High Quality IP Video Streaming with Adaptive Packet Marking

Sebastian Zander, Georg Carle

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
{zander, carle}@fokus.fhg.de
http://www.fokus.fhg.de/glone

Abstract. The transmission of high quality video streams over IP networks becomes more and more attractive for providing IP based TV or video on demand services. Still a reasonable high bandwidth is required for achieving high quality streaming. Instead of using hard reservations to guaranty the quality of the video transmissions we present an adaptive streaming algorithm based on adaptive packet marking which provides soft guarantees. A prototype streaming server and client have been developed supporting RTP-encapsulated MPEG-2 transmission, into which the adaptive streaming algorithm has been integrated. The software has been successfully used in a testbed, demonstrating that the algorithm is effective and has very promising properties.

1 Introduction

IP-based transmission of high quality video streams becomes more and more attractive especially for applications like video on demand, remote teaching, surveillance and Internet TV. Despite the wide distribution of MPEG-4 and its impressive compression capabilities still a reasonably high bandwidth is required for achieving a high video quality. With MPEG-4 a bandwidth of about 1-2 Mbit/s is needed for good quality. A higher quality can be achieved with MPEG-2 encoded video at a bandwidth of approximately 8 Mbit/s, as is used for DVD encoding. This is a relatively high data rate compared to typical access and wide-area data rates. In the case of network congestion, the video playback will either have dropouts if UDP is used as transport protocol, or the video will freeze when the playout buffer is empty if TCP is used. Such a quality degradation is not acceptable in many cases.

To guarantee the transmission quality, resource reservation such as IntServ [RFC2205] or DiffServ [RFC2475] can be used. These solutions are typically used in combination of a hard reservation of the maximum bandwidth of the flow to be transported. In the case of high bandwidth requirements such as MPEG-2 streaming with 8 Mbit/s, using resource reservation typically will be a costly alternative to a low price best effort service.

This paper describes an approach for providing high quality video streaming which uses expensive guaranteed resources only when necessary to maintain quality. Cheap bandwidth (i.e. a best effort service) is used as much as possible. The adaptive streaming algorithm presented in this paper is based on adaptive packet marking. It operates with two service classes. Best effort bandwidth is used for video streaming as much as possible. If there is not enough bandwidth available (which is detected by low fill level of the receiver playout buffer), the algorithm uses additional bandwidth from a guaranteed service based on the currently available best effort bandwidth and the receivers playout buffer fill level. The paper is focused on applications with a fairly large end-to-end delay budget such as video on demand, which allow for a playout buffer delay of a few seconds to compensate for the bandwidth fluctuations in the network.

We decided to implement and try our algorithm in a real DiffServ [RFC2475] enabled test network. We implemented a prototype streaming server and client software which is able to stream DVD quality MPEG-2 videos, as there is now free open source software available for providing high quality video streaming. The adaptive streaming algorithm has been integrated into this prototype.

The remaining parts of the paper are structured as follows. Section 2 discusses related work. Section 3 introduces the algorithm used for adaptive video streaming. Section 4 outlines the prototype implementation which adaptively streams MPEG-2 videos in a DiffServ enabled network. Section 5 presents evaluation results obtained by using the prototype and section 6 concludes the paper and outlines future work.

2 Related Work

The approach to use a mixture of best effort and guaranteed service for video streams was postulated in [RaTh98]. Besides the basic idea, this paper only describes a general network architecture supporting layered video but contains no solution for realizing adaptive streaming. [FeKS98] examines adaptive priority marking for providing soft bandwidth guarantees. Their proposed algorithm is similar to the one proposed in this paper. The main difference is that [FeKS98] proposes a general algorithm for use with TCP while our algorithm is specific for video transmission considering application feedback. [FeKS98] is focused on the optimal marking and overload behavior. In contrast, the question we want to answer is the achievable ratio of best effort and guaranteed packets. In [FeKS98], simulation is used to prove that the generic adaptive packet marking algorithm works. Our goal is to demonstrate that the algorithm works in a real network in combination with a real streaming application, and that the algorithm reacts fast enough on typical changes of network conditions.

A related but better understood problem is the smoothing of variable bit rate video streams. [SZKT96] and [ReTo99] describe possible solutions for smoothing the video stream to be transmitted by calculating a bandwidth plan before the video is sent over the network. [RSFT97] contains a proposal for a smoothing technique applicable for live streams.

[VFJF99] provides a good overview about the existing techniques for adaptive video streaming. It describes layered encoding, adaptive forward error correction and smoothing. [RNKA97] describes an adaptive video streaming service which adapts to the quality of the transmission. An algorithm chooses the video adaptation which achieves the best quality under the current conditions. No paper known to the authors describes an algorithm like the one proposed in this paper.

Our prototype software is based on protocols standardized by the IETF such as the Real Time Streaming Protocol (RTSP) [RFC2236] for controlling the playback and the Real Time Protocol (RTP) [RFC1889] for transporting the real-time data. For the transport of MPEG-2 over RTP our work is based on the MPEG-2 standard and the MPEG-2 RTP encoding as defined in [RFC1890], [RFC2250] and [RFC2343].

3 Adaptive Video Streaming

For guaranteeing the quality of a video transmission a simple solution is to reserve network resources according to the maximum video bandwidth for the full duration of the video transmission. Such a hard reservation will guarantee the quality of the video transmission but is much more expensive than using a best effort service. If the video stream is of variable bit rate some of the reserved resources will not be used but typically must be paid for. We assume that costs of using the guaranteed service class will depend on the maximum bandwidth of the guaranteed service class that can be used, and on the data volume sent using the guaranteed service class. By using a smoothing scheme, the maximum data rate which has to be reserved can be reduced, thereby reducing costs.

In existing streaming approaches that use a guaranteed service class, the complete video is transmitted over the guaranteed service (1. in Figure 1). When using a best effort service where the available bandwidth is lower than the required bandwidth for certain time intervals, buffer underuns occur at the receiver, impairing the visual quality (2. in Figure 1).

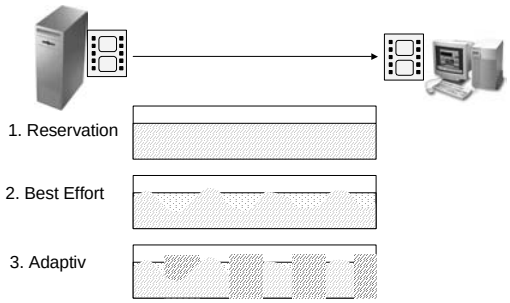

Figure 1: Adaptive Streaming

During the times when the best effort service class provides insufficient bandwidth, the required additional bandwidth is sent over the guaranteed service class (3. in Figure 1). If there is not enough guaranteed bandwidth available the video can not be

transmitted without using some quality degradation scheme but this is considered out of scope of this paper. The main properties of our adaptive streaming are:

- As much data as possible is sent using cheap best effort traffic.
- When the available bandwidth of the best effort service is insufficient (i.e., low fill level of the playout buffer) the missing bandwidth is sent using a guaranteed channel.
- As long as sufficient guaranteed bandwidth is available (i.e. no blocking happens when using reserved bandwidth), no quality degradation becomes visible, and the video has full DVD-like quality.

The algorithm we developed for realizing the adaptive video streaming consists of two parts: rate control and adaptive marking. The sender controls both the sender rate and the adaptive marking based on the video which is streamed and the feedback information from the receiver.

Sender Rate Control

The sender estimates the data rate it has to send over time based on the timestamps contained in the video file. In the simplest case the send rate s is constant (see Figure 2) which is the case for DVD encoded MPEG-2. In case the send rate is not constant we use linear regression to estimate the gradient s. Videos with non constant bitrate could also be smoothed before transmission ([SZKT96]). Figure 2 illustrates the algorithm. At time T the current position in the video is L. Since L is below $s \cdot T$ we are below the average video rate. $\triangle T$ is the time until the rate control function is called again and is estimated during runtime. The amount of data to send within $\triangle T$ is $\triangle L$ and can be calculated in a straightforward manner.

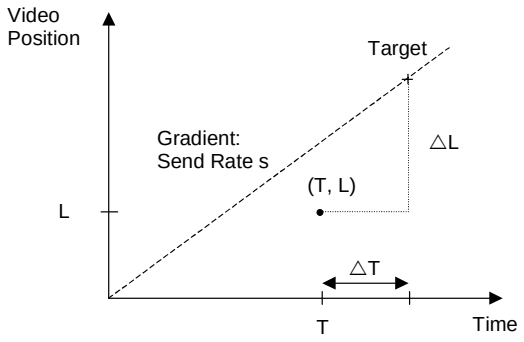

Figure 2: Sender Timing Algorithm

In a real network the current possible sending rate may be larger or smaller than s depending on the congestion. To compensate for the bandwidth fluctuation, a playout buffer is used at the receiver. At the beginning of a video transmission the playout buffer is filled to level P_0. This level can be adjusted by selecting a maximum startup delay. Afterwards the receiver starts the playback. The current buffer level is P and P_T

is the desired target fill level during playback. The deviation from the target buffer fill level is $\triangle P=P_T-P$. This value is send from the receiver to the sender in regular intervals $\triangle T_F$. The rate for the additional data to send is $s_A=\triangle P/\triangle T_F$. Figure 3 shows the sender algorithm in meta code. The amount of data to send S is divided into n packets of equal size. The size of the packets should be near but below the Maximum Transfer Unit (MTU) of the path between sender and receiver to avoid fragmentation.

```
SEND(T,L)
      S0 = s·(T+△T)-L
      SA= sA·△T
      S=S0+SA
      READ_AND_SEND(S)
```

Figure 3: Sender Rate Control Algorithm

Adaptive Marking

We assume the availability of two service classes: guaranteed and best effort. The best effort service class should be used as much as possible but in case the receiver runs out of data we transmit data over the guaranteed service. A packet will be marked with probability p as guaranteed class. Therefore p is the ratio of guaranteed packets while $1-p$ is the ratio of best effort packets. We set $p=min(0, S_A)/S_0, 0\leq p\leq 1$ at each interval $\triangle T_F$. The rationale for this is that S_0 is the number of bytes in an ideal network where we can send with constant rate s and S_A gives us an indication whether our real rate is smaller or larger. Using the ratio of S_A and S_0 means that we increase the percentage of guaranteed packets with increasing non negative S_A which means the receiver buffer level is decreasing below the target fill rate.

Note that in case there is not enough bandwidth available in the guaranteed class a frame based marking scheme would lead to much better performance at the receiver. However in the opposite case the probabilistic scheme is much easier to implement, needs less CPU time and allows a smoother partitioning.

4 Implementation

Instead of simulation we decided to implement and try the algorithm in a real network. We have developed a prototype implementation of a MPEG-2 video server and client in which we integrated the adaptive streaming algorithm introduced in section 3. Our implementation is based on standard protocols such as RTSP [RFC2326] and RTP [RFC1889]. It supports the basic RTSP capabilities and MPEG-2 program and transport stream encoding over RTP. Both UDP and TCP can be selected as transport protocols. The adaptive streaming algorithm works only with TCP. For sending RTP packets over TCP connections we send the length of each RTP packet followed by the

packet as proposed in [UDPT]. For feedback information we use the RTCP protocol [RFC1889]. Since RTCP as defined in the standard has non-deterministic and fairly long feedback intervals we have implemented a fast feedback mechanism which allows to have regular feedback in the order of 100 ms. The receiver sends the standard RTCP feedback messages, and additional feedback messages about the playout buffer state. The adaptive streaming algorithm which is built into the server uses this information. If TCP is used as transport protocol there are two separate connections: one for RTP and one for RTCP.

For adaptive streaming the sender has to mark packets as either best effort or guaranteed service. To be generic the software implemented by us has no direct interface to some QoS functionality. Instead we use the RTP marker bit (m-bit) which is included in every packet header to signal the service class. Obviously this only works with two classes. If more than two classes would be needed a custom RTP header extension could be defined. The m-bit is examined at the edge router by a NetFilter [NetFilter] classifier module running in the Linux kernel which has been implemented by us. We use the DiffServ [RFC2475] model for providing QoS. Based on the RTP m-bit the DiffServ Code Point (DSCP) will be set for each of the packets. The DSCP is later used for scheduling at the core routers. The classifier is optimized so that all retransmitted TCP packets are marked as guaranteed class. The software has been implemented in C/C++ and is running under the Linux operation system. Figure 4 shows a screenshot with the video and the statistics generated (see section 5).

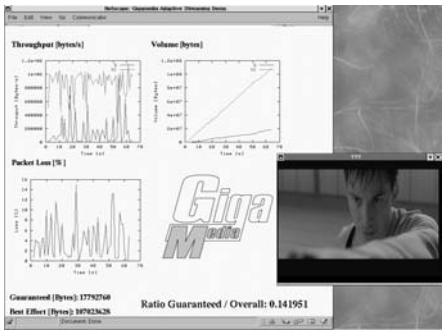

Figure 4: Adaptive Streaming Implementation

Since the Internet is a highly heterogeneous network it is questionable whether all routers will ever support QoS. Our application uses TCP as transport protocol and relies on DiffServ enabled router's for marking and scheduling. If no end-to-end QoS is supported there is no negative impact on the network because a normal best effort TCP connection is used. In our solution there is only one TCP connection for the video data which is multiplexed on two service classes. The advantage is that considering the number of connections it is not different to traditional applications and therefore it does not behave more aggressive as usual. However it needs to be investigated how this influences TCP congestion control. As shown in [FeKS98], pure marking results in a percentage of marked packets that is too high considering the fair share.

Our algorithm prevents bursts of guaranteed packets by incrementally increasing and decreasing the percentage to be marked as suggested in [FeKS98].

5 Evaluation

The implementation has been tested in a DiffServ setup in our testbed. All machines in the testbed are Linux PCs equipped with Fast Ethernet network adapters. The server runs the MPEG-2 video server and streams MPEG-2 videos encoded with 8 Mbit/s bit rate. The client runs the prototype video client and has a MPEG-2 hardware decoder card. The routers run Linux 2.4 and are DiffServ enabled. One router is configured as edge router and marks the packets with a DSCP according to the RTP m-bit. The second router is a DiffServ core router performing the scheduling. Two classes are configured: Expedited Forwarding (EF) as guaranteed class, and a best effort class. In our setup we do not use background traffic. Instead we emulate congestion by randomly dropping packets in the best effort class. Our loss emulation can drop packets with uniform or exponential loss distribution. We only use exponential loss in the evaluation. To measure the distribution of the data onto the two classes we use a separate meter connected to the same hub as the server. Figure 5 shows the testbed setup.

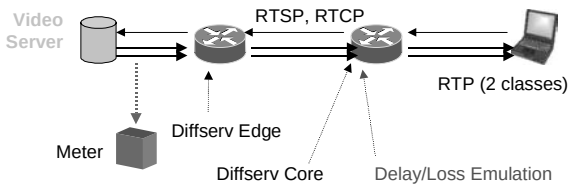

Figure 5: Adaptive Streaming Testbed

We show the results of three different tests. In the first test we prove that the implementation works as desired and show the behavior of the adaptive streaming algorithm. In the second test we show how the ratio of guaranteed class to overall volume behaves in case of different loss rates within the network. In the third test we investigate the impact of the receiver feedback frequency on the guaranteed/overall ratio.

Algorithm Behavior

In this trial an 8 Mbit/s video with a duration of approximately 4.5 minutes is transmitted over TCP from the sender to the receiver using the adaptive streaming algorithm. The receiver has a playout buffer of size 8 Mbyte and sends a feedback message every second. The mean loss was set to 3% and the maximum loss to 6% (exponential distribution). Figure 6 shows the throughput over time for both service classes (G = Guaranteed, BE = Best Effort) and Figure 7 shows the absolute volume transmitted. The ratio of guaranteed to overall volume is only 0.36. This means that only 36% of the traffic was sent using the guaranteed service class instead of 100% in case of a

hard reservation. In this trial all video frames arrived at the receiver in time; no errors occurred during the playback.

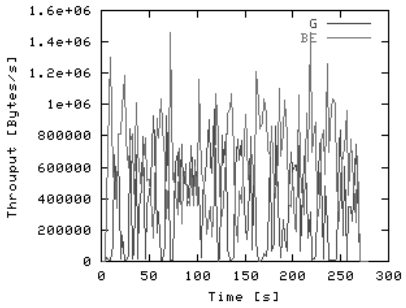

Figure 6: Throughput

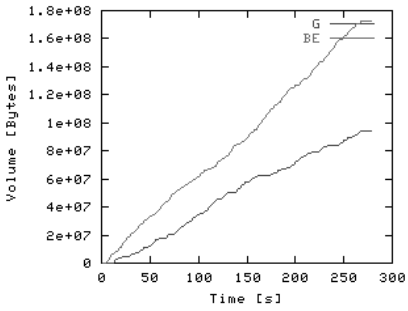

Figure 7: Volume

Impact of Loss Rate

In this test the same video as before is transmitted while emulating different loss rates (congestion) in the network. The receiver buffer size is 8 Mbyte and the receiver feedback is sent every second. We investigated the algorithm with mean loss rates of 0%, 2%, 3%, 5%, 10% and 15%. The maximum loss rate is always twice the mean loss rate, and exponential distribution is used. Figure 8 shows the ratio of guaranteed to overall volume depending on the loss rate. For each loss rate we conducted three tests. In all tests all video frames arrived at the receiver in time; no playback errors occurred. However at 15% mean loss rate the receiver buffer size had to be increased to 12 Mbyte to avoid loosing data. To prevent data loss the buffer size must be increased with increasing mean loss rate but we have not performed investigations for loss rate higher than 15%.

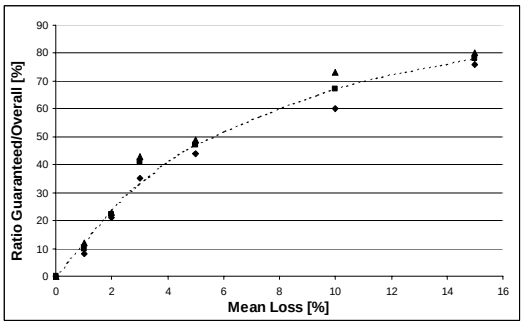

Figure 8: Impact of Mean Loss Rate

The figure shows that for small loss rates the gain of using adaptive streaming is quite high (only 20% needs to be transmitted over guaranteed service) while for large loss rates (>10%) the ratio is still over 70% but moving asymptotically to 100%.

Impact of Receiver Feedback Frequency

In this test we examined how the feedback frequency affects the performance of the algorithm. We transmitted the same video as before, the receiver buffer size is 8 Mbyte and the mean loss rate is 3% (maximum 6%). We vary the feedback frequency between 0.2 and 20 (0.05 s to 5 s interval). Figure 9 shows the ratio of guaranteed to overall volume depending on the receiver feedback frequency. For each frequency two independent tests have been performed.

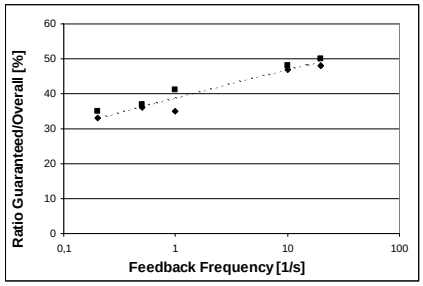

Figure 9: Impact of Feedback Frequency

The figure shows that the ratio is increasing with increasing feedback frequency (plotted with log scale). This means that the algorithm looses performance in case of frequent feedback because it is reacting too fast. We found that the smallest frequency which can be used is the playout time of half of the receiver buffer size.

6 Conclusions and Future Work

We have developed a mechanism for optimized delivery of high quality video streams over an IP based network. The algorithm uses adaptive packet marking for guaranteeing soft QoS while trying to minimize the use of an expensive guaranteed network service. The algorithm has been integrated into a real MPEG-2 streaming implementation. The evaluation in a DiffServ testbed shows the efficiency of the proposed algorithm. For mean loss rates up to 10% a substantial amount of bandwidth can be obtained from a best effort service.

In the future we plan to perform additional measurements, and to improve the algorithm. In particular we want examine how aggressive our algorithm behaves in the presence of TCP background traffic and investigate on how much guaranteed bandwidth is needed to satisfy a certain number of clients to be able to create admission control rules. Furthermore we want to make the algorithm to be aware of the MPEG-2 frame structure, in order to improve the mapping of data onto the different service classes and to allow an open loop solution without feedback messages from the receiver. Another goal is to better smooth the usage of guaranteed bandwidth by estimating the network conditions of future time intervals based on measurements of past time intervals.

7 Acknowledgements

This work was partially funded by the German Research Network (DFN Verein) project “Gigamedia” (see http://www.fokus.fhg.de/glone/gigamedia/).

The authors would like to thank Alexander Pikovski for his contribution to the implementation of the adaptive video streaming application described in this paper and the anonymous reviewers for their valuable comments.

References

[FeKS98] W. Feng, Dilip D. Kandlur, D. Saha, K. Shin: "Adaptive Packet Marking for Providing Differentiated Services in the Internet", Proc. of Int. Conf. on Network Protocols, October 1998.

[NetFilter] http://netfilter.samba.org/

[RaTh98] R. Ramanujan, K. Thurber: "An Active Network Based Design for a QoS Adaptive Video Multicast Service", SCI'98, 1998.

[ReTo99] J. Rexford, D. Towsley: "Smoothing Variable-Bit-Rate Video in an Internetwork", IEEE/ACM, 1999.

[RFC 2205] Braden, B., Ed., et. al., "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, September 1997.

[RFC1889] H. Schulzrinne, S.Casner, R. Frederick, V. Jacobson: „RTP: A Transport Protocol for Real-Time Applications“, RFC1889, Januar 1996.

[RFC1890] H. Schulzrinne: „RTP Profile for Audio and Video Conferences with Minimal Control“, RFC1890, Januar 1996.

[RFC2250] D. Hoffman, G. Fernando, V. Goyal, M. Civanlar: „RTP Payload Format for MPEG1/MPEG2 Video“, RFC2250, Januar 1998.

[RFC2326] H. Schulzrinne, A. Rao, R. Lanphier: „Real Time Streaming Protocol (RTSP)”, RFC2326, April 1998.

[RFC2343] M. Civanlar, G. Cash, B. Haskell: „RTP Payload Format for Bundled MPEG“, RFC2343, Mai 1998.

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss: "An Architecture for Differentiated Service", RFC 2475, December 1998.

[RNKA97] R. Ramanujan, J. Newhouse, M. Kaddoura, A. Ahamad, E. Chartier, K. Thurber: "Adaptive Streaming of MPEG Video over IP networks", IEEE LCN'97, November 1997.

[RSFT97] J. Rexford, S. Den, J. Dey, W. Feng, J. Kurose, J. Stankovic, D. Towsley: "Online Smoothing of live variable-bit-rate video" in Proc. Network and OS Support for Digital Audio and Video, pp. 249-257, May 1997.

[SZKT96] J. Salehi, Z. Zhang, J. Kurose, D. Towsley: "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing", ACM SIGMETRICS, 1996

[UDPT] http://www.cs.columbia.edu/~lennox/udptunnel/

[VFJF99] B. Vandalore, W. Feng, R. Jain, S. Fahmy: "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia", Journal of Real Time Systems, April 1999.