COMBINING STREAM SWITCHING WITH FINE-GRAINED INTRA-STREAM ADAPTATION FOR ADAPTIVE VIDEO STREAMING

Michael Kropfberger and Hermann Hellwagner

Department of Information Technology, Klagenfurt University
Universitätsstraße 65-67, A–9020 Klagenfurt, Austria
{michael.kropfberger,hermann.hellwagner}@itec.uni-klu.ac.at

This work was partially supported by:

FWF (Fonds zur Förderung der wissenschaftlichen Forschung) P14788 and by KWF (Kärntner Wirtschaftsförderungsfonds)

ABSTRACT

Video streaming systems in best effort networks have to somehow cope with dynamically changing bandwidth. Various scalable video codecs allow *intra-stream adaptation* by use of temporal, spatial, or quality (SNR) scalability; optimizations for finer grained scalability are available as layered coding and FGS techniques. However, if there is no scalable video stream at hand, *stream switching* among pre-encoded stream versions of different bitrates and qualities allows at least coarse-grained adaptation.

Those different approaches compete to be the most efficient solution for adaptive video streaming. However, this paper will show that the efficacy is significantly increased by *combining* those approaches. As will be discussed, the combination of coarse-grained stream switching and temporal intra-stream adaptation offers better visual results and more stable client buffer behavior than the denoted approaches used separately.

1. INTRODUCTION

Most modern scalable video codecs offer support for fast intra-stream adaptation. This not only includes the possibility of temporal adaptation by dropping bi-directionally predicted B-frames [1], but also even more fine-grained adjustment of bitstreams based on, e.g., MPEG-4 Fine Granular Scalability (FGS) [2].

Those intra-stream adaptation methods can be applied to compensate fluctuations in the available network bandwidth. Still, all known adaptation methods have their limits. For instance, dropping more than 50% of all available frames will lead to very choppy presentation results. This even ignores the fact that enough B-frames have to be available, otherwise I- or P-frames will be dropped, which also leads to massive decoding errors.

These limitations in adaptation range do exist for all available adaptation methods that work on a certain stream, either by reaching the base layer (after having dropped all available enhancement data) or by resulting in intolerable quality results. If not quality is the limiting factor, then inherent coding overhead for adaptable content forces codecs to stay within reasonable adaptation ranges. E.g., MPEG-4 FGS, when coded with a very small base layer, suffers of over 2 dB PSNR loss at high bitrates (when using very much of the available enhancement layer), in comparison to the non-scalable MPEG-4 codec [2].

To overcome those limitations and offering better user experience when the available network bandwidth falls below or exceeds a certain threshold, the actual stream should be changed to another one, better fitting the given network situation. This is common practice in modern commercial video streaming systems like Real Network's RealPlayer or Microsoft's MediaPlayer in connection with the appropriate servers. [3] analyzes the usage and functionality of Real's stream switching support called *SureStream*, where the sender has multiple (statically pre-encoded) versions of the same video content available.

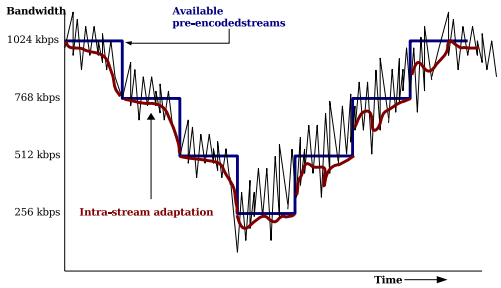

Figure 1: Combining adaptation and stream switching

Figure 1 illustrates our approach. A hypothetical changing network bandwidth curve is shown, leading to video streams being switched accordingly. Further, a more fine-grained intra-stream adaptation is applied, to react further to the inherent bandwidth fluctuations. When bandwidth falls or rises too much and intra-stream adaptation cannot cope with the fluctuation any more, a new stream has to be chosen. This approach will be discussed and evaluated in detail in the following sections.

2. ADAPTATION AND SWITCHING IN THE TEMPORAL, SPATIAL, AND SNR DOMAINS

In the following, we propose a new way of combining switching among streams with dynamic intra-stream adaptation. The *switch set* consists of different video streams with different quantization parameters or/and spatial resolutions. *Switching* among those streams is used as a coarse-grain reaction to network and client buffer problems. At the same time, fine-grain dynamic intra-stream *temporal adaptation* is applied to compensate small bandwidth fluctuations. For optimal dropping decisions, various priori-

tization algorithms, e.g., based on PSNR, are introduced and evaluated. We will show that this novel approach of combining adaptation methods substantially improves visual quality and keeps the client buffer in a more stable state than just coarse-grain stream switching.

The main challenge for video streaming systems in lossy environments is to optimize user perception. The most important rule is to never let the client buffer run out of data. To ensure this, when the buffer level gets critical, the system switches to a lower bandwidth stream, so more, yet lower-quality video data can be sent within one streamout second, which fills up the buffer again.

If the buffer is within acceptable bounds but the available network bandwidth does not reach the needed streamout bandwidth, we have to adapt the video data accordingly, to prevent a buffer underrun also in the future. Using today's available codecs, the cheapest (in terms of processing at streamout time) intra-stream adaptation method is temporal adaptation. When temporal adaptation cannot compensate a severe bandwidth reduction (the remaining frame rate is too choppy at, e.g., below 15 fps, or no more B-frames are available), stream switching has to be performed again.

On the other hand, when the buffer level is overly high, the system can switch to a higher bandwidth stream for actively draining the buffer but gaining better quality, which, under a normal buffer fill level, would not fit the given network bandwidth.

3. TEST ENVIRONMENT

The following evaluations were performed with the MPEG-4 reference video stream *Big Show One + Two*, encoded at various quantization levels and spatial resolutions using the Microsoft reference software for MPEG-4. The video has 13000 frames with one I-frame each 30 frames, using a static pattern with $P \overset{1B}{\rightarrow} P$, denoting the use of one B-frame between two reference frames (`IBPBPBP...PB`). Playout of the video was done at 25 fps, which results in a video length of 520 seconds (8:40 minutes).

Spatial resolution	Bitrate [kbps]	I-/P-quant. param.	B-quant. param.	Overall PSNR [dB]	Relative (absolute) PSNR difference
CIF	704	12	12	30.19	0 (0)
CIF	496	16	16	28.75	-1.44 (-1.44)
CIF	352	21	23	27.45	-1.30 (-2.74)
QCIF	241	12	16	22.54	-4.91 (-7.65)

Table 1: Switch set of streams with varying spatial resolutions and quantization parameters

Table 1 lists the stream variants being used (the switch set), representing the different variations available. Note that those streams where hand-chosen from hundreds of pre-encoded streams, so that the average stream bitrate is always decreased by close to 30% for each consecutive variant, where the PSNR-wise quality decreases relatively to the preceding stream, and absolutely to the highest quality stream.

It was not possible to encode a CIF version to reach a bandwidth of 241 kbps (which is 30% below 352 kbps), so some other means of reduction had to be taken. Since this work is focussing on the MPEG-4 video codec using the reference encoder, it was not an option to change the codec to, e.g., H.263 for this single stream. So spatial reduction is obviously the most useful way, when further SNR reduction fails (quantization is already at its lowest bounds). Further, having (at least one) QCIF version in the switch set will enable the server to better fulfill special requirements from low resolution clients like PDAs or cell phones.

Production of the QCIF version was performed as follows: the original CIF video was downscaled to QCIF in the uncompressed (YUV) domain using the simple and fast *nearest neighbor search* algorithm as described in [4]. After decoding at the client side, this QCIF video was upscaled again to CIF (using *nearest neighbor search*) and then compared to the original CIF version. This results in an effective relative PSNR loss of -4.91 dB to the next higher quality in CIF resolution. Please note that better scaling methods will lead to significantly better results, but those were out of scope and computational power of this work and test environment.

Similarly to the steppy bandwidth curve in Figure 1, the gradual degradation and increase of the available bandwidth was simulated with the Linux traffic shaping class *leaky bucket filter*, simulating an approx. 50% bandwidth variation range from 700 kbps down to 343 kbps. Starting at 700 kbps, the bandwidth is reduced by 30% every thirty seconds, so the next step is 490 kbps for 30 seconds, then the lowest step of 343 kbps is reached. After another 30 seconds, the bandwidth goes up again to 490 kbps, until it reaches the top level of 700 kbps. The whole process is repeated for the entire duration of the video streaming session.

In Figure 2, the spiky curve shows the measured streamout bandwidth, which coarsely follows the traffic shaped bandwidth steps. Please note that our streaming environment, which is implemented within ViTooKi, the open-source Video ToolKit [5], is capable of estimating the available bandwidth and the client buffer fill level using knowledge on sent packets and RTP NACK messages. Retransmissions were also enabled, so lost packets of important frames were sent again, which highly increases visual quality [6]. Further, since TCP-friendly [7] behavior is envisioned, bandwidth is always re-measured and then the streamout bandwidth is adjusted in an AIMD fashion (additive increase, multiplicative decrease). This leads even more to the displayed spiky streamout bandwidth and shortly delayed reaction.

4. RESULTS

4.1. Coarse-Grained Stream Switching in the Spatial and SNR Domains

In Figure 2, the bold line for stream switching[1] and the currently chosen stream is coarsely following the network bandwidth. The decision on stream switching is based on the available bandwidth and the client buffer. If the available bandwidth is very far off the necessary stream bandwidth (below 30 %), or if the available video seconds ($Vsecs$) stored in the client buffer fall below 5 video seconds, the current stream is not acceptable any more and a lower bandwidth stream has to be chosen.

In addition to normal down- and upstepping to select the stream best fitting the current network bandwidth, further switching behavior is noted. At streamout seconds ($Ssec$) 28, 264, and 440, Figure 2 shows a down-stepping to a lower quality stream version, triggered by very low client buffers. This nicely correlates with the buffer fill level shown in Figure 3, which drops below the limit of 5 video seconds.

If the client buffer fill level is overly high, this can be used to enforce the system to use a better quality video stream, even if

[1] All streams are tried to be streamed with an excess bandwidth of +15%, to compensate for bitrate variations.

network bandwidth is low. Our system is configured to switch up to a higher quality stream whenever the client buffer exceeds 25 video seconds. This is shown at $Ssecs$ 313 and 471, triggered by high client buffers shown in Figure 3.

When buffers are very low, the lowest quality stream of our switch set has to be chosen. Unfortunately, this is a QCIF stream, which has to be scaled up on the client side. This obviously gives very low PSNR results (see Figure 4), but, on the other hand, might be a perfect fit for a low-resolution client in other cases.

Anyway, the overall average PSNR loss for all 520 video seconds is -2.85 dB (in comparison to always streaming the best quality stream).

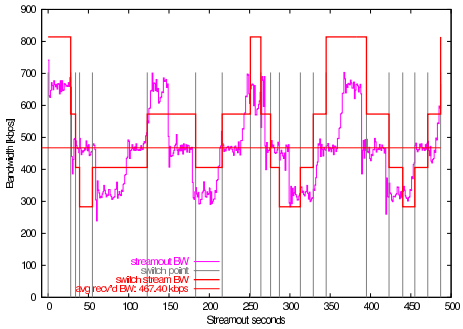

Figure 2: Network bandwidth and switching behavior (switching only)

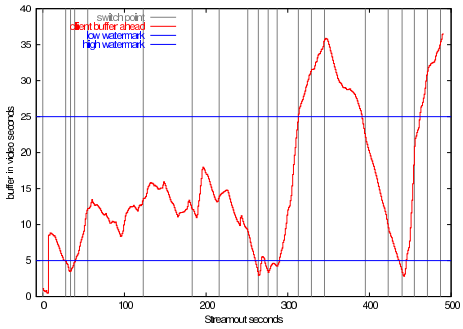

Figure 3: Client buffer fill level of available video seconds (switching only)

4.2. Adding Fine-Grained Temporal Adaptation

Instead of using a QCIF version, the chosen method for commercially available solutions like RealVideo [8] is to have low quality streams based on statically built coarse grained temporal adaptation (e.g., from 25 fps down to a 12.5 fps version). To also cover this behavior in our measurements, we simulated this by using the lowest quality $P \overset{1B}{\rightarrow} P$ CIF stream from our switch set with I- and P-quantization parameters set to 21 and a B-quantization parameter set to 23, which leads to an average bandwidth of 352 kbps. Then we removed all B-frames, which results in a framerate of 12.5 fps and an average bandwidth requirement of 232 kbps

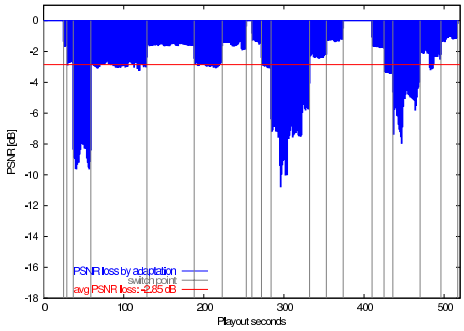

Figure 4: Qualitative reduction caused by adaptation (switching only)

(which approx. leads to the required 30% reduction). With an average PSNR value of 24.73 dB, this stream suffers a relative PSNR reduction of 2.72 dB (instead of -4.91 dB with the QCIF version) as compared to the original (higher quality) stream.

Although this low-framerate version gives better PSNR results than the QCIF upscaled version, it lacks flexibility on low-resolution clients. For this reason, this work will sustain from using coarse grained temporal versions, but will show the ability of improving switching strategies by including finer grained temporal adaptation within the actually chosen video stream. This will always give equal or better results than simple stream switching.

Further, the following examples will show that fine-grained frame dropping in some cases avoids the need to switch down to the very lowest quality stream, so it does not matter if that stream would have been QCIF or a statically frame-rate adapted version. Note, that our switch set was chosen to offer $P \overset{1B}{\rightarrow} P$ streams, so there is always the possibility of finer grained, dynamic temporal adaptation within the range of 25 to 12.5 fps.

All streaming tests with temporal adaptation where done by dropping B-frames in the compressed domain, using *timely uniform distribution* of B-frames (within a GOP) to be discarded. The $P \overset{1B}{\rightarrow} P$ GOP pattern of the streams turned out to be a good compromise. A $P \overset{4B}{\rightarrow} P$ pattern would offer more adaptation possibilities and would, especially when combined with a quality-based frame prioritization approach like QCTVA [9], also yield slightly better overall PSNR results. Still, timely uniform distribution is simpler and gives reasonable quality results with up to max. 50% frame rate reduction (e.g., from 25 fps down to 12.5 fps).

Adding dynamic temporal adaptation, Figure 5 shows the better switching behavior when the streaming environment is exposed to the exact same bandwidth curve as in the previous measurements. It was never necessary to fall back to the lowest bandwidth/quality stream. This was achieved by intermediate and dynamic temporal adaptation, where the frame rate never dropped below 18 fps. Furthermore, there were many situations where no temporal adaptation was necessary at all, so we were keeping the original frame rate of 25 fps.

Figure 6 shows that, because of temporal adaptation, the client buffer is more stable and always within safe bounds. The average PSNR loss (see Figure 7) is at -2.37 dB, so it is also better than simple stream switching shown in Figure 4.

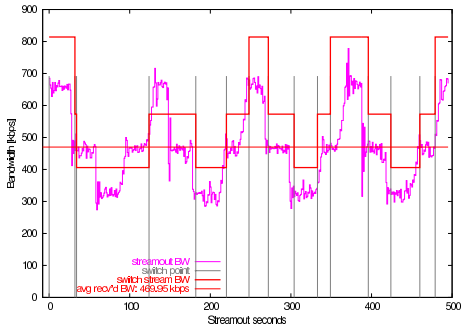

Figure 5: Network bandwidth and switching behavior (combined approach)

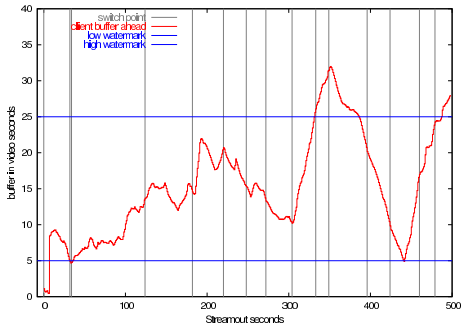

Figure 6: Client buffer fill level of available video seconds (combined approach)

5. CONCLUSION AND FUTURE WORK

Although the combination of stream switching and B-frame dropping offers better buffering behavior, the user is subjected to an ongoing fluctuation of the visual frame rate. First subjective evaluations have shown that these fluctuations are not critical, as long as variations always stay in the upper frame rate ranges (somewhere between 25 and 18 fps). Although PSNR in our experiments is severely impacted by missing frames (which does not show up in the real visual experience), the proposed combination of stream switching and temporal adaptation only results in an overall average quality loss of 2.37 dB PSNR, whereas the simple switching only scenario suffers from 2.85 dB PSNR loss. Still, this raises the question, if PSNR is a suitable measure for temporal differences in a video, so future subjective tests have to be conducted.

This work used temporal adaptation as a proof of concept, but all other (not yet widely) available intra-stream adaptation methods of various scalable video codecs (eg. MPEG-4 FGS or wavelet coding) are even more capable of stabilizing buffers and/or preventing unnecessary stream switches, because they work on even more fine grained steps.

Finally, we conclude that intra-stream adaptation using any scalable codec in connection with stream switching is a good combination, when there are enough streams available in the switch set, so the intra-stream adaptation can be performed in its optimal range (e.g., stay between 25 and 15 fps). Future work will have to investigate the impact of such a multi-facetted adaptation system with respect to server load and the number of possible parallel users. Further, evaluations with different codecs and with variable frame patterns have to be performed, to find reasonable combinations of stream switching and intra-stream adaptation under those conditions.

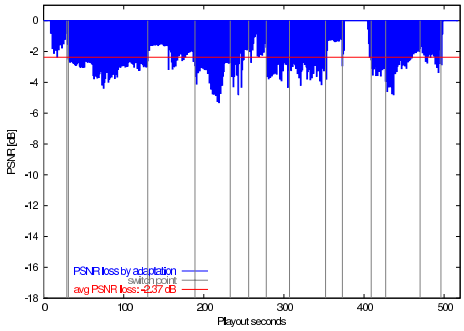

Figure 7: Qualitative reduction caused by adaptation (combined approach)

6. REFERENCES

[1] Jenq-Neng Hwang, Tzong-Der Wu, and Chia-Wen Lin, "Dynamic frame-skipping in video transcoding", Proceedings of the IEEE Workshop on Multimedia Signal Processing, pp. 616–621, December 1998.

[2] Weiping Li, "Overview of fine granularity scalability in MPEG-4 video standard", IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no. 3, March 2001.

[3] Jae Chung, Mark Claypool, and Yali Zhu, "Measurement of the congestion responsiveness of RealPlayer streaming video over UDP", Proceedings of the Packet Video Workshop, April 2003.

[4] Chuohao Yeo, "An investigation of methods for digital television format conversions," M.S. thesis, Massachusetts Institute of Technology, May 2002.

[5] Michael Kropfberger and Peter Schojer, "ViTooKi – The Video ToolKit," http://ViTooKi.sourceforge.net.

[6] Michael Kropfberger and Hermann Hellwagner, "Evaluation of RTP immediate feedback and retransmission extensions", Proceedings of ICME 2004, June 2004.

[7] Jörg Widmer, Robert Denda, and Martin Mauve, "A survey on TCP-friendly congestion control", IEEE Network, vol. 15, no. 3, pp. 28–37, 2001.

[8] Gregory Conklin, Gary Greenbaum, Karl Lillevold, Alan Lippman, and Yuriy Reznik, "Video coding for streaming media delivery on the Internet", IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 3, March 2001.

[9] Klaus Leopold, Hermann Hellwagner, and Michael Kropfberger, "QCTVA - quality controlled temporal video adaptation", Proceedings of SPIE Vol. 5242, pp. 163–174, 2003.