

Delay-Constrained Rate Adaptation for Robust Video Transmission over Home Networks

Peter van Beek
Sharp Laboratories of America
Camas, WA 98607, USA
pvanbeek@sharplabs.com

Mehmet Umut Demircin
Georgia Institute of Technology
Atlanta, GA 30332, USA
demircin@ece.gatech.edu

Abstract—In this paper, we consider rate adaptation for robust transmission of high-quality video over home networks, in particular IEEE 802.11 wireless LANs. Our approach includes on-line estimation of the time-varying bandwidth available at the application-/transport-layer. We propose a delay-constrained rate adaptation algorithm to select an optimal bit rate, subsequently enforced by a video transcoder. We examine the effectiveness of our novel delay-constrained rate adaptation algorithm, utilizing a simulation environment based on NS-2. Our simulation results show that transmission of MPEG-2 video over an IEEE 802.11b channel results in significantly better quality with rate adaptation compared to transmission without rate adaptation, and that delay-constrained rate adaptation performs significantly better than rate adaptation without a delay constraint.

I. INTRODUCTION

In this paper, we address the problem of transmission of high-quality video over home networks, in particular IEEE 802.11 wireless LANs, for display on SD and HD TVs. In this consumer application, the quality requirements are close to those of traditional broadcast television or DVD. However, the video data must be transported over an IP-based home network with highly unpredictable and time-varying throughput and delay. Also, multiple high bit rate video streams may contend for limited network resources. This application requires one-way transmission of stored and live audio and video, and may involve some interaction, such as channel changing. In this scenario, maximum end-to-end delay is required to be less than 500 ms $\sim$ 1 s. Furthermore, the video content is usually available in a single non-scalable bit stream in MPEG-2 format; therefore techniques based on scalable coding or stream switching currently are not applicable in practice.

WLANs based on IEEE 802.11 operate at several physical link rates: up to 11 Mbps for 802.11b, and up to 54 Mbps for 802.11a/g. However, actual bandwidth available for video transport is significantly lower due to protocol overhead. Furthermore, the wireless medium is prone to degradations due to path loss, fading and interference from other devices. Error correction mechanisms built into the 802.11 MAC and PHY, such as retransmissions and switching to lower link rates, reduce the inherent error rate, but effectively result in (time-varying) increases in packet delays and decreases in bandwidth available to higher protocol layers.

Maintaining the highest possible video quality and preventing interruptions to the video play-out requires real-time adaptation of the video stream, based on network conditions and content characteristics. Specifically, we investigate a rate adaptive approach, where the video bit rate is dynamically optimized at the sender [1].

The receiver includes a network monitor module, which monitors transmission performance and in particular estimates available bandwidth. We have proposed a technique for on-line estimation of available bandwidth in case of IEEE 802.11 channels in [1], based on measurement of packet arrival rates at the receiver. Estimates of available bandwidth are transmitted back to the sender using feedback packets, where a rate adaptation module selects an optimal video bit rate. A transcoder (or encoder, in case of analog video input) adapts the bit rate of the actual video bit stream given the selected target bit rate (also see Figure 1).

In this paper, we propose an algorithm for video rate adaptation, given the time-varying available network bandwidth, sender queue size, and given delivery deadlines for video data. Our system model includes an application transmission buffer and MAC transmission buffer at the sender, as well as a decoder buffer at the receiver. We first describe techniques to estimate overall expected delay for transmission of a video frame over a wireless LAN channel. We then propose an algorithm to compute or select an optimal bit rate for a frame to be transmitted, subject to the above delay constraint and expected delays. The resulting delay-constrained video rate adaptation algorithm responds to changes of current channel bandwidth and sender packet queue size, and, depending on the size of the decoder buffer, reduces the probability that video data arrives late at the receiver.

Much research has been reported recently on rate-distortion optimized packet scheduling [2]. Although more easily applied to streaming of stored video in scalable form, R-D optimized packet scheduling can be combined with our rate adaptation approach. However, our focus here is on examining the effectiveness of rate adaptation by itself.

We focus on application of robust transmission mechanisms at the application and transport layers. Cross-layer optimization for scalable video transmission over WLANs has been proposed in [3], involving interaction with the MAC and PHY layers. We believe our approach is more easily applied to home networks utilizing other transmission media, e.g. powerline-based (HomePlug), and home networks that consist of multiple links, potentially including combinations of Ethernet, wireless (802.11) and powerline.

Our overall approach to rate adaptation for wireless channels has similarities to the approach proposed in [4]. However, we utilize an explicit delay constraint, rather than a rate constraint. Also, we assume stricter separation between channel layers than is the case in [4].

The details of our rate adaptation algorithm are described in section II. Section III provides simulation results. We examine the effectiveness of our approach using a simulation testbed, providing results for the Mobile (352x288) and Crew (704x576) test video sequences. We conclude in section IV.

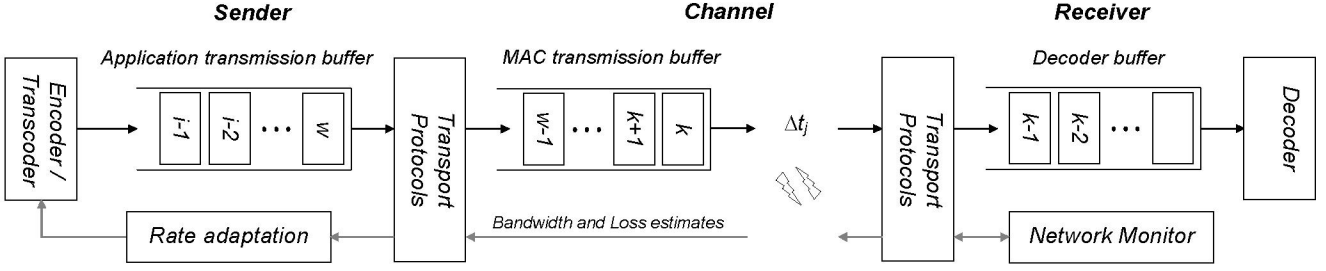


Figure 1: Overview of network-adaptive video transmission system.

II. DELAY-CONSTRAINED RATE ADAPTATION

In the following, we first describe our overall system model, then describe modeling and estimation of delays, and finally describe our rate adaptation algorithm.

We model the source as a sequence of compressed video frames, indexed by i . A single video frame is transported using multiple packets. Suppose that at time s_i (on the sender clock) the sender is about to encode (transcode) and submit frame i to the transmission buffer (assuming instantaneous encoding). Let d_i be the time difference between the instant that frame i is encoded (transcoded) and the moment it has arrived at the receiver. The delivery deadline (decoder timestamp) of frame i is denoted by $s_{DTS,i}$. We assume normal playback speed at the receiver at a constant frame rate, and let ΔT denote the frame interval. The following must hold for on-time delivery of this video frame:

$$s_i + d_i \leq s_{DTS,i} = s_{start} + \Delta T_E + i \cdot \Delta T, \quad (1)$$

where s_{start} corresponds to the time on the sender clock at which the first frame is encoded, and ΔT_E is the constant end-to-end delay between coding and decoding. The end-to-end delay ΔT_E is determined in large part by the time interval that the receiver chooses to wait before starting to decode and play out video, after it has received the first packet of video data. This time interval is specified based on the maximum tolerable delay perceived by the viewer. The receiver buffers incoming video frames to increase robustness to packet delays and delay jitter.

The state of the system at time s_i is illustrated in the diagram in Figure 1. Our system model includes an application transmission buffer and a MAC transmission buffer at the sender, together containing a queue of (packetized) video frames which have been submitted by the sender application (and transport) layer, but are still to be actually transmitted by the MAC and PHY layers. We model the sender MAC buffer explicitly, because it can contain a considerable amount of video data and has a significant impact on the performance. At time s_i , the sender is about to encode and submit frame i to the transmission buffer. The application and MAC transmission buffers contain frames $k, k+1, \dots, i-2, i-1$; where the frame in the transmission buffer that was submitted previously is frame $i-1$, and the frame with the lowest index still in the MAC buffer is frame k . We assume that video frames in the MAC buffer are not accessible to the application; furthermore, the number of packets in the MAC buffer is unknown and must be estimated. The frame with highest index that has already arrived at the decoding buffer, awaiting decoding and playout, is frame $k-1$. Note that our system currently does not use application-layer retransmissions.

Let Δt_j be the time it takes to transmit frame j over the (wireless) channel. The duration Δt_j depends on the number of bits r_j used to encode frame j and how it is packetized, in particular the number M_j and payload size P_j of packets, as well as on the (time-varying) bandwidth of the channel at the time this frame is actually being transmitted. We can now express d_i as follows:

$$d_i \approx \Delta t_k + \Delta t_{k+1} + \dots + \Delta t_{i-1} + \Delta t_i = \sum_{j=k}^i \Delta t_j.$$

As the delay depends on future, unknown, channel behavior, the channel may be modeled probabilistically.

One approach is to utilize estimates of the expected values of Δt_j and d_i . The expected duration of transmission of frame j is

$$E[\Delta t_j] = \sum_{m=1}^{M_j} \varphi(P_j, \theta_{jm}), \text{ where } \varphi(P, \theta) \text{ denotes the expected}$$

duration of transmission of a packet with payload size P over the channel, given the channel condition θ at the time this packet is transmitted. Values of $\varphi(P, \theta)$ can be computed and tabulated off-line for various discrete values of P and θ , given the assumption that the channel consists of a single link utilizing a particular transmission technology, for example an IEEE 802.11a/b/g wireless link, and given the values of a number of MAC and PHY parameters. An example of such computations in the case of IEEE 802.11a is [5].

However, we follow a more general and simplified approach here, in which knowledge of the channel condition is in the form of estimates of available bandwidth. The bandwidth is estimated at the receiver, based on measurements of Δt_j , where the data packets for a single frame are sent as a burst (see [1] for details). An estimate of the bandwidth is transmitted back to the sender after each burst of packets, using a feedback packet. Let $\hat{H}_i$ be the most recent estimate of available bandwidth at time s_i . We utilize this estimate to compute expected values of Δt_j and d_i as follows:

$$E[d_i] = \sum_{j=k}^i E[\Delta t_j] \approx \frac{1}{\hat{H}_i} \sum_{j=k}^i M_j \cdot P_j \quad (2)$$

Our rate adaptation approach is to modify the delay constraint in Eq. (1) using the expected delay value from Eq. (2), and to select the optimal values of M_i and P_i (hence the optimal rate) subject to the expected delay constraint. Note that video frames j with $j < i$ are already transcoded and will not be transcoded again.

In one case, the payload size for all packets for all frames is held constant, and we only select the optimal number of (fixed-length) packets allocated to video frame i , from a finite set of values. In practice, we select for M_i the value that best satisfies:

$$s_i + M_i \cdot \frac{P}{\hat{H}_i} + \sum_{j=k}^{i-1} M_j \cdot \frac{P}{\hat{H}_i} = s_{start} + i \cdot \Delta T + F \cdot \Delta T_E \quad (3)$$

where F is a constant between 0.0 and 1.0. F acts as a safety or sensitivity factor: the algorithm behaves conservatively (generally selecting lower values for M_i , thus lower rates) when F is small, and liberally (choosing higher values for M_i , thus higher rates) when F is large. Adapting the payload size in addition to the number of packets may be advantageous in case the channel has degraded significantly. However, lowering the payload size decreases the available bandwidth significantly in 802.11 wireless channels, since the protocol overhead for each packet is fixed and relatively large.

Note that the adaptation algorithm at the sender needs to know the value of k in Eq. (3), i.e., how many video frames and packets there are in the MAC transmission queue. One simple technique to estimate this number is by including the sequence number of the most recently received frame or packet in the feedback information from the receiver. The sender then assumes that all frames which it has submitted to the network and for which it has not received acknowledgement of receipt are still contained in the MAC transmission queue. The sender also keeps track of the number of packets used for each video frame, to derive the number of packets in the MAC queue.

III. SIMULATION RESULTS

A. Simulation Setup

We have implemented our rate adaptation algorithm, as well as bandwidth estimation and feedback techniques [1], in both a real-world client/server test-bed consisting of two laptops equipped with 802.11 cards, and a simulation environment based on the network simulator NS-2 [6]. NS-2 simulates the UDP, IP and MAC protocol layers for a single server and client. The MAC is IEEE 802.11 in DCF mode, with a retry limit of 16. We use channel traces to simulate the PHY and the medium, which were obtained from measurements of a real 11 Mbps 2.4 GHz (IEEE 802.11b) channel using our client/server test-bed. For the results included here, we obtained channel traces with 100 seconds duration using a sending and a receiving IEEE 802.11b station while generating interference with a frequency-hopping spread-spectrum cordless phone, operating in the same spectrum band as IEEE 802.11b. The available bandwidth (maximum throughput) at the application layer for such a channel is about 6.2 Mbps in ideal conditions; however, it fluctuates between 3.5 Mbps and 5.5 Mbps for the particular channel trace used in the simulations reported here, averaging about 4.5 Mbps.

The NS-2 simulator is also provided with a video trace, containing information about the average video input bit rate, the coding types (I, P or B) and number of bits for each frame, and the GOP length. For our simulations, we used the “Mobile” sequence at 352x288 pixels (CIF), and the “Crew” sequence at 704x576 (4CIF), both 30 frames per second progressive. Each sequence was looped 10 times for a total duration of 100 seconds and MPEG-2 encoded using the TM-5 reference software, with IBBP GOP pattern of length 15. Mobile was encoded at 4 Mbps, with an average luminance PSNR of 34.2 dB, while Crew was encoded at 6 Mbps, with an average PSNR of 38.6 dB.

The simulator generates a log file containing the selected bit rates at the sender, as well as information about the arrival of each packet and other statistics. This information is subsequently used to transcode the MPEG-2 bit stream according to the time-varying bit rates selected, and to apply losses to the bit stream corresponding to lost or late packets. In our simulations, we utilized a software MPEG-2 bit rate reducing transcoder, with open-loop requantizing architecture, see [1]. Losses are applied at the slice level, i.e., an entire slice is removed if it overlaps with a lost packet. Missing slices are replaced by slice data from the most recently reconstructed frame (in decoding order, *not* display order) by the decoder.

The quality of the final decoded output video is evaluated in terms of the PSNR with respect to the original (unencoded) video, and by visual comparison. In addition to computing the average PSNR over the sequence, we also compute the percentage of frames with a PSNR value lower than 20 dB. The latter is an indication of the number and duration of *glitches* (sharp drops of quality due to late or lost packets), which strongly influence the perceived visual quality.

B. Overview of Results

In our simulations, we compare the performance of three schemes. Scheme I is a non-adaptive scheme, which does not use rate adaptation, i.e., this scheme attempts to transport all video data at its original bit rate. Scheme II utilizes basic rate adaptation without knowledge of delivery deadlines, where the number of packets for a video frame is selected such that the resulting rate is adapted in proportion to the estimated available bandwidth: $(M_i \cdot P) / \Delta T \approx 0.9 \cdot \hat{H}_i$. Scheme III utilizes the proposed delay-constrained rate adaptation according to Eq. (3), for various values of the safety factor F .

Table 1 and Table 3 compare the resulting performance of these three schemes in terms of average luminance PSNR, for various values of the end-to-end delay ΔT_E , for Mobile and Crew. Table 2 and Table 4 compare these schemes in terms of the presence of visual glitches, for Mobile and Crew. The non-adaptive scheme performs very poorly as over 90% of all packets are either late or lost, even with 300 ms end-to-end delay. The simulation data shows that single packets may incur significant delay due to the retry mechanism of the 802.11 MAC (with exponential backoff), causing subsequent packets to arrive late as well. The resulting PSNR values for both Mobile and Crew are very low. The basic rate adaptation scheme performs reasonably well in the case of Mobile, but very poorly in the case of Crew. In the case of Mobile, encoded at 4 Mbps, the presence of B frames coded with very few bits provided some headroom. However, such is not the case for Crew, encoded at 6 Mbps; resulting in over 65% of packets being late or lost, as delay constraints are not taken into account. The resulting PSNR values are reasonable for Mobile, with a low number of glitches, while PSNR values are very low for Crew.

The delay-constrained rate adaptation scheme outperforms the non-adaptive scheme by a wide margin, with a much improved average PSNR and reduced number of glitches. The delay-constrained rate adaptation scheme also outperforms the basic rate adaptation scheme, by a wide margin in the case of Crew. The performance of the delay-constrained rate adaptation scheme varies with F : a small value causes lower bit rates to be selected, resulting in lower average PSNR values and a low number of glitches; while a higher value causes higher bit rates to be selected, resulting in higher

average PSNR values but also slightly increasing the number of glitches. This is illustrated for Mobile in Figure 2. For a wide range of values of F , the delay-constrained scheme outperforms the basic rate adaptation scheme in terms of both average PSNR and occurrence of glitches. For Mobile, the gain in average PSNR is about 2 to 3 dB in many cases, while for Crew, the gain in PSNR is much higher.

Also note the improvement of the PSNR (and decreased number of glitches) as the end-to-end delay is increased. We expect that the delay-constrained adaptation scheme will gradually select higher bit rates as the end-to-end delay is increased further, effectively phasing out bit rate reductions as they become unnecessary, even when the available bandwidth fluctuates. The basic rate adaptation scheme, however, is not expected to exhibit such behavior, as it adapts to bandwidth but is unaware of the delivery deadlines. Visual comparisons of the decoded sequences confirm these numerical results, and support the relevance of including a measure of glitches in our evaluation.

In [7], we report similar results for streaming video over an 802.11b link with 802.11e MAC enhancements, utilizing a different bandwidth estimation technique.

IV. CONCLUSIONS

In this paper, we have proposed an algorithm for delay-constrained bit rate adaptation for robust transmission of video over home networks. We have examined the performance of our algorithm using a network simulator, utilizing channel traces from real-world measurements of a two node IEEE 802.11b network. The simulation system includes techniques for on-line estimation of available bandwidth, utilized by the rate adaptation algorithm. Our simulation results show that transmission of MPEG-2 video over an IEEE 802.11b channel results in significantly better quality with rate adaptation compared to transmission without rate adaptation, and that delay-constrained rate adaptation performs significantly better than rate adaptation without a delay constraint.

REFERENCES

- [1] P. van Beek, S. Deshpande, H. Pan and I. Sezan, "Adaptive streaming of high-quality video over wireless LANs," *Visual Communications and Image Processing 2004 (VCIP 2004)*, Proc. SPIE vol. 5308, pp. 647-660.
- [2] P.A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Tech. Report MSR-TR-2001-35*, Microsoft Research, February 2001.
- [3] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. on Multimedia*, vol. 6, no. 2, April 2004.
- [4] C.-Y. Hsu, A. Ortega and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, May 1999.
- [5] D. Qiao, S. Choi and K. G. Shin, "Goodput analysis and link adaptation for IEEE 802.11a wireless LANs," *IEEE Trans. on Mobile Computing*, vol. 1, no. 4, Oct.-Dec. 2002.
- [6] NS-2 Network Simulator (version 2.27). URL: <http://www.isi.edu/nsnam/ns/>.
- [7] M.U. Demircin and P. van Beek, "Bandwidth estimation and robust video streaming over 802.11e wireless LANs," *IEEE Int.*

Conf. on Multimedia & Expo (ICME 2005), Amsterdam, the Netherlands, July 2005.

Table 1: Average PSNR (in dB) for scheme I (non-adaptive), scheme II (basic rate adaptation), and scheme III (delay-constrained rate adaptation), for Mobile (CIF).

ΔT_E (ms)	I	II	III $F=0.3$	III $F=0.5$	III $F=0.7$
100	6.78	27.77	27.27	29.77	29.56
200	6.98	28.44	30.63	30.77	31.55
300	6.98	28.62	30.46	32.08	31.57

Table 2: Percentage of frames with PSNR lower than 20 dB (occurrence of glitches) for scheme I, II and III, for Mobile (CIF).

ΔT_E (ms)	I	II	III $F=0.3$	III $F=0.5$	III $F=0.7$
100	95.67	5.50	1.73	2.00	3.97
200	95.23	1.63	1.13	1.47	1.47
300	95.23	0.23	0.00	1.20	2.50

Table 3: Average PSNR (in dB) for scheme I (non-adaptive), scheme II (basic rate adaptation), and scheme III (delay-constrained rate adaptation), for Crew (4CIF).

ΔT_E (ms)	I	II	III $F=0.3$	III $F=0.5$	III $F=0.7$
100	8.99	11.90	33.00	34.07	33.48
200	9.57	15.72	34.52	34.00	33.62
300	9.61	17.87	34.35	34.61	33.45

Table 4: Percentage of frames with PSNR lower than 20 dB (occurrence of glitches) for scheme I, II and III, for Crew (4CIF).

ΔT_E (ms)	I	II	III $F=0.3$	III $F=0.5$	III $F=0.7$
100	100.00	90.80	0.57	0.90	1.93
200	98.97	76.30	0.50	0.53	1.60
300	98.80	65.93	0.13	0.47	1.07

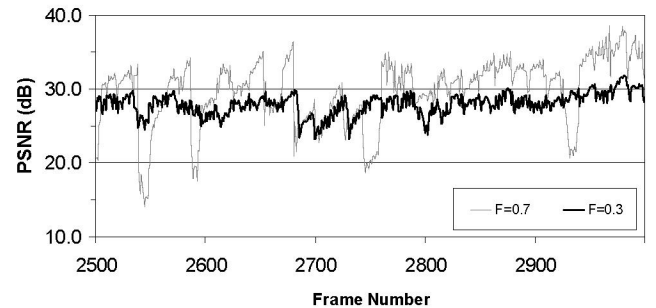


Figure 2: Traces of the luminance PSNR for the last 500 frames of the Mobile output sequence using delay-constrained rate adaptation, $\Delta T_E = 100$ ms, $F = 0.3$ and 0.7 .