

MULTI-LAYER VIDEO BROADCASTING WITH LOW CHANNEL SWITCHING DELAYS

Cheng-Hsin Hsu and Mohamed Hefeeda

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

ABSTRACT

Modern mobile devices, despite their small sizes, can run many multimedia applications that were only possible to stationary workstations. Mobile devices, however, have quite heterogeneous resources, which poses a challenge to mobile TV broadcast networks. We study the problem of broadcasting multi-layer video streams to mobile devices with heterogeneous resources. We propose broadcast schemes that allow each mobile device to selectively receive a few (or all) layers of the complete video streams, and achieve proportional energy saving. We also propose a broadcast scheme that achieves low channel switching delay, which is important to user experience. We analytically analyze the performance of the proposed schemes. Most importantly, we have implemented them in a real mobile TV testbed. We conduct extensive experiments to show the practicality and efficiency of the proposed schemes. The experimental results show that channel switching delays less than 200 msec and energy saving between 75% and 95% are possible under typical system parameters of mobile TV networks.

Index Terms— Mobile TV, DVB-H, time slicing, quality of service, energy saving, channel switching delay.

1. INTRODUCTION

Recent advances in technology have provided mobile devices tremendous communication and computational powers that were only possible to stationary computers. Modern mobile devices, despite their small sizes, can run many multimedia applications, including games and music players that are already popular nowadays, and video broadcast that is still emerging. Video broadcasting, better known as TV, services allow users to watch TV programs anytime at anyplace. Market forecasts reveal that mobile TV will catch up with gaming and music, and become the most popular application on mobile devices: more than 140 million subscribers worldwide, and multi-billion dollar revenues in North America by 2011 [1]. Mobile TV services can be delivered using cellular or dedicated broadcast networks. We only consider dedicated broadcast networks, because they achieve very high spectrum

efficiency: a large number of users can be supported by a single broadcast tower.

Mobile devices are heterogeneous in terms of resources, which include screen resolution, decoder capability, and battery capacity. For example, although notebook computers can display 720p (1280x720) videos, many smart phones only have QVGA (320x240) displays. It is challenging to *concurrently* support these mobile devices: broadcasting TV programs in QVGA resolution results in terrible video quality on notebook computers, while broadcasting in 720p resolution results in higher overhead, and thus higher energy consumption, on smart phones with no visible quality improvement. More importantly, broadcasting in 720p resolution could deny smart phones, that are not computationally powerful enough, from mobile TV services. To partially cope with this problem, network operators could broadcast every TV channel in multiple versions, where each version is suitable to a mobile device. This is called *multi-version* video broadcasting, which is not bandwidth efficient: fewer number of TV channels can be concurrently broadcast. This inefficiency becomes even more severe if we categorize mobile devices into groups by not only device models but also working conditions of the same device. For example, mobile devices with low battery levels or in poor wireless channel conditions may prefer to receive at a lower bit rate to save energy and/or reduce bit error rate.

Multi-layer video broadcasting, in contrast, enables network operators to support many mobile devices without exhausting network bandwidth. This is achieved by using scalable video coders to encode every TV channel into a single stream with multiple layers, and broadcast each layer only once. Such a coded stream is scalable because several substreams, consist of one or a few layers, can be extracted from the complete stream and are still decodable for meaningful playback. Each mobile device can then choose and render the substream that is most suitable to its capability. Multi-layer video broadcasting, however, is quite challenging for the base station, which concurrently broadcasts multiple TV channels. This is because the base station broadcasts every TV channel in bursts with a bit rate much higher than the encoding rate of that TV channel, which enables mobile devices to receive a burst of data and turn off their radio components until the

next burst in order to save energy. This is called *time slicing*, and is dictated by leading broadcast standards such as DVB-H (Digital Video Broadcast-Handheld) [2–4] and MediaFLO (Forward Link Only) [5]. To enable time slicing, the base station *has to* prepare bursts for all TV channels to ensure that: (i) mobile devices can receive enough data for smooth play-out, and (ii) no bursts intersect with each other in time. While preparing bursts for non-scalable videos is already hard [6], preparing bursts for multi-layer videos is even more complex, because the dependency among various layers must be carefully considered.

We study the multi-layer video broadcasting problem in mobile TV broadcast networks, where each TV channel is encoded into a scalable video stream with multiple layers, and several TV channels are concurrently broadcast over a shared air medium to many mobile devices with *heterogeneous* resources. Our problem is to design a time slicing scheme for the base station to allow each mobile device to receive and decode the substream that is most suitable to its capability, while the consumed energy is proportional to the perceived video quality and the quality of service is maintained. We formulate and solve this problem, and we analyze the performance of the proposed schemes. We implement them in a real mobile TV testbed to demonstrate their practicality and efficiency. The experimental results indicate that the proposed schemes allow mobile devices to trade perceived quality for energy saving, as they can opt to receive a smaller substream to prolong battery lifetime. Furthermore, very low channel switching delay, only a few hundred milliseconds, can be achieved.

The rest of this paper is organized as follows. We briefly review the previous works in Sec. 2. In Sec. 3, we define the considered problem. We present the solution in Sec. 4. We evaluate the solution in Sec. 5 using a real mobile TV testbed. Sec. 6 concludes the paper.

2. RELATED WORK

Scalable video coding has been adopted in mobile broadcast networks to support mobile devices under diverse reception conditions [7]. For example, unequal error protection (UEP) methods were proposed to improve video quality for mobile devices with bad radio receptions [8, 9]. Ghandi and Ghanbari [8] proposed to transmit the base layer and the enhancement layers with different modulation and coding schemes, so that the base layer is sent over a more robust low bit rate channel while the enhancement layer is sent over a less robust high bit rate channel. This is called hierarchical modulation and channel coding, and is supported by many broadcast networks including DVB-T [10]. Hellge et al. [9] considered the layering dependency in multi-layer video streams, and proposed to use parity bits of higher layers to protect lower layers, which need to be more resilient to errors as they are more critical to successful decoding. None of these works considers time slicing schemes, thus they are orthogonal to our work.

The energy saving of mobile devices has been studied for mobile TV networks, where TV channels are coded in non-scalable fashion. The authors of [11] and [12] estimated the energy saving achieved by time slicing. They showed that time slicing enables mobile devices to turn off their radio components for a significant fraction of the time. They, however, did not propose time slicing schemes: they only compute the achieved energy saving for *pre-determined* bursts. Balaguer et al. [13] proposed an energy saving strategy by not receiving more FEC (forward error correction) bytes once the data can be successfully reconstructed. In this way, mobile devices can turn off their radio components earlier, and achieve additional energy saving compared to receiving all FEC bytes. Zhang et al. [14] considered mobile devices with an auxiliary short range wireless interface and constructed a cooperative network over this short range network to share the IP packets received from the broadcast network. Assuming transmitting data over the short range network is more energy efficient than receiving data from mobile TV networks, this cooperative strategy can save energy. The proposals in [13, 14] are orthogonal and complementary to our work, as they reside in the mobile devices themselves and try to achieve additional energy saving on top of that achieved by time slicing. In contrast, we propose time slicing scheme that is implemented in the base station.

Finally, our previous works proposed time slicing schemes for mobile TV networks that broadcast to homogeneous mobile devices [15, 16]. In a more recent work [17], we considered heterogeneous mobile devices, and proposed a time slicing scheme that enables each mobile device to render a substream that is most suitable to its capability. However, the analytical and experimental results there did not consider channel switching events. In this paper, we consider the channel switching events and propose a new time slicing scheme that not only enables each mobile device to achieve energy saving proportional to the received quality, but also guarantees the maximal channel switching delay.

3. PROBLEM STATEMENT

In this section, we first define the problem considered in this paper. We also describe the operation of current mobile TV networks. We then show that such networks cannot properly support heterogeneous mobile devices.

3.1. Time Slicing Schemes for Heterogeneous Mobile Devices

We consider a mobile TV network in which a base station concurrently broadcasts multiple TV channels over a shared air medium with bandwidth R kbps to mobile devices with *heterogeneous* resources. The network bandwidth is divided among TV channels, where each TV channel is assigned a bit rate of r kbps. In order to support heterogeneous mobile

devices, each TV channel is encoded into C layers using scalable video coders, where each layer has a bit rate of $r_s = r/C$ kbps. The base station broadcasts each TV channel as a series of bursts, where each burst consists of b kb data. The burst size b is a system parameter. As each TV channel is sent in bursts, a mobile device can receive a burst of data and turn off its radio component till the next burst of the same TV channel in order to save energy. This is called time slicing.

Time slicing is critical to the quality of service in mobile TV networks for two reasons. First, time slicing enables mobile devices to save energy by turning off their radio components while not receiving bursts. Higher energy saving results in longer battery lifetime, which extends the mobile TV watch time. Second, time slicing also has an impact on channel switching delay, which refers to the time a user waits before s/he starts viewing a selected channel when a change of channel is requested by that user [18, 19]. High and varying channel switching delay degrades view experience because many users quickly flip through numerous TV channels before they decide to watch the specific ones.

Since energy saving and channel switching are two important quality of service metrics, we formally define them in the following. The energy saved by a mobile device because of time slicing is denoted by γ , and it is calculated as the ratio of time the radio component is in off mode to the total time [11, 12]. When computing the energy saving, we need to consider the wake-up time of the radio components on mobile devices to receive the next burst [12]. This is because it takes radio components some time to power up and resynchronize to the radio signals before data can be demodulated. This period is called overhead duration and is denoted as T_o . With current technology, T_o can be as high as 250 msec [3]. The channel switching delay is denoted by d , which is defined as the time difference between a user switches to a new TV channel and the first burst of that TV channel arrives. We notice that d is a function of the channel switching time. We let d_m be the channel switching delay in the worst case.

Using the above notations, we now state the multi-layer video broadcasting problem considered in this paper.

Problem 1 (Multi-Layer Video Broadcasting in Mobile TV Networks.). *Consider a mobile TV broadcast network with air medium bandwidth R kbps shared among S TV channels, where each TV channel has a bit rate of r kbps. Every TV channel is encoded into C layers, and each layer has a bit rate of $r_s = r/C$ kbps. Mobile devices are classified into C classes so that devices in class c ($c = 1, 2, \dots, C$) receive and decode all layers \bar{c} , where $\bar{c} \leq c$. The coded streams are encapsulated into bursts of size b kb, and broadcast over the shared air medium. Design a time slicing scheme such that, for mobile devices in any class c , the energy saving γ_c of mobile devices is proportional to the perceived video quality, and the maximal channel switching delay from any channel to any other channel is at most d_m . The time slicing scheme specifies the start time and the size of each burst for each layer*

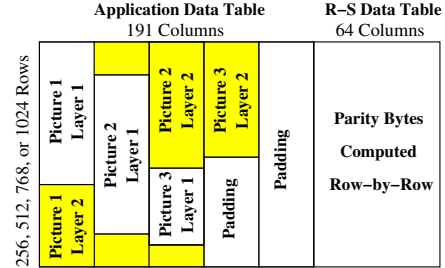


Fig. 1. The structure of MPE-FEC frame.

of individual TV channels. This time slicing scheme must be feasible: bursts are enough for smooth playout on all mobile devices, and no two bursts intersect with each other in time.

Solving this problem allows users to watch more TV programs (due to longer battery lifetime), while maintaining channel switching delay within an acceptable range. This will increase user satisfactory and eventually increase the number of subscribers as well as revenues of network operators.

3.2. Burst Preparation in Mobile TV Networks

Each mobile TV base station consists of three main components: a video server, an IP encapsulator, and an RF signal modulator. The video server encapsulates the video data, pre-encoded or live, into RTP packets, and sends these packets over an IP network to the IP encapsulator. The IP encapsulator receives these IP packets and puts them in MPE (multi-protocol encapsulation) frames. MPE frames are used by IP encapsulator for preparing transmission bursts of a specific TV channel, and each MPE frame is sent as *one* burst. The MPE frame can optionally be protected by Reed-Solomon (R-S) codes. This is achieved by extending the MPE frame into the MPE-FEC frame, which consists of FEC parity bytes. Since mobile devices are vulnerable to bad radio channel conditions, MPE-FEC frames are important as they provide better error resilience. We note that the name *IP encapsulator* is a bit misleading as it actually encapsulates IP packets into *bursts*, thus it should be better called *burst encapsulator*. We, however, use the term IP encapsulator in this paper, following previous works in the standardization documents [12] and in the literature [11, 13].

Fig. 1 reveals the structure of an MPE-FEC frame, which is divided into two parts: an application data table (ADT) that carries IP packets and an R-S data table (RDT) that carries the parity bytes. To compute the parity bytes, received IP packets are *sequentially* placed in the ADT column-by-column, from left to right. Zeros are padded in the remaining space of the ADT if there are not enough data to fill the ADT. Once the ADT is full (by data and/or zeros), the parity bytes are computed row-by-row, and stored in the RDT. Once the parity bytes are computed, the whole MPE-FEC frame is sent, column-by-column, as a burst. Notice that as multiple TV

channels are concurrently broadcast by the base station, the IP encapsulator applies a *time slicing scheme* to time multiplex bursts for multiple TV channels, and sends them over a shared air medium. We study the problem of designing time slicing schemes to optimize the quality of service, therefore our proposed schemes will be implemented in the IP encapsulator. We notice that existing IP encapsulators *do not* inspect the received IP packets by any means: they just sequentially encapsulate IP packets one after another.

3.3. Broadcasting Scalable Video Streams

Traditional, non-scalable, coded streams must be decoded in their entirety, and are not efficient for mobile devices with heterogeneous resources. This is because all mobile devices have to receive the complete video streams even though some of them do not have enough resources to render the complete streams. Multi-version video broadcasting partially copes with this issue by sending *multiple* coded streams of the same TV channel, which allows each mobile device to choose the version that is most suitable to it. Multi-version video broadcasting is not bandwidth efficient, thus can only support limited number of mobile devices. In contrast, scalable video coders (SVCs) encode each TV channel into a single video stream that can be sent and decoded at various bit rates. This is achieved by simple manipulations, which extract substreams from the original stream, where each substream can be decoded and displayed at a lower perceived quality than the original (complete) stream. Several scalability modes, including spatial, temporal, and SNR (signal-to-noise ratio) are supported by recent video coding standards such as H.264/SVC [20, Annex. G]. Readers interested in more details about the H.264/SVC coding standard are referred to [21].

In order to support mobile devices with heterogeneous resources, network operators can use scalable video coders to encode each TV channel into multiple layers. These layers are then concurrently broadcast over the shared air medium to mobile devices. To do so, network operators may only need to upgrade the video server to support new RTP payload types, and can keep other components of the network unchanged. In this way, existing time slicing schemes can be used to broadcast multi-layer video streams. Unfortunately, this “patched” base station is not efficient as we will show in the following illustrative example.

We consider a small time window of three pictures, where each picture is coded into two layers. We assume that each layer is encapsulated into a single IP packet, and these packets are sent by the video server and received by the IP encapsulator in the following order: (picture 1, layer 1), (picture 1, layer 2), (picture 2, layer 1), (picture 2, layer 2), (picture 3, layer 1), (picture 3, layer 2). Since IP encapsulators *sequentially* place IP packets in the receiving order within the ADT part of MPE-FEC frames, these packets are stored into

a frame as illustrated in Fig. 1. This MPE-FEC frame is then sent over the broadcast network as a burst. Consider a mobile device that can only render the base layer (layer 1), this mobile device, unfortunately, still has to receive and process the *complete* burst for two reasons. First, IP packets belonging to the base layer (unshaded in the figure) are scattered all over the MPE-FEC frame, and a deep inspection (at RTP or video-coding layer) is required to locate them. Second, the parity bytes are computed over IP packets from various layers, and are useless if some IP packets are not received. Receiving complete bursts degrades the energy saving achieved by this mobile device, because it has to open the radio component for a longer time period to receive some data that will be dropped eventually.

The above example shows that simply broadcasting multi-layer video streams using current mobile TV networks lead to *no additional* energy saving for mobile devices that cannot render the complete video streams. More importantly, we need to design IP encapsulators that take the layering structure of video streams into considerations while preparing transmission bursts, such that mobile devices do not need to open their radio components for data that will be discarded later. We refer to such IP encapsulators as *layer-aware IP encapsulators*, and the time slicing schemes used in them as *layer-aware time slicing schemes*. We propose layer-aware time slicing schemes and analyze their performance in the next section.

4. PROPOSED SOLUTIONS

In this section, we first review a layer-aware time slicing scheme that was initially proposed in [17]. We analyze this scheme using a quality of service metric that was not considered there. The analysis reveals the potential weakness of that scheme, and we propose a new time slicing scheme that fully addresses the considered problem.

4.1. Layer-Aware Time Slicing Scheme: LATS

We propose a time slicing scheme that considers the layering structure of video streams while preparing bursts, which is called Layer-Aware Time Slicing (LATS) scheme. Fig. 2 illustrates this scheme, which prepares a different MPE-FEC frame (and thus a burst) for each coded layer of individual TV channels. For example, all IP packets in the left-most burst belong to layer 1 of TV channel 1, while IP packets in the fourth burst belong to layer 2 of TV channel 2. The main merit of LATS scheme is that, even before receiving any data of a burst, mobile devices know which layer the IP packets in that burst belong to. This frees mobile devices from opening the radio components for extracting a suitable substream, and increases the energy saving as no received data are discarded. LATS can be easily implemented: no additional signaling from the base station to mobile devices is required. To

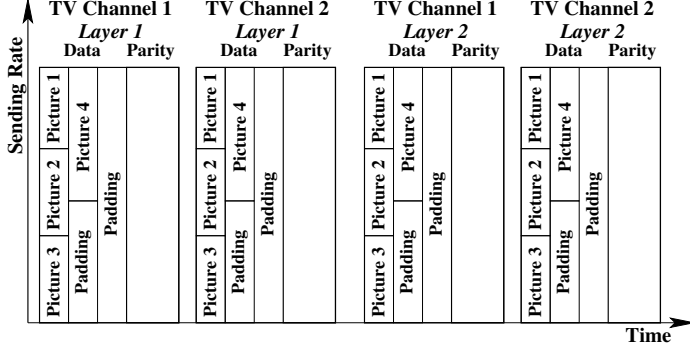


Fig. 2. Proposed Layer-Aware Time Slicing Scheme: LATS.

locate the bursts (layers) that need to be received, mobile devices only need to know the total number of layers (C), which is already sent to them for decoding the scalable stream.

We present details on the LATS scheme as follows. We first compute the number of TV channels that can be concurrently broadcast as $S = \lfloor R/r \rfloor$. LATS is defined for a recurring window of CS bursts, where each burst contains b kb data. Since the radio channel bandwidth is R , the window size is given as $(b/R)CS$ sec. As illustrated in Fig. 2, LATS allocates bursts for layer c of all TV channels earlier than those for layer $c+1$ of all TV channels. Therefore, the start time of the burst for layer c ($c = 1, 2, \dots, C$) of TV channel 1 is given as $(b/r)(c-1)S$ because there are S bursts in each layer. Furthermore, the start time of burst for layer c ($c = 1, 2, \dots, C$) of TV channel s ($s = 1, 2, \dots, S$) is given as:

$$(b/R)[(c-1)S + (s-1)]. \quad (1)$$

The next Lemma shows that LATS scheme is correct and allows mobile devices in different classes to render at various perceived quality and achieve proportional energy saving. The proof was given in [17].

Lemma 1. *LATS scheme (Eq. (1)) specifies a feasible time slicing scheme for a recurring window of $(b/R)CS$ sec, where (i) no two bursts intersect with each other, and (ii) bursts are long enough to send data for smooth playout on all mobile devices. Furthermore, the energy saving achieved by mobile devices in class c is given by:*

$$\gamma_c = 1 - \frac{c}{CS} - \frac{RT_o c}{bCS} \quad \text{where } c = 1, 2, \dots, C. \quad (2)$$

We extend the analysis in [17], and consider the channel switching delay of LATS scheme in the next Lemma.

Lemma 2. *The Layer-Aware Time Slicing (LATS) scheme (Eq. (1)) specifies a feasible time slicing scheme that achieves a worst case channel switching delay of:*

$$d_m = (b/r)C. \quad (3)$$

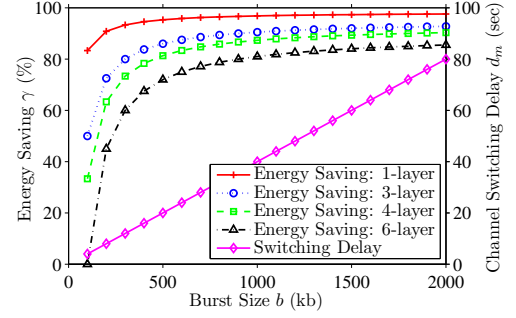


Fig. 3. Diverse energy saving supported by LATS scheme, and resulting channel switching delay.

Proof. We first consider the worst case channel switching delay for mobile devices in class 1. The worst case happens when a user switches to a channel s right after the burst for layer 1 of channel s is broadcast: the user has to wait for the complete recurring window, i.e., $d_m = (b/R)CS$. Following the definition of S , we write:

$$d_m = (b/R)CS < (b/R)CR/r = (b/r)C,$$

which yields Eq. (3). This result can be extended to any class c : mobile devices, despite how many layers they receive, can start playing out with the base layer (layer 1) whenever it arrives. Mobile devices gradually add enhancement layers whenever they are available for incremental quality improvements. \square

We next show the implication of the burst size on energy saving and channel switching delay using typical system parameters. We consider a mobile TV network with radio bandwidth 9 Mbps, where each TV channel is encoded into 6 layers: each layer has a bit rate of 150 kbps. We assume the overhead duration $T_o = 100$ msec. Fig. 3 presents sample energy saving of four classes of mobile devices, and resulting channel switching delay. This figure shows that LATS scheme allows mobile devices to achieve a wide range of energy saving. However, this flexibility comes at an expense of high channel switching delay. For example, as illustrated in this figure, the channel switching delay is about 7.5 sec when $b = 200$ kbps, and is even larger when b is larger. Such a long channel switching delay may drive subscribers away from the mobile TV network. To address this problem, we develop a new time slicing scheme in Sec. 4.2, which allows us to achieve diverse energy saving without incurring excessive channel switching delay.

4.2. Layer-Aware Time Slicing Scheme with Switching Delay Bound: LATSb

To control the channel switching delay, we propose to insert short and frequent *bootstrap* bursts between any two adjacent

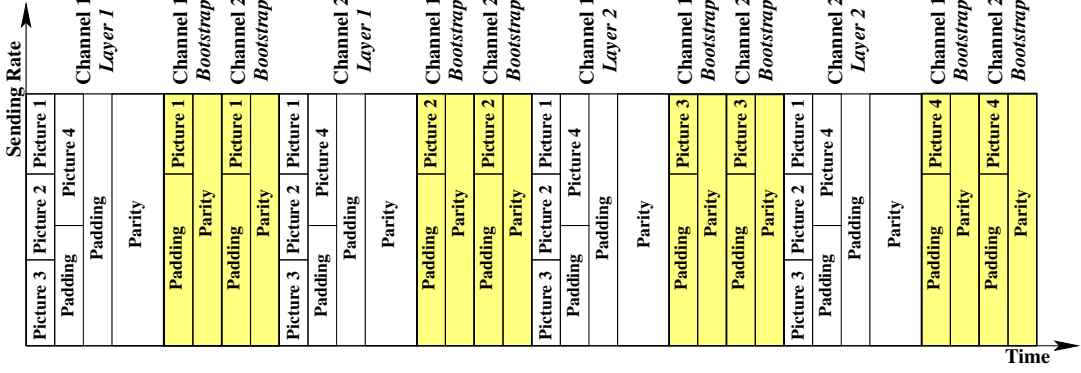


Fig. 4. The proposed Layer-Aware Time Slicing Scheme with switching delay bound. The shaded bursts are bootstrap bursts optimized for fast channel switching.

bursts defined in LATS scheme, and send the base layers of all TV channels using these bootstrap bursts. We refer to the bursts defined in LATS scheme as *normal* bursts in the rest of this paper. Fig. 4 illustrates how we insert the bootstrap bursts: a small piece of base layer from all TV channel is sent between any two normal bursts. Since the bootstrap bursts are sent very often, the user who switches to a new channel can receive the bootstrap bursts and start playing the base layer very quickly. Upon reaching the next normal burst for layer 1 of the selected TV channel, mobile devices switch over to normal bursts. Normal bursts provide higher energy saving, because they are longer thus the overhead duration T_o is relatively insignificant to them. Since this new time slicing scheme provides channel switching delay guarantees, as we will show in a moment, we call it layer-aware time slicing scheme with delay bound (LATSb).

We develop LATSb scheme by specifying the start time and the size of each burst. Since we send the base layer twice, the number of TV channels that can be concurrently broadcast is given as: $S = \lfloor \frac{R}{r+r_s} \rfloor = \lfloor \frac{R}{r+r/C} \rfloor$. The size of normal bursts is b kb, which is a system parameter, while the size of bootstrap bursts is smaller as they are sent more often. Observe that there are CS bootstrap bursts in each recurring window for a TV channel, and these bootstrap bursts should aggregately carry the base layer that is also carried by one normal burst. Therefore, the bootstrap burst size is given as $b/(CS)$. We let $\bar{b} = S \times b/(CS) = b/C$ be the aggregate burst size of all bootstrap bursts (belonging to all TV channels) between any two normal bursts.

Next, we give the start times for all bursts. The start time of the normal burst for layer c ($c = 1, 2, \dots, C$) of TV channel s ($s = 1, 2, \dots, S$) is written as:

$$\frac{\bar{b} + b}{R} [(c-1)S + (s-1)] = \frac{b(C+1)}{RC} [(c-1)S + (s-1)]. \quad (4)$$

The start times of bootstrap bursts for TV channel s ($s =$

$1, 2, \dots, S$) are given as:

$$\frac{b(C+1)}{RC} [(l-1)S + (k-1)] + (s-1) \frac{b}{CSR}, \quad (5)$$

for all l ($l = 1, 2, \dots, C$) and k ($k = 1, 2, \dots, S$).

We show the correctness of LATSb scheme and analyze its performance in the next Lemma.

Lemma 3. *LATSb scheme (Eqs. (4) and (5)) specifies a feasible time slicing scheme for a recurring window of $\frac{b(C+1)}{R}S$ sec, where (i) no two bursts intersect with each other, and (ii) bursts are long enough to send data for smooth playout on all mobile devices. Furthermore, the energy saving achieved by mobile devices in class c ($c = 1, 2, \dots, C$) is given by:*

$$\gamma_c = 1 - \frac{c}{(C+1)S} - \frac{RT_o c}{b(C+1)S}. \quad (6)$$

The energy saving achieved by mobile devices that receive bootstrap bursts is:

$$\gamma_b = 1 - \frac{R}{C+1} - \frac{RCT_o}{(C+1)b}. \quad (7)$$

Finally, the maximal channel switching delay is:

$$d_m = \frac{b + b/C}{R}. \quad (8)$$

Proof. First, the construction of LATSb guarantees that no two bursts intersect with each other because the air medium (of bit rate R kbps) is fast enough to send all bursts within the reserved time slots. Second, consider two adjacent bootstrap bursts for the same TV channel, which are apart from each other by time $\frac{\bar{b}+b}{R}$. The bootstrap burst transmits up to $b/(CS)$ kb data, and the needed data amount for smooth playout is $\frac{\bar{b}+b}{R}r_s$. Following the definition of S , we have :

$$\frac{b}{CS} \geq \frac{b(r+r/C)}{CR} = \frac{b}{CR}(C+1)r_s = \frac{\bar{b}+b}{R}r_s, \quad (9)$$

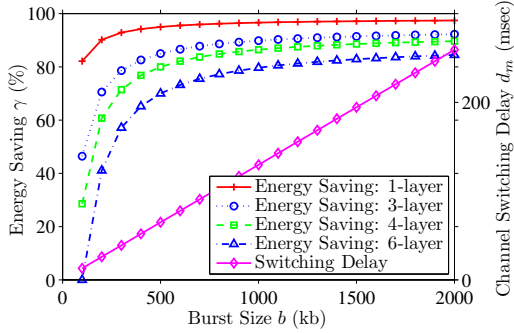


Fig. 5. Diverse energy saving supported by LATSB scheme, and resulting channel switching delay.

which shows that bootstrap bursts are long enough to support smooth payout. We then consider two adjacent normal bursts for the same layer of a specific TV channel. The burst transmits up to b kb data, and the needed data amount for smooth payout is $\frac{b(C+1)}{RC}CSr_s$. Following the definition of S , we have:

$$\frac{b(C+1)}{RC}CSr_s \leq \frac{b(C+1)}{R} \frac{R}{r+r/C}r_s = b, \quad (10)$$

which shows that normal bursts are also long enough to support smooth payout. This shows the correctness of LATSB.

Next, based on the definition of energy saving, the energy saved for mobile devices that receive c layers is written as:

$$\gamma_c = 1 - \frac{(b/R + T_o)c}{\frac{b(C+1)}{RC}CS},$$

while the energy saved for mobile devices that receive bootstrap bursts is written as:

$$\gamma_b = 1 - \frac{b/C + T_o}{(C+1)b/(RC)}.$$

Manipulating these equations leads to Eqs. (6) and (7). Finally, the maximal channel switching delay is the time difference between two bootstrap bursts for the same TV channel, i.e., $d_m = \frac{\bar{b}+b}{R}$. This yields Eq. (8). \square

We next study the implication of the burst size on energy saving and channel switching delay using the same system parameters used in Sec. 4.1 and plot the results in Fig. 5. This figure shows that LATSB scheme enables flexible energy saving for mobile devices with heterogeneous resources. More importantly, this figure also reveals that, compared to LATS, LATSB scheme dramatically reduces the channel switching delay: less than 260 msec delay can be achieved. Notice that, in this analysis, we assume mobile devices receive bootstrap bursts for short, *transient*, time periods, and only consider mobile devices that receive normal bursts. In Sec. 5, we will remove this assumption.

Last, the following Corollary allows network operators to precisely bound the channel switching delay. This is a direct result of Eq. (8).

Corollary 1. For a given maximal channel switching delay d_m , applying LATSB scheme (Eqs. (4) and (5)) with any normal burst size b , $b \leq b_m$ guarantees that mobile devices never suffer from channel switching delay longer than d_m , where $b_m = \frac{C}{1+C}d_mR$.

5. EVALUATION

We evaluate the proposed time slicing schemes using a real mobile TV testbed. We first describe the testbed and experimental setup. We then present the results.

5.1. Mobile TV Testbed

We have implemented a mobile TV testbed in our Lab, which provides a realistic platform for analyzing the performance of the proposed time slicing schemes. The mobile TV testbed has two parts: base station and receivers. We use a commodity Linux box as the base station, and runs video server, IP encapsulator, and modulator software on it. Most of the software components are developed by us and few drivers are leveraged from open-source projects. We installed a PCI modulator in the base station, which implements the physical layer of the DVB-H protocol and transmits DVB-H standard compliant signals via a low-power amplifier and an indoor antenna. We use Nokia N96 cellular phones as receivers to assess the visual quality of videos. For detailed analysis of the mobile TV network, we add a DVB-H analyzer to the testbed. This analyzer is attached to a PC via a USB port and comes with a visualization software for analysis. The analyzer records traffic streams as well as provides a very detailed information on radio signal, MPEs, jitter, time slicing, and so on. More details on the testbed are given in [22].

5.2. Setup

We have implemented both LATS and LATSB schemes in the testbed. For comparison, we have also implemented the current, sequential scheme, which is denoted as CUR in figures. Using the H.264/SVC reference software [23], we have encoded the City video sequence, into four SNR scalability layers, where each layer has a bit rate of 192 kbps and is in CIF (352x288) resolution, at 30 fps frame rate. We configured the modulator card to use 8 MHz bandwidth, QPSK (quadrature phase-shift keying) modulation, 3/4 code ratio, and 1/8 guard interval. This leads to the channel bandwidth of 8.289 Mbps [24]. We broadcast 4 TV channels for 5 min using LATS and LATSB schemes, and repeated the same test using CUR scheme.

To gather statistically meaningful results, we have instrumented the testbed to save log files for offline analysis. The

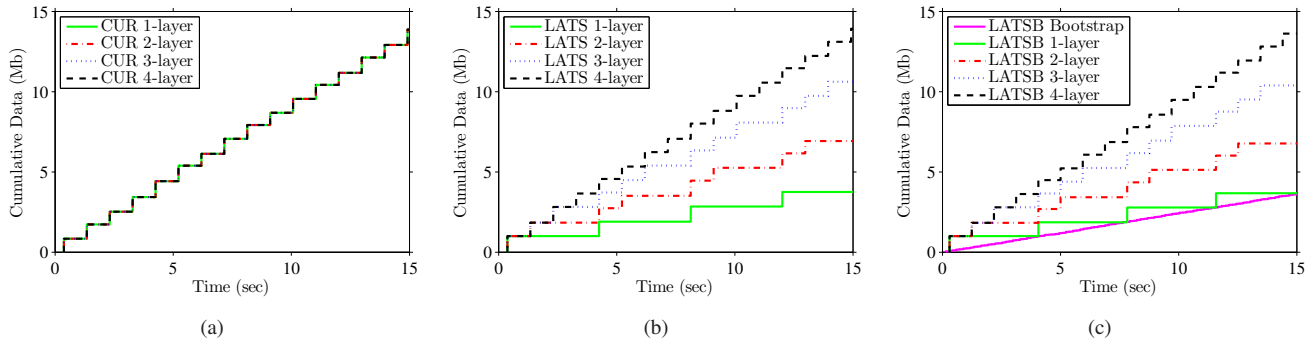


Fig. 6. Cumulative received data: (a) CUR, (b) LATS, and (c) LATSBS scheme.

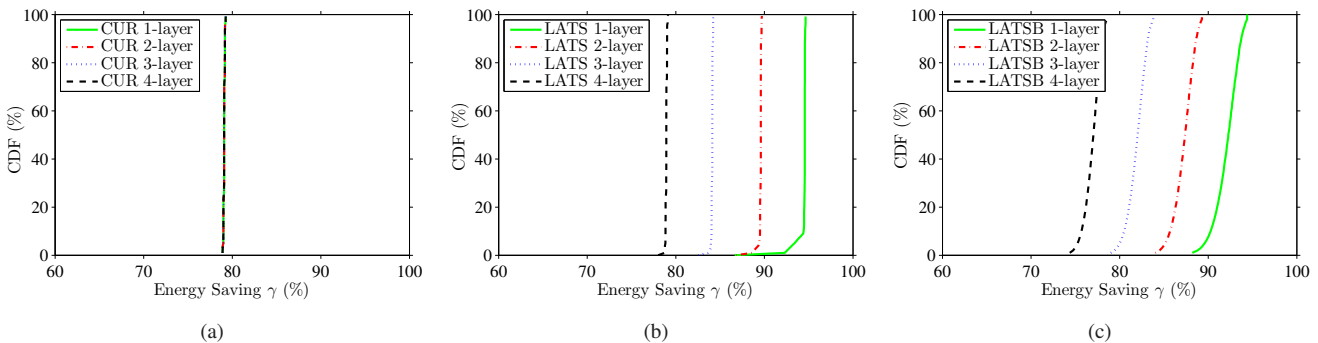


Fig. 7. Energy saving achieved by mobile devices in different classes: (a) CUR, (b) LATS, and (c) LATSBS scheme.

log files contain start and end times of each burst, its size, and the layer it belongs to. Using these logs, we wrote a software utility to emulate the channel switching behavior of a large number (1 million) of mobile devices. We generate random channel switching events using Bernoulli trials. For every mobile device, we toss a biased coin every second and issue a channel switching command if the trial is success. The new selected channel is randomly chosen from all broadcast channels other than the currently watched one. We configured the probability of success to vary the average watch time of each channel from 1 sec to 60 sec. We also varied the burst size from 500 to 1500 kb. If not otherwise specified, we present sample results for 60-sec average watch time, and 1000 kb burst size.

We ran the simulation against every log file collected from the real testbed and we computed the channel switching delay d and energy saving γ . We measured the channel switching delay by searching for the next burst of the selected TV channel and computing the time difference between it and the channel switching event. We set the overhead duration $T_o = 100$ msec, and measured the energy saving by calculating the fraction of time that the radio component is on between every two channel switching events. We emphasize that while computing the energy saving for LATSBS scheme, we divide the watching period into two parts: when a device

receives bootstrap bursts, and when it receives normal bursts. We calculated the energy saving in both periods and reported the weighted average of them. Since we consider both time periods in the experiments, we no longer assume that mobile devices receive bootstrap bursts only for a transient time period as we did in Sec. 4.2.

5.3. Results

We present sample results for TV channel 1, while results for other TV channels are similar.

Cumulative Data Dynamics: We first plot the cumulative received data in Fig. 6 for a 15-sec time period. In this figure, every staircase step represents a received burst. This figure reveals that the CUR scheme does not allow mobile devices to receive substreams that are smaller than the complete video streams, while both LATS and LATSBS enable mobile devices to skip those bursts that are of no use to them. In addition Fig. 6(c) illustrates that in LATSBS, bootstrap bursts come more often, and the accumulated bit rate of bootstrap bursts is equivalent to that of the base layer¹. This shows that mobile devices who receive bootstrap bursts can quickly render the video at the base layer quality.

¹The line of bootstrap bursts consists of staircases, but in smaller scale.

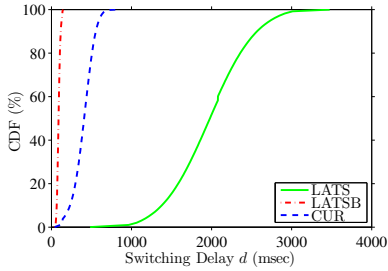


Fig. 8. Delay of different schemes.

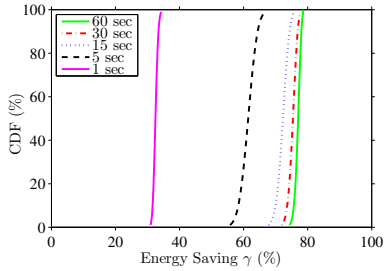


Fig. 9. Implication of watch time.

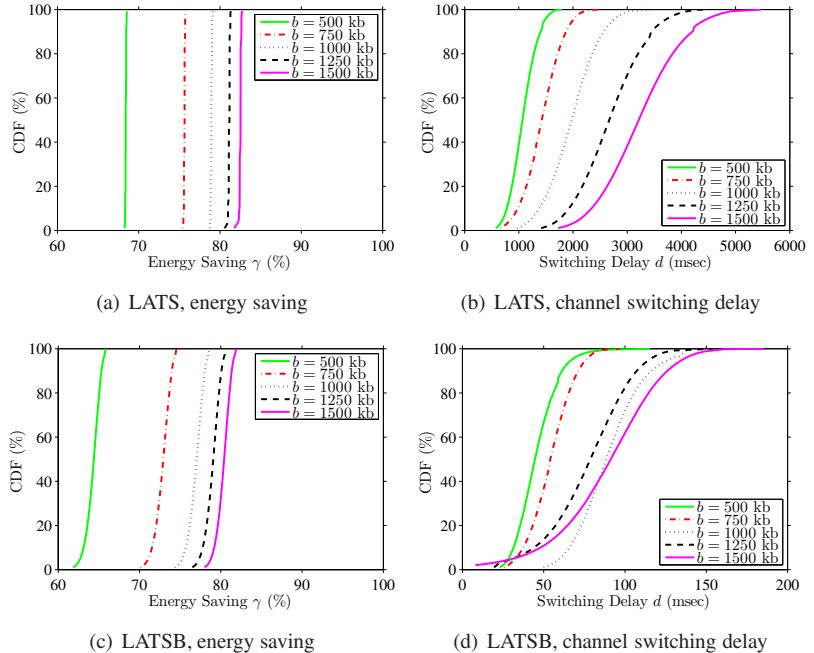


Fig. 10. Implication of burst size on energy saving and channel switching delay.

Diverse energy saving: We plot the average energy saving achieved by mobile devices in Fig. 7. This figure shows that CUR scheme leads to no energy saving differentiation, while LATS and LATSb schemes enable proportional energy saving for mobile devices in different classes. This confirms that broadcasting multi-layer streams using CUR scheme cannot support heterogeneous mobile devices.

Channel Switching Delay: We plot the channel switching delay in Fig. 8. This figure shows that LATS scheme may lead to high channel switching delay: up to 4-sec delay in this experiment. In contrast, LATSb scheme results in *negligible* channel switching delay: about 200 msec is achieved.

Implication of burst size: We next vary burst size b from 500 to 1500 kb. This covers the whole practical range of b values, because 1,565 kb is the maximal burst size specified by DVB-H standard documents [12]. Due to space limitations, we present sample results for the mobile devices that receive the complete streams. We plot the energy saving and channel switching delay in Fig. 10. Fig. 10(a) shows that increasing burst size allows LATS scheme to achieve higher energy saving. However, Fig. 10(b) reveals that larger burst size also increases the channel switching delay: letting $b = 1500$ kb leads to as high as 6-sec delay. These two figures show that LATS scheme uses b to control the tradeoff between energy saving and channel switching delay, which is inefficient. In contrast, Figs. 10(c) and 10(d) show that increasing b in LATSb scheme also leads to higher energy saving, however, it does not result in excessive channel switching delay: the delay is shorter than 200 msec in all cases.

Implication of per channel watch time: In LATSb scheme, mobile devices that receive bootstrap bursts incur lower energy saving. Hence, the frequency of channel switching events can affect the average energy saving. To quantify this impact, we vary the time that a user would watch a TV channel from 1 sec to 60 sec. We plot energy saving of mobile devices that receive the complete streams in Fig. 9. This figure shows that frequent channel switching events only degrade the energy saving in extreme cases, where users constantly change TV channels.

6. CONCLUSIONS

In this paper, we studied the problem of broadcasting multi-layer video streams over a shared air medium to mobile devices with *heterogeneous* resources. We proposed two time slicing schemes, LATS and LATSb, which enable every mobile device to render a substream that is most suitable to its capability, and achieve proportional energy saving. Compared to LATS, LATSb further achieves low, and controllable, channel switching delay, which is critical to user experience. We analytically analyzed the performance of LATS and LATSb, and we implemented them in a real mobile TV testbed to show their practicality and efficiency. Our extensive experiments showed that both LATS and LATSb enable energy saving differentiation: between 75% and 95% were observed. Moreover, LATSb achieves low channel switching delays: 200 msec is possible with typical system parameters.

7. REFERENCES

- [1] "Mobile TV predicted to be a hit," <http://news.bbc.co.uk/2/hi/technology/6639249.stm>.
- [2] G. May, "The IP Datacast system - overview and mobility aspects," in *Proc. of IEEE International Symposium on Consumer Electronics (ISCE'04)*, Reading, UK, September 2004, pp. 509–514.
- [3] M. Kornfeld and G. May, "DVB-H and IP Datacast – broadcast to handheld devices," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 161–170, March 2007.
- [4] "Digital Video Broadcasting (DVB); transmission system for handheld terminals (DVB-H). ETSI Standard EN 302 304 Ver. 1.1.1," November 2004.
- [5] "FLO technology overview," http://www.mediaflo.com/news/pdf/tech_overview.pdf.
- [6] M. Hefeeda and C. Hsu, "Energy optimization in mobile TV broadcast networks," in *Proc. of IEEE Innovations in Information Technology (Innovations'08)*, Al Ain, United Arab Emirates, December 2008.
- [7] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile video transmission using scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1204–1217, September 2007.
- [8] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier, "Layered H.264 video transmission with hierarchical QAM," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 451–466, April 2006.
- [9] C. Hellge, T. Schierl, and T. Wiegand, "Mobile TV using scalable video coding and layer-aware forward error correction," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME'08)*, Hannover, Germany, April 2008, pp. 1177–1180.
- [10] U. Ladebusch and C. Liss, "Terrestrial DVB (DVB-T): A broadcast technology for stationary portable and mobile use," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 183–193, January 2006.
- [11] X. Yang, Y. Song, T. Owens, J. Cosmas, and T. Itagaki, "Performance analysis of time slicing in DVB-H," in *Proc. of Joint IST Workshop on Mobile Future and Symposium on Trends in Communications (SympoTIC'04)*, Bratislava, Slovakia, October 2004, pp. 183–186.
- [12] "Digital Video Broadcasting (DVB); DVB-H implementation guidelines. ETSI Standard EN 102 377 Ver. 1.3.1," May 2007.
- [13] E. Balaguer, F. Fitzek, O. Olsen, and M. Gade, "Performance evaluation of power saving strategies for DVB-H services using adaptive MPE-FEC decoding," in *Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'05)*, Berlin, Germany, September 2005, pp. 2221–2226.
- [14] Q. Zhang, F. Fitzek, and M. Katz, "Cooperative power saving strategies for IP-services supported over DVB-H networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'07)*, Hong Kong, China, March 2007, pp. 4107–4111.
- [15] C. Hsu and M. Hefeeda, "Bounding switching delay in mobile TV broadcast networks," in *Proc. of ACM/SPIE Multimedia Computing and Networking (MMCN'09)*, San Jose, CA, January 2009.
- [16] C. Hsu and M. Hefeeda, "Time slicing in mobile TV broadcast networks with arbitrary channel bit rates," in *Proc. of IEEE INFOCOM'09*, Rio de Janeiro, Brazil, April 2009.
- [17] C. Hsu and M. Hefeeda, "Video broadcasting to heterogeneous mobile devices," in *Proc. of IFIP Networking 2009 (Networking'09)*, Aachen, Germany, May 2009.
- [18] M. Rezaei, M. Hannuksela, and M. Gabbouj, "Tune-in time reduction in video streaming over DVB-H," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 320–328, March 2007.
- [19] M. Rezaei, I. Bouazizi, and M. Gabbouj, "Joint video coding and statistical multiplexing for broadcasting over DVB-H channels," *IEEE Transactions on Multimedia*, vol. 10, no. 7, pp. 1455–1464, December 2008.
- [20] Joint Video Team, "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, November 2007.
- [21] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.
- [22] M. Hefeeda, C. Hsu, and Y. Liu, "Testbed and experiments for mobile TV (DVB-H) networks," in *Proc. of ACM Multimedia '08, Technical Demonstrations*, Vancouver, Canada, October 2008.
- [23] Joint Video Team, "Joint scalable video model reference software," JSVM 14.0, May 2008.
- [24] "Digital Video Broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television. ETSI Standard EN 300 744 Ver. 1.5.1," June 2004.