Achieving Viewing Time Scalability in Mobile Video Streaming Using Scalable Video Coding

Cheng-Hsin Hsu
Deutsche Telekom R&D Lab USA
5050 El Camino Real #221
Los Altos, CA 94022

Mohamed Hefeeda
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

ABSTRACT

We propose a general quality-power adaptation framework that controls the perceived video quality and the length of viewing time on battery-powered video receivers. The framework can be used for standalone video devices (e.g., DVD players and notebooks) as well as mobile receivers obtaining video signals from wireless networks (e.g., mobile TV and video streaming over WiMAX). Furthermore, the framework supports both live streams (e.g., live TV shows) and pre-encoded video streams (e.g., DVD movies). We present an adaptation algorithm for each mobile device to determine the optimal substream that can be received, decoded, and rendered to the user at the: (i) highest quality for a given viewing time, and (ii) longest viewing time for a given quality without exceeding the battery level constraint. We instantiate this framework and work out its details for mobile video broadcast networks. In particular, we propose a new video broadcast scheme that enables mobile video devices to efficiently adapt scalable video streams and achieve power saving proportional to the bit rates of the received streams. We implement the proposed framework in an actual mobile video streaming testbed and we conduct experiments using real video streams broadcast to mobile phones. These experiments show the practicality of the proposed framework and the possibility of achieving viewing time scalability. For example, on a mobile phone receiving and decoding the same video program, a viewing time in the range from 4 to 11 hours can be achieved by adaptively controlling the frame rate and visual quality of the video stream.

Categories and Subject Descriptors
C.4 [Performance of Systems]: Modeling Techniques

General Terms
Design

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

1. INTRODUCTION

Video streaming to mobile devices has become very popular, as modern mobile devices are powerful enough to render video contents that were only feasible to stationary workstations a few years ago. In fact, we have been witnessing a new usage pattern of mobile devices, which clearly indicates that more and more users consume video contents using their mobile devices [27]. Several studies, e.g., [22], reveal that unlike the CPU speed, memory size, and disk capacity, which achieved exponential growth in the past few decades, the battery technology has been lagging and only achieved a linear growth. Therefore, insufficient battery capacity imposes stringent energy constraints on mobile video streaming, which requires mobile device designers to take energy consumption as one of the main design considerations.

Since most traditional multimedia equipments have no energy constraints, users typically seek the optimal viewing experience in terms of resolution, frame rate, and picture quality. On mobile devices, however, users must consider the battery lifetime as a new dimension of their viewing experience because it determines the maximum viewing time. We define the battery lifetime as the amount of time a user can continuously watch mobile video streaming without charging or replacing the battery. To illustrate the importance of battery lifetime, consider a user, Amy, who wants to watch a 30-min TV episode using her cellular phone that only has remaining battery capacity for watching the show for 25 min. Most current mobile devices cannot adapt to the energy constraint, because video streams coded by traditional, nonscalable, coders must be received and decoded in their entirety. Consequently, Amy would watch the episode for 25 min, and then miss the (most important) ending, which significantly degrades her viewing experience and may drive her away from the mobile video streaming service. From Amy’s point of view, finishing this episode, but at a slightly lower video quality is probably a much better experience. Nonetheless, such adaptation is not possible on current mobile devices.

In this paper, we study the problem of: (i) predicting battery lifetime of mobile devices for video streaming services, and (ii) allowing users to opt for longer battery lifetime by watching the video at a lower picture quality. Our goal is to design a systematic and intuitive method for users to decide on the desired perceived quality and viewing time. To achieve this, we leverage scalable video coders (SVCs) that
encode each video content into a single video stream with multiple layers [24]. Scalable coded streams can be transmitted and decoded at various bit rates, and this is done by simple manipulations that extract substreams of a few layers from the original stream. Each substream can be rendered at a lower perceived quality than the original (complete) stream. SVC coded streams are conventionally used to support heterogeneous devices in terms of communication bandwidth, display resolution, and CPU power. In this paper, we highlight another important benefit of SVC: enabling viewing time scalability. That is, we show that SVC can provide users with a control knob to prolong viewing time of videos on mobile devices by reducing the perceived video quality. We present quantitative models that map basic energy consumption and video bit rate to intuitive, and easy-to-understand, performance metrics such as the length of viewing time in hours and expected perceived quality in MOS (mean opinion score). More importantly, we propose an adaptation algorithm for each mobile device to determine the optimal substream that can be received, decoded, and rendered to the user at the: (i) highest quality for a given viewing time, and (ii) longest viewing time for a given quality without exceeding the battery level constraint.

The contributions of this paper are as follows.

- We propose a general quality-power adaptation framework, which can be used to systematically control the perceived video quality and the length of viewing time on battery-powered video receivers. The framework is general because it can be used for standalone receivers (e.g., DVD players and notebooks) as well as mobile receivers obtaining video signals from wireless networks (e.g., mobile TV and video streaming over WiMAX). Furthermore, the framework supports both live streaming (e.g., live TV shows) and pre-encoded streams (e.g., DVD movies).

- We propose a novel video broadcast scheme that efficiently enables mobile video devices to implement the adaptation framework. We analytically analyze this scheme and its power consumption. This broadcast scheme is of interest in its own right, as it allows heterogeneous mobile devices to receive different versions of the video stream and achieve power saving commensurate to the bit rates of the received streams. The scheme is designed to broadcast SVC video streams that efficiently support temporal and visual quality scalabilities at the same time.

- We propose a CPU power consumption model for decoding scalable video streams. Our model is an integration of previous models that were proposed in different contexts.

- We implement the proposed framework and models in a real mobile TV testbed and we conduct experiments using actual video streams broadcast to mobile phones. These experiments show the practicality of the proposed framework and the possibility of achieving viewing time scalability using scalably-coded video streams.

The rest of this paper is organized as follows. We provide an overview of the proposed framework in Sec. 2. We present the details of the proposed video broadcast scheme in Sec. 3. We study the communication power consumption in the same section. In Sec. 4, we describe the CPU power consumption and perceived quality models. We give our quality-power adaptation algorithm in Sec. 5. We present experimental results in Sec. 6. We summarize the related work in the literature in Sec. 7, and we conclude the paper in Sec. 8.

## 2. OVERVIEW

In this section, we present an overview of the proposed quality-power adaptation framework as well as the quality and power models needed for the framework. For quick reference, we list all symbols used in the paper in Table 1, where bold symbols denote sets, e.g., \( t \) denotes the set of frame rates while \( t \) is a specific frame rate.

### 2.1 Quality-Power Adaptation Framework

We propose an adaptation framework, which gives users with a control knob to prolong viewing time of videos on mobile devices by reducing the video quality. The framework uses scalable video coders to encode each video into a scalable stream that can be decoded at various bit rates. That is, several substreams can be extracted from the scalable stream, where each substream has a lower bit rate, incurs a lower decoding complexity, and is rendered at a lower perceived video quality than those of the complete (original) stream. Hence, receiving, decoding, and rendering a substream at a lower video quality consumes less energy, and prolongs battery lifetime and viewing time.

The goal of our framework is to take users’ desired viewing time (or perceived quality) as input, and systematically compute the optimal video substream to receive and render so that the desired viewing time (or perceived quality) is met without violating the given energy constraint. Notice
that the framework takes one desired performance metric, either viewing time or perceived quality and computes the expected value of the other metric. With this framework, Amy, in the illustrative example in Sec. 1, can specify the desired viewing time as 30 mins, and would not miss the ending of the TV episode, which is currently not possible on mobile devices. Fig. 1 illustrates the proposed framework. In the core of the framework is the Quality-Power Adaptation Algorithm (QPAA), which collaborates with four system models in order to produce the desired video adaptation strategy.

The adaptation framework takes several inputs, which we categorize into two classes: (i) video characteristics that describe the structure of coded streams, and (ii) mobile device characteristics that describe the conditions and efficiency of mobile devices. Examples of the inputs in the first class include supported frame rates \( t \) and quantization steps \( \delta \), which can be specified at encoding time or extracted from pre-encoded streams. Examples of mobile device characteristics include current battery level \( \omega \), background power consumption \( p_b \), communication power consumption \( \lambda \), and CPU efficiency factor \( \psi \), which are device dependent and can either be directly measured or inferred by some experiments. The adaptation framework generates the adaptation strategy that consists of two parts: (i) the optimal version of substream to render specified by the chosen frame rate \( t \) as well as the quantization step \( \delta \), which determines the amount of consumed resources, and (ii) the output to control the amount of supplied resources, which include CPU voltage level \( v \) and available CPU cycles \( y \). Applying these outputs to proper components leads to optimal stream adaptation.

### 2.2 Power and Quality Models

We briefly introduce the models used in the adaptation framework; more details are given in later sections.

The first model is the perceived quality model, which maps a given substream to its perceived video quality. The other three models specify the total power consumption of a mobile device, and they are: CPU, communication, and I/O and background models. The CPU power consumption \( p_c \) is the power consumed for decoding video streams. More CPU cycles in general lead to higher power consumption, and thus we write \( p_c(y) \) as a function of \( y \), which is the average number of CPU cycles per second used by video decoders. The communication power consumption \( p_n \) is the energy consumed for receiving video streams over communication networks. We consider mobile devices that can put their network interfaces into sleep for a fraction of time, and we define the power saving \( \gamma \) as the fraction of the time a network interface is in sleep mode over the total time. Since higher power saving \( \gamma \) means lower communication power consumption, we write \( p_n(\gamma) \) as a function of \( \gamma \).

The third part for the system power consumption is the I/O and background. The I/O power model calculates the...
In a mobile video broadcast network, a base station broadcasts multiple video streams (alternately referred to as TV channels) in layers. This allows the efficient support of scalable video streams, and can lead to energy saving because of time slicing, which is a technique used to save energy by turning off devices temporarily when they are not in use.

### 3.2 Flexible Mobile Video Broadcast (FMVB)

Flexible Mobile Video Broadcast (FMVB) is a scheme that enables different mobile devices to adapt scalability modes and can lead to efficient support of scalable video streams. However, our previous works present optimal/near-optimal broadcast schemes to send multiple scalable video streams. However, our previous works do not construct burst time for each stream, so that all video data can be delivered on time.

The broadcast scheme cannot have burst collisions, which happen when two or more bursts have nonempty intersection. Thus, we estimate the effectiveness of time slicing on energy saving achieved by mobile devices because of time slicing as nontrivial compared to burst lengths. We denote the power overhead duration and is denoted by $T$.

In this paper, we focus on dedicated video broadcast networks, such as MediaFLO (Forward Link Only Layer 16), which concurrently offer TV and mobile video services to many users. We propose a new broadcast scheme that enables the adaptation of scalable video streams along both the temporal and quality scalability dimensions. The second limitation is that video streams are assumed to have simple, linear dependencies among layers in order to achieve higher coding efficiency, and it is required in several broadcast standards such as DVB-H. Mobile devices must turn on their network interfaces slightly before the burst time, because it takes some time to wake up and synchronize the circuitry itself.

#### FMVB Scheme

We now present the FMVB scheme. We consider a quite simple case, in which the current burst is the current layer of the scalable stream. To model this dependency, we define $R_l$ as the sending rate of layer $l$ that can be extracted from the original (complete) stream. Let $r$ be the bit rate of the original scalable stream with $R/r$ be the substream sending rate. We let $L$ be the set of all supported layers. We no-longer require $R_l$ to be the bit rate of the original scalable stream.

![Figure 2: The proposed broadcast scheme.](image_url)

The broadcast network bandwidth is $R_l$. We consider the following constraints:

1. $R_l \leq R$ for all $l \in L$.
2. $R_l \geq r$ for all $l \in L$.
3. $R_l = R$ for all $l \in L$.

The dependency model among layers is defined by $\delta_l$, which also includes $\delta_l \in L$. We no-longer require $\delta_l$ to be the layer that supports frame rates and $\delta_l$ to represent the layer that supports frame rates.

We let $\delta_l$ be the set of all supported layers. The dependency among layers is defined by $\delta_l$. That is, $\delta_l = \{l \in L : R_l = R\}$ and $\delta_l \subseteq L$. We let $\delta_l = \{l \in L : R_l = R\}$. We no-longer require $\delta_l$ to be the set of all supported layers.

### Sending Rate Calculation

The sending rate $R_l$ for layer $l$ is calculated as follows:

$$ R_l = R \cdot \frac{\delta_l}{\delta_l} $$

where $\delta_l$ is the subset of layers that support the same bit rate $R$.

### Complex Inter-layer Dependencies

Flexible Mobile Video Broadcast (FMVB) allows layers to be encoded at different bit rates and does not assume that layers are linearly cumulative. Thus, FMVB offers substantial flexibility in encoding rates.

More importantly, FMVB is more suitable to modern scalable video coders, such as H.264/SVC [24], which support multiple scalability modes, and complex inter-layer dependencies. This limits inter-dependency among layers, which was not possible in our previous schemes in [13–16].

Flexible Mobile Video Broadcast (FMVB), also allows layers to be encoded at different bit rates and does not assume that layers are linearly cumulative. Thus, FMVB offers substantial flexibility in encoding rates.

The inter-layer dependencies among layers in order to achieve higher coding efficiency, and it is required in several broadcast standards such as DVB-H. Mobile devices must turn on their network interfaces slightly before the burst time, because it takes some time to wake up and synchronize the circuitry itself.
smallest quantization step, and

Among the considered \( \delta \) scalable video coders. Various frame rates and quantization steps render a substream with frame rate \( R_S \), where \( \delta \) is the layer bit rate of layer \( l \), and quantization step \( r \) is given as

\[
\delta = \bar{\delta}/r
\]

where \( \bar{\delta} \) is the layer bit rate. This inequality shows that the aggregate bit rate of all dependent layers of layer \( l \) is given as:

\[
\sum_{i=1}^{E} \delta_{i} \leq \bar{\delta}/r
\]

This is because the start time of each burst is ensured to be from Eq. (4), we need to know the layer bit rate. This differentiation is not possible with other broadcast schemes, as mentioned in Sec. 3.1.

This is because the start time of each burst is ensured to be from Eq. (4), we need to know the layer bit rate. This differentiation is not possible with other broadcast schemes, as mentioned in Sec. 3.1.

Furthermore, the energy saving achieved by mobile devices that turn off their network interfaces. We also show that the saving in terms of the fraction of time mobile devices can stream. In the next theorem, we derive the relative power adaptation framework.

Theorem

Proof.

From Eq. (4), we need to know the layer bit rate. This differentiation is not possible with other broadcast schemes, as mentioned in Sec. 3.1.

Furthermore, the energy saving achieved by mobile devices that turn off their network interfaces. We also show that the saving in terms of the fraction of time mobile devices can stream. In the next theorem, we derive the relative power adaptation framework.

Theorem

Proof.

This is because the start time of each burst is ensured to be from Eq. (4), we need to know the layer bit rate. This differentiation is not possible with other broadcast schemes, as mentioned in Sec. 3.1.

Furthermore, the energy saving achieved by mobile devices that turn off their network interfaces. We also show that the saving in terms of the fraction of time mobile devices can stream. In the next theorem, we derive the relative power adaptation framework.

Theorem

Proof.

This is because the start time of each burst is ensured to be from Eq. (4), we need to know the layer bit rate. This differentiation is not possible with other broadcast schemes, as mentioned in Sec. 3.1.

Furthermore, the energy saving achieved by mobile devices that turn off their network interfaces. We also show that the saving in terms of the fraction of time mobile devices can stream. In the next theorem, we derive the relative power adaptation framework.

Theorem

Proof.

This is because the start time of each burst is ensured to be from Eq. (4), we need to know the layer bit rate. This differentiation is not possible with other broadcast schemes, as mentioned in Sec. 3.1.

Furthermore, the energy saving achieved by mobile devices that turn off their network interfaces. We also show that the saving in terms of the fraction of time mobile devices can stream. In the next theorem, we derive the relative power adaptation framework.

Theorem

Proof.
The number of CPU cycles per second as:

scalability is typically achieved by using different quantization steps during video compression: smaller steps produce higher visual quality scalability. The temporal scalability is considered, and the model parameters in Eq. (5) are estimated using curve fitting. For example, the model parameters for the Crew sequence are estimated, and the model parameters in Eq. (5) are estimated using curve fitting.

Finally, since the power saving device can turn off its mobile TV chip, the communication voltage scaling (DVS) mechanism, and can adjust the clock frequency, and the effective capacitance is linearly proportional to the voltage demand.

We adopt one variation that supports temporal decoders. We consider several variations of a complexity model for scalable video decoders. One reason is that traditional, nonscalable, video decoders have little room for reducing computational complexity: the whole video stream can be decoded for playout. This, however, is not true for scalable coded streams, where several substreams have to be decoded in order to save CPU cycles. A recent work [20] proposes a rate model allows us to compute the power saving of CPU cycles.

\[ P = \frac{1}{t} \left( \gamma \delta Y + (1 - \gamma Y) \right) \]

where \( \gamma \) is the fraction of time a layer is considered, and the model parameters in Eq. (5) are estimated using curve fitting. For each video sequence, 16 substreams are considered, and the model parameters in Eq. (5) are estimated using curve fitting. For example, the model parameters for the Crew sequence are estimated, and the model parameters in Eq. (5) are estimated using curve fitting.

In the past few years, several quality metrics for scalable video quality have been proposed. Most of these metrics, such as PSNR (peak signal-to-noise ratio), assume that the resolution and frame rate are fixed. In this paper, we propose a quality model that considers two scalability modes: quality scalability and visual quality scalability. The quality model considers two scalability modes: quality scalability and visual quality scalability. The temporal scalability is considered, and the model parameters in Eq. (5) are estimated using curve fitting.

\[ Q = \frac{1}{t} \left( \gamma \delta Y + (1 - \gamma Y) \right) \]

where \( \gamma \) is the fraction of time a layer is considered, and the model parameters in Eq. (5) are estimated using curve fitting. For each video sequence, 16 substreams are considered, and the model parameters in Eq. (5) are estimated using curve fitting. For example, the model parameters for the Crew sequence are estimated, and the model parameters in Eq. (5) are estimated using curve fitting. For each video sequence, 16 substreams are considered, and the model parameters in Eq. (5) are estimated using curve fitting.
Parameter Estimation and Results

We explain how to derive them in the following. We design the background power consumption, \( b \), in Eq. (1), which is the power consumption of the mobile TV chip, (ii) in Eq. (8), which represent the power efficiency, (iii) in Eq. (4), which is the overhead duration, (iv) \( p \) of the CPU. We explain how to derive them in the following.

We conduct the experiment to infer the mobile TV chip power consumption using a monitoring program called Juice, which has been shown to be fairly accurate and is comparable to external instruments [2]. After each 3.5-hr experiment, we fully charge the battery, and repeat the experiment by capturing a 10-minute news clip from a digital cable service.

We encode the video using an H.264/AVC encoder at bit rate 450 kbps, and the audio using an eAAC+ encoder at 32 kbps. We make the base station broadcast the coded video to watch this video, and we measure its power consumption at 700 mW [3]. To systematically infer the power consumption, we compute the CDF (cumulative distribution function) of all power consumption measurements for each broadcast scheme. We then plot CDF curves of three sample broadcast schemes in Fig. 5(b). This figure shows the power consumption difference between any two of the broadcast schemes is completely attributed to the number of overhead spikes.

This figure confirms that the broadcast schemes with inter-burst periods: 250, 500, 1000, 2000, and 3000 msec. We collect the power consumption of each broadcast scheme throughout the experiment.

To derive the mobile TV chip power consumption, we compute the CDF (cumulative distribution) against that of other schemes. We derive the overhead duration, \( o \), as \( 2(\lambda - \mu) \) second, which is much more often than the chosen inter-burst time period. We consider five different broadcast schemes with different parameters in or-
We use Eq. (7) to compute the number of CPU cycles required for decoding video streams, and calculate the CPU power efficiency factor $\psi$ in two steps. We first deduct the background and communication power consumption over 3.5-hr broadcast in above derivations, a much shorter, only a few secs, measurement period would be sufficient.

We empirically compute the CPU power efficiency factor $\psi$, which is the average $\psi$ values in Fig. 6(b). This figure shows that the CPU power efficiency factor $\psi$ are fairly close. We let $\psi = 3$ for individual broadcast schemes with different inter-burst periods, and plot the value derived from different broadcast schemes.

Figure 7: The tradeoff between viewing time and perceived quality for different scalabilities.

Figure 6: Estimating the background and CPU related parameters of the Nokia N96 phone.

(g) Temporal scalability 

(h) Quality scalability 

We study the tradeoff between quality and viewing time in the experiments, because they are not available on mobile devices yet. Developing a scalable video decoder for mobile TV networks, and (ii) decode video streams.

Developing a scalable video decoder for mobile devices is one of our future works.
enables users to use scalable streams, where each substream is specified by the frame rate and the quantization step. We then connect all substreams that lead to the complete scalable streams, where each substream is specified by the frame rate and the quantization step.

The authors of [29] propose a battery prediction method for mobile devices based on the observation that battery discharge curves under different workloads have a displacement of the actual discharge curve from the reference curve. This reference curve is then used at run-time to compute a polynomial function that describes the battery lifetime.

A number of works, e.g., [23, 29], empirically derive the battery depletes. This reference curve is then used at run-time to compute a polynomial function that describes the battery lifetime. The authors of [29] propose a battery prediction method for mobile devices based on the observation that battery discharge curves under different workloads have a displacement of the actual discharge curve from the reference curve. This reference curve is then used at run-time to compute a polynomial function that describes the battery lifetime.

In summary, the experiments illustrate that the proposed quality-power adaptation framework can derive the perceived quality and the viewing time of each substream from the system parameters of N96 phones and the video related to the service. We first plot the video quality and viewing time of Service I in Fig. 7(a). This figure clearly shows that reducing frame rate is a more effective way to prolong viewing time. For example, considering a frame rate of 75 fps and a quantization step of 3, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

We plot a similar figure for Service II in Fig. 7(b). Note that higher layers have smaller quantization steps, and this figure also shows that more layers lead to higher video quality and shorter viewing time. For example, considering a quantization step of 75 fps and a frame rate of 30 fps, Service II can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

In summary, the experiments illustrate that the proposed quality-power adaptation framework can derive the perceived quality and the viewing time of each substream from the system parameters of N96 phones and the video related to the service. We first plot the video quality and viewing time of Service I in Fig. 7(a). This figure clearly shows that reducing frame rate is a more effective way to prolong viewing time. For example, considering a frame rate of 75 fps and a quantization step of 3, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

We plot a similar figure for Service II in Fig. 7(b). Note that higher layers have smaller quantization steps, and this figure also shows that more layers lead to higher video quality and shorter viewing time. For example, considering a quantization step of 75 fps and a frame rate of 30 fps, Service II can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.

The figure shows that mobile devices that receive fewer layers can prolong viewing time: mobile devices that receive one layer can watch TV for 11 hrs, while mobile devices that receive one layer can watch TV for 4 hrs. Furthermore, the four layers can only watch TV for 4 hrs. Moreover, the authors of [29] empirically derive the quality-power adaptation framework, we can derive the perceived quality and the viewing time, e.g., the substream with the target MOS score of 50, Service I achieves viewing time of 11 hrs, while Service II only achieves 9 hrs.
our framework predicts battery lifetime and may take some time to converge. In contrast, these two works are not suitable to mobile video streaming applications at an expense of lower quality of service (QoS). A similar approach is proposed by the authors of [23].

Resource provision to support this tradeoff occurs either through (i) adapting the application to the available resource provided by hardware, and (ii) scaling the resource upon request. They can be classified into two groups [8]: (i) scaling the quality of the video stream to render in order to optimize the user experience and predicts the CPU requirements of decoding the coded video stream, and insert control messages in these streams. We employed scalable video streams to support this tradeoff. We studied the problem of controlling the viewing time to aid users in finding the most appropriate version of the video stream to render in order to optimize the user experience. The proposed framework is quite general, and can be applied to TV, multimedia, and interactive applications to trade quality for battery life. They conduct experiments to support this tradeoff. We studied the problem of controlling the viewing time to aid users in finding the most appropriate version of the video stream to render in order to optimize the user experience. The proposed framework is quite general, and can be applied to TV, multimedia, and interactive applications to trade quality for battery life. They conduct experiments to support this tradeoff.

The work in this paper can be extended in multiple directions. The authors of [1] propose to save energy. The stream analysis is done by decoding the video streams, and insert control messages in these streams to instruct mobile devices adjusting their CPU frequencies based on the decoder complexity model. This model takes account of modeling energy consumption under different QoS levels. The authors of [31, 32] observe that most CPUs support a few discrete frequencies and power voltages, and they design a DVS algorithm that can periodically derive energy consumption curves. The authors of [21] present quality adaptation algorithms based on the empirically derived energy consumption curves. The authors of [25, 31, 32] and offline [1, 17] algorithms. Briefly described are the following approaches:

- Based on a decoder complexity model. This model takes into account the perceived quality. We employed scalable video streams to support this tradeoff. We studied the problem of controlling the viewing time to aid users in finding the most appropriate version of the video stream to render in order to optimize the user experience. The proposed framework is quite general, and can be applied to TV, multimedia, and interactive applications to trade quality for battery life. They conduct experiments to support this tradeoff.

- DVS algorithms can further be categorized into online and offline algorithms. DSOM algorithms have been proposed in several works to aid users in finding the most appropriate version of the video stream to render in order to optimize the user experience. The proposed framework is quite general, and can be applied to TV, multimedia, and interactive applications to trade quality for battery life. They conduct experiments to support this tradeoff.

- Based on profiling. The profiling data is then used by a middle-ware for obtaining video signals from wireless networks. In addition, we plan to implement our quality-power adaptation framework on various mobile wireless platforms, and conduct more experiments. The experimental results will allow us to improve our framework and to identify limitations. In addition, we plan to implement our quality-power adaptation framework on various mobile wireless platforms, and conduct more experiments. The experimental results will allow us to improve our framework and to identify limitations.