Video Broadcasting to Heterogeneous Mobile Devices

Cheng-Hsin Hsu and Mohamed Hefeeda

School of Computing Science, Simon Fraser University
250-13450 102nd Ave, Surrey, BC, Canada
{cha16, mhefeeda}@cs.sfu.ca

Abstract. We study the problem of broadcasting multiple scalable video streams to heterogeneous mobile devices, which have limited energy budgets. We show that scalable video streams should be broadcast in a different manner than nonscalable streams; otherwise energy of mobile devices could be wasted. We propose an efficient broadcast scheme for mobile TV networks that explicitly supports heterogeneous mobile devices, and we show its correctness as well as performance in terms of energy saving. We implement the proposed scheme in a real mobile TV testbed to evaluate its performance. Our results indicate that, with the proposed broadcast scheme, significant energy savings can be achieved by different heterogeneous devices. For example, using the proposed broadcast scheme allows mobile devices to achieve energy saving between 62% to 92%, while using the current broadcast scheme only allows them to achieve energy saving 62% despite how many layers they can (or opt to) receive and decode.

Keywords: TV Broadcast Networks, Energy Saving, Time Slicing

1 Introduction

Modern mobile devices are lightweight to be carried all the time, and provide communication and computational powers that were only available to stationary computers a few years ago. These mobile devices can run many multimedia applications including mobile TV, which allows users to watch TV programs anywhere, anytime. Mobile TV is expected to be a huge market: up to 20 billion Euros with 500 million subscribers worldwide by 2011 [1]. Mobile TV can be sent over cellular networks or dedicated broadcast networks. In this paper, we only consider dedicated broadcast networks, because they can support a very large number of subscribers using a single broadcast tower.

Mobile devices are heterogeneous from several aspects, including screen resolutions, decoder features (coding standards, spatial dimensions, and frame rates), and battery capacities. For example, GSmart t600 PDA phone is equipped with a VGA (640x480) display [2], while Nokia N96 cellular phone only has a QVGA (320x240) display [3]. Mobile TV operators who wish to simultaneously support these two mobile devices will face a dilemma: broadcasting TV channels in QVGA resolution leads to lower perceived quality on GSmart t600, while broadcasting in VGA resolution results in higher communication and computational

overhead on Nokia N96 (and thus shorter watch times) with no visible quality improvement. One way to cope with this dilemma is to encode and broadcast every TV channel into two versions: one for each device. This multi-version approach, however, does not scale because it incurs huge bandwidth overhead, thus reduces the number of TV channels that can be concurrently broadcast. Moreover, mobile devices can be categorized into classes by not only different mobile devices but also different working conditions of the same mobile device, e.g., mobile devices with low battery levels or in poor wireless channel conditions may prefer to receive lower bit rate streams to save energy and/or reduce bit error rate. Therefore, the number of classes can be quite large, which renders the multi-version approach less practical.

To eliminate the bandwidth overhead of multi-version approach, operators can adopt *scalable video coders (SVCs)* to encode each TV channel into a single stream with multiple layers, where each layer is broadcast exactly *once*. Mobile devices can then selectively receive and decode a few (or all) layers for perceived quality that are the most suitable to them. Broadcasting scalable video streams, however, poses a challenge for the base station. This is because the base station broadcasts each TV channel in bursts with a bit rate much higher than the encoding rate of that TV channel. Mobile devices can then receive a burst of traffic and turn off their radio frequency (RF) circuits until the next burst in order to save energy. This is called *time slicing*, and it is dictated in major broadcast standards such as DVB-H (Digital Video Broadcast-Handheld) [4, 5] and MediaFLO (Forward Link Only) [6]. Preparing bursts of TV channels encoded in scalable manner is much more complex than preparing these bursts for nonscalable TV channels, because of the dependency among various layers.

In this paper, we study the burst transmission problem in mobile TV networks, where several TV channels are concurrently broadcast as scalable streams over a shared air medium to many mobile devices with *heterogeneous* resources. To the best of our knowledge, there is no existing solution in the literature to efficiently broadcast scalable video streams in mobile TV networks. We formulate and solve the burst transmission problem in mobile TV networks with scalable video streams. We show the correctness of our solution. We implement the proposed solution in a mobile TV testbed and we demonstrate its practicality and efficiency. We also empirically show that the proposed solution allows mobile devices to save energy and receive only the appropriate layers of the video streams. Solving this problem enables mobile devices to obtain the most suitable resolution and frame rate without increasing energy consumption of the devices. Moreover, solving the problem allows mobile devices to trade perceived quality for energy consumption, as they can opt to receive fewer layers to prolong battery lifetime.

The rest of this paper is organized as follows. We review the previous works in Sec. 2. In Sec.3, we formally state the considered problem and discuss methods for broadcasting scalable streams in mobile TV networks. We propose the solution in Sec.4 and evaluate the solution in Sec.5 using a real mobile TV testbed. We conclude the paper in Sec. 6.

2 Related Works

Multicast of scalable video streams over the Internet has been studied in the literature and many protocols and algorithms have been proposed to support multicast routing, resource reservation, robustness, and flow and congestion controls [7,8]. None of these proposals is applicable to mobile TV networks, because they are single-hop broadcast networks, rather than the multi-hop Internet. For example, in RLM (Receiver-driven Layered Multicast) [9], different layers of a video stream are sent to different multicast groups, and receivers periodically join the next higher layer's group until experiencing excessive packet loss. RLM is not useful in mobile TV networks because of their broadcast nature: *more* receivers do not incur *higher* network loads, and packet loss ratio on a mobile device is independent of how much data it receives. Most importantly, previous works in multicasting scalable video streams do not consider energy consumption on clients, as we do in this paper.

Unequal error protection (UEP) methods were proposed to improve video quality for mobile devices with bad radio receptions in mobile TV networks [10,11]. Ghandi and Ghanbari [10] proposed to transmit the base layer and the enhancement layers with different modulation and coding schemes. This is called hierarchical modulation and channel coding, and is supported by broadcast networks like DVB-T [12]. Hellge et al. [11] proposed to use FEC bits of higher layers to protect data bits in the lower layers. The rational is that lower layers are more critical to successful decoding, and they need to be more resilient to errors. None of these works considers construction of bursts, and they are orthogonal to our work.

Previous works have studied the energy saving in mobile TV networks that broadcast TV channels in nonscalable manner. For example, it has been shown that time slicing enables mobile devices to turn off their RF circuits for a significant fraction of the time [13,14]. The works in [13,14] did not solve the burst transmission problem. Balaguer et al. [15] proposed an energy saving strategy by not receiving more FEC bytes once the data can be successfully reconstructed. Zhang et al. [16] 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. The proposals in [15,16] did not consider the burst transmission problem, and are complementary to our work.

Finally, our previous works studied the burst transmission problems and proposed time slicing schedules for mobile TV networks that broadcast nonscalable video streams to a single class of mobile devices [17–19]. That is, our previous works assumed that all mobile devices receive the entire video streams. In this paper, we solve the burst transmission problem for scalable video streams and we explicitly consider heterogeneous mobile devices that can only (or opt to) receive parts of video streams. We show that naive transmission of scalable streams could lead to wasting the energy of mobile devices.

3 Problem Statement

In this section, we formally describe the considered problem. We then show the limitations of the current mobile broadcast networks.

3.1 Burst Transmission to 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 many mobile devices with heterogeneous capability. Each TV channel is allocated a bit rate of r kbps, and is divided into C layers using scalable video coders. A TV channel is encapsulated and broadcast as a series of bursts, where each burst is in size b kb. Each mobile device receives a burst of data and turns off its RF circuit till the next burst of the same TV channel to save energy. This is called time slicing. 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 RF circuit is in off mode to the total time [13, 14]. When computing γ, we need to consider the overhead of waking up the RF circuits on mobile devices to receive the next burst [14]. This is because it takes RF circuits some time to power up and resynchronize before data can be demodulated. This period is called overhead duration T_o, which can be as high as 250 msec [4]. The problem considered in this paper can be stated as follows.

Problem 1 (Burst Transmission in Multi-Layer Broadcast Networks)
Consider a mobile TV broadcast network with air medium bandwidth R kbps shared among S TV channels, where every TV channel has a bit rate of r kbps. Each TV channel is encoded into C layers, where 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 ($c = 1, 2, \ldots, C$) receive and render all layers $\bar{c}$, where $\bar{c} \leq c$. Video streams are put into IP packets and then encapsulated into bursts of size b kb. Design a burst transmission scheme to maximize energy saving of mobile devices in all classes. The burst transmission scheme assigns IP packets to individual bursts, and specifies the start time of each burst.

Solving the above problem is critical to the quality of service in mobile TV networks, because it increases battery lifetimes for heterogeneous mobile devices. Longer battery lifetimes enable subscribers to watch more TV and provide network operators more opportunities for higher revenues due to subscription fees and advertisements.

3.2 Encapsulating and Broadcasting Scalable Video Streams

Video streams coded by traditional, nonscalable coders must be transmitted and decoded in their entirety, and thus may not be suitable to be broadcast to heterogeneous mobile devices. This is because all mobile devices will have
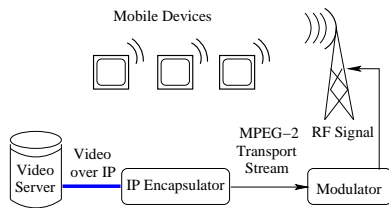
Fig. 1. The main components of mobile TV broadcast networks.
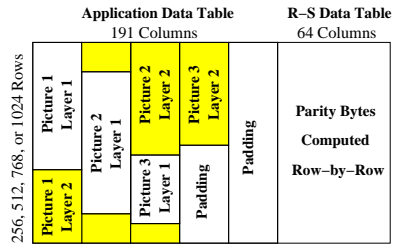
Fig. 2. The structure of MPE-FEC frame.

to receive complete video streams even though some of them may not have resources to decode and render those streams. Scalable video coders (SVC), on the other hand, can 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. Modern scalable coders support several scalability modes, including spatial, temporal, and SNR (signal-to-noise ratio) scalability. With spatial scalability, several picture resolutions can be supported. With temporal scalability, different frame rates can be supported. Finally, SNR scalability supports various picture fidelities. These scalability modes are not mutually exclusive: the combined scalability is supported by recent coders such as H.264/SVC [20, Annex. G]. Interested readers are referred [21] for more details on SVC.

While SVC is promising to many applications with heterogeneous networks and clients, it is quite challenging to broadcast SVC streams over mobile TV networks in order to maximize energy saving for heterogeneous mobile devices. This is because unlike other network applications, where SVC substreams can be extracted by MANEs (Media Aware Network Elements) based on the requests from downstream subnets, there is only one air medium in each mobile TV network. Since there exists no network devices between the base station and mobile devices, substream extractions *must* be done at mobile devices, which can result in high overhead. In the following illustrative example, we show that broadcasting SVC streams in current mobile TV networks leads to no energy saving for mobile devices that cannot render the complete video stream!

As illustrated in Fig. 1, a mobile TV base station consists of several components, including a video server, an IP encapsulator, and a modulator. The video server puts the video data in RTP packets and sends these packets to the IP encapsulator. The IP encapsulator receives and encapsulates the IP packets in MPE (multiprotocol encapsulation) frames. The IP packets can be FEC-protected using Reed-Solomon (R-S) codes, which result in MPE-FEC frames. FEC is important because it provides better error resilience to bad channel conditions that are common in mobile systems. Fig. 2 shows the structure of an MPE-FEC frame, which can be divided into two parts: an application data ta-

ble (ADT) carries IP packets and an R-S data table (RDT) carries the parity bytes. To compute the parity bytes, IP packets received by the IP encapsulator are *sequentially* stored column-by-column, from left to right. If there are not enough IP packets to fill ADT, zeros are padded in the remaining space. Once the ADT is full, the parity bytes are computed row-by-row, and store in the RDT. The whole MPE-FEC frame is then sent as a burst. We note that the zeros are padded in ADT to facilitate the generation of parity bytes. The padded zeros are *not* broadcast over the air, thus do not incur any communication overhead [22]. Existing IP encapsulators are not media-aware: they just encapsulate IP packets one after another. We refer to this type of burst transmission as *sequential transmission*.

To support heterogeneous mobile devices, network operators may upgrade the video server to support scalable video coding. IP encapsulator and the modulator can still encapsulate and send SVC streams by treating them as ordinary IP streams. Let us consider a small time window of 3 pictures, where each picture is encoded into 2 layers. Without loss of generality, we assume that each layer of each picture is put in a single IP packet, and these IP packets are sent by the video server 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). We further assume that the IP packets are not reordered in the network, so that they are stored in the same order within an MPE-FEC frame as illustrated in Fig. 2. This MPE-FEC frame is then broadcast. Consider a mobile device that can only display 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 are scattered all over the frame, and a deep inspection (at RTP or video coding layer) is required to identify them. Second, each parity byte is computed over IP packets from various layers, thus it is useless if some IP packets are not received.

This illustrative example shows that simply upgrading the video server to support SVC streams results in *no energy saving* for mobile devices, and calls for new burst transmission schemes that treat IP packets of various SVC layers differently so that mobile devices can extract SVC substreams *without* receiving and processing complete bursts. We call such burst transmission schemes as *layer-aware* schemes.

4 Solutions: Layer-Aware Burst Transmission

We propose and analyze layer-aware burst transmission schemes in this section.

4.1 Parallel Services: PS

One way to achieve layer-aware burst transmission is to send each layer of a TV channel as a parallel service (PS), which can be implemented using several IP streams sent to different multicast IP addresses, or using multiple parallel elementary streams [14, Sec. 8.6]. Fig. 3 shows an example of broadcasting two TV
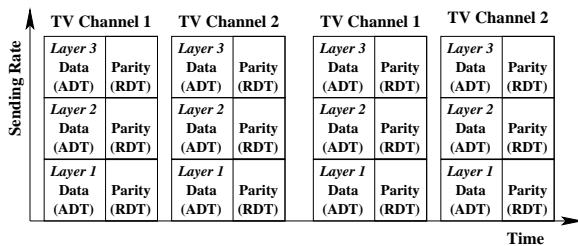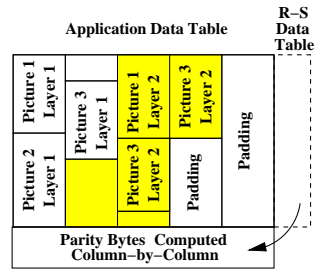
Fig. 3. Parallel Services.

Fig. 4. Layer-Aware FEC frame.

channels with three layers, where each block inside bursts is a parallel service and carries IP packets of a specific layer only. Compared to sequential transmission, parallel service approach supports efficient demultiplexing of IP packets to individual SVC layers based on either IP addresses or MPEG-2 PIDs (packet IDs). This frees mobile devices from inspecting IP packets and reduces their processing overhead on extracting substreams. However, as all services are sent in parallel, mobile devices still have to open their RF circuits for the complete burst duration. Therefore, all mobile devices achieve the same energy saving despite how many layers they receive and decode.

4.2 Layer-Aware FEC: LAF

In order to allow mobile devices that only receive a few layers to close their RF circuits earlier than each burst ends, the IP encapsulator must rearrange the received packets so that packets belonging to layer l are sent before packets belonging to layer $l+1$. If we reuse the illustrative example given in Sec. 3.2, 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. In order to allow mobile devices to efficiently determine the boundaries between layers (in this example, between (picture 3, layer 1) and (picture 1, layer 2)), we propose to prepend the SVC layer number as a one-byte extension header before the MPE section header. Mobile devices can then demultiplex the IP packets based on this extension header.

However, even after reordering IP packets, mobile devices still have to receive complete bursts in order to perform error corrections, which again prevents them from getting higher energy saving. To address this issue, we propose to compute parity bytes *column-by-column* as illustrated in Fig. 4. Furthermore, the parity bytes of each column are sent immediately after each column of the 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 RF circuits to save energy. We call this new frame format as Layer-Aware FEC (LAF) frame. Although LAF frame allows mobile devices to receive and
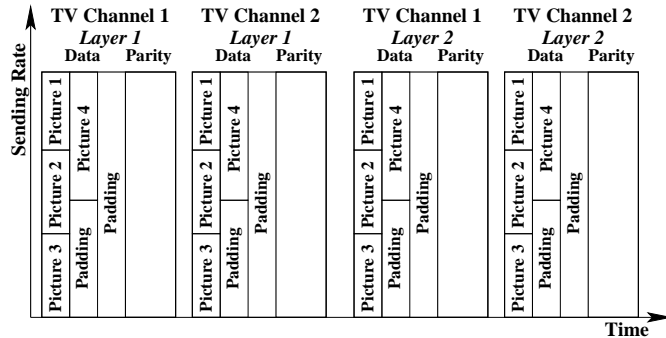
Fig. 5. Layer-Aware Time Slicing.

extract substreams while achieving proportional energy saving, it has disadvantages. First, LAF does not comply to mobile TV standards, which will cause 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, and more importantly, 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 [22].

4.3 Layer-Aware Time Slicing: LATS

Layer-aware burst transmission can also be implemented using time slicing schemes. Fig. 5 presents an illustrative example of such a time slicing scheme, which we call Layer-Aware Time Slicing (LATS). As shown in this figure, the IP encapsulator prepares a different MPE-FEC frame for each SVC layer of every TV channel: e.g., all the IP packets in the left-most burst belong to the base layer of TV channel 1, while the IP packets in the third burst belong to layer 2 of TV channel 1. Note that, as we mentioned in Sec. 3.2, the IP encapsulator adds zero padding in MPE-FEC frames only to compute parity bytes: these padded zeros are not transmitted [22]. Since each burst consists of IP packets from the same layer of the same TV channel, mobile devices know which layer those IP packets are in, even before receiving the burst. This frees mobile devices from opening the RF circuits and inspecting IP packets for substream extractions. LATS naturally works with existing MPE-FEC frame, because whenever a mobile device decides to decode layer l, it has to receive all IP packets in layer l for successful video reconstruction. Therefore, all IP packets in ADT will be received before error corrections, and the FEC decoder (implemented in hardware/firmware) can work as-is. Last, we note that 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 layer (C), which is already sent to them for decoding SVC streams.

We develop the LATS scheme in the following by giving the burst start time to each layer of individual TV channels. We first compute the number of TV channels that can be concurrently broadcast as $S = \lfloor R/r \rfloor$. LATS works in a recurring window, where each window consists of CS bursts of size b kb. Given that the radio channel bandwidth is R, the window size can be computed by $(b/R)CS$ sec. For a layer c, where $c = 1, 2, \ldots, C$, the starting time is given as $(b/R)(c-1)S$ because there are S bursts in each layer. Finally, LATS schedules a burst start at:

$$(b/R)[(c-1)S + (s-1)] \tag{1}$$

to layer c ($c = 1, 2, \ldots, C$) of TV channel s ($s = 1, 2, \ldots, S$).

Lemma 1 *The Layer-Aware Time Slicing (LATS) scheme (Eq. (1)) specifies a feasible time slicing scheme for a recurring window of $(b/R)CS$ sec, where (i) no two bursts overlap with each other, and (ii) 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 is given by:*

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

Proof. First, since sending a burst of b kb takes b/R sec to transmit, by definition of Eq. (1) the resulting time slicing scheme leads to no overlapping bursts. Second, because the recurring window size is $(b/R)CS$ and the bit rate of any layer is $r_s = r/C$, the required amount of data in any layer for smooth playout is $(b/R)CSr_s = (b/R)Sr \leq (b/R)R = b$, where the inequality comes from the definition of S. This inequality shows that the allocated time period for each burst is long enough to carry the playout data for a layer till the next burst of the same layer.

For energy saving, since mobile devices in class c receive c bursts of size b in every recurring window, the energy saving can be computed by $\gamma_c = 1 - \frac{(bc/R) + T_o c}{(b/R)CS}$. Manipulating this equation yields Eq. (2).

This lemma shows that LATS scheme is correct and allows mobile devices in different classes to receive and render at different perceived quality, while achieving proportional energy saving.

In the next lemma, we show that LAF scheme leads to lower energy savings than LATS scheme.

Lemma 2 *The Layer-Aware Time Slicing (LATS) scheme achieves higher energy saving than the Layer-Aware FEC (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. LAF works in a recurring window of S bursts of size b kb. The window time is $(b/R)S$ sec. In every window, class c devices receive a burst prefix of

length bc/C and turn off their RF circuits. Hence, we write the energy saving achieved of mobile devices in class c as:

$$\hat{\gamma}_c = 1 - \frac{(bc)/(CR) + T_o}{(b/R)S} = 1 - \frac{c}{CS} - \frac{RT_o}{bS} \quad , \text{where } c = 1, 2, \ldots, C. \qquad (3)$$

Comparing Eq. (3) against Eq. (2) yields the lemma.

Summary: Compared to sequential scheme, Parallel Service (PS) scheme only saves processing overhead, and does not lead to energy savings for heterogeneous devices. While Layer-Aware FEC (LAF) scheme achieves proportional energy savings, implementing it requires modifying broadcast protocols and could make broadcast networks more sensitive to bursty channel errors [22]. Moreover, LAF scheme results in lower energy savings than LATS as we proved in Lemma 2. Since LATS enables us to achieve the highest energy savings among all proposed schemes, we recommend LATS scheme and do not consider the other two in the rest of this paper. Last, we mention that although LATS scheme allocates each TV channel multiple (C) bursts in a recurring window (bCS/R sec), these bursts are placed apart enough for the base station to fill up them. Hence, LATS scheme does not result in under-utilized bursts.

5 Evaluation

We first briefly describe the testbed and the experimental setup. We then present the results.

5.1 Mobile TV Testbed

We have implemented a testbed in our Lab for one of the most popular mobile TV standard: DVB-H [4, 5]. The testbed provides a realistic platform for analyzing the performance of the proposed burst transmission scheme. The 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. We installed a PCI modulator [23] 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 [3] as receivers to assess the visual quality of videos. For detailed information on the signals, we add a DVB-H analyzer [24] 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 the RF signal, the MPEs, jitter, time slicing, and so on. More details on the testbed are given in [25].

5.2 Setup

We have implemented the Layer-Aware Time Slicing (LATS) scheme in the testbed. For comparison, we have also implemented the current, sequential burst
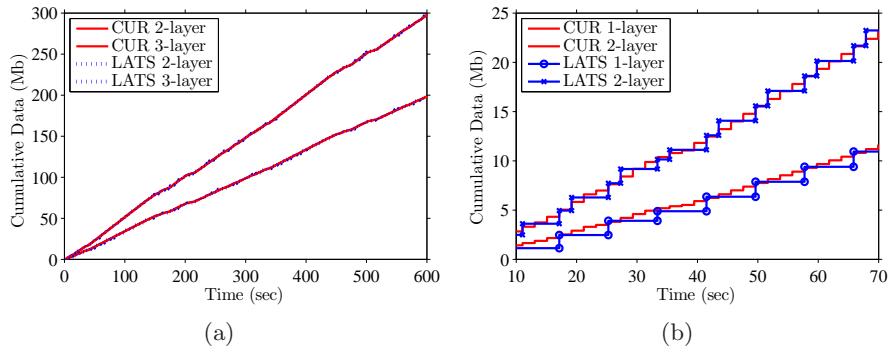
Fig. 6. Cumulative received data: (a) 10-min broadcast, and (b) zoom-in 60-sec period.

transmission scheme, which is denoted as CUR in the figures. We encode several video sequences at 768 kbps, which are then partitioned into four layers, where each layer has a bit rate of 192 kbps. We then configure the modulation card to use 8 MHz bandwidth, QPSK (quadrature phase-shift keying) modulation, 3/4 code ratio, 1/8 guard interval. This leads to channel bandwidth of 8.289 Mbps [26]. We varied the burst size b from 200 to 1600 kb to study the implications of b on the energy saving γ. We broadcast 8 TV channels for 10-min using LATS scheme, and we repeat the same test using the CUR scheme.

To gather statistically meaningful results, we have instrumented the testbed to save log files for offline analysis. The log files contain start and end times of each burst as well as its size. Moreover, the log files indicate the distribution of burst data among SVC layers. For example, a burst produced by CUR scheme contains IP packets for all layers, while a burst produced by LATS only contains IP packets for a specific layer. We have developed a script to emulate mobile devices in various classes based on the log files. This script computes the cumulative size of IP packets received by each mobile device and the achieved energy saving.

5.3 Results

While we concurrently broadcast 8 TV channels, we only present sample results for TV channel 1. Results for other TV channels are similar and are not shown for brevity.

Cumulative Data Dynamics: We plot the cumulative received data in Fig. 6 for two classes of mobile devices. Results for other classes are similar. In Fig. 6(a), we plot the cumulative received data for the complete experiment. This figure shows that CUR and LATS are both feasible, and transmit the same amount of data for mobile devices in the same class. In Fig. 6(b), we zoom into a short time period. In this figure, every staircase step represents a received burst. This figure reveals that mobile devices receive many more bursts when CUR scheme is used, which leads to higher processing overhead.
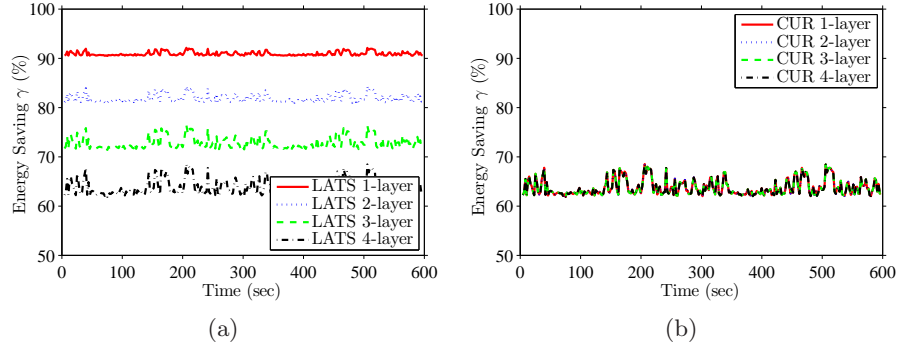
Fig. 7. Energy saving achieved by mobile devices in different classes: (a) LATS and (b) CUR.

Proportional Energy Saving: We plot the energy saving achieved by various mobile device classes. We present a sample result with $b = 400$ kb in Fig. 7. Fig. 7(a) illustrates that LATS enables mobile devices to receive a subset of layers, and achieve *proportional energy saving*. For example, mobile devices which receive all four layers achieve 65%, while mobile devices which receive the base layer achieve more than 90% energy saving. Meanwhile, Fig. 7(b) depicts that no matter how many layers they receive, mobile devices in CUR scheme achieve the same energy saving. This shows that LATS is required to support proportional energy saving for heterogeneous classes.
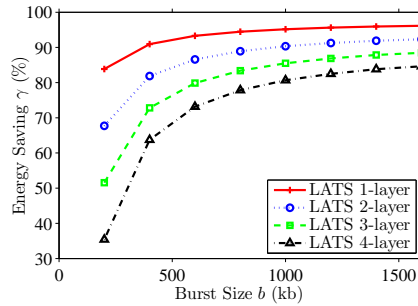
Fig. 8. Range of average energy saving achieved by different classes with various burst sizes.

Implication of Burst Size on Energy Saving: We compute the average energy saving of each mobile device class under different burst sizes. Fig. 8 shows the results. This figure reveals that larger burst sizes lead to higher energy saving. This is because larger burst sizes means fewer number of bursts, where each

burst incurs an overhead duration T_o. However, mobile devices have to reserve more memory to receive and process bursts when the burst size is large. This figure shows that $b = 1000$ kb is a sweet spot: larger burst sizes only leads to marginal increases on energy saving.

6 Conclusions

We have studied the problem of efficiently broadcasting multiple scalable video streams to heterogeneous mobile devices. We showed that network operators should broadcast scalable video streams in a different manner than nonscalable streams; otherwise the energy of mobile devices could be wasted. We have presented three broadcast schemes: PS, LAF, and LATS. They explicitly support heterogeneous mobile devices. In the PS scheme, layers of the same TV channel are multiplexed by either IP addresses or MPEG-2 PIDs into a series of bursts. In the LAF scheme, layers of the same TV channel are sequentially placed in Layer-Aware FEC frames. This allows mobile devices to receive partial bursts for desired layers and turn off their RF circuits earlier to save energy. In the LATS scheme, every layer of a TV channel forms a series of bursts. This enables mobile devices to locate layers in video streams without opening their RF circuits, such that they can receive desired layers efficiently in terms of energy saving. We proved that the LATS scheme is the most efficient broadcast scheme among the three proposed scheme. Hence, we recommend the LATS scheme.

We implemented the proposed broadcast scheme in a real testbed in our Lab for DVB-H networks. We also implemented the current, sequential broadcast scheme for comparison. Our experimental results show that, with the proposed broadcast scheme, significant energy savings can be achieved by different heterogeneous devices. For example, energy saving between 62% and 92% can be achieved by receiving different number of layers. In contrast, with the current broadcast scheme, all mobile devices achieve energy saving 62% despite how many layers they can (or opt to) receive and decode.

References

1. EU: Mobile TV across Europe: Commission endorses addition of DVB-H to EU list of official standards (2008) `http://europa.eu/rapid/pressReleasesAction.do?reference=IP/08/451&format=PDF`.
2. GSmart: GSmart website (2008) `http://www.gigabytecm.com/`.
3. Nokia: Nokia Nseries website (2008) `http://www.nseries.com/`.
4. Kornfeld, M., May, G.: DVB-H and IP Datacast – broadcast to handheld devices. IEEE Transactions on Broadcasting **53**(1) (March 2007) 161–170
5. ETSI: Digital Video Broadcasting (DVB); transmission system for handheld terminals (DVB-H). EN 302 304 Ver. 1.1.1 (November 2004)
6. Qualcomm: FLO technology overview (2008) `http://www.qualcomm.com/common/documents/brochures/tech_overview.pdf`.
7. Li, B., Liu, J.: Multirate video multicast over the Internet: An overview. IEEE Network Magazine **17**(1) (January/February 2003) 24–29

8. Ganjam, A., Zhang, H.: Internet multicast video delivery. Proc. of the IEEE **93**(1) (January 2005) 159–170
9. McCanne, S., Jacobson, V., Vetterli, M.: Receiver-driven layered multicast. In: Proc. of ACM SIGCOMM'96, Palo Alto, CA (August 1996) 117–130
10. Haddadi, H., Rio, M., Iannaccone, G., Moore, A., Mortier, R.: Layered H.264 video transmission with hierarchical QAM. Journal of Visual Communication and Image Representation **17**(2) (April 2006) 451–466
11. Hellge, C., Schierl, T., Wiegand, T.: 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) 1177–1180
12. Ladebusch, U., Liss, C.: Terrestrial DVB (DVB-T): A broadcast technology for stationary portable and mobile use. Proceedings of the IEEE **94**(1) (January 2006) 183–193
13. Yang, X., Song, Y., Owens, T., Cosmas, J., Itagaki, T.: 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) 183–186
14. ETSI: Digital Video Broadcasting (DVB); DVB-H implementation guidelines. EN 102 377 Ver. 1.3.1 (May 2007)
15. Balaguer, E., Fitzek, F., Olsen, O., Gade, M.: 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) 2221–2226
16. Zhang, Q., Fitzek, F., Katz, M.: 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) 4107–4111
17. Hsu, C., Hefeeda, M.: Bounding switching delay in mobile TV broadcast networks. In: Proc. of ACM/SPIE Multimedia Computing and Networking (MMCN'09), San Jose, CA (January 2009)
18. Hefeeda, M., Hsu, C.: Energy optimization in mobile TV broadcast networks. In: Proc. of IEEE Innovations in Information Technology (Innovations'08), Al Ain, United Arab Emirates (December 2008) 430–434
19. Hsu, C., Hefeeda, M.: Time slicing in mobile TV broadcast networks with arbitrary channel bit rates. In: Proc. of IEEE INFOCOM'09, Rio de Janeiro, Brazil (April 2009)
20. Joint Video Team: Advanced video coding for generic audiovisual services. ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC (November 2007)
21. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Transactions on Circuits and Systems for Video Technology **17**(9) (September 2007) 1103–1120
22. Faria, G., Henriksson, J., Stare, E., Talmola, P.: DVB-H: Digital broadcast services to handheld devices. Proceedings of the IEEE **94**(1) (January 2006) 194–209
23. Dektec: DTA-110T PCI modulator (2008) `http://www.dektec.com/Products/DTA-110T/`.
24. Enensys: Divi Catch RF-T/H analyzer (2008) `http://www.enensys.com/`.
25. Hefeeda, M., Hsu, C.: Design and evaluation of a testbed for mobile TV networks. Technical Report TR 2009-03, Simon Fraser University (February 2009)
26. ETSI: Digital Video Broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television. EN 300 744 Ver. 1.5.1 (June 2004)