and the system transits to a new state s' according to the transition probability $P(s'|s, a)$. In video streaming, the MDP is used to model the interaction between the rate adaptation agent, employed at the server (i.e., an agent) and

Table 4 Summary of notations used in MDP formulation

Notation	Definition	Value
L	set of available video bit rate levels	$\{l1, ..., l8\}$={0.2 Mbps, ..., 8.0 Mbps}
m	set of consecutive video segments	$\{1, 2, ..., M\}$
s_m	system state at segment m	
l_m	video bit rate level assigned at current state s_m	
$b1_m$	channel condition based on CQI index at current state s_m	
$b2_m$	battery level at current state s_m	[0, 1] 0 = 0% battery level 1 = 100% battery level
Υ_m	video bit rate consistency at current state s_m	{0, 1} based from (6)
a_m	action taken at current state s_m	$\{a1, a2, a3\}$={increase, decrease, maintain} video bit rate
$\mathbb{R}, \mathbb{R}^\pi, r_{qi}, r_m$	immediate reward, long-term reward, reward based on quality factors, expected reward at current state s_m based on action taken a_m	(11) (13) (14)
p	MDP's state trans. prob.	
γ	MDP's discount parameter	[0, 1]
π, π^*	MDP's adaptive streaming policy, MDP's optimal policy	(12)
Ω	learning rate in Q-learning algorithm	[0, 1]
Φ	parameter to control the prob. of selecting non-optimal or random actions in Q-learning algorithm	[0, 1]

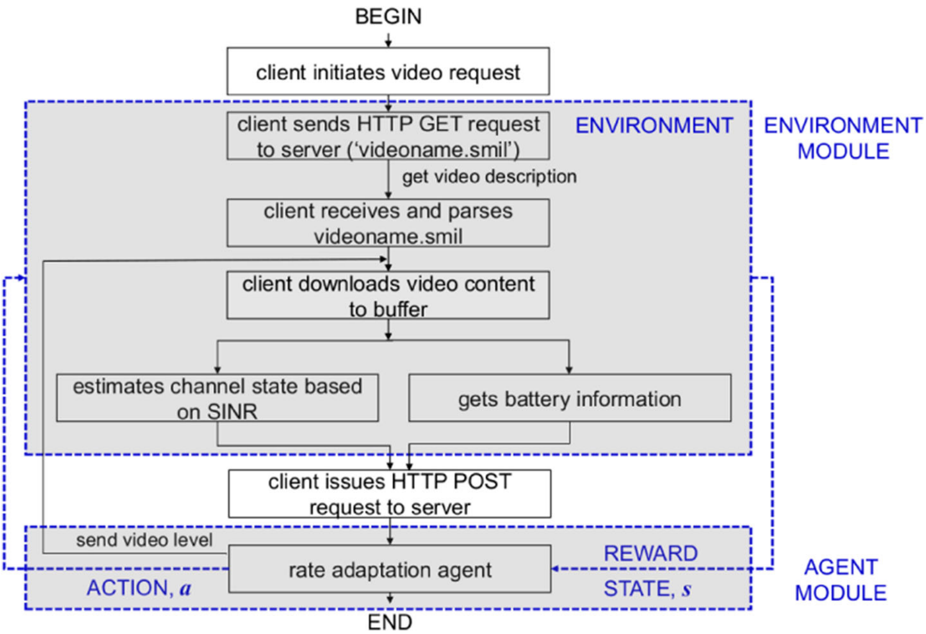


Fig. 2 MDP-based rate adaptation process in BARA

its surrounding environment (i.e., the channel condition and the device's or client's battery information) to achieve an objective. In our case, the MDP aims to optimize the video transmission rate selection as to extend the device's battery lifetime. For easier reference, we summarize the main notations used for MDP formulation in this section in Table 4.

4.1 MDP problem formulation

The mapping of our proposed rate adaptation model to the MDP is shown in Fig. 2. In general, as mentioned previously, the video streaming process could be viewed as an interaction between two modules. In Fig. 2, all the processes included in the top shaded region are considered as an integrated environment module, and the rate adaptation agent in the bottom shaded region is considered as an agent module. The environment sends a state signal for each video segment to the agent, and the agent then decides the best action to take correspondingly. The environment then assigns a reward to the agent for each decided action. This video streaming process could be formulated as a Reinforcement Learning (RL) task [44]. RL allows the agent to learn the best strategy or action to take through trial-and-error based on feedback from the environment [15]. With a properly formulated rate adaptation problem, the RL could efficiently converge to the optimal solution that maximizes the cumulative reward, even after a short training period. In this paper, we adopt the commonly used RL method known as Q-learning to determine the optimal policy.

In Fig. 2, the system is modeled in discrete time slots 1, 2, ..., t. At the beginning of each time slot t, the MDP algorithm decides which video quality level to download at the next slot $t+1$. Initially, the client connects to the server and selects the video to be played. Once the video is selected, a HTTP GET request is sent to the server which points to a SMIL[1]

[1] http://www.w3.org/TR/2005/REC-SMIL2-20050107/

compliant file. The base URL of the video, the available video levels and the corresponding encoding bit rates are provided in the SMIL file [13]. Next, the client parses the SMIL file to reconstruct the complete URLs of the available video levels. All the videos available are encoded at different bit rates. After parsing the SMIL file, the client issues HTTP POST request to the server, specifying parameters including buffer size, CQI, current received video level bit rate and battery information. Then, the rate adaptation algorithm at the server decides and selects the suitable video level based on the parameters received from the client. Once the request decision is made, the video segment with the appropriate video level is sent to the client. This process continues until the last video segment has been downloaded or the video has been terminated by the user.

Considering the Markov property of the system states, the MDP could be formulated for the streaming process. To apply the MDP, we need to devise the state transition model of the Markov process. A rate decision is made at stage m for the next segment $m + 1$. Thus, the total number of stages equals the number of segments M. For stage m, we denote the state as s_m, which consists of all the information gathered from the network once segment m has been completely downloaded. We also define each state vector s_m to contain three system parameters in the current system situation, including the channel state based on CQI $b1_m$, the battery levels $b2_m$ and finally the video bit rate consistency function Υ_m.

$$s_m = (b1_m,\ b2_m, \Upsilon_m)\,, \tag{5}$$

The video bit rate consistency function Υ_m, is used to avoid any abrupt rate switching during viewing. For instance, a sudden rate drop from the highest to the lowest level will cause a visual disturbance to the client. A smooth transition from one bit rate level to another bit rate level during adaptation process is critical in ensuring a good QoE. Thus, we use the Υ_m parameter to control the rate consistency in the MDP. To formulate the video bit rate consistency function, we define a video bit rate level vector $\mathbf{l}_m = \left[l_{m-N+1}, l_{m-N+2}, ..., l_{m-1}, l_m\right]$, where l_m represents the video bit rate level assigned in the current state s_m. The $\Upsilon_m = 1$, if all the latest N segments have the same video rate and 0 otherwise, as summarized in the following equation [47]:

$$\Upsilon_m = \begin{cases} 1, & \text{if } l_{m-N+1} = l_{m-N+2} = ... = l_{m-1} = l_m \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Based on the information gathered by these three parameters, the controller then applies control action a_m (i.e., either to increase or to decrease or to maintain the current bit rate), to decide the video rate for segment $m + 1$ based on the information in state s_m. The output of a control action is the video bit rate to be downloaded, i.e., $l_{m+1} = a_m(s_m)$.

4.2 State transition probabilities

We now derive the transition probabilities of the joint channel and battery states, given an arbitrary video quality selection decision. Figure 3 illustrates a fundamental aspect of transition probabilities in MDP, using eight different video level bit rates and the option of choosing from two actions, $a1$ and $a2$. It shows that the probability of reaching to a particular video bit rate depends on the action taken. Due to the Markovian property, the state at any time instance depends only on its immediate previous state. Additionally, considering that the system parameters are independent of each other, thus, given any state s_m and

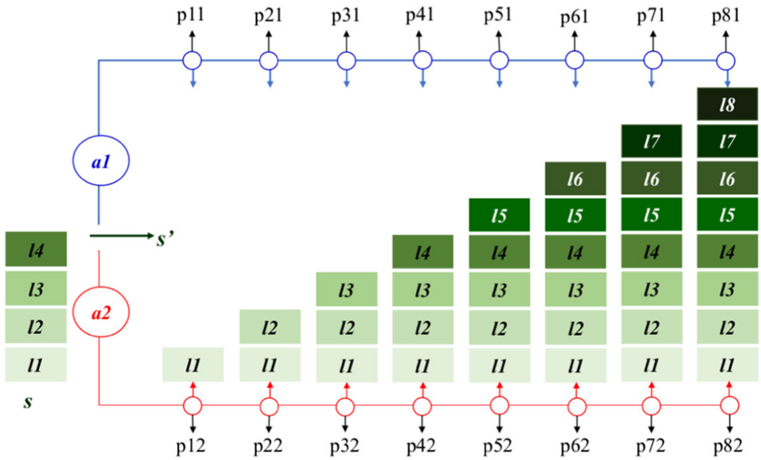


Fig. 3 Transition probabilities in MDP for choosing different actions

action a_m at time t, the MDP's transition probability between consecutive states $s_m = s$ to $s_{m+1} = s'$ could be derived as [47]:

$$\begin{aligned} p(s_{m+1} \mid s_m, a_m) & \\ &= p_r\left(s_{m+1} \mid s_m, a_m(s_m)\right), \\ &= p_r\left(b1_{m+1}, b2_{m+1}, \Upsilon_{m+1} \mid b1_m, b2_m, \Upsilon_m, a_m(s_m)\right) \\ &= p_r\left(b1_{m+1} \mid b1_m, a_m(s_m)\right) \times p_r\left(b2_{m+1} \mid b2_m, a_m(s_m)\right) \\ &\times\ p_r\left(\Upsilon_{m+1} \mid \Upsilon_m, a_m(s_m)\right) \end{aligned} \tag{7}$$

Note that, the first term in the right hand side of (7), could be obtained from the SINR probability distribution based on (1), the second term could be obtained from the battery level probability distribution based on (2) and the final term, the video rate consistency function, is defined by (6).

Considering M number of states, the transition matrix $p(s_{m+1} \mid s_m, a_m)$ could then be modeled by an M x M matrix as follows:

$$p^m = \begin{pmatrix} p_{11}^m & \cdots & p_{1M}^m \\ \vdots & \ddots & \vdots \\ p_{M1}^m & \cdots & p_{MM}^m \end{pmatrix}, \tag{8}$$

where element p_{ij}^m denotes the transition probability from state i to state j. The matrix is initialized with $p_{ij}^0 = 1/M, \forall i \leq M, j \leq M$. The transition probability from state i to state j under a control or action a by $p_{ij}^m(a)$ is defined and updated as follows:

$$\begin{gathered} \text{if } a = a1 \text{ (increase)} \\ p_{ij}^{m+1}(a1) = \begin{cases} p_r(b_1) \cdot p_r(b_2) \cdot p_r(\Upsilon) & \text{if } i \leq M \text{ and } j > i \\ 0 & \text{otherwise} \end{cases} \\ \text{if } a = a2 \text{ (decrease)} \\ p_{ij}^{m+1}(a2) = \begin{cases} p_r(b_1) \cdot p_r(b_2) \cdot p_r(\Upsilon) & \text{if } i \leq M \text{ and } j < i \\ 0 & \text{otherwise} \end{cases} \\ \text{if } a = a3 \text{ (maintain)} \\ p_{ij}^{m+1}(a3) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \end{gathered} \tag{9}$$

An example of state transition probability matrix for action $a1$ (increase) is given in the equation below:

$$p_{ij}^{m+1}(a1) = \begin{pmatrix} 0 & 0.04 & 0.04 & 0.46 & 0.23 & 0.23 & 0 & 0 \\ 0 & 0 & 0.04 & 0.42 & 0.21 & 0.21 & 0.12 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.2 & 0.12 & 0.08 \\ 0 & 0 & 0 & 0 & 0.33 & 0.33 & 0.2 & 0.14 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.3 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{10}$$

The number of states M could be increased to provide a more accurate predication, since a large M will generate smaller quantification intervals. However, this action will increase the number of state-variables and computational complexity [47].

4.3 Reward for rate adaptation derivation

We derive a reward function to measure the effectiveness of an action. We define a reward r_m associated with action a_m, at stage m, as a function of state s_m, i.e., $r_m = \mathbb{R}(s_m)$. Then, we define π as the adaptive streaming policy as a mapping of the action taken at each state. The long-term reward $\mathbb{R}^\pi(s_m)$ under policy π could be formulated by using the rewards associated with the individual states using the following equation [47]:

$$\mathbb{R}^\pi(s_m) = \sum_{s_{m+1}} p\left(s_{m+1} \mid s_m, a_m(s_m)\right)\left(r_m + \gamma \mathbb{R}(s_{m+1})\right) \tag{11}$$

where $\gamma \in [0, 1]$ is a discount parameter reflecting the present value of future reward. The goal is to find the optimal policy π^* that maximizes the reward during streaming. The video rate adaptation process could be formulated as the following optimization problem:

$$\pi^* = \arg\max\ \mathbb{R}^\pi(s_m) \tag{12}$$

The rates selected should consider three main quality factors: (i) maintain an acceptable viewing perception (less degradation), denotes by first reward r_{q1}, (ii) provide smooth rate transition from one video segment to another, denotes by r_{q2} and lastly (iii) minimize mobile device battery consumption, denotes by r_{q3}. Based on these factors, we can formulate r_{q1}, r_{q2} and r_{q3} as follows:

$$\begin{aligned} r_{q1} &= \begin{cases} 1 & \text{if } l_m = l_{max} \\ \frac{1}{M-m} & \text{if } l_{m-1} \le l_m < l_{max} \\ 0 & \text{otherwise (i.e., } l_m < l_{m-1}) \end{cases} \\ r_{q2} &= \begin{cases} 1 & \text{if } l_m = l_{m-1} \\ 0 & \text{otherwise} \end{cases} \\ r_{q3} &= \begin{cases} 1 & \text{if } l_m = l_{min} \\ \frac{1}{m} & \text{if } l_{min} < l_m \le l_{m-1} \\ 0 & \text{otherwise (i.e., } l_m > l_{m-1}), \end{cases} \end{aligned} \tag{13}$$

where l_m represents the video bit rate level assigned in the current state s_m.

- With r_{q1}, the aim is to achieve a high video quality by assigning a high reward if the action taken in state s_m results in higher bit rate than in the previous state s_{m-1}. For

example, maximum reward = 1 is assigned if the video bit rate in state s_m, l_m is equal to l_{max}, where $l_{max} = 8.0$ Mbit/s (from Table 3). The higher the video bit rate level, the higher reward is assigned to that particular state.
- With r_{q2}, the aim is to avoid an abrupt bit rate change from the previous state to the current state. Therefore, maximum reward is assigned if the action taken leads to the same bit rate level or no reward is assigned otherwise.
- Finally, with r_{q3}, the aim is to reduce battery consumption by rewarding the system with maximum value, if the action taken results in selecting the minimum video bit rate level, $l_{min} = 0.2$ Mbit/s (from Table 3). In contrast to r_{q1}, higher rewards are assigned for lower bit rate levels.

The overall reward of action a_m under state s_m is defined as the linear combination of the factors discussed above, as in the following equation:

$$r_m = \mathbb{R}(s_m) = r_{q1} + r_{q2} + r_{q3} \tag{14}$$

where $r_{qi} \in [0, 1]$ with $i = 1, 2, 3$, as defined in (13). The optimal policy π^* is then determined based on the derived state transition probability $p(s_{m+1} \mid s_m, a_m)$ in (7) and the devised reward function r_m in (14), for each state s_m associated with action a_m.

4.4 MDP solution

There are few methods for solving an MDP; value iteration, policy iteration and Q-learning. Value iteration and policy iteration assume a priori knowledge of state transition probabilities and use it to derive the optimal policy before the system starts its operation. Meanwhile, Q-learning assumes no priori knowledge of transition probabilities, but learns the optimal policy on-line, such that it only needs to know what states exist and what actions are possible in each state. Due to this property, in this paper, we adopt the Q-learning method to derive the MDP solution, as proposed in [10].

In Q-learning technique, a Q matrix, which defines the values of every state s if action a is taken, is initialized with zero. Whenever the video server starts out in state s_m, takes action a_m, and ends up in state s_{m+1}, it updates $Q(s_m, a_m)$ as [10]:

$$\begin{aligned} Q(s_m, a_m) \leftarrow (1 - \Omega) Q(s_m, a_m) + \Omega \big[\mathbb{R}(s_m, a_m, s_{m+1}) + \\ \gamma \max\nolimits_{a_{m+1}} \left[Q(s_{m+1}, a_{m+1}) \right] \big] \end{aligned} \tag{15}$$

where $\Omega \in [0, 1]$ is the learning rate. In a given state s_m, the function should decide and choose an action that provides the highest Q value based on the current estimates in Q for most of the time, but a random action the rest of the time. The probability of selecting the highest Q value action should increase over time. This could be achieved using the Boltzmann distribution as follows:

$$f(s_m, a_m) = \frac{e^{Q(s_m, a_m)/\Phi}}{\sum_j e^{Q(s_m, a_{mj})/\Phi}} \tag{16}$$

where $f(s_m, a_m)$ is the probability of selecting action a when in state s, and Φ parameter controls the probability of selecting non-optimal or random actions. If Φ is large, all actions are selected fairly and uniformly. If Φ is close to zero, the best action is always selected with high probabilities. Since no priori knowledge of the channel state is available at the beginning of the video download, we begin with a large value of Φ and gradually decrease it over time. Through this method, the algorithm initially makes random decisions and eventually learns from its observations. As learning continues, the Q values are used more often to make the decision. Our proposed MDP-based algorithm which uses Q-learning for optimal decision is depicted in Algorithm 1.

Algorithm 1 Q-learning algorithm

```
1: Initialize Q to 0
2: Download first segment using the lowest quality and observe next state
3: for m = 1:M do
4:   Update Q based on (15)
5:   Calculate Boltzmann probability based on (16) and use it to select video level l of next
     segment
6:   Download the next segment using video level l and observe next state
7: end for
```

5 Results and performance evaluation

This section investigates and compares the performance of the proposed scheme with the existing channel-based adaptation streaming scheme such as in DASH. The results are divided into two parts: actual test experiment and simulation. Experiment is conducted with the aim of investigating the impact of viewing different video resolutions on battery power consumption in continuous real-time streaming. Simulation is carried out with the aim of extending the analysis for multiple UEs in different scenarios (stationary and with mobility) with varying channel conditions and battery levels. MDP-based algorithm is also executed and tested in MATLAB.

5.1 Experimental analysis

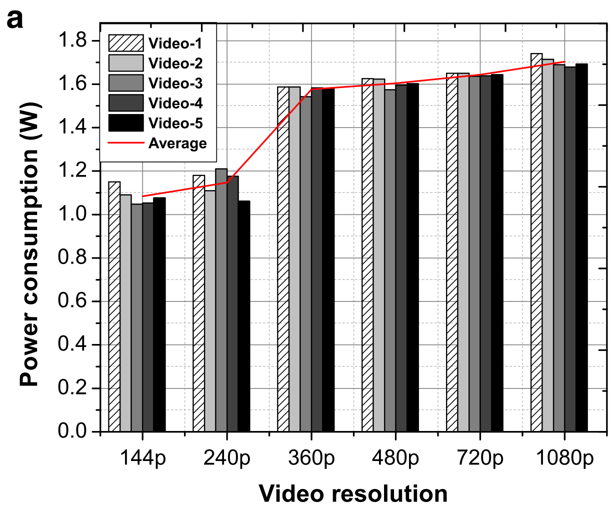
5.1.1 Power consumption in channel-based adaptation streaming

We measure the power consumption of video streaming on the mobile device in real LTE networks. We select five different open-source videos to investigate the effect of varying video resolutions on mobile's power consumption. The video files parameters used in the experiment are listed in Table 5. Video-1 is a CGI-enhanced movie, which contains fast-paced scenes (high motion scenes). Video-2 is an animation movie (medium motion scenes), featured in several published experiments. Video-3 is a news clip with less action and static shots (slow motion scenes). Video-4 contains fast-paced action (high motion scenes) and Video-5 is a CGI-enhanced short movie (medium motion scenes), also featured in several publications. All original videos are re-encoded into six different bit rate representations, ranging from 200 kbit/s for low resolution until 4 Mbit/s for high resolution, as shown in Table 5. The mobile device's information used for testing is shown in Table 1. The measurements are taken with Bluetooth/GSM/Wi-Fi interfaces on the device are disabled with minimal background application activity. The device's battery is not recharged during the whole streaming duration. PowerTutor software [35] is installed on the device and is used for measuring the average power consumption.

Table 5 Video files used in experiment

Video file	Label	Resolution	Bit rate (Mbit/s)	File size (MB)	Video duration (min)
Video-1 transformers 4 [42]	144p	256×144	0.2	15.3	8:92
	240p	426×240	0.4	24.1	
	360p	640×360	0.7	45.0	
	480p	854×480	1.1	71.1	
	720p	1280×720	2.9	184.0	
	1080p (Original)	1920×1080	4.0	243.0	
Video-2 Big buck bunny [8]	144p	256×144	0.2	17.0	9:93
	240p	426×240	0.4	26.7	
	360p	640×360	0.6	41.6	
	480p	854×480	0.8	56.4	
	720p	1280×720	2.0	142.0	
	1080p (Original)	1920×1080	2.7	190.0	
Video-3 BBC news clip [6]	144p	256×144	0.2	14.0	8:15
	240p	426×240	0.37	22.0	
	360p	640×360	0.39	23.4	
	480p	854×480	0.7	39.9	
	720p	1280×720	1.2	70.4	
	1080p (Original)	1920×1080	2.3	135.0	
Video-4 Motorcycle vs. Car Drift battle [25]	144p	256×144	0.2	14.5	8:29
	240p	426×240	0.4	22.8	
	360p	640×360	0.6	34.4	
	480p	854×480	1.0	59.6	
	720p	1280×720	1.8	108.0	
	1080p (Original)	1920×1080	3.2	192.0	
Video-5 Tears of steel [9]	144p	256×106	0.2	20.9	12:14
	240p	426×178	0.4	32.7	
	360p	640×266	0.5	41.9	
	480p	854×356	0.8	71.8	
	720p	1280×534	1.4	124.0	
	1080p (Original)	1920×800	2.6	229.0	

Figure 4a demonstrates the mobile battery power consumption during streaming in different video resolutions. Intuitively, the power consumption increases with the increasing video resolution, as increased resolution requires higher data rate processing. The power consumption difference between the low resolution video (144p) and the high resolution video (1080p), for example in Video-2, is 600 mW, more than 50% increase in power consumption. Higher resolution videos not only increase the computational power but also the



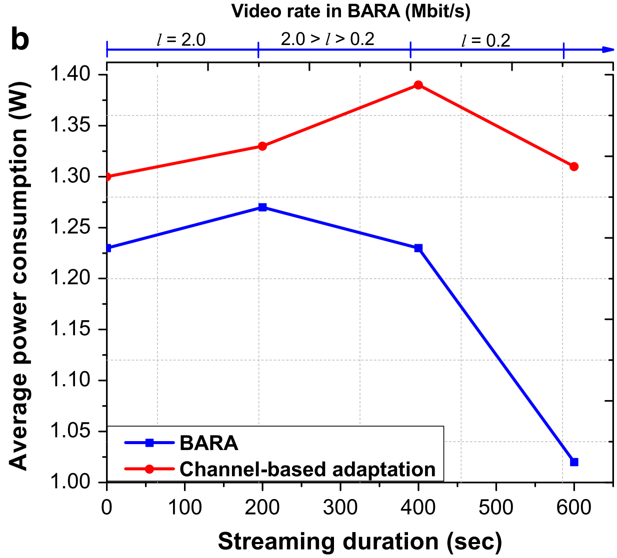


Fig. 4 Power consumption (experiment). **a** With different video resolutions **b** Average power consumption comparison (Video-2)

network usage, due to higher data rate of the video. Varying the video bit rate changes the battery consumption especially during online streaming. These results serve as another motivation to further optimize the video bit rate to conserve battery life. However, we observe that the power consumption increases at slower rate after 360p resolution. This is due to the video display attaining or spanning across the maximum screen of the device when viewing at 360p resolution. Beyond this point, the battery power is consumed mostly by the decoding process and radio interface elements in the device rather than the device display.

5.1.2 Power consumption in BARA streaming

To verify the gain of BARA implementation, experiment is conducted to measure the power consumption in real-time streaming. The experimental setup includes a mobile device installed with an Akamai Play [3] video streaming software application. "Akamai Play" application allows video content to be downloaded and streamed from the geologically nearest content delivery networks or Akamai server installed at the local BS [19, 36]. Video-2 (Big Buck Bunny) is selected for the experiment. The device's battery power is monitored at every 200 seconds and the video quality is changed from high (2 Mbit/s) to low (200 kbit/s) during the streaming process. We repeated the experiment eight times and measured the average power consumption in both streaming methods. Our experiment setup is as illustrated in Fig. 5.

Figure 4b shows the comparison of the measured average power consumption between BARA and the conventional video streaming technique. In BARA, since the rate is adjusted according to the depleting battery level, initially, between 0 to 200 sec, the video rate is maintained at 2 Mbit/s. No rate adaptation happens during this duration due to a high battery level still remaining on the device. When we monitor the battery power consumption at 200 sec, the power consumed is high at this time since it is calculated based on the high video rate received. Then, between 200 sec to 400 sec, since the battery level has depleted to a certain value, the video rate is now reduced to a lower rate than 2 Mbit/s, causing a lower power consumption. Finally, at 600 sec, as we reach the end of the video, less number of video segments are received by the user. In addition, the video rate has dropped to 0.2 Mbit/s to match the depleting battery on mobile due to the continuous video streaming. Thus, this scenario leads to a lower power consumption at the end of the streaming

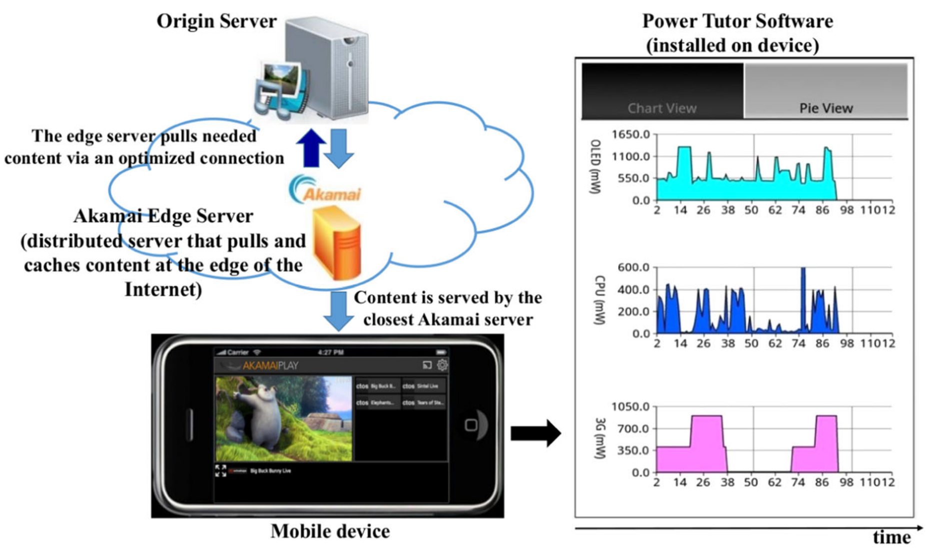


Fig. 5 Experiment setup to compare power consumption

session. As illustrated in Fig. 4b, for a duration of 600 seconds, it is observed that BARA-based streaming scheme can reduce the power consumption by 10% as compared to the conventional streaming scheme.

5.2 Simulation analysis

In simulation analysis, BARA performance is evaluated under two scenarios; stationary and mobility analysis. The network profile and simulation parameters are provided in Table 6. In both stationary and mobility scenarios, the power consumption calculations are based on the LTE power consumption model presented in (3) and (4) in Section 3.4.

5.2.1 Stationary analysis

For the stationary analysis, assume that the UE streaming a content over a 300 seconds duration without changing its location. By implementing BARA, the bit rate is reduced according to the residual battery level, β.

- Case 1: UE located closer to the BS (high CQI index) but with low battery level ($\beta = 0.06$) downloads in low bit rate and thus consumes less energy and extends the battery life further. After 5 minutes of continuous streaming, Fig. 6a shows, with BARA scheme, the total UE energy consumption is only 0.8 kJ. This is almost 64% less energy consumption comparing to 2.3 kJ using downloading scheme without BARA.
- Case 2: UE located in the middle range distance from BS (average CQI index) with high battery level ($\beta = 0.86$) receives almost similar bit rate and consumes around 0.7 kJ, almost the same amount of energy consumed as in Case 1. The results presented in Fig. 6a prove that a mobile user with low residual battery level, is still capable of streaming for a prolonged duration, almost similar to a mobile user with high residual battery level, albeit at a reduced rate. Thus, the comparisons made validate that BARA has the potential to prolong battery lifetime.

Figure 6b shows the energy efficiency comparison based on the UE's distance from the BS. A higher energy efficiency is obtained in BARA, when a UE is located near to the BS.

Table 6 Simulation parameters

Simulation parameters	
Number of UE(s)	1 - 5
BS radius coverage	300 m
Number of neighboring BSs	8
Number of RB allocated	[5, 95]
RB scheduling (for multiple UEs)	Round Robin
Modulation schemes	QPSK, 16QAM, 64QAM
Antenna configuration	STBC: 2×2 MIMO
Simulation time	300.0 sec
Android mobile device model	SAMSUNG Galaxy S4, GT-I9505
Nominal device battery voltage	3.8 V
Device battery capacity	2600 mAh
Carrier frequency, f_c	20.0 MHz

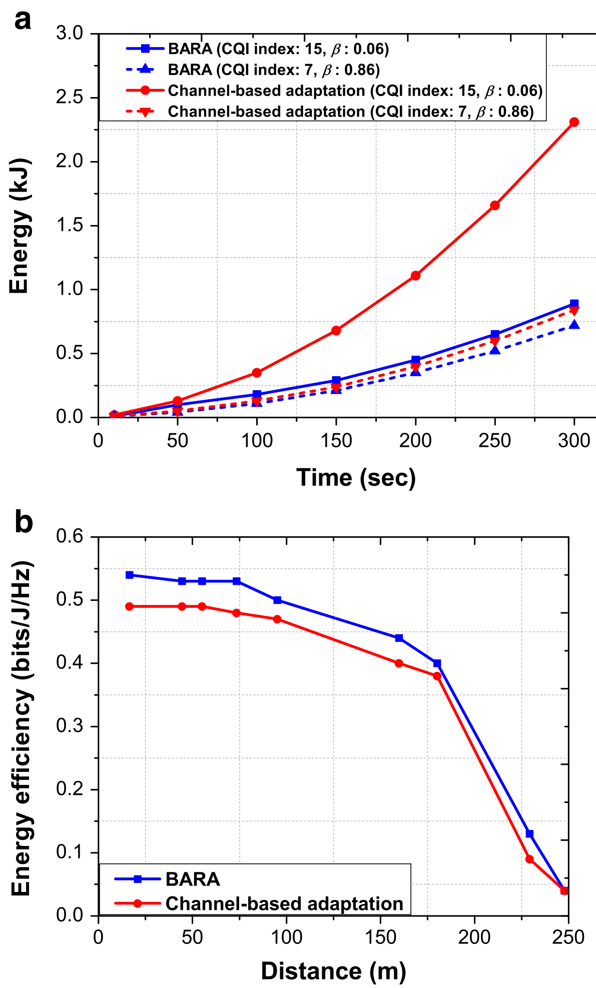


Fig. 6 Performance comparison (stationary analysis). **a** Energy consumption with different CQI indexes and battery levels **b** Energy efficiency based on distance from BS

In this case, the UE always experiences a good channel condition and is able to maintain a high SINR during streaming. The stable environment helps to avoid frequent rate switching from happening and the video bit rate is maintained at a higher level.

5.2.2 Analysis with real mobility trace

Mobility in outdoor settings affect transmission rates due to varying signal strength and hand-off operations. In our work, for a more accurate BARA performance evaluation, simulations based upon a real-time CRAWDAD [40] mobility trace record is performed. The data set selected contains six months of a personal mobile phone records stored by Deutsche Telekom in 2009 - 2010. However, in this analysis, we only evaluate data recorded on Aug.

31st., 2009 for five hours duration between 8.09 am until 1.07 pm. The mobility trace for the mentioned duration is provided in Fig. 7 [7].

BARA's performance is analyzed under different initial residual battery levels ranging between $0.1 \leq \beta_{ini} \leq 0.5$. The analysis is also based upon assumption that the user does not recharge his or her mobile device during the entire downloading process. Figure 8 shows BARA performance with added mobility trace. In Fig. 8a, a significant difference can be seen in the average power consumption on mobile device with and without BARA implementation. In Fig. 8b, by implementing BARA, depending on the residual battery level of mobile device or UE, up to 50% of battery power could be saved during downloading. BARA always chooses the optimum bit rate for transmission to maintain a lower power consumption on mobile, thus resulting in longer battery lifetime.

5.2.3 Impact of varying channel conditions and battery levels

We evaluate the results in the case that the channel condition, denotes by the channel signal strength, changes over the streaming duration. We analyze the rate adaptation process happening in both the conventional streaming scheme and the MDP-based BARA. We run the analysis using Video-2 as the test video running for 300 seconds. The initial β is set to be 0.5.

Figure 9 shows the effects of the varying channel conditions and the decreasing residual battery level over video bit rate selection. Since BARA (Fig. 9a) selects the video bit rate based on MDP algorithm, which is dependent on the system state and reward function, it achieves a smoother adaptation as compared to the conventional channel-based adaptation (Fig. 9b). It can be observed from Fig. 9a, BARA is able to maintain a fixed bit rate at the beginning of the streaming session without being affected by the varying channel strengths.

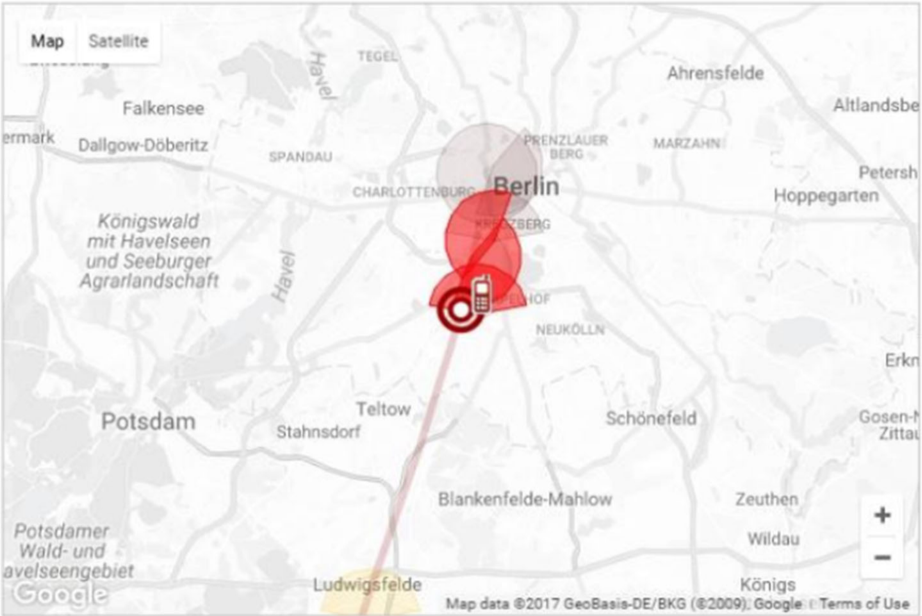


Fig. 7 Mobility trace of mobile phone records [7]

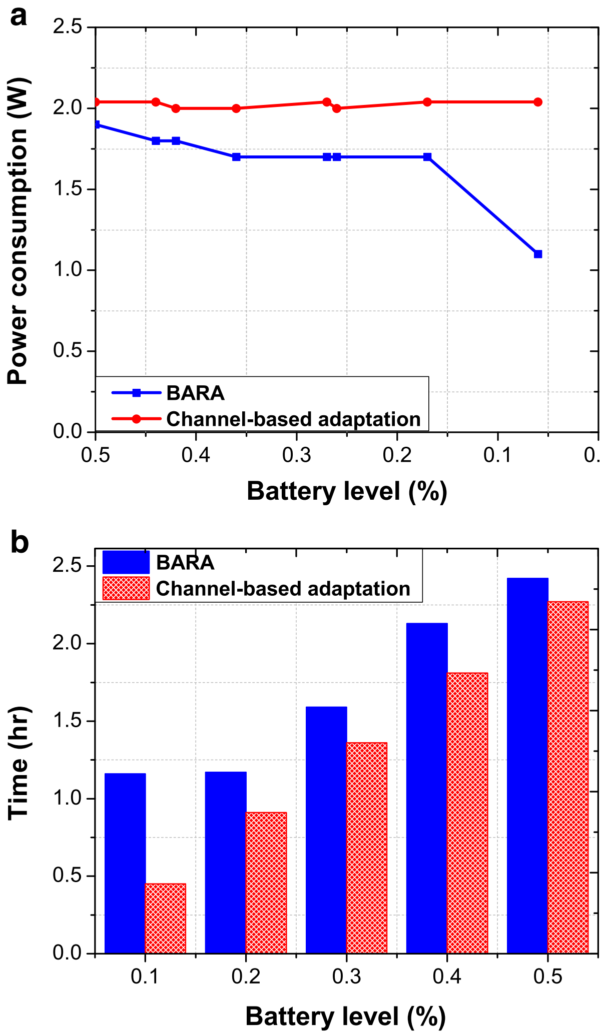


Fig. 8 Performance comparison (with mobility). **a** Power consumption comparison ($\beta_{ini} = 0.5$) **b** Estimated battery lifetime comparison

Within 200 seconds after the streaming session begins, the channel signal strength varies between −73 dBm to −69 dBm and the battery level decreases from 0.5 to 0.48. Within this duration, the download rate in BARA is maintained at 2.5 Mbit/s, whereas streaming via conventional method, experiences unstable and frequent rate switching, ranging between 1.48 Mbit/s to 2.08 Mbit/s. This proves BARA's potential in avoiding frequent adaptation, thus avoiding significant degradation on the video quality. The rate switching decision in BARA tries to minimize the battery power consumption by maintaining lower bit rate to match the depleting battery level, especially during the last remaining streaming session when the battery level has dropped to 0.37. It is also important to highlight that with BARA, a better bandwidth utilization can be achieved since MDP-based approach allows a video bit rate selection to be higher than the available bandwidth [47].

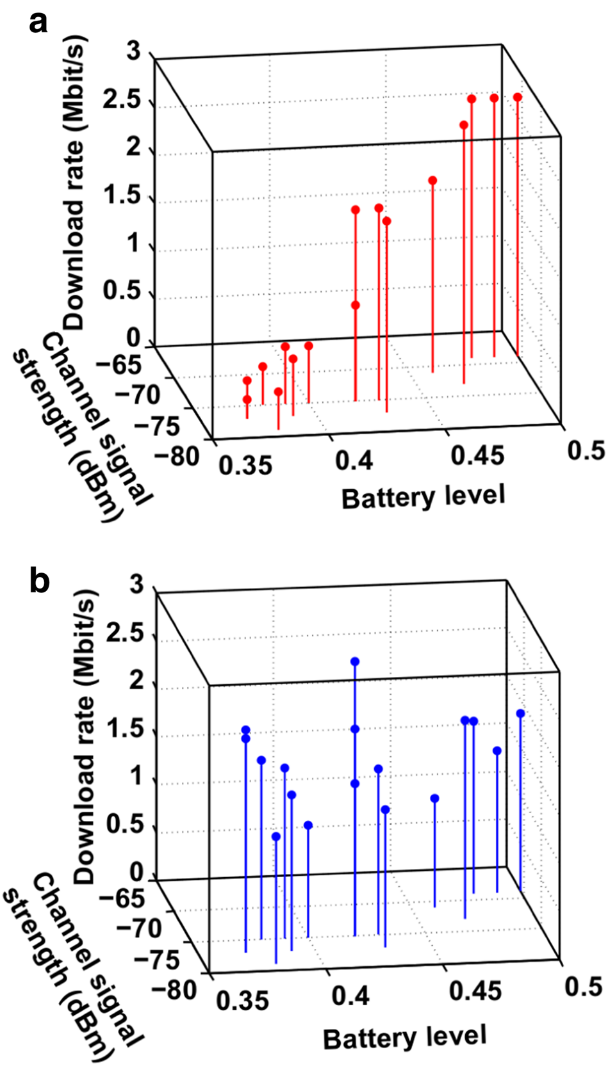


Fig. 9 Rate adaptation under varying channel conditions and battery levels. **a** BARA **b** Channel-based adaptation

The gain of BARA implementation is further verified by the results shown in Fig. 10. In comparison to the conventional scheme, MDP-based BARA does not only provide a smoother, more stable and more gradual rate switching, but also, it could effectively adjust the video bit rate depending on the residual battery. The bit rate adjustment can be analyzed using the time intervals t_1, t_2, t_3 and t_4 depicted in Fig. 10 (Bottom). The initial residual battery level, $\beta_{ini} = 0.5$ at the beginning of the viewing session.

– In the initial phase of the video playback (up to t_1), when the residual battery is high, BARA achieves a higher bit rate, comparing to existing channel-based adaptation.
– Between t_1 to t_2 period, as residual battery drops below a certain threshold, BARA starts to reduce the video bit rate to increase the battery lifetime. Consequently, we begin to

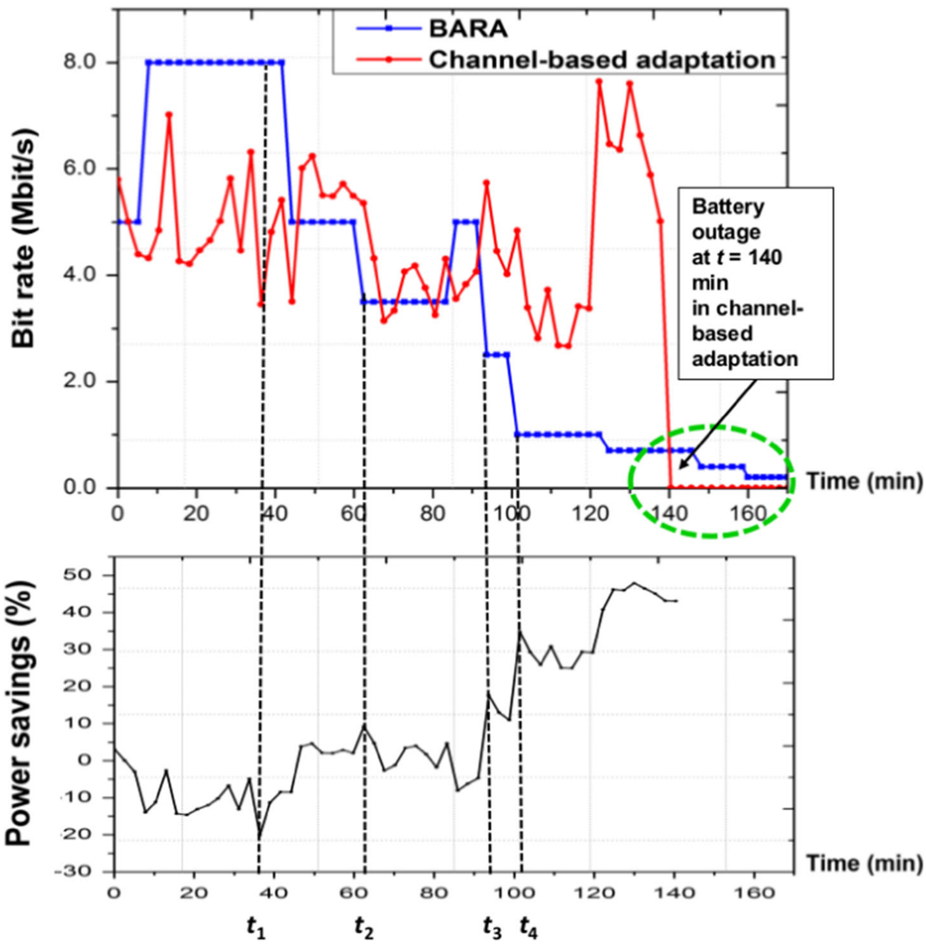


Fig. 10 Performance analysis. (Top) Battery lifetime comparison (Bottom) Power savings in BARA

see an increment in the power savings after t_1, and a sudden spike in power savings at t_2, as BARA makes another decision to drop the video rate.

- Between t_2 to t_3 period, both BARA and existing channel-based rate adaptation achieve almost similar video bit rates, possibly due to the fluctuation in channel conditions.
- After t_3, as the residual battery drops even more, BARA also reduces the video bit rate at par with it. From t_4 onwards, BARA adjusts its video bit rate even lower than the existing channel-based rate adaptation strategy. This action leads to an increased power savings after t_3. A significant increase up to 40% can be seen especially after t_4, as BARA keeps reducing the rate to extend the battery lifetime as long as possible and prolong the video playback.
- The ultimate gain is observed after 140 minutes of video playback. At this point, the battery outage in existing channel-based rate adaptation scheme results in a complete stop of the video playback. However, our proposed BARA successfully completes the entire video playback (up to around 170 minutes) by dynamically reducing the video bit rate up to the minimum allowed playback rate.

In summary, Fig. 10 points out that BARA has the potential to elongate the video viewing time by 20% and is able to save up to 40% of battery power, in comparison to the channel-based adaptation scheme.

6 Power-video quality trade-off

In this section we analyze the relation between battery power consumption and user QoE. Our attempt is to quantify the video quality level as perceived by the user using the Mean Opinion Score (MOS). As the SVC encoding scheme adopted in BARA, supports switching from a video layer to another, BARA strategy is to select the maximum number of layers based on the available battery. Each additional layer selected, provides different QoE level and power consumption, as each layer has different bit rate representations. We analyze the battery power versus QoE trade-off by considering either power or quality as constraint. We divide our optimization problem to achieve two objectives: (1) Minimizing power consumption for a given video with certain minimum acceptable visual quality and (2) Maximizing user QoE for a given video with certain maximum energy constraint. Each objective aims at selecting the best possible combination of input parameters to optimize the one considered criterion.

We adopt the method proposed in [32] to perform the trade-off analysis. In the analysis, we consider six layers or levels to match the six different video resolutions shown in Table 5. For a video duration of T sec, BARA performs the rate adaptation and selects the video level at every time period t_i. For simplicity purposes, we limit our analysis for a duration of $T = t_1 + t_2 + .. + t_6$. In the optimization problem $\mathbf{t} = [t_1 \ t_2 \ ... t_6]$ are the input variables. Using the MOS metric, the perceived visual quality, $\widehat{QoE}$, can be defined as [32]:

$$\widehat{QoE} = \frac{1}{T} \sum_{i=1}^{6} \int_{t_i} \mathrm{MOS}_i(t) \cdot dt, \tag{17}$$

where $\mathrm{MOS}_i(t)$ is the evolution of the QoE measured in the MOS scale along the ith period. The MOS scale is defined between 1 – 5, according to the ITU-T Recommendation ITU-T P.862.1 [26], with the values 1 = bad, 2 = poor, 3 = fair, 4 = good and 5 = excellent. If the normalized quality coefficients are defined as:

$$\mathbf{Y} = [Y_1 \ \ Y_2 \ \ Y_3 \ \ Y_4 \ \ Y_5 \ \ Y_6], \quad Y_i = \frac{MOS_i}{T} \tag{18}$$

Then, (17) can be expressed as:

$$\widehat{QoE} = \mathbf{Y} \cdot \mathbf{t}^T \tag{19}$$

As the battery consumption depends on the selected number of layers or video levels, the total power consumption, E_q for the whole duration of T can be expressed as:

$$E_q = \mathbf{Ba} \cdot \mathbf{t}^T, \tag{20}$$

where $\mathbf{Ba} = [Ba_1 \ Ba_2 \ ... Ba_6]$ is the vector of battery consumption normalized coefficients.

6.1 Minimizing power consumption with minimum acceptable visual quality

In this case, the objective is to minimize the power consumption. The minimum power consumption will be constrained by the QoE threshold, Z, that the user can tolerate, such that $\widehat{QoE} \geq Z$, where $Z_{min} \leq Z \leq Z_{max}$. According to the MOS metric, we define the $Z_{min} = \text{MOS}_1$ and $Z_{max} = \text{MOS}_6$. The optimization problem can be formulated as follows [32].

$$\begin{aligned} &\max \left\{-\mathbf{Ba} \cdot \mathbf{t}^T\right\} \\ &\text{s.t.} \begin{bmatrix} [\mathbf{I}] \\ \mathbf{Y} \end{bmatrix} \cdot \mathbf{t}^T \geq \begin{bmatrix} [0] \\ Z \end{bmatrix} \\ &\mathbf{i} \cdot \mathbf{t}^T = T, \end{aligned} \tag{21}$$

where $\mathbf{i} = [1\ 1\ 1\ 1\ 1\ 1]$ and $[\mathbf{I}]$ is 6×6 identity matrix, with $t_i \geq 0$ for all i.

We performed the optimization and the parameters gathered are shown in Table 7. To avoid specific details about battery characteristics (i.e., voltage and capacity in mAh), we measure the normalized available battery in seconds, as proposed in [32]. Figure 11a depicts the minimum power consumption by varying the QoE thresholds, generated based from Table 7. From Fig. 11a, it can be observed that, by selecting an appropriate quality threshold, good battery power-saving opportunities can be achieved. For example, by setting $Z = 3$ (fair visual quality) that is also equivalent to layer 3 video representation for Video-1, Video-2 and Video-4, layer 5 for Video-3 and layer 4 for Video-5 (bit rate up to 1.2 Mbit/s), the average battery power consumption is 800 seconds (20% reduction). The minimum power consumed in videos varies due to the different video content types. Video-1 and Video-4 contain higher and faster motion scenes thus are encoded in higher bit rates at each layer, whereas, other videos have medium or fewer motion scenes and are encoded in lower bit rates at each layer.

Table 7 Parameters for minimizing battery power consumption

Parameter	Value	
Video length, T (sec)	300	
QoE threshold, Z	$Z_{min} \leq Z \leq Z_{max}$	
Available battery (sec)	1000	
	Video	Values per layer
MOS_i	Video-1	[2.00 2.69 3.25 3.71 4.68 4.99]
	Video-2	[2.22 2.74 3.05 3.27 3.96 4.19]
	Video-3	[2.23 2.54 2.57 2.86 3.13 3.45]
	Video-4	[2.00 2.69 3.10 3.61 4.20 4.78]
	Video-5	[2.22 2.74 2.91 3.27 3.69 4.16]
Ba	[2 3 4 5 6 7]	

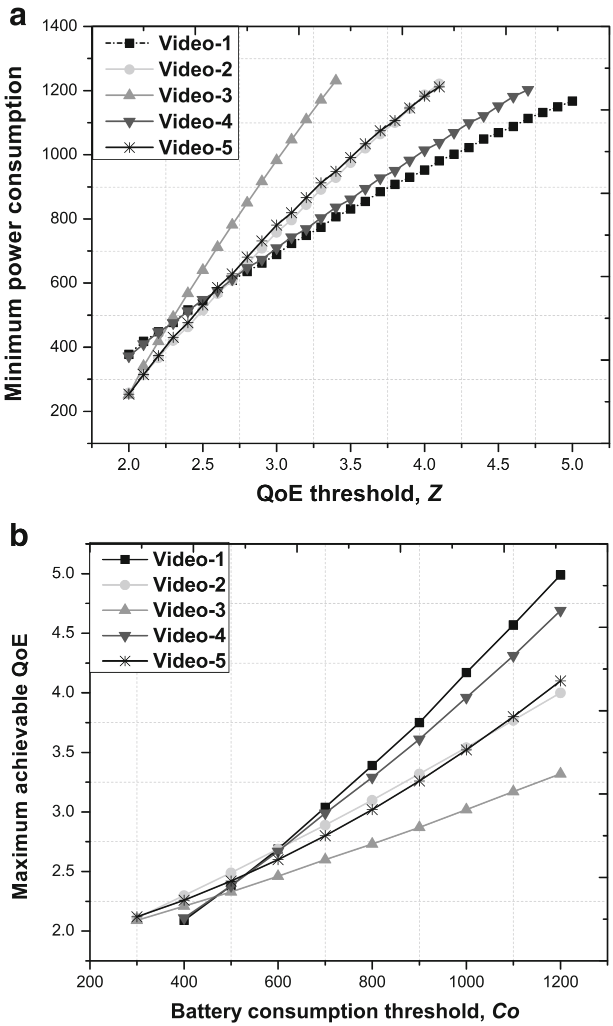


Fig. 11 Power-video quality trade-off. **a** Minimizing battery power with quality constraints **b** Maximizing quality with battery power constraints

6.2 Maximizing visual quality with maximum energy constraint

In this case, the objective is to maximize the MOS. The maximum achievable QoE will be constrained by the maximum battery consumption, Co. The optimization problem can be formulated as:

$$\max\left\{\mathbf{Y}\cdot\mathbf{t}^T\right\}$$

$$\text{s.t.}\begin{bmatrix}[\mathbf{I}]\\ -\mathbf{Ba}\end{bmatrix}\cdot\mathbf{t}^T\geq\begin{bmatrix}[0]\\ -Co\end{bmatrix} \tag{22}$$

$$\mathbf{I}\cdot\mathbf{t}^T=T,$$

Figure 11b presents the maximum quality achievable by limiting the battery power consumption. Similar to the concept explained in the previous Subsection 6.1, it is observed that the power consumption limitation directly affects the visual quality of the resulting video. With the focus on improving the video quality, the optimization process risks of having to consume higher battery power due to the higher layer or bit rate selection. Analyzing the obtained results from Fig. 11, it is possible to conclude that BARA selection strategy could reduce power consumption at the end users, thus help to improve the device battery lifetime, while maintaining reasonable QoE level.

7 Conclusion

As an alternative to DASH and other current adaptive streaming schemes, we have proposed a mobile video streaming scheme which jointly considers both the channel conditions as well as the mobile battery level. The proposed scheme aims at increasing the mobile devices battery lifetime and consequently extending the video playback duration for video streaming. We have successfully implemented our MDP-based rate adaptation approach while maintaining a certain QoE level. Our actual experiments and simulation results show that the proposed scheme can save more than 40% of battery power.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF−2016R1D1A1B03935633).

References

1. Ahmad H, Saxena N, Roy A, De P (2016) Extending video playback time with limited residual battery. IEEE Commun Lett 20(8):1659–1662
2. Akamai Technologies Inc. (2012) Encoding Best Practices for Akamai HD for iPhone and iPad. Akamai HD Network: White Paper
3. Akamai Play version 2.0 (2016) [Android Application Software], Octoshape Aps
4. Akamai Technologies (2017) CDN Learning Center. https://www.akamai.com/uk/en/cdn/. Accessed 5 Dec. 2017
5. Alsheikh MA, Hoang DT, Niyato D, Tan HP, Lin S (2015) Markov decision processes with applications in wireless sensor networks: a survey. IEEE Comm Surv Tutorials 17(3):1239–1267
6. BBC News (2014) Jimmy Page: How Stairway to Heaven was Written - BBC News. https://www.youtube.com/watch?v=DDo4CA13LbY. Accessed 9 Jan. 2016
7. Biermann VK (2011) Data Protection: Betrayed by our own data. http://www.zeit.de/digital/datenschutz/2011-03/data-protection-malte-spitz. Accessed 10 Jan. 2016
8. Blender Foundation (2008) Big Buck Bunny. http://www.bigbuckbunny.org/. Accessed 9 Jan. 2016
9. Blender Foundation (2012) Tears of Steel - Blender VFX Open Movie. https://www.youtube.com/watch?v=R6MlUcmOul8&list=PL6B3937A5D230E335&index=6. Accessed 9 Jan. 2016
10. Bokani A, Hassan M, Kanhere SS, Zhu X (2015) Optimizing HTTP-based adaptive streaming in vehicular environment using markov decision process. IEEE Trans Multimed 17(12):2297–2309
11. Bui DH, Lee K, Oh S, Shin I, Shin H, Woo H, Ban D (2013) Greenbag: Energy-efficient bandwidth aggregation for real-time streaming in heterogeneous mobile wireless networks. In: IEEE Real-Time Systems Symposium (RTSS), pp 57–67
12. Cao Y, Jiang T, Chen X, Zhang J (2016) Social-aware video multicast based on device-to-device communications. IEEE Trans Mob Comput 15(6):1528–1539
13. Cicco LD, Mascolo S, Palmisano V (2011) Feedback Control for Adaptive Live Video Streaming. In: ACM Conference on Multimedia Systems, pp 145–156

14. Cisco (2016) Cisco Visual Networking Index: Forecast and Methodology, 2015-2020 White Paper. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. Accessed 13 Oct. 2016
15. Claeys M, Latre S, Famaey J, Wu T, Leekwijck WV, Turck FD (2013) Design of a Q-learning-based client quality selection algorithm for http adaptive video streaming. In: Adaptive and Learning Agents Workshop, part of AAMAS2013 (ALA-2013), pp 30–37
16. Ding R, Muntean GM (2013) Device characteristics-based differentiated energy-efficient adaptive solution for video delivery over heterogeneous wireless networks. In: Proceedings of the IEEE Wireless Commission and Networking Conference, pp 4588–4593
17. Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Comm Mob Comput 13(8):1587–1611
18. Ejembi O, Bhatti SN (2014) Help save the planet: please do adjust your picture. In: ACM Proceedings of the ACM International Conference on on Multimedia, pp 427–436
19. Farahbakhsh R, Cuevas A, Ortiz AM, Han X, Crespi N (2015) How far is facebook from me? facebook network infrastructure analysis. IEEE Comm Mag 53(9):134–142
20. Grant AE, Meadows JH (2006) Communication Tech. Update, 10th edn. Focal Press, Boston, pp 126–127
21. Guruprasad R, Dev S (2015) Battery aware video delivery techniques using rate adaptation and base station reconfiguration. IEEE Trans Multimed 17(9):1630–1645
22. Hoque MA, Siekkinen M, Nurminen JK (2014) Energy efficient multimedia streaming to mobile devices - a survey. IEEE Commun Surv Tutorials 16(1):579–597
23. Hu W, Cao G (2015) Energy-aware video streaming on smartphones. In: IEEE International Conference on Computer Commission, pp 1185–1193
24. Kennedy M, Ventakaraman H, Muntean GM (2010) Battery and stream-aware adaptive multimedia delivery for wireless devices. In: IEEE 35th Conference on Local Computer Networks (LCN), pp 843–846
25. ICON Motosports (2012) Motorcycle vs. Car Drift Battle 2. https://www.youtube.com/watch?v=Te0V71sGoxA. Accessed 9 Jan. 2016
26. ITU-T Rec. P.862.1 (2003) Mapping Function for Transforming P.862 Raw Result Scores to MOS-LQO
27. Khan S, Schroeder D, El Essaili A, Steinbach E (2014) Energy-efficient and QoE-driven Adaptive HTTP Streaming over LTE. In: IEEE Wireless Commission and Networking Conference on Mobile and Wireless Networks, pp 2354–2359
28. Lauridsen M, Noel L, Sorensen TB, Mogensen P (2014) An empirical LTE smartphone power model with a view to energy efficiency evolution. Intel Tech J 18(1):172–193
29. Lee H, Lee Y, Lee J, Lee D, Shin H (2009) Design of a mobile video streaming system using adaptive spatial resolution control. IEEE Trans Consum Electron 5(3):1682–1689
30. Li Y, Markopoulou A, Apostolopoulos J, Bambos N (2008) Content-aware playout and packet scheduling for video streaming over wireless links. IEEE Trans Multimed 10(5):885–895
31. Li X, Dong M, Ma Z, Fernandes FC (2012) GreenTube: Power Optimization for Mobile Video Streaming via Dynamic Cache Management. In: Proceedings of the 20th. ACM International Conference on Multimedia, pp 279–288
32. Liberal F, Taboada I, Fajardo JO (2013) Dealing with energy-QoE trade-offs in mobile video. J Comput Netw Comm 2013:1–12
33. Mkwawa I, Lingfen S (2013) Battery voltage discharge rate prediction and video content adaptation in mobile devices over 3G access networks. J ZTE Commun 11(1):44–50
34. Penttinen JTJ (2016) LTE-A radio network. In: The LTE-Advanced Deployment Handbook: The Planning Guidelines for the Fourth Generation Networks. 1st edn. Wiley, UK, pp 131-133
35. PowerTutor (2013) [Open Source Application Software], PowerTutor.org
36. Ronen A (2014) Akamai Expands CDN to the RAN with Saguna. http://broabandtrafficmanagement.blogspot.kr/2014/09/akamai-expands-cdn-to-ran-with-saguna.html. Accessed 9 Jan. 2016
37. Roy A, De P, Saxena N (2015) Location-based social video sharing over next generation cellular networks. IEEE Comm Magaz 53(10):136–143
38. Saw LH, Somasundaram K, Ye Y, Tay AAO (2014) Electro-thermal analysis of lithium iron phosphate battery for electric vehicles. J Power Sources 249:231–238
39. Saxena N, Roy A (2015) Exploiting Multicast in LTE Networks for Smart Grids Demand Response. In: IEEE International Conference on Commission (ICC), pp 3155–3160
40. Spitz M (2011) CRAWDAD dataset spitz/cellular (v. 2011-05-04). http://crawdad.org/spitz/cellular/20110504. Accessed 10 Jan. 2016

41. Sun Y, Yin X, Jiang J, Sekar V, Lin F, Wang N, Liu T, Sinopoli B (2016) CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction. In: Proceedings of the 2016 ACM Special Interest Group on Data Commission (SIGCOMM) Conference, pp 272–285
42. Transformers 4 - Optimus vs. Lockdown (2014) https://www.youtube.com/watch?v=LMMP4ILcalI. Accessed 9 Jan. 2016
43. Wang X, Chen M, Kwon TT, Yang LT, Leung VCM (2013) AMES-cloud: A framework of adaptive mobile video streaming and efficient social video sharing in the clouds. IEEE Trans Multimed 15(4):811–820
44. Xing M, Xiang S, Cai L (2014) A real-time adaptive algorithm for video streaming over multiple wireless access networks. IEEE J Sel Areas Comm 32(4):795–805
45. Yan Z, Chen CW (2016) RnB: Rate and brightness adaptation for rate-distortion-energy tradeoff in HTTP adaptive streaming over mobile devices. In: Proceedings of the 22nd. ACM International Conference on Mobile Computing and Networking (MobiCom), pp 308–319
46. Yin X, Jindal A, Sekar V, Sinopoli B (2015) A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. ACM SIGCOMM Comput Comm Rev 45(4):325–338
47. Zhou C, Lin CW, Guo Z (2016) mDASH: A markov decision-based rate adaptation approach for dynamic HTTP streaming. IEEE Trans Multimed 18(4):738–751
48. Zhu H, Cao Y, Wang W, Liu B, Jiang T (2015) QoE-aware resource allocation for adaptive device-to-device video streaming. IEEE Netw 29(6):6–12

Hasnah Ahmad is currently working towards her Ph.D. as a researcher with the Mobile Ubiquitous System Information Center in Sungkyunkwan University, South Korea. She received her B. Eng. degree in Electrical and Electronic Engineering and M. Eng. degree in Electronic Engineering from University of Science, Malaysia and Royal Melbourne Institute of Technology (RMIT), Australia, in 2000 and 2008, respectively. Her research interests include analysis and optimization of wireless and mobile communications networks, as well as energy and qualityaware adaptive multimedia delivery.

Navrati Saxena is an Associate Professor in the Department of Software, Sungkyunkwan University (SKKU), South Korea. She is also the Director, Mobile Ubiquitous System Information Center (MUSIC) at SKKU. Prior to joining SKKU, she was an Assistant Professor in Amity University India and a visiting researcher at the University of Texas at Arlington. She completed her Ph.D. from the Department of Information and Telecommunication, University of Trento, Italy. Her research interests involve 4G/5G wireless, IoT and smart environments. She serves as a Guest Editor and Technical Program Committee member of international journals and conferences. She has co-authored one book (Taylor & Francis) and published more than 40 international journals.

Abhishek Roy is currently working in the Networks Division, Samsung Electronics, South Korea. He received his Ph.D. in 2010 from Sungkyunkwan University and his M.S. in 2002 from the University of Texas at Arlington. His research interests include mobility and resource management in 4G/5G wireless systems, IoT, and smart grids. He serves as a Guest Editor and Technical Program Committee member of many international journals and conferences. He has co-authored one book (Taylor & Francis) and published more than 40 international journals.

Pradipta De is an Assistant Professor at the Department of Computer Science, Georgia Southern University. He is also affiliated to StonyBrook University (SUNY) Korea, a remote campus of StonyBrook University in Songdo, South Korea, as a Research Assistant Professor. He received his Ph.D. from StonyBrook University in 2007 and did his undergraduate in Computer Science and Engineering from Jadavpur University, Kolkata. Pradipta worked as a Research Staff Member at IBM Research – India from 2005 to 2012.