EVALUATION OF RTP IMMEDIATE FEEDBACK AND RETRANSMISSION EXTENSIONS

Michael Kropfberger and Hermann Hellwagner

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

ABSTRACT

Modern video streaming servers should adapt, and switch quality levels of, the streamed data according to precise and timely feedback about the network conditions, and should also incorporate selective retransmissions of important reference frames (I- and P-VOPs). This paper evaluates two recent IETF Internet Drafts on RTP extensions for immediate feedback and retransmission and shows, in conjunction with temporal video adaptation, how a substantial visual quality gain can be achieved by using those extensions (up to 4.4 dB PSNR under lossy conditions).

1. INTRODUCTION

Nowadays, many video streaming solutions like RealPlayer choose TCP for sending and receiving their content, despite many drawbacks of a connection oriented protocol. Channel congestion may lead to packet loss, when routers drop arbitrary packets. TCP retransmits them and, for each lost packet, halves the sending bandwidth. This bandwidth reduction incurs the risk of draining client-side buffers. In case of an empty buffer, the application will react by blocking the video play-out and by starting to buffer again; it can take the application several seconds to regenerate from this situation and to continue playing the video. This behavior is unacceptable for the user who expects smooth video streaming.

The solution to eliminate those drawbacks of such a rigid protocol like TCP is to use a more light-weight and therefore less powerful protocol like UDP. UDP does, however, not support automatic retransmission, has no provisions to detect and signal packet loss and, hence, cannot adjust to changing network conditions. The Real-Time Transport Protocol (RTP) [1] can be used on top of UDP to obtain the missing features. It supports (sampling) timestamps, packet sequence numbers, and rudimentary feedback support via the Real-Time Control Protocol (RTCP).

This work will discuss and evaluate more elaborate support in RTP for adaptive behavior, namely two RTP extensions proposed in recent IETF Internet Drafts: *RTCP-Based Immediate Feedback [2]* and *RTP Retransmission [3]*.

Sections 2 to 4 of this paper will give a short overview on ideas behind adaptation and switching for video streaming, our adaptive streaming environment used for the evaluation, and standard RTP with RTCP behavior. Sections 5 and 6 will discuss the two RTP extensions on immediate feedback and retransmission and present our evaluation results. We will give a conclusion and an outlook on future work in Section 7.

2. ADAPTATION AND SWITCHING FOR VIDEO STREAMING

Regardless of any protocol used, there are three obvious measures to allow seamless video streaming under varying network conditions:

- Using a long prefetching interval before displaying the first video frame, thus compensating bad network conditions by a large buffer. This is not possible if the video length is unknown a priori. The longer the video, the more data has to be prefetched to avoid buffer underrun. Furthermore, users will hardly tolerate a startup latency longer than about ten seconds.
- Seamlessly switching to different-quality video streams. This requires the availability of multiple pre-encoded streams. Too few streams will imply a coarse granularity of the bandwidth fluctuation steps that can be compensated. This will lead to underutilization of the available bandwidth and therefore reduced quality.
- Using more fine-grained adaptation to reduce the data volume of short-term video intervals (e.g., each a second worth of play-out time) to be transferred under bad network conditions. This will lead to lower-quality video in short-term intervals only and will result in better bandwidth utilization than switching to a lower-quality stream. However, this mechanism requires timely feedback about network conditions.

Obviously, minimal reduction of quality by adaptation and/or switching is still more tolerable to the user than total blocking and intermediate prebuffering of a video session.

We have designed and are implementing a video streaming system that combines coarse-grained switching with fine-grained adaptation, based on various video scaling facilities. This combination enlarges the range of possible network bandwidths that can be supported: large bandwidth fluctuation steps are captured by switching, and fine-grained temporal adaptation takes place in between those steps.

3. ADAPTIVE STREAMING TEST ENVIRONMENT

For the following experiments, we ignore stream switching because of the following assumption: harddisk constraints will force content providers to keep the number of stored streams low. So different stream versions could be prepared at every 20% quality reduction level. Since the average needed adaptation in the following measurements is always below 20%, we stay within one of those levels where we have to adapt more fine-granular.

In this paper, we are using only the *temporal* video adaptation capability of our system, since the goal of this investigation is to exercise and evaluate the extended RTP mechanisms (immediate feedback and retransmission) implemented in our streaming system, rather than the efficacy of its stream-out smoothing and switching mechanisms.

The video stream used for the evaluation is the MPEG-4 reference stream “Big Show One + Two” with 13,000 frames and a frame rate of 25 fps in CIF resolution. The average bitrate is 400 kbps, with quantization levels of 28 for B-VOPs and 16 for I- and P-VOPs. This leads to an average PSNR value of 27.8 dB. See Figure 1 for the unadapted PSNR chart for the encoded stream. To gain temporal scalability, the stream was encoded with four B-VOPs between I- and P-VOPs in each *one-second GOP*, yielding a fixed frame pattern of `IBBBBPBBBBPBBBBPBBBBPBBBB`.

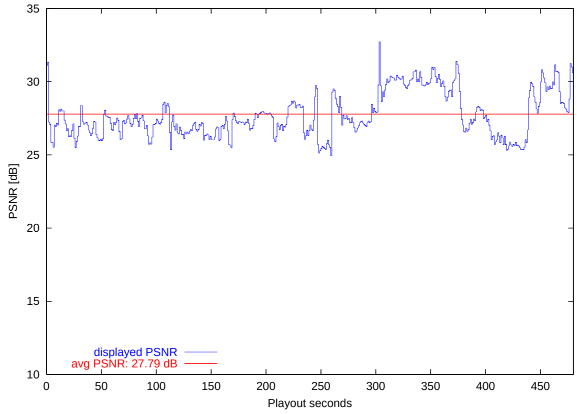

Figure 1: PSNR values for the unadapted video with 400 kbps

All of the following measurements were performed on a traffic-shaped 100 Mbps Ethernet network through an intermediary Internet Service Management Device (ISMD) from Rether Networks, Inc., which changes the available bandwidth every 30 seconds within a 20% range. Starting with 410 kbps, bandwidth is reduced to 370 kbps after 30 seconds, then degraded to a minimum of 320 kbps, which enforces 20% quality reduction compared to the original stream; subsequently, bandwidth is increased to 370 kbps and, finally, back to 410 kbps. This bandwidth fluctuation pattern is repeated within an infinite loop, yielding an average available bandwidth of about 350 kbps. The video was stored and streamed off a commodity PC under Linux and sent via the ISMD to another Linux PC using RTP/UDP with the different extensions.

Note that, without retransmission, arbitrary frames are lost, which may also include important I- and P-VOPs. This might make it impossible to decode correctly received frames. This fact is displayed in some graphs where we compare the received number of frames versus the number of decodable frames per second, which might largely diverge, if a reference frame was missing. Thus, exploiting the proposed Internet draft extensions, our streaming system can retransmit I- and P-VOPs to allow optimum decoding results at the receiver.

Adaptation and quality reduction is shown by displaying the PSNR loss in dB to the unadapted stream shown in Figure 1. So the less adaptation is done to the video, the more we converge to the zero line of the unadapted stream (see Figures 4, 7 and 9).

4. BASIC RTP FEEDBACK

If the sender has to react to changing network conditions, *feedback* from the receiver is indispensable. The standard RTCP receiver report includes the percentage of lost packets since the last RTCP receiver report and the number of totally lost and sent packets since the beginning of the session.

Those reports are sent minimally every 5 seconds and are not allowed to exceed 5% of the overall corresponding RTP session traffic. With network utilization and the amount of participating senders/receivers in an RTP session, the reporting interval will increase or RTCP receiver reports may even be turned off. Still, applications can use this rarely sent information to adjust their streaming bandwidth.

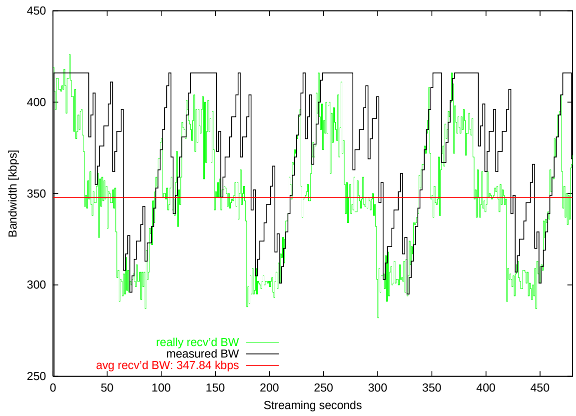

Figure 2: Bandwidth measurements with standard RTCP feedback

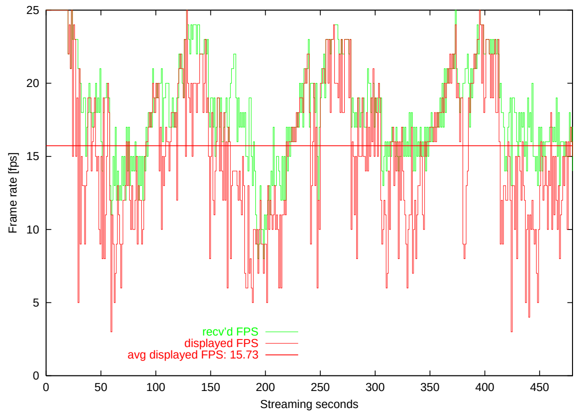

Figure 3: Frame rate adjustments with standard RTCP feedback

Fig. 2 nicely shows the steps of the traffic shaper and the late reaction on bandwidth changes due to long intervals between RTCP reports. Furthermore, all estimations on available bandwidth are too high, further aggravating packet loss. This fact and the unavailability of retransmission result in an average frame rate of 15.7 fps and adaptation rates of up to 30%. Fig. 3 shows the received frames per second, but since referencing frames like I- and

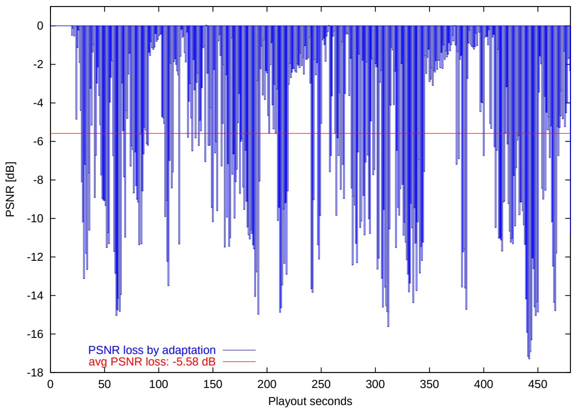

Figure 4: Quality loss with standard RTCP feedback

P-VOPs might be missing, the really displayable number of frames per second might decrease substantially. According to Fig. 4, we lose up to 16 dB PSNR quality because of lost I- and P-VOPs. The average quality reduction is about 5.6 dB.

5. RTCP-BASED FEEDBACK EXTENSION

Standard RTCP does not give any information about *which* packets were lost, just the measured loss ratio. To enable more accurate and immediate action on network problems, the immediate feedback extension to RTCP was proposed by Ott *et al.* [2]. In the best case, this allows information on loss (NACK) or receipt (ACK) of RTP packets in a round-trip time. Important data can be retransmitted and/or the original stream can be adapted to a lower bandwidth.

The draft proposes three modes of operation depending on the group size of participating hosts in an RTP setup:

1. The *immediate feedback mode* is used when the group size is small enough so that every receiving party has enough bandwidth to immediately send all RTCP feedback packets.
2. In the *early RTCP mode*, the group size or other parameters do not allow receivers to react on each event that would be worth (or is required) to be reported, but they are allowed to send RTCP packets before their regularly scheduled RTCP interval.
3. For a very large group size, it is no longer useful to provide feedback from individual receivers at all. Here, *normal rules for RTCP* intervals and packaging apply.

The draft not only introduces different possible sending modes of RTCP packets, but also an additional RTCP packet extension to cover more detailed feedback on single entities like packets or video frames. Feedback messages are classified as follows:

- *Transport layer feedback messages* for general purpose feedback information. These messages are based on packets and RTP sequence numbers, so they are independent from the particular codec or application.
- *Payload-specific feedback messages* are highly dependent on the used payload type; they are codec specific.
- *Application layer feedback messages* are totally handled by the application and are not further specified in the draft.

The draft defines the packet formats for NACK and ACK, which are based on the generic RTCP packet format. It further defines extensions to SDP within RTSP, so that the capability of extended feedback can be signalled by all participating servers and clients.

For this evaluation, we only use the *immediate feedback mode* in a unicast scenario with one client and one server. The feedback type employed is the simple *transport layer feedback*.

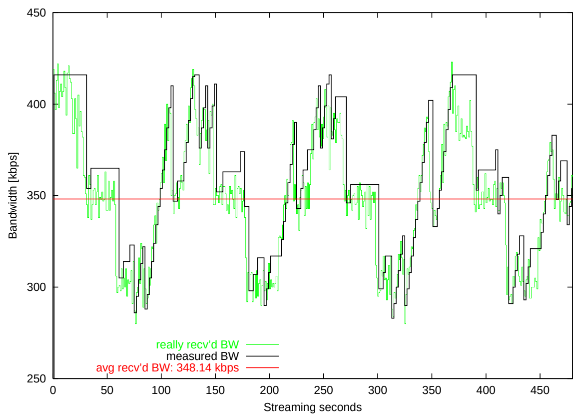

Figure 5: Bandwidth measurements with extended RTCP feedback

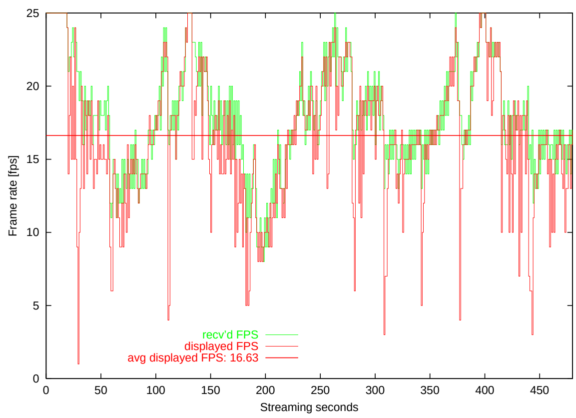

Figure 6: Frame rate adjustments with extended RTCP feedback

Fig. 5 again shows the steps of the traffic shaper and the better reaction on bandwidth changes because of the shortened RTCP reporting intervals. Also, the bandwidth fluctuations are better met. Still, the unavailability of retransmission leads to an average frame rate of 16.6 fps (see Fig. 6). According to Fig. 7, we lose up to 14 dB PSNR quality because of lost I- and P-VOPs. The average quality reduction is about 3.6 dB. Hence, the extended RTCP feedback brings about a gain of 2 dB under the same network conditions.

6. RTP RETRANSMISSION

The previously introduced RTCP feedback only informs about lost packets, but it does not specify how a server should react on this packet loss. Retransmissions are defined in another RTP extension called the *RTP retransmission payload format [3]*.

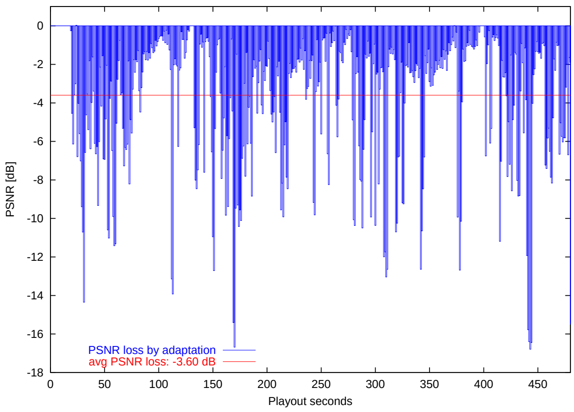

Figure 7: Quality loss with extended RTCP feedback

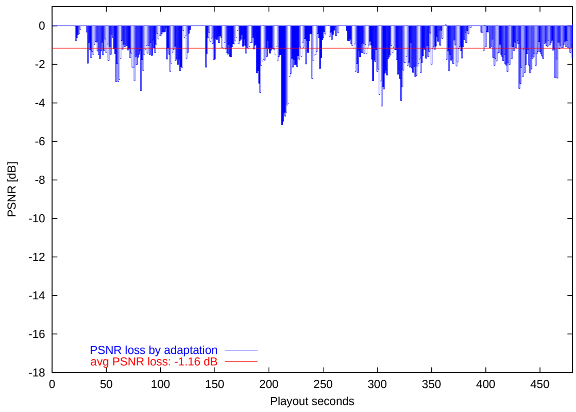

Figure 9: Quality loss after decoded retransmissions

This new scheme fulfills the following requirements:

- It does not break general RTP and RTCP mechanisms.
- It is suitable for unicast and small multicast groups.
- It works with mixers and translators.
- It works with all known payload types.
- It allows the use of multiple payload types within a session.
- Sequence number preservation is guaranteed.

Every retransmitted packet has to store its old RTP sequence number, so it can easily be re-inserted into the right place in the received data stream. Original and retransmission packets are sent in two separate streams. Thereby, the retransmitted packets are not in the same sequence number space as the normal data packets, so all packets can be distinguished and RTCP statistics are working properly.

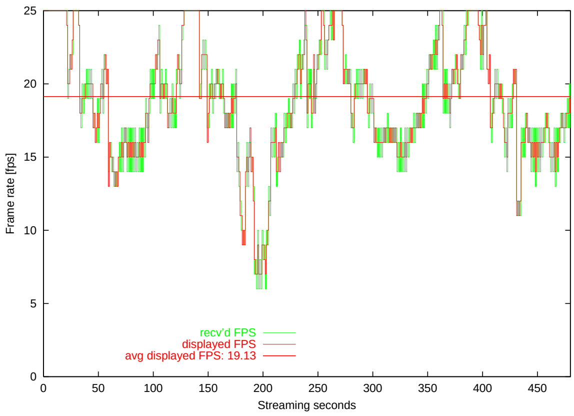

Figure 8: Frame rate adjustments with retransmissions

The measured bandwidth steps are identical to those in Fig. 5, but since we use retransmission on all packets (as long as they arrive in time), we obtain a higher frame rate of 19.1 fps (see Fig. 8). According to Fig. 9, we only lose up to 5 dB PSNR quality since we retransmit all lost I- and P-VOPs. The average quality reduction is less than 1.2 dB. Eventually, under the same network conditions, we achieve an average quality increase of 2.4 dB just by retransmission and an average quality increase of 4.4 dB as compared to the standard RTCP and RTP, without extensions.

7. CONCLUSION AND FUTURE WORK

The two proposed IETF drafts on extended RTP feedback and retransmission have proven to offer substantial benefit for unicast streaming environments over best effort networks employing IP. Further research and evaluation has to be done in the area of multicast scenarios. Also, all benchmarks have to be repeated in real networks like the global Internet.

Mainly, we have to evaluate the efficacy of feedback and retransmission, as compared to other quality-ensuring measures like forward error correction (FEC) or adding redundancy to packets, so that lost packets can be (partially) regenerated [4, 5]. Still, advantages of packet retransmission will always be the low complexity on the receiver side and the bandwidth efficiency in a nearly error-free network.

8. REFERENCES

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RFC 1889: RTP: a transport protocol for real-time applications,” January 1996.

[2] Joerg Ott, Stephan Wenger, Noriyuki Sato, Carsten Burmeister, and Jose Rey, “Extended RTP profile for RTCP-based feedback (RTP/AVPF),” draft-ietf-avt-rtcp-feedback-04.txt, October 2002, expires April 2003.

[3] Jose Rey, David Leon, Akihiro Miyazaki, Viktor Varsa, and Rolf Hakenberg, “RTP retransmission payload format,” draft-ietf-avt-rtp-retransmission-04.txt, December 2002, expires May 2003.

[4] Christian Leicher, “Hierarchical encoding of MPEG sequences using priority encoding transmission (PET),” Tech. Rep. TR-94-058, International Computer Science Institute, Berkeley, CA, November 1994.

[5] Andres Albanese and Giancarlo Fortino, “Robust transmission of MPEG video streams over lossy packet-switching networks by using PET,” Tech. Rep. TR-99-014, International Computer Science Institute, Berkeley, CA, June 1999.