HAL open science

Adaptive Transmission of Variable-Bit-Rate Video Streams to Mobile Devices

Farid Molazem Tabrizi, Joseph Peters, Mohamed Hefeeda

To cite this version:

Farid Molazem Tabrizi, Joseph Peters, Mohamed Hefeeda. Adaptive Transmission of Variable-Bit-Rate Video Streams to Mobile Devices. 10th IFIP Networking Conference (NETWORKING), May 2011, Valencia, Spain. pp.213-224, 10.1007/978-3-642-20798-3_16 . hal-01597971

HAL Id: hal-01597971

https://inria.hal.science/hal-01597971v1

Submitted on 29 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Adaptive Transmission of Variable-Bit-Rate Video Streams to Mobile Devices

Farid Molazem Tabrizi, Joseph Peters, and Mohamed Hefeeda

School of Computing Science, Simon Fraser University
250-13450 102nd Ave, Surrey, BC, Canada
{fma20, peters, mhefeeda}@cs.sfu.ca

Abstract. We propose a novel algorithm to efficiently transmit multiple variable-bit-rate (VBR) video streams from a base station to mobile receivers in wide-area wireless networks. The algorithm transmits video streams in bursts to save the energy of mobile devices. A key feature of the new algorithm is that it dynamically controls the buffer levels of mobile devices receiving different video streams according to the bit rate of the video stream being received by each device. Our algorithm is adaptive to the changes in the bit rates of video streams and allows the base station to transmit more video data on time to mobile receivers. We have implemented the proposed algorithm as well as two other recent algorithms in a mobile video streaming testbed. Our extensive analysis and results demonstrate that the proposed algorithm outperforms two other algorithms and it results in higher energy saving for mobile devices and fewer dropped video frames.

Key words: Mobile Video Streaming, Mobile Broadcast Networks, Energy Saving

1 Introduction

Due to advances in mobile devices such as increased computing power and screen size, the demand for mobile multimedia services has been increasing in recent years [1]. However, video streaming to mobile devices still has many challenges that need to be addressed. For example, mobile devices are small and can only be equipped with small batteries that have limited lifetimes. Thus, conserving the energy of mobile devices during streaming sessions is needed to prolong the battery lifetime and enable users to watch videos for longer periods. Another challenge for mobile video is the limited wireless bandwidth in wide-area wireless networks. The wireless bandwidth is not only limited, but it is also quite expensive. For instance, Craig Wireless System Ltd. agreed to sell one quarter of its wireless spectrum to a joint venture of Rogers Communication and Bell Canada for $80 million [2], and AT&T sold a 2.5 GHz spectrum to Clearwire Corporation in a $300 million transaction [3]. Thus, for commercially viable mobile video services, network operators should maximize the utilization of their license-based wireless spectrum bands.

In this paper, we consider the problem of multicasting multiple video streams from a wireless base station to many mobile receivers over a common wireless channel. This problem arises in wide-area wireless networks that offer multimedia content using multicast and broadcast services, such as DVB-H (Digital Video Broadcast-Handheld) [4], ATSC M/H (Advanced Television Systems Committee-Mobile/Handheld) [5], WiMAX [6], and 3G/4G cellular networks that enable the Multimedia Broadcast Multicast Services (MBMS) [7]. We propose a new algorithm to efficiently transmit the video streams from the base station to mobile receivers. The transmission of video streams is done in bursts to save the energy of mobile devices. Unlike previous algorithms in the literature, e.g. [8–12], the new algorithm adaptively controls the buffer levels of mobile devices receiving different video streams. The algorithm adjusts the buffer level at each mobile device according to the bit rate of the video stream being received by that device. The algorithm also uses variable-bit-rate (VBR) video streams, which are statistically multiplexed to increase the utilization of the expensive wireless bandwidth. By dynamically adjusting the receivers' buffers, our algorithm enables finer control of the wireless multicast channel and allows the base station to transmit more video data on time to mobile receivers.

We have implemented our new algorithm in an actual mobile video streaming testbed that fully complies with the DVB-H standard. We have also implemented the recent algorithm proposed in [8] and an algorithm used in some commercial base stations for broadcasting VBR video streams [13], which we know through private communication [14]. Our empirical results demonstrate the practicality of our new algorithm. Our results also show that our algorithm outperforms other algorithms as it delivers more video frames on-time to mobile receivers and it achieves high energy saving for mobile receivers.

The rest of this paper is organized as follows. We review related work in Section 2. In Section 3, we describe the wireless network model that we consider and we state the problem addressed in this paper. We present the proposed algorithm in Section 4. We empirically evaluate our algorithm and compare it against others in Section 5. We conclude the paper in Section 6.

2 Related work

Energy saving at mobile receivers using burst transmission (aka time slicing) has been studied in [9, 10]. Simulations are used in these studies to show that time slicing can improve energy saving for mobile receivers. However, no burst transmission algorithms are presented. Burst transmission algorithms for constant-bit-rate (CBR) video streams are proposed in [11, 15]. These algorithms cannot handle VBR video streams with fluctuating bit rates. In this paper, we consider the more general VBR video streams which can achieve better visual quality and bandwidth utilization [16]. In [12], we presented a burst transmission technique designed for scalable video streams. The algorithm in [12] adjusts the number of transmitted quality layers based on the available bandwidth. However, unlike the algorithm presented in this paper, our previous algorithm does not support

VBR streams, nor does is dynamically adjusts the receivers' buffers. In addition, the new algorithm in this paper is designed for single-layer VBR video streams, which are currently the most common in practice.

Transmitting VBR video streams over a wireless channel while avoiding buffer overflow and underflow at mobile devices is a difficult problem [17]. Rate smoothing is one approach to reducing the complexity of this problem. The smoothing algorithms in [18, 19] reduce the complexity by controlling the transmission rate of a single video stream to produce a constant bitrate stream. The minimum requirements of rate smoothing algorithms in terms of playback delay, lookahead time, and buffer size are discussed in [20]. The on-line smoothing algorithm in [21] reduces the peak bandwidth requirements for video streams. However, none of these smoothing algorithms is designed for mobile multicast/broadcast networks with limited-energy receivers. A different approach to handling VBR video streams is to employ joint rate control algorithms. For example, Rezaei et al. [22] propose an algorithm to assign bandwidth to video streams proportional to their coding complexities. This algorithm, however, requires expensive joint video encoders. A recent algorithm for transmitting VBR video streams to mobile devices without requiring joint video encoders is presented in [8]. This algorithm, called SMS, performs statistical multiplexing of video streams. SMS divides the receivers' buffers into two equal parts, one for receiving data, and the other for playing out video. The proposed algorithm in this paper outperforms SMS because it dynamically controls the wireless channel and the buffer levels of the receivers. This allows the base station to exploit the varying nature of VBR streams in order to transmit more video frames on time.

The VBR transmission algorithms deployed in practice are simple heuristics. For example, in the Nokia Mobile Broadcast Solution (MBS) [13,14], the operator determines a bit rate value for each video stream and a time interval based on which bursts are transmitted. The time interval is calculated on the basis of the size of the receiver buffers and the largest bit rate among all video streams, and this time interval is used for all video streams to avoid buffer overflow instances at the receivers. In each time interval, a burst is scheduled for each video stream based on its bit rate. In practice, it is difficult for an operator to assign bit rate values to VBR video streams to achieve good performance while avoiding buffer underflow and overflow instances at the receivers. In this paper, we compare our proposed algorithm to the SMS algorithm [8] (which represents the state-of-the-art in the literature) as well as to the algorithm used in the Nokia Mobile Broadcast Solution (which represents one of the state-of-the-art algorithms in practice).

3 System Model and Problem Statement

We study the problem of transmitting several video streams from a wireless base station to a large number of mobile receivers. We focus on multicast and broadcast services enabled in many recent wide-area wireless networks such as DVB-H [4], MediaFLO [23], WiMAX [6], and 3G/4G cellular networks that

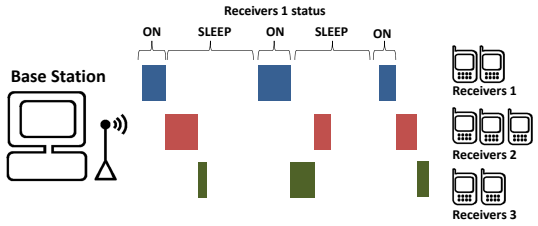


Fig. 1. The wireless network model considered in this paper.

offer Multimedia Broadcast Multicast Services (MBMS) [7]. In such networks, a portion of the wireless spectrum can be set aside to concurrently broadcast multiple video streams to many mobile receivers. Since the wireless spectrum in wide-area wireless networks is license-based and expensive, maximizing the utilization of this spectrum is important. To achieve high bandwidth utilization, we employ the variable-bit-rate (VBR) model for encoding video streams. Unlike the constant-bit-rate (CBR) model, the VBR model allows statistical multiplexing of video streams [8], and yields better perceived video quality [16]. However, the VBR model makes video transmission much more challenging than the CBR model in mobile video streaming networks [17].

Mobile receivers are typically battery powered. Thus, reducing the energy consumption of mobile receivers is essential. To save energy, we employ the burst transmission model for transmitting video streams, in which the base station transmits the data of each video stream in bursts with a bit rate higher than the encoding bit rate of the video. The burst transmission model allows a mobile device to save energy by turning off its wireless interface between the reception of two bursts [15, 17]. The arrival time of a burst is included in the header of its preceding burst. Thus, the clocks at mobile receivers do not need to be synchronized. In addition, each receiver is assumed to have a buffer to store the received data. Figure 1 shows a high-level depiction of the system model that we consider.

To achieve the burst transmission of video streams described in Figure 1, we need to create a transmission schedule which specifies for each stream the number of bursts, the size of each burst, and the start time of each burst. Note that only one burst can be transmitted on the broadcast channel at any time. The problem we address in this paper is to design an algorithm to create a transmission schedule for bursts that yields better performance than current algorithms in the literature.

In particular, we study the problem of broadcasting S VBR video streams from a base station to mobile receivers in bursts over a wireless channel of bandwidth R Kbps. The base station runs the transmission scheduling algorithm every Γ sec; we call Γ the scheduling window. The base station receives the video data belonging to video streams from streaming servers and/or reads it from local video databases. The base station aggregates video data for Γ sec. Then,

it computes for each stream s the required number of bursts. We denote the size of burst k of video stream s by b_k^s (Kb), and the transmission start time for it by f_k^s sec. The end time of the transmission for burst k of stream s is $f_k^s + b_k^s/R$ sec.

After computing the schedule, the base station will start transmitting bursts in the next scheduling window. Each burst may contain multiple video frames. We denote the size of frame i of video stream s by l_i^s (Kb). Each video frame i has a decoding deadline, which is i/F, where F is the frame rate (fps). The goals of our scheduling algorithm are: (i) maximize the number of frames delivered on time (before their decoding deadlines) for all video streams, and (ii) maximize the average energy saving for all mobile receivers. We define the average energy saving as $\gamma = \sum_{s=1}^{S} \gamma_s/S$, where γ_s is the fraction of time the wireless interfaces of the receivers of stream s are turned off.

4 Proposed Algorithm

4.1 Overview

We start by presenting an overview of the proposed algorithm and then we present the details. We have proved that our algorithm produces near-optimal energy saving and that it runs in time $O(NS)$, where S is the number of video streams and N is the total number of control points defined for the video streams, but we have omitted the proofs due to space limitations.

We propose a novel algorithm, which we call the Adaptive Data Transmission (ADT) algorithm, to solve the burst transmission problem for the VBR video streams described in Section 3. The key idea of the algorithm is to *adaptively* control the buffer levels of mobile devices receiving different video streams using dynamic control points. The buffer level at each mobile device is adjusted as a function of the bit rate of the video stream being received by that device. Since we consider VBR video streams, the bit rate of each video is changing with time according to the visual characteristics of the video. This means that the buffer level at each mobile device is also changing with time. The receiver buffer level is controlled through the sizes and timings of the bursts transmitted by the base station in each scheduling window. The sizes and timings of the transmitted bursts are computed by the proposed ADT algorithm at dynamically defined control points. Dynamic control points provide flexibility to the base station so that the algorithm has more control over the bandwidth when the bit rates of video streams increase. This results in transmitting more video data on time to mobile receivers.

The ADT algorithm defines control points at which it makes decisions about which stream should have access to the wireless medium and for how long it should have access. Control points for each video stream are determined based on a parameter α, where $0 < \alpha < 1$. This parameter is the fraction of B, the receiver's buffer, that is played out between two control points of a video stream. The parameter α can change dynamically between scheduling windows but is the

same for all video streams. At a given control point, the base station selects a stream and computes the buffer level of the receivers for the selected stream and can transmit data as long as there is no buffer overflow at the receivers. Our algorithm is designed for broadcast networks where there is no feedback channel, so we cannot know the buffer capacity of every receiver. We assume B to be the minimum buffer capacity for the receivers. Therefore, the receivers have a buffer capacity of at least B Kb and our solution still works if some of them have larger buffer capacities.

For small values of α, control points are closer to each other (in time) which results in smaller bursts. This gives the base station more flexibility when deciding which video stream should be transmitted to meet its deadline. That is, the base station has more opportunities to adapt to the changing bit rates of the different VBR video streams being transmitted. For example, the base station can quickly transmit more bursts for a video stream experiencing high bit rate in the current scheduling window and fewer bursts for another stream with low bit rate in the current scheduling window. This dynamic adaptation increases the number of video frames that meet their deadlines from the high-bit rate stream while not harming the low-bit rate stream. However, smaller bursts may result in less energy saving for the mobile receivers.Each time that a wireless interface is turned on, it incurs an energy overhead because it has to wake up shortly before the arrival of the burst to initialize its circuits and lock onto the radio frequency of the wireless channel. We denote this overhead by T_o, which is on the order of msec depending on the wireless technology.

4.2 Details

The proposed ADT algorithm is to be run by the wireless base station to schedule the transmission of S video streams to mobile receivers. The algorithm can be called periodically every scheduling window of length Γ sec, and whenever a change in the number of video streams occurs.

We define several variables that are used in the algorithm. Each video stream s is coded at F fps. We assume that mobile receivers of video streams have a buffer capacity of B Kb. We denote the size of frame i of video stream s by l_i^s (Kb). We denote the time that the data in the buffer of a receiver of stream s can be played out by d_s sec. This means that it takes d_s sec until the buffer of a receiver of video stream s is drained. We use the parameter M (Kb) to indicate the maximum size of a burst that could be scheduled for a video stream in our algorithm when there are no other limitations like buffer size. In some wireless network standards, there might be limitations on the value of M. A control point is a time when the scheduling algorithm decides which stream should be assigned a burst. A control point for video stream s is set every time the receiver has played out αB Kb of video data.

Let us assume that the algorithm is currently computing the schedule for the time window t_{start} to $t_{start} + \Gamma$ sec. The algorithm defines the variable $t_{schedule}$ and sets it equal to t_{start}. Then, the algorithm computes bursts one by one and keeps incrementing $t_{schedule}$ until it reaches the end of the current scheduling

window, i.e., $t_{start} \leq t_{schedule} \leq t_{start} + \Gamma$. For instance, if the algorithm schedules a burst of size 125 Kb on a 1 Mbps bandwidth, then the length of this burst will be 0.125 sec and the algorithm increments $t_{schedule}$ by 0.125 sec. The number of video frames of video stream s that are sent until time $t_{schedule}$ is denoted by m_s. The number of frames of stream s that are played out at the receiver side of stream s until time $t_{schedule}$ is $p_s = \min(m_s, t_{schedule} \times F)$. If $p_s < m_s$, then frames $p_s + 1$ to m_s are still in the receivers' buffers. If $p_s = m_s$, then the buffers of the receivers of video stream s are empty. In this case, the receivers of video stream s are waiting for data and if $m_s < t_{schedule} \times F$, some video frames were not received on time. We define the playout deadline for stream s as:

$$d_s = (m_s - p_s)/F. \tag{1}$$

The next control point h_s of stream s after time $t_{schedule}$ will be when the receivers of stream s have played out αB Kb of video data. We compute the number of frames g_s corresponding to this amount as follows:

$$\sum_{i=p_s}^{p_s+g_s} l_i^s \leq \alpha B < \sum_{i=p_s}^{p_s+g_s+1} l_i^s. \tag{2}$$

The control point h_s is then given by:

$$h_s = t_{schedule} + g_s/F. \tag{3}$$

The high-level pseudo-code of the ADT algorithm is given in Figure 2. The algorithm works as follows. At each control point, the algorithm finds the stream s' which has the closest deadline $d_{s'}$. Then it finds the stream s'' with the closest control point $h_{s''}$. Then the algorithm schedules a burst for stream s' until the control point $h_{s''}$ if this does not exceed the maximum burst size M or the available buffer space at the receivers of s'. Otherwise, the size of the new burst is set to the minimum of M and the available buffer space. The algorithm repeats the above steps until there are no more bursts to be transmitted from the video streams in the current scheduling window, or until $t_{schedule}$ exceeds $t_{start}+\Gamma$ and there remains data to be transmitted. The latter case means that some frames will not be transmitted. In this case, the algorithm tries to find a better schedule by increasing the number of control points to introduce more flexibility. This is done by dividing α by 2. After decreasing α, the algorithm resets $t_{schedule}$ to be t_{start} and computes a new schedule. The algorithm keeps decreasing α until it reaches a preset minimum value α_{min} or a schedule with all frames transmitted on time is found. If α is reduced in a scheduling window, then it will be gradually increased in the following scheduling windows.

5 Evaluation

5.1 Testbed and Setup

The testbed that we used to evaluate our algorithm consists of a base station, mobile receivers, and data analyzers. The base station includes an RF signal

Adaptive Data Transmission (ADT) Algorithm

```
// Input: S VBR video streams
// Output: Burst schedule to transmit S video streams
1.  compute control points and deadlines for each stream s
2.  while there is a video stream to transmit {
3.    create a new scheduling window from t_start to t_start + Γ
4.    while the current scheduling window is not complete {
5.      pick video stream s having the earliest deadline
6.      schedule a burst until the next control point or until the buffer is full
7.      update t_schedule based on the length of scheduled burst
8.      //gradually increase α if it was reduced in previous scheduling windows
9.      if α < α_max and α was not reduced in the current scheduling window
10.        update α (increase linearly)
11.     update d_s and h_s (control point) for video stream s
12.     if there is a stream s' which is late and α > α_min
13.        //go back within the scheduling window and reschedule bursts
14.        update t_schedule to t_start
15.        update α (decrease by a factor of 2)
16.     else
17.        move to next control point greater than or equal to t_schedule
18.   }
19. }
```

Fig. 2. The proposed transmission scheduling algorithm.

modulator which produces DVB-H standard-compliant signals. The signals are amplified to around 0 dB before transmission through a low-cost antenna to provide coverage to approximately 20m for cellular phones. The mobile receivers in our testbed are Nokia N96 cell phones that have DVB-H signal receivers and video players. Two DVB-H analyzers are included in the testbed system. The first one, a DiviCatch RF T/H tester [24], has a graphical interface for monitoring detailed information about the received signals, including burst schedules and burst jitters. The second analyzer, a dvbSAM [25], is used to access and analyze received data at the byte level to monitor the correctness of the received content.

We implemented our ADT algorithm, the SMS algorithm [8], and the Nokia Mobile Broadcast Solution (MBS) [13, 14], in the testbed and integrated them with the IP encapsulator of the transmitter. We set the modulator to a QPSK (Quadrature Phase Shift Keying) scheme and a 5 MHz radio channel, and the overhead T_o was set to 100 msec. We fixed the maximum receiver buffer size B to be 4 Mb (500 KB). We prepared a test set of 17 diverse VBR video streams to evaluate the algorithms. The different content of the streams (TV commercials, sports, action movies, documentaries) provided a wide range of video charac-

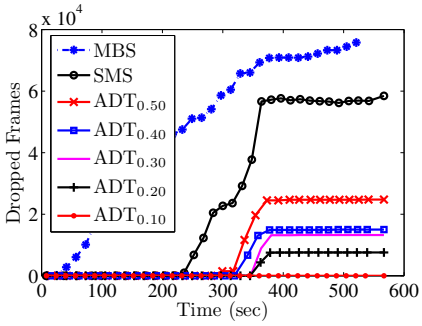


Fig. 3. Total number of dropped video frames.

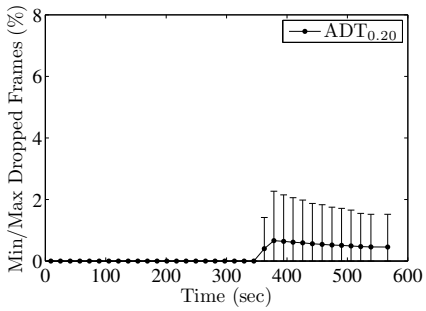
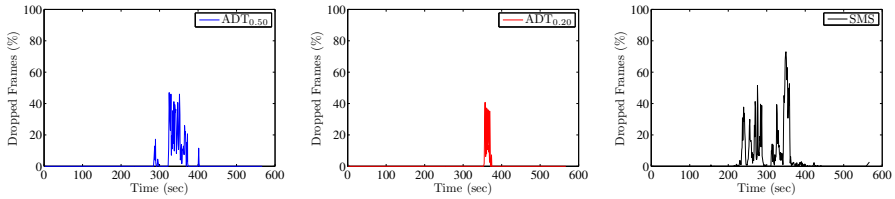


Fig. 4. Minimum and maximum number of dropped video frames.



(a) ADT with $\alpha = 0.50$ (b) ADT with $\alpha = 0.20$ (c) SMS

Fig. 5. Dropped video frames over 1 sec periods.

teristics with average bitrates ranging from 25 kbps to 750 kbps and a ratio of minimum frame size to maximum frame size of nearly 300. Each video stream was played at 30 fps and had length of 566 sec. We transmitted the 17 VBR video streams concurrently to the receivers and we collected detailed statistics from the analyzers. Each experiment was repeated for each of the three algorithms (ADT, SMS, and MBS).

5.2 Results for Dropped Video Frames

Dropped frames are frames that are received at the mobile receivers after their decoding deadlines or not received at all. The number of dropped frames is an important quality of service metric as it impacts the visual quality and smoothness of the received videos. Figure 3 shows the cumulative total over all video streams of the numbers of dropped frames for ADT with fixed values of α ranging from 0.10 to 0.50, and for SMS and MBS. The figure clearly shows that our ADT algorithm consistently drops significantly fewer frames than the SMS and MBS algorithms. The figure also shows the effect of decreasing the value of α for our ADT algorithm. The total number of dropped frames decreases from 24,742 with $\alpha = 0.50$ to 7,571 with $\alpha = 0.20$. No frames are dropped when α is reduced

to 0.10. On the other hand, the SMS algorithm is significantly worse with 58,450 dropped frames. The results for MBS in Figure 3 were obtained by running the algorithm for each video stream with different assigned bit rates ranging from 0.25 times the average bit rate to 4 times the average bit rate of the video stream and then choosing the best result for each video stream. Even in this total of best cases, the number of dropped frames is more than 75,700. In practice, an operator heuristically chooses the assigned bit rates for video streams, so the results in practice likely will be worse.

We counted the number of dropped frames for each video stream to check whether the ADT algorithm improves the quality of some video streams at the expense of others. A sample of our results is shown in Figure 4; others are similar for different values of α. The curve in the figure shows the average over all streams of the percentage of dropped frames; each point on the curve is the average percentage of frames transmitted to that point in time that were dropped. The bars show the ranges over all video streams. As shown in the figure, the difference between the maximum and minimum dropped frame percentages at the end of the transmission period is very small. Therefore, the ADT algorithm does not sacrifice the quality of service for some streams to achieve good aggregate results.

We further analyzed the patterns of dropped video frames for each algorithm by plotting the total number of dropped frames during each 1 sec interval across all video streams. Some samples of our results are shown in Figure 5. For the ADT algorithm, reducing α from 0.50 to 0.20 resulted in finer control over the bandwidth allocation and significantly fewer frames were dropped. Further reducing α to 0.10 eliminated all dropped frames as we have already seen in Figure 3. The SMS algorithm on the other hand dropped up to 72% of the frames during the period in which the aggregate bit rate of all streams is high.

5.3 Results for Energy Saving

We compute the average energy saving γ achieved across all video streams, which represents the average amount of time that the wireless interface is in off mode. This is done based on the formulation for average energy saving described in Section 3. The results are shown in Figure 6. The figure also shows the impact of changing α on the energy saving achieved by the ADT algorithm. The average over all video streams of the energy saving is 87.59% for the ADT algorithm when $\alpha = 0.10$ which is approximately the same as the SMS algorithm. Increasing α to 0.50 increases the energy saving to 93.08%. The small improvement of 5.49% in energy saving is non-trivial but it might not be large enough in many practical situations to offset the advantage of minimizing the number of dropped frames by setting $\alpha = 0.10$. Also, our experiments show that the energy saving achieved by ADT is considerably higher than MBS, which achieves average energy saving of 49%. We measured the energy saving for the receivers of each individual stream to check whether the ADT algorithm unfairly saves more energy for some receivers at the expense of others. A sample of our results is shown in Figure 7. The figure confirms that ADT does not sacrifice energy saving in some streams to achieve good average energy saving.

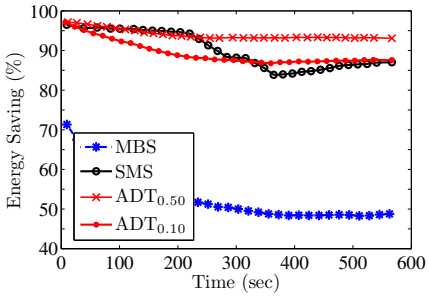


Fig. 6. Average energy saving.

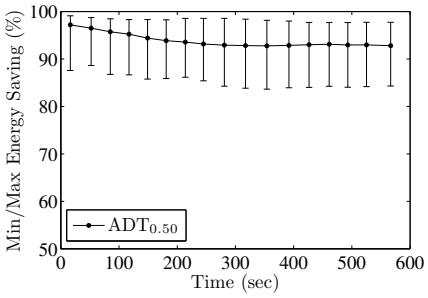


Fig. 7. Min and max energy saving.

6 Conclusions

We have presented a new algorithm for transmitting multiple VBR video streams to energy-constrained mobile devices. The algorithm is to be used by wireless base stations in wide-area wireless networks that offer multimedia services in broadcast/multicast modes such as the MBMS (Multimedia Broadcast Multicast Service) of 3G/4G cellular networks, WiMAX networks, and DVB-H (Digital Video Broadcast–Handheld) networks. One of the novel aspects of the proposed algorithm is its ability to dynamically adjust the levels of receivers' buffers according to the bit rates of the video streams being received by each receiver. We presented a proof-of-concept implementation of the proposed algorithm in a mobile video streaming testbed. We also compared the new algorithm to recent algorithms in the literature and used in practice. We conducted an extensive empirical analysis using a large number of VBR video streams with diverse visual characteristics and bit rates. Our results show that the proposed algorithm yields high energy saving for mobile receivers and reduces the number of video frames that miss their deadlines. The results also demonstrate that the proposed algorithm outperforms the current state-of-the-art algorithms.

References

1. : Global IPTV market analysis (2006–2010) `http://www.rncos.com/Report/IM063.htm`.
2. : Craig wireless to sell canadian spectrum for $80m (2010) `http://www.cbc.ca/fp/story/2010/03/26/2729450.html`.
3. : AT&T sells wireless spectrum in southeast to Clearwire corporation `http://www.att.com/gen/press-room?pid=4800&cdvn=news&newsarticleid=23428`.
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. : ATSC mobile DTV standard (2009) `http://www.openmobilevideo.com/about-mobile-dtv/standards/`.
6. : IEEE 802.16: Broadband Wireless Metropolitan Area Network (2009) `http://standards.ieee.org/getieee802/802.16.html`.

7. Parkvall, S., Englund, E., Lundevall, M., Torsner, J.: Evolving 3G mobile systems: Broadband and broadcast services in WCDMA. IEEE Communications Magazine **44**(2) (February 2006) 30–36
8. Hsu, C., Hefeeda, M.: On statistical multiplexing of variable-bit-rate video streams in mobile systems. In: Proc. of ACM Multimedia'09, Beijing, China (October 2009) 411–420
9. : Digital Video Broadcasting (DVB); DVB-H implementation guidelines. European Telecommunications Standards Institute (ETSI) Standard EN 102 377 Ver. 1.3.1 (May 2007)
10. 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
11. 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) 2231–2239
12. Tabrizi, F.M., Hsu, C.H., Hefeeda, M., Peters, J.G.: Optimal scalable video multiplexing in mobile broadcast networks. In: Proceedings of the 3rd Workshop on Mobile Video Delivery (MoViD'10), ACM (2010) 9–14
13. : Nokia mobile broadcast solution `http://press.nokia.com/PR/200510/1018770_5.html`.
14. Private communication with Nokia's engineers managing mobile TV base stations
15. Hefeeda, M., Hsu, C.: On burst transmission scheduling in mobile TV broadcast networks. IEEE/ACM Transactions on Networking **18**(2) (April 2010) 610–623
16. Lakshman, T., Ortega, A., Reibman, A.: VBR video: Tradeoffs and potentials. Proc. of the IEEE **86**(5) (May 1998) 952–973
17. Rezaei, M.: Video streaming over DVB-H. In Luo, F., ed.: Mobile Multimedia Broadcasting Standards. Springer US (November 2009) 109–131
18. Lai, H., Lee, J., Chen, L.: A monotonic-decreasing rate scheduler for variable-bit-rate video streaming. IEEE Transactions on Circuits and Systems for Video Technology **15**(2) (February 2005) 221–231
19. Lin, J., Chang, R., Ho, J., Lai, F.: FOS: A funnel-based approach for optimal online traffic smoothing of live video. IEEE Transactions on Multimedia **8**(5) (October 2006) 996–1004
20. Thiran, P., yves Le Boudec, J., Worm, F.: Network calculus applied to optimal multimedia smoothing. In: Proc. of IEEE INFOCOM'01, Anchorage, Alaska (April 2001) 1474–1483
21. Sen, S., Rexford, J., Dey, J., Kurose, J., Towsley, D.: Online smoothing of variable-bit-rate streaming video. IEEE Transactions on Multimedia **2**(1) (March 2000) 37–48
22. Rezaei, M., Bouazizi, I., Gabbouj, M.: Joint video coding and statistical multiplexing for broadcasting over DVB-H channels. IEEE Transactions on Multimedia **10**(7) (December 2008) 1455–1464
23. : FLO technology overview (2009) `http://www.mediaflo.com/news/pdf/tech_overview.pdf`.
24. : Divi Catch RF-T/H transport stream analyzer (2008) `http://www.enensys.com/`.
25. : dvbSAM DVB-H solution for analysis, monitoring, and measurement (2008) `http://www.decontis.com/`.