Flexible Broadcasting of Scalable Video Streams to Heterogeneous Mobile Devices

Cheng-Hsin Hsu, *Member*, *IEEE*, and Mohamed Hefeeda, *Senior Member*, *IEEE*

Abstract—We study the scalable 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 goal is to encapsulate and broadcast video streams encoded in scalable manner to enable heterogeneous mobile devices to render the most appropriate video substreams while achieving high energy saving and low channel switching delay. The appropriate streams depend on the device capability and the target energy consumption level. We propose two new broadcast schemes, which are flexible in the sense that they allow diverse bit rates among layers of the same stream. Such flexibility enables videos to be optimally encoded in terms of coding efficiency, and allows the coded video streams to be better matched with the capability of mobile devices. We analyze the performance of the proposed broadcast schemes. In addition, we have implemented the proposed schemes in a real mobile TV testbed to show their practicality and efficiency. Our extensive experiments confirm that the proposed schemes enable energy saving differentiation: between 75 and 95 percent were observed. Moreover, one of the schemes achieves low channel switching delays: 200 msec is possible with typical system parameters.

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

1 Introduction

Technology advances have tremendously increased the communication and computational powers of mobile devices. Modern mobile devices, despite their small sizes, can run many multimedia applications, including games and music players that are already popular nowadays, and mobile TV that is still emerging. Mobile TV services allow users to watch TV programs any time at any place. 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 multibillion dollar revenues in North America by 2011 [1].

Mobile TV programs can be delivered using 3G cellular or dedicated broadcast networks. Streaming TV programs over existing 3G cellular networks incurs bandwidth requirement that grows linearly with the number of mobile devices, because current 3G cellular networks are designed for unicast. Therefore, delivering TV programs over a 3G cellular network may easily overload the network, and does not scale well. Analysis, by a market research group, predicted that 3G cellular networks would be overloaded with only 40 percent of cellular phone users watching 8 minutes of video per day [2]. In fact, a recent news report reveals that smart phone users are already very close to saturate the bandwidth of 3G cellular networks as of 2009 [3]. In contrast, streaming videos from a base station using one-to-many broadcast service achieves high spectrum efficiency, and can support a large number of mobile devices. Therefore, we only consider TV programs delivered over dedicated broadcast networks.

Mobile devices have heterogeneous resources such as screen resolution, decoder capability, and battery capacity. For example, while laptop computers can display 720p ($1{,}280 \times 720$) videos, most smart phones only have QVGA (320×240) displays. It is challenging to *concurrently* support these mobile devices: broadcasting TV programs in QVGA resolution results in unacceptable video quality on laptop 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 appropriate for a mobile device. This is called simulcast, which is inefficient in terms of bandwidth as it effectively reduces the number of TV channels that can be concurrently broadcast. This inefficiency becomes even more severe if we categorize mobile devices into groups not only by device models but also by working conditions of the same model. 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.

Scalable 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 each TV channel into a single stream with multiple layers, and broadcast each layer only once. Such a coded stream is scalable because several substreams, with one or a few layers, can be extracted from the complete stream and are still decodable. Each mobile device can then choose and render the substream that is most appropriate to

- *C.-H. Hsu is with the Deutsche Telekom R&D Laboratories, 5050 El Camino Real 221, Los Altos, CA 94022. E-mail: cheng-hsin.hsu@telekom.com.*
- *M. Hefeeda is with the School of Computing Science, Simon Fraser University, 250-13450 102nd Ave, Surrey, BC V3T 0A3, Canada. E-mail: mhefeeda@cs.sfu.ca.*

Manuscript received 27 June 2009; revised 5 Jan. 2010; accepted 31 Mar. 2010; published online 7 Sept. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-06-0257. Digital Object Identifier no. 10.1109/TMC.2010.173.

its capability and condition. Scalable video broadcasting, however, is quite challenging for the base station broadcasting 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*, which is dictated in leading broadcast standards such as Digital Video Broadcast-Handheld (DVB-H) [4], [5], [6] and Forward Link Only (MediaFLO) [7]. Time slicing is important because energy saving is critical to battery-powered mobile devices, and recent commercial mobile TV chips consume more than 400 mW [8] in continuous mode, which is nontrivial compared to the total power consumption of many mobile devices, e.g., our measurement study [9] indicates that Nokia N96 phones only consume 1 W in total while receiving mobile TV signals. To enable time slicing, the base station *has to* prepare bursts for all TV channels to ensure that: 1) mobile devices can receive enough data for smooth playout, and 2) no bursts intersect with each other in time. Since preparing bursts for nonscalable videos is already hard [10], preparing bursts for scalable videos is even more complex, as the dependency among layers must be carefully considered.

We study scalable video broadcasting 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 encapsulate and broadcast video streams encoded in scalable manner, so that heterogeneous mobile devices can render the most appropriate video substreams to achieve high energy saving and low channel switching delay. The appropriate substreams depend on the device capability and the target energy consumption level.

Scalable video broadcasting has been proposed in the literature [11], [12], [13], [14]. Schierl et al. [11] propose to broadcast SVC streams in mobile TV networks to provide video-on-demand service. The authors of [12], [13], [14] propose to apply unequal error protection (UEP) on scalable video streams to cope with bad radio receptions. These four works do not consider the issue of interlayer synchronization and may lead to long *frame refresh delay*, which refers to the time period between receiving the first bit of video data and reaching the next random access point, typically an intracoded frame, of that video. To address this issue, we propose to insert instantaneous decoder refresh (IDR) frames at the beginning of individual bursts, so that mobile devices can receive decodable data of all scalable layers in timely fashion. We mention that frequently inserting intracoded H.264/AVC [15] frames in bursts has been studied in the literature [16], [17], [18], [19], [20]. We propose to apply similar techniques to scalable, multilayer video streams. Note that the IDR frames are typically larger than intercoded frames, and thus, inserting more IDR frames may result in data amount exceeding the capacity of individual bursts. We propose three approaches to accommodate the larger sizes of IDR frames, we can 1) reserve a small portion of bursts for IDR frames, 2) employ reduced quality (and thus smaller) IDR frames, and/or 3) drop some quality-scalability packets without affecting the decodability. By frequently inserting IDR frames in the bursts without exceeding their capacity, we can minimize the frame refresh delay and address the interlayer synchronization issue.

The contribution of this paper are as follows:

- We analyze current mobile TV systems and we show that they are not efficient for scalable coded streams. This is done by first presenting several broadcast schemes to enable scalable video broadcasting in current systems, and then pointing out their drawbacks. Moreover, we analytically show that these broadcast schemes lead to lower energy saving compared to our proposed schemes.
- We propose two new broadcast schemes for scalable video streams with arbitrary layer bit rates, and we analyze their performance. The two proposed schemes are flexible in terms of bit rates of individual layers, which enable network operators to optimally encode videos in terms of coding efficiency, and allow the coded video streams to be better matched with the capability of mobile devices. We prove that both schemes achieve high energy saving, and one of them also results in low channel switching delays.
- We implement the broadcast schemes in a real mobile TV testbed in our Lab to show their practicality and efficiency. We conduct extensive experiments to evaluate these schemes. The experimental results confirm our analysis, and 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 short channel switching delay, as low as a few hundred milliseconds, can be achieved using one of the proposed schemes.

The rest of this paper is organized as follows: Section 2 surveys the literature. We define the considered problem in Section 3. In Section 4, we study the limitations of current mobile TV networks on broadcasting scalable video streams. We present and analyze a broadcast scheme for high energy saving in Section 5, and we present another broadcast scheme that also reduces the channel switching delay in Section 6. We implement and evaluate both schemes in Section 7. We conclude the paper in Section 8.

2 Related Work

The energy saving of mobile devices has been studied for mobile TV networks, where TV channels are coded in nonscalable fashion. The authors of [21] and [22] estimate the energy saving achieved by time slicing. They show that time slicing enables mobile devices to turn off their radio components for a significant fraction of time. They, however, did not propose broadcast schemes: they only compute the achieved energy saving for *predetermined* bursts. Balaguer et al. [23] propose an energy saving strategy by not receiving more forward error correction (FEC) 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. [24] consider mobile devices with an auxiliary short range wireless interface and construct 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 [23], [24] 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 broadcast schemes that are implemented in the base station.

Our previous works study the burst transmission problems and propose time slicing schemes for mobile TV networks that broadcast nonscalable video streams to a single class of mobile devices [9], [10], [25]. That is, these works assume that all mobile devices have the same capability. We consider heterogeneous mobile devices that can only (or opt to) receive parts of video streams in [26], [27], where we assume the scalable video streams are coded into several layers with the same bit rate. However, such assumption may limit the applicability of our preliminary solutions in [26], [27]. For example, uniform bit rates may prevent the video coders from optimally allocating bit rate among layers for higher coding efficiency, and may result in scalable video streams that do not properly match the capability of mobile devices. To address these shortcomings, we propose two new broadcast schemes in this paper, which are general in the sense that layers can take any arbitrary bit rates.

3 Problem Statement

In this section, we describe the considered problem and define the notations that are used in the rest of this paper. 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, and each TV channel is assigned a bit rate of r Kbps. To support heterogeneous mobile devices, every TV channel is encoded into C layers using scalable video coders, where each layer c $(1 \leq c \leq C)$ has a bit rate of r_c Kbps. The base station broadcasts each TV channel as a series of bursts, and a mobile device receives a burst of data and turns off its radio component until the next burst of the same TV channel to save energy, which 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 has an impact on channel switching delay, which refers to the time a user waits before she/he starts viewing a selected channel when a change of channel is requested by that user [17], [18]. Shorter channel switching delays result in better view experience as many users flip through numerous TV channels before they decide to watch specific ones.

We next formally define the energy saving and the channel switching delay. 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 [21], [22]. 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. This is because it takes radio components sometime to power up and lock on to the radio signals before data can be demodulated [22]. This period is called overhead duration and is denoted as T_o. T_o is not negligible, and can be as high as 250 msec [5]. The channel switching delay d consists of several components, in which the frame refresh delay and time slicing delay are the two dominating contributors [17], [18]. In Section 1, we proposed inserting IDR frames to minimize frame refresh delay. Therefore, we assume that frame refresh delay is negligible, and only consider time slicing delay. The time slicing delay refers to the time period between locking on to a mobile TV channel and reaching the first burst of that TV channel. We let d_m be the maximal (the worse case) channel switching delay.

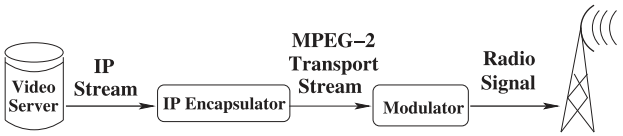

Fig. 1. The main components of mobile TV networks.

Using the above notations, we next state the scalable video broadcasting problem considered in this paper.

Problem 1. *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 layer c $(1 \leq c \leq C)$ has a bit rate of r_c Kbps, where $\sum_{c=1}^{C} r_c = r$. Mobile devices are classified into C classes so that devices in class c receive and decode all layers $\bar{c}$, where $\bar{c} \leq c$. The coded streams are encapsulated into bursts, and broadcast over the shared air medium. Design a broadcast scheme such that mobile devices in any class c achieves high energy saving and low channel switching delay from any channel to any other channel. A broadcast scheme assigns IP packets to individual bursts, and specifies the start time of each burst.*

Solving this problem at a base station allows users to watch more TV programs (due to longer battery lifetime), while keeping channel switching delay short. This will improve user satisfaction and eventually increase the number of subscribers as well as revenues for network operators.

4 Broadcasting Scalable Streams in Current Systems

In this section, we present the burst preparation in the current mobile TV networks, and discuss their limitations.

4.1 Burst Preparation in Mobile TV Networks

In DVB-H base stations, each mobile TV base station consists of three main components: video server, IP encapsulator, and modulator, as illustrated in Fig. 1. The video server encapsulates 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 (multiprotocol encapsulation) frames.

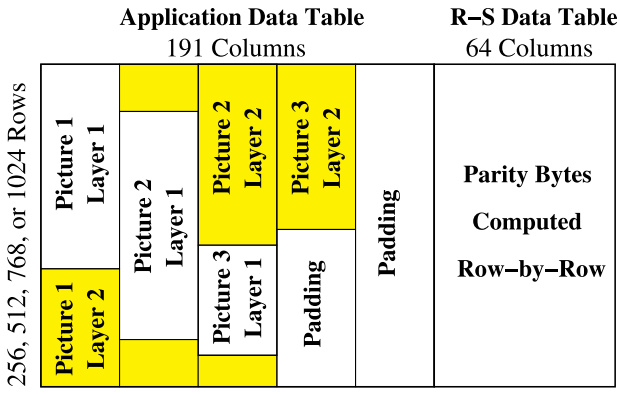

Fig. 2. The structure of MPE-FEC frame.

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.

Fig. 2 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. After the parity bytes are computed, the whole MPE-FEC frame is sent, column-by-column, as a burst. Note that, the padded zeros are for computing parity bytes only; they are not transmitted over the wireless medium [28]. We study the problem of designing broadcast schemes to optimize the quality of service, therefore the proposed schemes will be implemented in the IP encapsulator.

Despite different terminology, MediaFLO base stations also have to prepare transmission bursts in time-slicing manner [29]. More specifically, MediaFLO base stations transmit signals in a superframe structure, where each superframe consists of four frames and each frame has 250 OFDM symbols. A TV channel is assigned several OFDM symbols in every frame, and this assignment defines a period of time that mobile devices *must* turn on their radio components to receive data. Therefore, MediaFLO base stations prepare transmission bursts in the form of assigning OFDM symbols to individual TV channels, and most of our works in this paper are also applicable to MediaFLO networks.

4.2 Limitations of Current Systems

Traditional, nonscalable, 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. In contrast, scalable video coders encode each TV channel into a single video stream that can be sent and decoded at various bit rates. This is achieved by extracting substreams from the complete (original) stream, where each substream can be decoded and displayed at a lower perceived quality.

While scalable coded streams enable efficient substream extractions, broadcasting them in mobile TV networks is challenging because of the dependency among layers of the same coded stream. In the following, we give a brief summary of three possible schemes to broadcast scalable video streams in current mobile TV networks, and we provide insights on their shortcomings. We note that more detailed analysis on these schemes was given in our previous works [26], [27].

4.2.1 Single Service (SS)

To support heterogeneous mobile devices, network operators can upgrade the video server (see Fig. 1) to support scalable streams, and keep other components of the network unchanged. That is, each scalable stream is transmitted as a TV channel, and we refer to this scheme as single service (SS). With the SS scheme, existing broadcast systems can be used to broadcast scalable video streams. Unfortunately, this "patched" base station is not efficient as we 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 the IP encapsulator *sequentially* places 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. 2. 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 *must* 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 of that mobile device, because it has to open the radio component for a longer time period to receive some data that will be *dropped* eventually.

4.2.2 Parallel Services (PS)

We show that the SS scheme leads to *no* additional energy saving for mobile devices that cannot render the complete video streams. To cope with this issue, we need to design a better IP encapsulator that takes the layering structure of scalable streams into considerations when preparing transmission bursts. One way to achieve this is to send each layer of a TV channel as a parallel service, which can be implemented using multiple IP streams with different multicast IP addresses, or using multiple parallel elementary streams [22, Section 8.6]. Fig. 3 shows an example of broadcasting two TV channels with three layers, where each block inside bursts is a parallel service and carries IP packets of a specific layer only. We refer to this scheme as

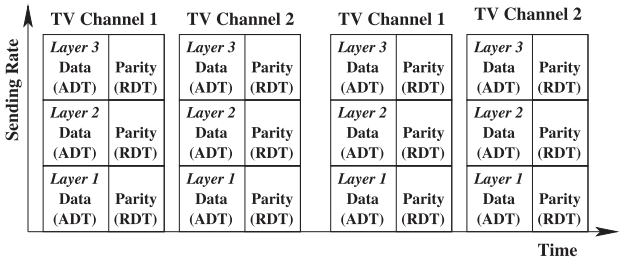

Fig. 3. Parallel Services.

parallel service (PS). Compared to SS, the PS scheme supports efficient demultiplexing of IP packets to individual layers based on either IP addresses or MPEG-2 PIDs (packet IDs). This allows mobile devices to extract substreams without inspecting the complete stream, and thus reduces its processing overhead. Unfortunately, as all services are sent in parallel, mobile devices have to open their radio components for the complete burst duration. Therefore, all mobile devices achieve the same energy saving despite how many layers they receive and decode. Observe that PS is inefficient because mobile devices may receive and discard data (in irrelevant layers) that are *useless* to them. This leads to longer on time of radio components, and thus lower energy saving.

4.2.3 Layer-Aware FEC (LAF)

To cope with the inefficiency of PS, the IP encapsulator may rearrange the IP packets received from streaming servers, so that packets belonging to layer c are sent before packets belonging to layer $c+1$. This allows mobile devices that do not decode the complete scalable streams to turn off their radio components before each burst ends. If we reuse the illustrative example of SS, the IP packets should be sent in the following order: (picture 1, layer 1), (picture 2, layer 1), (picture 3, layer 1), (picture 1, layer 2), (picture 2, layer 2), (picture 3, layer 2), as illustrated in Fig. 4. The layer number can be prepended before the MPE header as an one-byte extension header, which allows mobile devices to efficiently determine the boundaries between layers, e.g., between (picture 3, layer 1) and (picture 1, layer 2).

However, even after reordering IP packets, mobile devices still have to receive complete bursts in order to perform error correction, which again prevents them from getting higher energy saving. This is because the FEC parity bytes are sent after all the IP packets, at the end of each burst. To address this issue, we can compute parity bytes *column-by-column*, and send these bytes right after each column of data bytes. This allows mobile devices to perform error corrections without receiving complete bursts. That is, mobile devices can receive partial bursts and turn off the radio components to save energy. We call this new frame format as Layer-Aware FEC (LAF) frame, which is illustrated in Fig. 4.

While LAF frame allows mobile devices to efficiently receive and extract substreams, it has several drawbacks. First, LAF does not comply with mobile TV standards, which causes compatibility issues between the base station and mobile devices. Second, implementing LAF requires significant changes as error corrections are usually done in hardware/firmware for the sake of performance. Third, computing parity bytes column-by-column makes the FEC decoder vulnerable to bursty channel errors because it does not provide virtual time interleaving as by MPE-FEC frames [28]. Most importantly, in Section 5, we prove in Lemma 1 that the LAF scheme achieves lower energy saving compared to the scheme developed in that section.

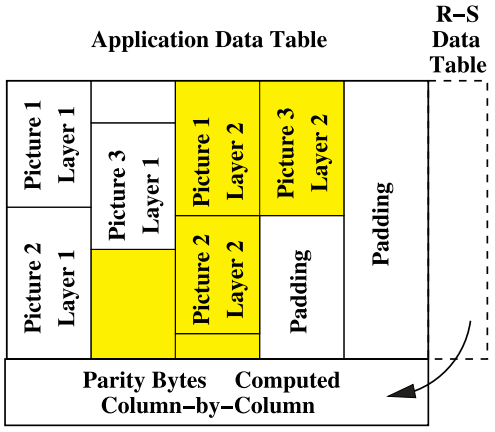

Fig. 4. Layer-Aware FEC frame.

4.2.4 Summary

We have presented three schemes to broadcast scalable video streams in current mobile TV networks: SS, PS, LAF. The SS and PS schemes require all mobile devices, despite which classes they are in, to receive complete scalable streams. Therefore, they result in *no* energy saving differentiation as all mobile devices turn on their radio components for the same amount of time. While the LAF scheme enables energy saving differentiation, it is:

1. not standard compliant,
2. hard to implement,
3. vulnerable to bad channel conditions, and
4. achieves lower energy saving than the scheme proposed in Section 5.

Hence, we conclude that current mobile TV broadcast networks cannot efficiently support scalable video streams, and we need to design better broadcast schemes to solve Problem 1.

5 Generalized Layer-Aware Time Slicing: GLATS

We propose a new broadcast scheme which we call Generalized Layer-Aware Time Slicing (GLATS).

5.1 Overview

The GLATS scheme works on a recurring window, and its key feature is that the IP encapsulator prepares a different burst (or MPE-FEC frame) in the recurring window for each layer of every TV channel. More specifically, every burst in the GLATS scheme consists of IP packets from the same layer of the same TV channel, which allows mobile devices to safely skip bursts that contain IP packets for irrelevant layers and save more energy. To illustrate, Fig. 5 shows an example of the GLATS scheme, in which all IP packets in

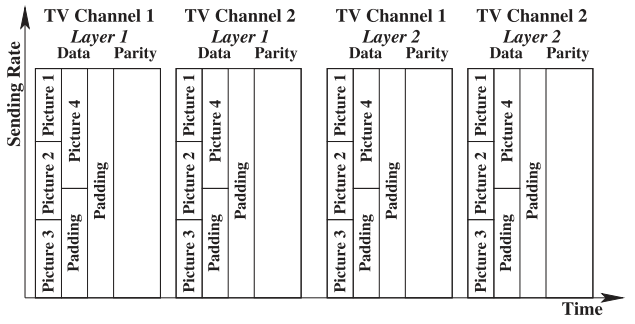

Fig. 5. Proposed broadcast scheme: GLATS.

the leftmost burst belong to the base layer of TV channel 1, while IP packets in the third burst belong to layer 2 of TV channel 1. With the GLATS scheme, mobile devices that are rendering TV channel 1 at the base layer quality do *not* need to receive the third burst, because it is of no use to them. More importantly, mobile devices know which layer the IP packets are in, even before receiving a burst. Therefore, mobile devices need *not* open their radio components and inspect IP packets for substream extractions, and thus more energy can be saved. Moreover, no additional signaling from the base station to mobile devices is required to determine which bursts (layers) to receive, mobile devices only need to know the total number of layers C, which is already sent to them for decoding the scalable stream. Furthermore, the GLATS scheme naturally works with existing MPE-FEC frame, because whenever a mobile device decides to decode layer c, it has to receive all IP packets in layer c for successful video reconstruction. Therefore, all IP packets in ADT will be received before error correction, and the FEC decoder implemented in hardware/firmware can still be used.

We next derive the burst start time and the burst size of each layer of individual TV channels. First, the number of TV channels that can be concurrently broadcast is $S = \lfloor R/r \rfloor$. We let b Kb be the burst size of base layers. The burst size for layer c is proportionally set to br_c/r_1 Kb. Since the recurring window consists of a burst for every layer of each TV channel, the total burst size of the complete stream of a TV channel within the window is $\sum_{c=1}^{C} br_c/r_1 = b\sum_{c=1}^{C} r_c/r_1 = br/r_1$ Kb. Since the broadcast bandwidth is R Kbps, the recurring window size is $(\frac{br/r_1}{R}) \times S = (brS)/(r_1R)$, where $\frac{br/r_1}{R}$ is the amount of time the GLATS scheme *must* reserve for a TV channel, and there are S TV channels in total. Finally, the GLATS scheme produces the time slicing schedule with bursts

GLATS:

$$\left\{ \left\langle s,\ c,\ \frac{b\sum_{i=1}^{c-1} r_i/r_1}{R} S + \frac{br_c/r_1}{R}(s-1),\ br_c/r_1 \right\rangle \ \middle|\ \forall s = 1, 2, \ldots, S,\ \ c = 1, 2, \ldots, C \right\}, \tag{1}$$

where the four elements are the TV channel, device class, burst start time, and burst size.

The GLATS scheme in (1) is general, as it supports layers with *diverse* bit rates. In contrast, the Layer-Aware Time Slicing (LATS) scheme proposed in our earlier work [26] assumes all layers have the same bit rate.

5.2 Analysis

We next prove the correctness of the GLATS scheme, and we quantify its performance.

Theorem 1. *The GLATS scheme (1) specifies a feasible time slicing scheme for a recurring windows of $\frac{brS}{r_1R}$ sec, where 1) no two bursts overlap with each other, and 2) bursts are long enough to send data for all mobile devices to playout until the next burst. Furthermore, the energy saving achieved by mobile devices in class c is given by*

$$\gamma_c = 1 - \frac{\sum_{i=1}^{c} r_i}{rS} - \frac{RT_o cr_1}{brS} \quad \text{where c} = 1, 2, \ldots, \text{C}. \tag{2}$$

Finally, the maximum channel switching delay is

$$d_m = b/r_1. \tag{3}$$

Proof. First, sending a burst of br_c/r_1 Kb takes time $\frac{br_c/r_1}{R}$ sec to transmit. Thus, by the definition of (1), the resulting scheme has no overlapping bursts. Second, consider any arbitrary layer c, where $1 \le c \le C$, the required amount of data for smooth playout is $\frac{brS}{r_1R} \times r_c \le \frac{br}{r_1R} \times \frac{R}{r} \times r_c = b\frac{r_c}{r_1}$, where the inequality comes from the definition of S. This inequality shows that the scheduled time period is long enough to carry the playout data.

For energy saving, observe that mobile devices in class c turn on their radio components for c times in each recurring window. Combining this with the total burst size, we have

$$\gamma_c = 1 - \frac{\frac{b\sum_{i=1}^{c} r_i/r_1}{R} + cT_o}{\frac{brS}{r_1R}}.$$

In this equation, the first term of the numerator accounts for the time to receive actual video data, the second term of the numerator represents the overhead durations, and the denominator is the recurring window size. Manipulating this equation yields (2).

For channel switching delay, we first consider 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., $d_m = \frac{brS}{r_1R}$. Following the definition of S, we write $d_m = \frac{brS}{r_1R} \le \frac{br}{r_1R} \times \frac{R}{r} = b/r_1$, which yields (3). This result can be extended to any class c: mobile devices, s 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. □

We then compare the performance of the GLATS scheme against and the LAF scheme presented in Section 4.2.

Lemma 1. *The GLATS scheme achieves higher energy saving than the LAF scheme for class c mobile devices if $c \neq C$. These two schemes lead to the same energy saving for class C mobile devices.*

Proof. With the LAF scheme, mobile devices in class c ($c = 1, 2, \ldots, C$) receive a fraction/prefix of every burst, and turn off their radio components earlier to save energy. Since b is the base layer burst size in the GLATS

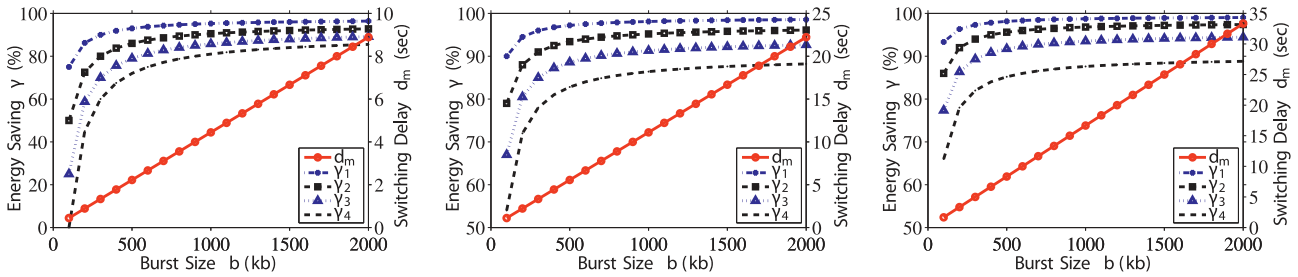

Fig. 6. Diverse energy saving supported by the GLATS scheme, and resulting channel switching delay. (a) Uniform. (b) Linear. (c) Exponential.

scheme, we consider a LAF scheme, where each TV channel has C bursts and their total size is br/r_1, for a fair comparison. Since the network bandwidth is R, class c mobile devices must open their radio components for

$$\frac{b\sum_{i=1}^{c} r_i/r_1}{R}$$

to receive the actual video data. The class c mobile devices need to turn on their radio components C times in every recurring window of $(brS)/(r_1R)$. This is because, as Fig. 4 illustrates, packets belong to layers $c \leq C$ scatter over all C bursts. Therefore, we write the energy saving achieved by class c mobile devices in the LAF scheme as

$$\hat{\gamma}_c = 1 - \frac{\sum_{i=1}^{c} r_i}{rS} - \frac{RT_oCr_1}{brS}, \text{ where c} = 1, 2, \ldots, \text{C}. \quad (4)$$

Comparing (4) against (2) yields the lemma. □

This lemma shows that the GLATS scheme outperforms the LAF scheme in terms of energy saving, as aforementioned in Section 4.2.

The following corollary is a direct result of Theorem 1:

Corollary 1. *Consider a special case of the GLATS scheme, where all layers have the same bit rate, i.e., $r_c = r/C$ for all $c = 1, 2, \ldots, C$. The energy saving achieved by class c mobile devices is given as $\gamma_c = 1 - \frac{c}{CS} - \frac{RT_oc}{bCS}$. Moreover, the maximal channel switching delay is given as $d_m = bC/r$.*

This corollary considers a simplified scheme with scalable streams where layers have uniform bit rates. Notice that we call this simplified scheme as the LATS scheme in our previous results [26], and this corollary is consistent with the analysis presented in [26].

5.3 Discussion

We apply typical network parameters to numerically study the performance of the GLATS scheme. More precisely, we consider a mobile TV network with bandwidth $R = 9$ Mbps, and several TV channels that are encoded into scalable streams at full rate of $r = 900$ Kbps. Each of these scalable streams is divided into four layers. That is, we consider four classes of mobile devices. We note that the GLATS scheme supports layers with diverse bit rates, and we consider three different scenarios to assign bit rates to individual layers, which are 1) uniform, in which $r_1 = r_2 = \cdots = r_C = r/C$, 2) linear, in which $r_c = c \times \frac{2r}{(1+C)C}$, and 3) exponential, in which $r_c = 2^{c-1} \times \frac{r}{2^C-1}$. We then assume the overhead duration $T_o = 100$ msec, and we vary the base layer burst size b. We compute the energy saving and channel switching delay of mobile devices in different classes using formulas derived in Theorem 1, and we plot the results in Fig. 6, which shows the energy saving γ_c for class c devices ($c = 1, 2, 3, 4$) and channel switching delay d_m. This figure clearly shows that the GLATS scheme allows mobile devices to achieve a wide range of energy saving values. For example, letting $b = 1{,}000$ Kb, 75-95 percent energy saving is possible in uniform scenario, and 85-95 percent energy saving can be achieved in the other two considered scenarios. The range of supported energy saving is even larger when b is smaller. This energy saving diversity enables mobile devices that are short of resources to be conservative on energy for longer watch time, while others can render TV channels at higher perceived quality.

We mention that this diversity may come at an expense of high channel switching delay. For example, as illustrated in Fig. 6a, the channel switching delay is about 5 sec when $b = 1{,}000$ Kbps, which may not be desirable for some users. A simple way to cope with long delays is to send bursts more often, which is equivalent to reducing b. Smaller b values, however, may lead to low energy saving as illustrated by Fig. 6. For example, in Fig. 6a, setting $b =$ 200 Kb results in about 40 percent energy saving for mobile devices that render the complete video streams, which is significantly smaller than the 85 percent energy saving that is achieved when $b = 2{,}000$ Kb. Therefore, reducing b is not an efficient way to control channel switching delays, and we need to develop a better solution.

6 Generalized Layer-Aware Time Slicing with Delay Bound: GLATSB

We propose a new broadcast scheme to achieve energy saving diversity without incurring long channel switching delays, and we refer to it as Generalized Layer-Aware Time Slicing with Delay Bound (GLATSB).

6.1 Overview

The GLATSB scheme is an extension of the GLATS scheme presented in Section 5, and it aims to reduce channel switching delays. The delay reduction is based on the following observation. Long channel switching delays are partially due to the dependency among different layers.

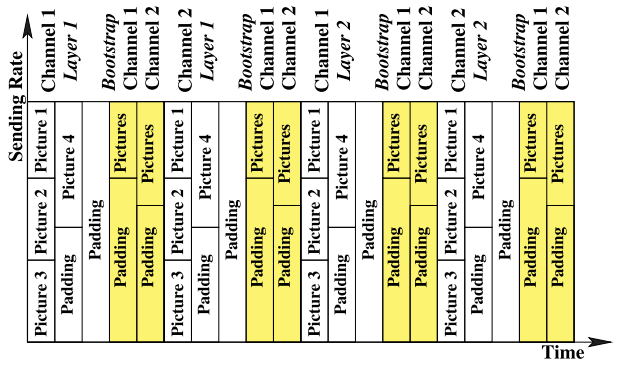

Fig. 7. Proposed broadcast scheme with switching delay bound: GLATSB.

This is because, despite how many layers a mobile device plans to receive, it cannot decode the coded stream until the base layer (layer 1) arrives, and the waiting time can be as long as the whole recurring window. To better control the channel switching delay, we propose to insert short and frequent *bootstrap* bursts between any two adjacent bursts defined in the GLATS scheme, and send the base layers of TV channels using these bootstrap bursts. We refer to the bursts defined in the GLATS scheme as *normal* bursts.

Fig. 7 illustrates how we insert the bootstrap bursts: a small piece of base layer from each TV channel is sent between two adjacent normal bursts. This figure shows that two bootstrap bursts for TV channel 1 and 2, respectively, are added 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, and thus the overhead duration T_o is relatively insignificant to them.

We next derive the burst start time and the burst size of each layer of individual TV channels. First, the number of TV channels that can be concurrently broadcast is $S = \lfloor (R)/(r + r_1) \rfloor$. This is because we send each base layer *twice* in the recurring window: both in bootstrap and normal bursts. We let b be the normal burst size for base layers. The normal burst size for layer c is proportional to r_c, and is written as br_c/r_1 where $c = 1, 2, \ldots, C$. Then, the recurring window size is $(b(r_1 + r)S)/(r_1 R)$ sec. In each recurring window, the total bootstrap burst size of any TV channel $s (s = 1, 2, \ldots, S)$ is b. Since the bit rate of layer c is given as r_c, for any $c = 1, 2, \ldots, C$, the total bootstrap burst size of any TV channel s between its layer c normal burst and its layer $c + 1$ normal burst is

$$b \frac{r_c}{\sum_{i=1}^{C} r_i} = br_c/r.$$

We write $\bar{b}(c) = br_c/r$ in the rest of our derivation. Last, the total bootstrap burst size between the layer c normal burst and the layer $c + 1$ normal burst is equally divided into S bootstrap bursts, where each bootstrap burst size is $\bar{b}(c)/S = (br_c)/(rS)$. Fig. 8 shows an illustrative example of the derivation of bootstrap burst sizes. Finally the GLATSB scheme produces the time slicing schedule with bursts

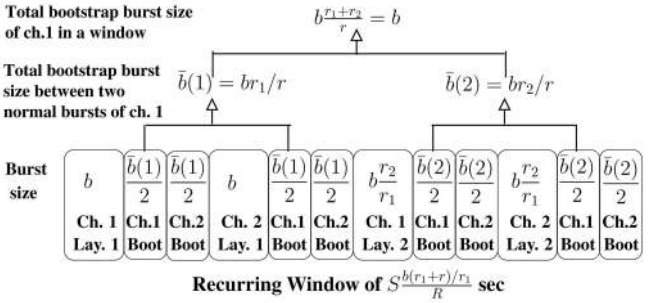

Fig. 8. Derivation of bootstrap burst sizes in the proposed GLATSB scheme.

GLATSB (normal):

$$\left\{ \left\langle s,\ c,\ b\frac{\sum_{i=1}^{c-1} r_i/r_1}{R} S + \frac{\sum_{i=1}^{c-1} \bar{b}(i)}{R} S + \frac{br_c/r_1}{R}(s-1) + \frac{\bar{b}(c)(s-1)}{R},\ \frac{br_c}{r_1} \right\rangle \middle| \forall s = 1, 2, \ldots, S,\ c = 1, 2, \ldots, C \right\}, \tag{5}$$

GLATSB (bootstrap):

$$\left\{ \left\langle s,\ c,\ b\frac{\sum_{i=1}^{c-1} r_i/r_1}{R} S + \frac{\sum_{i=1}^{c-1} \bar{b}(i)}{R} S + \frac{br_c/r_1}{R} k + \frac{\bar{b}(c)(k-1)}{R} + (s-1)\frac{\bar{b}(c)}{S},\ \frac{\bar{b}(c)}{S} \right\rangle \middle| \forall s = 1, 2, \ldots, S, \ c = 1, 2, \ldots, C, \text{and } k = 1, 2, \ldots, S \right\}, \tag{6}$$

where the four elements are the TV channel, device class, burst start time, and burst size, respectively. In (6), we use c and k for the bootstrap burst indices. We notice that the second and the fourth terms of the burst start time in (5) consider the amount of air medium time occupied by the bootstrap bursts. The last term in (6) splits the time between two normal bursts among bootstrap bursts of individual TV channels. These two equations define the proposed GLATSB scheme.

The GLATSB scheme is general, as it supports layers with *diverse* bit rates. In contrast, the Layer-Aware Time Slicing with Delay Bound (LATSB) scheme proposed in our earlier work [27] assumes all layers have the same bit rate.

6.2 Analysis

We prove the correctness of the proposed GLATSB scheme in the next theorem. We also quantify its performance in the same proof.

Theorem 2. *The GLATSB scheme (5) and (6) specifies a feasible time slicing scheme for a recurring window of $\frac{b(r_1+r)S}{r_1 R}$ sec, where 1) no two bursts overlap with each other, and 2) bursts are long enough to send data for all mobile devices to playout till the next burst. Furthermore, the energy saving achieved by mobile devices in class c ($c = 1, 2, \ldots, C$) is given by*

$$\gamma_c = 1 - \frac{\sum_{i=1}^{c} r_i}{(r_1 + r)S} - \frac{RT_o c r_1}{b(r + r_1)S}. \tag{7}$$

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

$$\gamma_b = 1 - \frac{r_1}{(r_1+r)S} - \frac{RCT_o r}{(r_l+r)b}. \quad (8)$$

Finally, the maximal channel switching delay is

$$d_m = \frac{b(r+r_1)\max_{c=1}^{C} r_c}{Rrr_1}. \quad (9)$$

Proof. First, the construction of the GLASTB scheme guarantees that no bursts intersect with each other in time as the air medium of R Kbps is fast enough to transmit all bursts within allocated time slots. Second, consider all S bootstrap bursts for an arbitrary TV channel between its normal bursts of layer c and $c+1$. Each of these bootstrap bursts is $\bar{b}(c)/S$ long, and thus the total burst size is $\bar{b}(c) = \frac{br_c/r}{S}$ Kb. Furthermore, the bootstrap bursts after the normal burst of layer $c+1$ arrives $\frac{\bar{b}(c)+br_c/r_1}{R}$ sec later. Because the bootstrap bursts carry the base layer with a bit rate of r_1 Kbps, each bootstrap burst can support a playout time of $\frac{br_c/r}{Sr_1} \geq \frac{br_c/(rr_1)}{R/(r+r_1)} = \frac{\bar{b}(c)+br_c/r_1}{R}$, where the inequality follows the definition of S (i.e., $S \leq \frac{R}{r+r_1}$). Since the playout time is no shorter than the time period to the next burst, the bootstrap bursts are long enough for smooth playout. We next consider two adjacent normal bursts for the same layer c of a specific TV channel. The burst transmits up to $b\frac{r_c}{r_1}$ Kb data, and the time difference between them is $S\frac{b(r_1+r)}{r_1R}$ sec. Since these two normal bursts carry the layer c with a bit rate of r_c Kbps, each such normal burst can support a playout time of $\frac{b(r_1+r)S}{r_1R} \leq \frac{b(r_1+r)R}{r_1R(r+r_1)} = b\frac{r_c}{r_1}$, which indicates that the normal bursts are long enough for smooth playout. This proves the correctness of the GLATSB scheme.

Next, following the definition of energy saving, the energy saved for mobile devices that receive c layer is

$$\gamma_c = 1 - \frac{\frac{b\sum_{i=1}^{c} r_i/r_1}{R} + cT_o}{S\frac{b(r_1+r)}{r_1R}},$$

and the energy saved for mobile devices that receive bootstrap bursts is

$$\gamma_b = 1 - \frac{b/R + CST_o}{S\frac{b(r_1+r)}{r_1R}}.$$

In these two equations, the first term of the numerator is the time to receive actual video data, the second term of it accounts for the total overhead duration, and the denominator is the recurring window size. Simplifying these two equations leads to (7) and (8).

Finally, the channel switching delay is the maximal time difference between two bootstrap bursts for the same TV channel. Note that, the time difference between two bootstrap bursts after a layer c normal burst is $\frac{\bar{b}(c)+br_c/r_1}{R}$, where the first term in the numerator accounts for the bootstrap bursts and the second term of it accounts for the normal burst. Hence, the maximal channel switching delay is

$$d_m = \frac{\max_{c=1}^{C}[\bar{b}(c) + br_c/r_1]}{R} = b\frac{(r+r_1)\max_{c=1}^{C} r_c}{Rrr_1},$$

which leads to (9). □

The next corollary enables network operators to bound the channel switching delay precisely. This is a direct result of (9).

Corollary 2. *For a given maximal channel switching delay d_m, applying the GLATSB scheme (5) and (6) 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 $b_m = d_m\frac{Rrr_1}{(r+r_1)\max_{c=1}^{C} r_c}$.*

The following corollary is a direct result of Theorem 2:

Corollary 3. *Consider a special case of the GLATSB scheme, where the layers have uniform bit rates, i.e., $r_c = r/C$ for all $c = 1, 2, \ldots, C$. The energy saving achieved by mobile devices in class c ($c = 1, 2, \ldots, C$) is given by $\gamma_c = 1 - \frac{c}{(C+1)S} - \frac{RT_o c}{b(C+1)S}$. The energy saving achieved by mobile devices that receive the bootstrap bursts is*

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

Finally, the maximal channel switching delay is $d_m = \frac{b+b/C}{R}$.

This corollary considers simplified scalable streams where layers have uniform bit rates. Notice that we call this simplified scheme as the LATSB scheme in our previous results [27], and this corollary is consistent with the analysis presented in [27].

6.3 Discussion

We apply the same network parameters used in Section 5.3 to numerically study the performance of the GLATSB scheme. We consider three different scenarios to assign bit rate to individual layers, which are uniform, linear, and exponential. We vary the base layer burst size b, and we compute the energy saving and channel switching delay of mobile devices in different classes using formulas derived in Theorem 2. We plot the results in Fig. 9, which shows the energy saving γ_c for class c device ($c = 1, 2, 3, 4$) and channel switching delay d_m. This figure indicates that the GLATSB scheme enables mobile devices to achieve a wide range of energy saving values. More importantly, this figure reveals that, compared to the GLATS scheme, the GLATSB scheme dramatically reduces the channel switching delay: less than 300 msec delay can be achieved in the uniform scenario, and the delays never exceed 2 sec in all considered scenarios. We mention that, in the analysis of energy saving, we assume that mobile devices receive bootstrap bursts for short, *transient* time periods, and only consider mobile devices that receive normal bursts. We eliminate this assumption when conducting the the empirical evaluation in Section 7.

Last, we notice that low channel switching delays achieved by GLATSB scheme comes at an expense of bandwidth overhead of Sr_1 Kbps, since the base layers

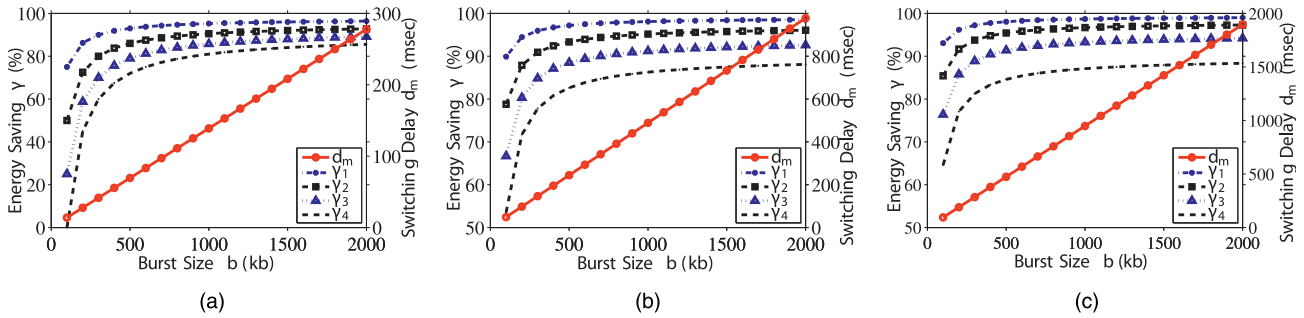

Fig. 9. Diverse energy saving supported by the GLATSB scheme, and resulting channel switching delay. (a) Uniform. (b) Linear. (c) Exponential.

are transmitted *twice*. This overhead is, however, controllable: network operators may decide to use a lower base layer rate.

7 Evaluation

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

7.1 Mobile TV Testbed

We have implemented a DVB-H testbed network in our Lab. Our mobile TV testbed has two main parts: base station and receivers, which are described below. We use a commodity Linux box as the base station, which runs our broadcast software that consists of video streaming server, IP encapsulator, and DVB-H modulator. The video streaming server supports scalable streams and sends RTP packets to the IP encapsulator. The IP encapsulator puts these RTP packets into MPE and MPE-FEC sections, and fits these sections into time slicing bursts. The encapsulated MPE and MPE-FEC sections form MPEG-2 traffic streams, which are sent to the DVB-H modulator and then transmitted over the air to mobile devices. We installed a DVB-H PCI modulator in our base station, which implements the physical layer of the protocol stack and transmits signals via an indoor antenna. The RF output level of the modulator, however, is quite low (~ -29 dBm). We use a low-power amplifier to boost the RF signals to about 0 dBm, which gives us more than 20-meter range in our lab environment with a regular cellular phone antenna.

We use Nokia N96 mobile phones as receivers, which come with the receiver-side of the DVB-H protocol and video player. While mobile phones help in assessing the visual quality of videos, they do not provide detailed logging functions of the low-level signals, which are needed to evaluate the performance of our proposed schemes. To address this limitation, we added a DVB-H protocol analyzer to the testbed. This analyzer is attached to a PC via a USB port and comes with visualization software for analysis. The analyzer records traffic streams as well as provides detailed information on the RF signals, the MPEs, jitter, time slicing, and so on. More details about our testbed can be found in [30].

7.2 Setup

We have implemented both the GLATS and GLATSB schemes in the testbed. We notice that, to the best of our knowledge, there exists no other broadcast schemes for broadcasting scalable video streams in current systems. For comparison, we implement the SS and PS schemes presented in Section 4.2, respectively. These two schemes work in current systems, and achieve the same energy saving and channel switching delay. Hence, we denote them as CUR in the figures. Using the H.264/SVC [31] reference software [32], 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 (352×288) resolution at 30 fps frame rate. Given that the City sequence is quite short, we concatenated it several times to form a 5-min video sequence. We broadcast the same sequence in all TV channels, because different video characteristics do not significantly affect the quality of service metrics considered in this section. Furthermore, as there is no effective rate control algorithms built in the H.264/SVC reference software, we had to search for appropriate quantization parameter (QP) values by encoding the same video many times. This is very time consuming even for a single video sequence, e.g., encoding a 10-sec sequence could take more than 12 hours in our experience. With the coded stream, we configured the modulator card to use 8 MHz bandwidth, quadrature phase-shift keying (QPSK) modulation, 3/4 code ratio, and 1/8 guard interval. This leads to the channel bandwidth of 8.289 Mbps [33]. We broadcast 4 TV channels for 5 min using the GLATS and GLATSB schemes, and repeated the same test using the CUR scheme.

We have instrumented the testbed to save log files for offline analysis. The 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 to 60 sec. We also varied the burst size from 500 to 1,500 Kb. If not otherwise specified, we present sample results for 60-sec average watch time, and 1,000 Kb burst size.

We ran the simulation against every log file collected from the 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

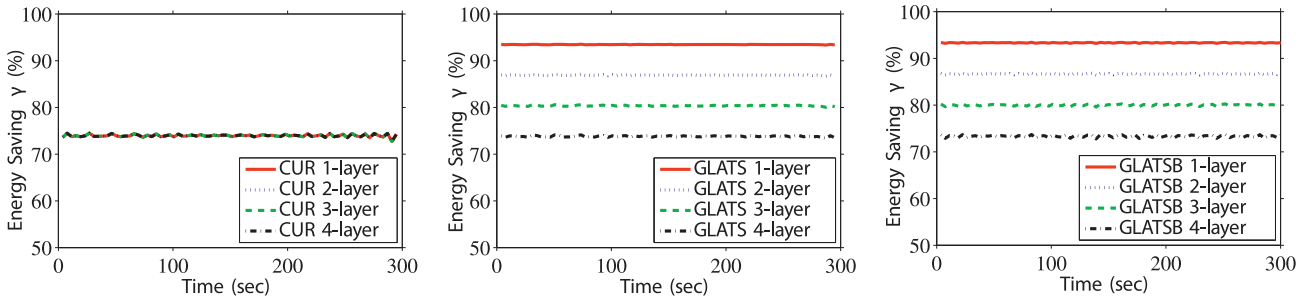

Fig. 10. Sample energy saving achieved by a mobile device in each classes: (a) CUR, (b) GLATS, and (c) GLATSB.

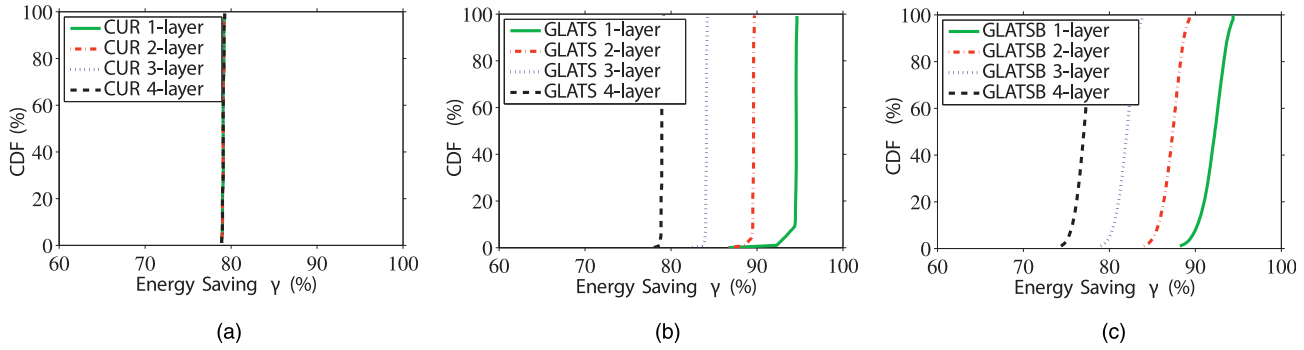

Fig. 11. Energy saving achieved by (all) mobile devices in different classes: (a) CUR, (b) GLATS, and (c) GLATSB.

the selected TV channel and computing the time difference between it and the channel switching event. We let the overhead duration $T_o = 100$ msec, and measured the energy saving by calculating the fraction of time that the radio component is in between every two channel switching events. We emphasize that when computing the energy saving for the GLATSB 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 Section 6.3.

7.3 Results

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

7.3.1 Diverse Energy Saving

We first plot the energy saving achieved by a sample mobile device in each mobile device class in Fig. 10. We then compute the average energy saving of all mobile devices, and report the cumulative distribution function (CDF) curves in Fig. 11. These two figures show that the CUR scheme leads to no energy saving differentiation, while the GLATS and GLATSB schemes enable proportional energy saving for mobile devices in different classes. This confirms that broadcasting scalable streams using the CUR scheme cannot support heterogeneous mobile devices.

7.3.2 Channel Switching Delay

We plot the CDF curves of channel switching delays in Fig. 12. This figure shows that the GLATS scheme may lead to high channel switching delay: up to 4-sec delay in this experiment. In contrast, GLATSB scheme results in *negligible* channel switching delay: about 200 msec is achieved.

7.3.3 Implication of per Channel Watch Time

In the GLATSB 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 to 60 sec. We plot the CDF curves of average energy saving of mobile devices that receive the complete streams in Fig. 13. This figure shows that frequent channel switching events only degrade the energy saving in extreme cases, in which users constantly change TV channels.

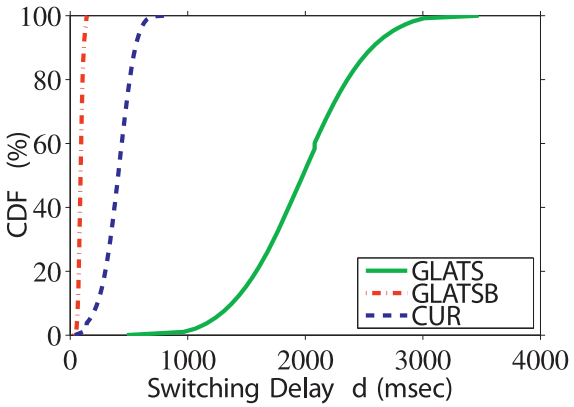

Fig. 12. Delay of different schemes.

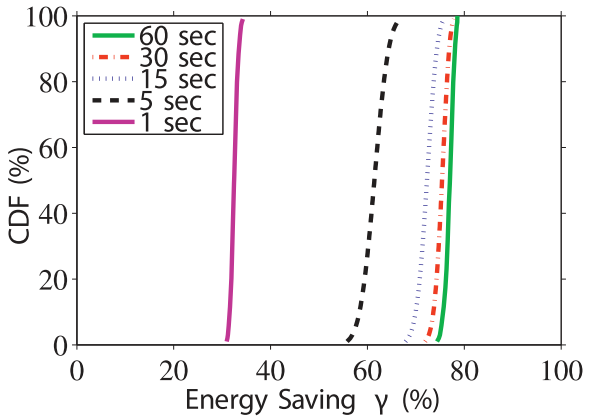

Fig. 13. Implication of per-channel watch time.

7.3.4 Implication of Burst Size

We next vary burst size b from 500 to 1,500 Kb. This covers the whole practical range of b values, since 1,565 Kb is the maximal burst size specified by DVB-H standard documents [22]. We first present results for the mobile devices that receive the complete streams. We plot the CDF curves of energy saving and channel switching delay in Fig. 14. Fig. 14a shows that increasing burst size allows the GLATS scheme to achieve higher energy saving. However, Fig. 14b reveals that larger burst size also increases the channel switching delay: letting $b = 1{,}500$ Kb leads to as high as 6-sec delay. These two figures show that the GLATS scheme uses b to control the tradeoff between energy saving and channel switching delay, which is inefficient. In contrast, Figs. 14c and 14d show that increasing b in the GLATSB 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. We note that the above discussions are applicable for all mobile devices despite how many layers they receive and decode. Figures for other classes are not shown due to the space limitations.

8 Conclusions

In this paper, we studied the problem of broadcasting scalable video streams over a shared air medium to mobile devices with heterogeneous resources, such that mobile devices can render the most appropriate substreams while achieving high energy saving and low channel switching delay. We analyzed current mobile TV networks, and showed that they cannot efficiently broadcast scalable coded streams. We then proposed two scalable broadcast schemes: GLATS and GLATSB. We formally proved the correctness of the proposed schemes, and we analytically quantified their performance in terms of energy saving and channel switching delay. The proposed schemes can be implemented in current base stations, and they produce bursts of traffic that are compliant with current mobile video broadcasting standards such as DVB-H. The main difference between GLATS and GLATSB is that the latter ensures short channel switching delays, but at a small cost of lower bandwidth utilization. We implemented the GLATS and GLATSB schemes in a real mobile TV testbed to show their practicality and efficiency. Our extensive experiments showed that both the GLATS and GLATSB schemes enable energy saving differentiation: between 75 and 95 percent were observed. Moreover, the GLATSB scheme also achieves low channel switching delays: 200 msec is possible with typical system parameters.

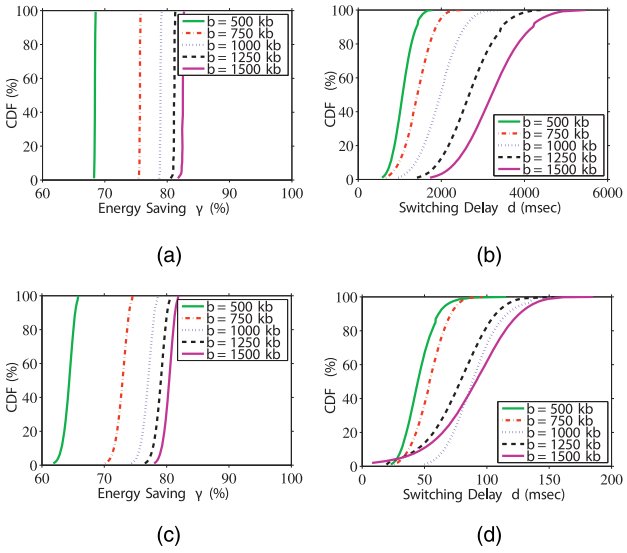

Fig. 14. Implication of burst size on energy saving and channel switching delay: mobile devices that receive all 4 layers. (a) GLATS, energy saving. (b) GLATS, channel switching delay. (c) GLATSB, energy saving. (d) GLATSB, channel switching delay.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by the British Columbia Innovation Council.

References

[1] "Mobile TV Predicted to Be a Hit," http://news.bbc.co.uk/2/hi/technology/6639249.stm, 2007.

[2] X. Liang, B. Zhang, and R. Taylor, "Development of the Mobile Phone Television Market in China," *Proc. Conf. Pacific Telecomm. Council (PTC '08)*, pp. 1-22, Jan. 2008.

[3] "Customers Angered as iPhones Overload AT&T," http://www.nytimes.com/2009/09/03/technology/companies/03att.html, Sept. 2009.

[4] G. May, "The IP Datacast System—Overview and Mobility Aspects," *Proc. IEEE Int'l Symp. Consumer Electronics (ISCE '04)*, pp. 509-514, Sept. 2004.

[5] M. Kornfeld and G. May, "DVB-H and IP Datacast—Broadcast to Handheld Devices," *IEEE Trans. Broadcasting*, vol. 53, no. 1, pp. 161-170, Mar. 2007.

[6] *Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H), European Telecommunications Standards Institute (ETSI) Standard EN 302 304 Ver. 1.1.1*, Nov. 2004.

[7] "FLO Technology Overview," http://www.mediaflo.com/news/pdf/tech_overview.pdf, 2009.

[8] "Evolution of DVB-T Front-End Receivers through Integration," http://www.dibcom.info/Images/Upload/pdf/whitePaper2_integration_MD_V2. pdf, June 2007.

[9] C. Hsu and M. Hefeeda, "Broadcasting Video Streams Encoded with Arbitrary Bit Rates in Energy-Constrained Mobile TV Networks," *IEEE/ACM Trans. Networking*, vol. 18, no. 3, pp. 681-694, June 2010.

[10] M. Hefeeda and C. Hsu, "On Burst Transmission Scheduling in Mobile TV Broadcast Networks," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 610-623, Apr. 2010.

[11] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile Video Transmission Using Scalable Video Coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1204-1217, Sept. 2007.

[12] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier, "Layered H.264 Video Transmission with Hierarchical QAM," *J. Visual Comm. and Image Representation*, vol. 17, no. 2, pp. 451-466, Apr. 2006.

[13] C. Hellge, T. Schierl, and T. Wiegand, "Mobile TV Using Scalable Video Coding and Layer-Aware Forward Error Correction," *Proc. IEEE Int'l Conf. Multimedia and Expo (ICME '08)*, pp. 1177-1180, Apr. 2008.

[14] H. Mansour, V. Krishnamurthy, and P. Nasiopoulos, "Channel Aware Multiuser Scalable Video Streaming over Lossy Under-Provisioned Channels: Modeling and Analysis," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1366-1381, Nov. 2008.

[15] T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.

[16] V. Vadakital, M. Hannuksela, and M. Gabbouj, "Time-Interleaved Simulcast and Redundant Intra Picture Insertion for Reducing Tune-In Delay in DVB-H," *Proc. IEEE Int'l Packet Video Workshop (PV '07)*, pp. 123-132, Nov. 2007.

[17] M. Rezaei, M. Hannuksela, and M. Gabbouj, "Tune-In Time Reduction in Video Streaming over DVB-H," *IEEE Trans. Broadcasting*, vol. 53, no. 1, pp. 320-328, Mar. 2007.

[18] M. Rezaei, I. Bouazizi, and M. Gabbouj, "Joint Video Coding and Statistical Multiplexing for Broadcasting over DVB-H Channels," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1455-1464, Dec. 2008.

[19] M. Rezaei, M. Hannuksela, and M. Gabbouj, "Video Encoding and Splicing for Tune-In Time Reduction in IP Datacasting (IPDC) over DVB-H," *Proc. IEEE Int'l Conf. Multimedia and Expo (ICME '06)*, pp. 601-604, July 2006.

[20] M. Rezaei, M. Hannuksela, and M. Gabbouj, "Video Splicing and Fuzzy Rate Control in IP Multi-Protocol Encapsulator for Tune-In Time Reduction in IP Datacasting (IPDC) over DVB-H," *Proc. IEEE Int'l Conf. Image Processing (ICIP '06)*, pp. 3041-3044, Oct. 2006.

[21] X. Yang, Y. Song, T. Owens, J. Cosmas, and T. Itagaki, "Performance Analysis of Time Slicing in DVB-H," *Proc. Joint IST Workshop Mobile Future and Symp. Trends in Comm. (SympoTIC '04)*, pp. 183-186, Oct. 2004.

[22] *Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines. European TeleComm. Standards Inst. (ETSI) Standard EN 102 377 Ver. 1.3.1*, May 2007.

[23] E. Balaguer, F. Fitzek, O. Olsen, and M. Gade, "Performance Evaluation of Power Saving Strategies for DVB-H Services Using Adaptive MPE-FEC Decoding," *Proc. IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC '05)*, pp. 2221-2226, Sept. 2005.

[24] Q. Zhang, F. Fitzek, and M. Katz, "Cooperative Power Saving Strategies for IP-Services Supported over DVB-H Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '07)*, pp. 4107-4111, Mar. 2007.

[25] C. Hsu and M. Hefeeda, "Bounding Switching Delay in Mobile TV Broadcast Networks," *Proc. ACM/SPIE Multimedia Computing and Networking (MMCN '09)*, pp. 72530A:1-72530A:12, Jan. 2009.

[26] C. Hsu and M. Hefeeda, "Video Broadcasting to Heterogeneous Mobile Devices," *Proc. Int'l Conf. IFIP Networking*, pp. 600-613, May 2009.

[27] C. Hsu and M. Hefeeda, "Multi-Layer Video Broadcasting with Low Channel Switching Delays," *Proc. IEEE Int'l Packet Video Workshop (PV '09)*, May 2009.

[28] G. Faria, J. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital Broadcast Services to Handheld Devices," *Proc. IEEE*, vol. 94, no. 1, pp. 194-209, Jan. 2006.

[29] K. Daoud, "Performance Comparison of the DVB-H and FLO Mobile Broadcasting Systems," *Proc. IEEE Int'l Symp. Consumer Electronics (ISCE '07)*, pp. 1-7, June 2007.

[30] M. Hefeeda, C. Hsu, and Y. Liu, "Testbed and Experiments for Mobile TV (DVB-H) Networks," *Proc. ACM Multimedia Demo Session*, Oct. 2008.

[31] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sept. 2007.

[32] "Joint Scalable Video Model Reference Software,"Joint Video Team, JSVM 14.0, May 2008.

[33] *Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television. European Telecomm. Standards Inst. (ETSI) Standard EN 300 744 Ver. 1.5.1*, June 2004.

Cheng-Hsin Hsu received the BSc and MSc degrees from National Chung Cheng University, Taiwan, in 1996 and 2000, respectively, the MEng degree from the University of Maryland, College Park, in 2003, and the PhD degree from Simon Fraser University, Canada, in 2009. He is a senior research scientist at Deutsche Telekom R&D Lab, Los Altos, California. His research interests are in the area of multimedia networking and distributed systems. He is a member of the IEEE.

Mohamed Hefeeda received the BSc and MSc degrees from Mansoura University, Egypt, in 1994 and 1997, respectively, and the PhD degree from Purdue University, West Lafayette, Indiana, in 2004. He is an associate professor in the School of Computing Science, Simon Fraser University, Surrey, British Columbia, Canada, where he leads the Network Systems Lab. His research interests include multimedia networking over wired and wireless networks, peer-to-peer systems, mobile multimedia, and Internet protocols. Dr. Hefeeda won the Best Paper Award at the IEEE Innovations 2008 conference for his paper on the hardness of optimally broadcasting multiple video streams with different bitrates. In addition to publications, he and his students have developed actual systems, such as pCache, svcAuth, pCDN, and mobile TV testbed. The mobile TV testbed software developed by his group won the Best Technical Demo Award at the ACM Multimedia 2008 conference. He serves as the Preservation Editor of the ACM Special Interest Group on Multimedia (SIGMM) web magazine. He served as the program chair of the ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2010) and as a program cochair of the International Conference on Multimedia and Expo (ICME 2011). In addition, he has served on many technical program committees of major conferences in his research areas, including ACM Multimedia, ACM Multimedia Systems, and the IEEE Conference on Network Protocols (ICNP). He is on the editorial boards of the *ACM Transactions on Multimedia Computing, Communications and Applications* (ACM TOMCCAP), the *Journal of Multimedia*, and the *International Journal of Advanced Media and Communication*. He is a senior member of the IEEE.