Predictive Video Scaling – Adapting Source Coding to Early Network Congestion Indicators

Fabian Jäger, Thomas C. Schmidt
fabian.jaeger@haw-hamburg.de, t.schmidt@ieee.org
iNET Research Group – Department Informatik
Hamburg University of Applied Sciences
Berliner Tor 7, 20099 Hamburg, Germany

Matthias Wählisch
waehlisch@ieee.org
Institut für Informatik
Freie Universität Berlin
Takustr. 9, 14195 Berlin, Germany

***Abstract*—A major trend in current Internet communication augments voice conversation with video. Video conferencing over IP (VCoIP) has rapidly spread in the mobile realm, where it faces the problem of heterogeneous, fluctuating network conditions. Scalable video coding enables bandwidth adaptation, but requires guidance by appropriate resource estimators.**

This work focuses on the analysis and design of an adaptive, bandwidth-aware transmission strategy for real-time multimedia applications like video conferencing. We present an early indicator of network congestion based on jitter variation along with our implementation of a new lightweight sender-based approach to adapt the video codec.

I. Introduction

Modern video applications like conferencing are capable to produce high-quality video streams that can be scaled individually on a per packet base [1], [2]. Scalability is of particular importance when mobile participants join in [3]. Wireless links introduce heterogeneous conditions that often change quickly over time. While packet loss can be proactively compensated by enhanced coding techniques or transport layer repair, the early detection of network congestion is difficult for IP-based applications that do not interact with lower layers directly.

Internet connectivity may suddenly degrade and no longer offer enough bandwidth to deliver a video stream in the foreseen quality. Congestion may drastically reduce the QoE for the participants, when the sender does not react promptly by adjusting bandwidth demands. To ensure undisturbed video flows to all participants, a scalable video codec needs triggers for early adjustment and is then able to scale the visual quality for each participant individually.

In an end-to-end communication environment, the adaptation of the video codec happens at the sender and therefore requires a sender-based approximation of the available bandwidth. In contrast to the tardy common approaches of bandwidths estimation, we analyze an early trigger based on the variation of the jitter. Rising jitter in correlation with a positive delay drift serve as the first available sign for an impending network unbalance. A fast reaction on this early congestion indicator is shown to reduce unwanted network load early enough to fit the requirements of real-time multimedia applications.

The remainder of this paper is structured as follows. We first introduce the problem of scalable video adaptation along with related work in Section II. Section III discusses indicators of network congestion that guide our video adaptation outlined in Section IV. Experimental results that test our scheme are presented in Section V. Finally, we give a conclusion and an outlook.

II. Problem Statement & Related Work

In multimedia applications with multiple participants, we often need to scale the video for each participant individually. This holds in particular for mobile regimes [4]. Scaling can be done with the SVC (Scalable Video Codec) extension to H.264 [1], [2]. Most importantly, the video transmission rate must be adapted to the available bandwidth to avoid congestion of links. However, it is not easy to determine the effective bandwidth of a path, especially when the available bandwidth is not constant. Common approaches mostly use PGM (Probe Gap Model) or PRM (Probe Rate Model) to measure the available bandwidth, which requires extra probing packets [5]. The accuracy depends on the amount of probing packets, the nature of side traffic and duration of the measurement. Overall, these techniques are intrusive and rather slow.

Video transmission itself is a major cause of traffic load and quickly fills buffers and queues along a congested network path. Once accumulated, in-network buffering will boost packet delays and may stall audio-visual dialogs completely. It is therefore important to detect over-utilization of links as early as possible and to quickly reduce the data rate of the video. The price of imposing a controlled decrease in video quality is compensated by the gain of a continuous visual flow.

III. Network Congestion Indicators

It is neither easy nor fast to determine the effective bandwidth on a path. Instead, we try to predict a congestion on a link and to avoid it [6]. This is done at the sender side, since we want to scale the transmission rate of encoded video to meet available resources.

In regular network communication, packets traverse a path in a mean round trip time (RTT) and experience some delay variation (jitter) that is typically of the same order of magnitude. The RTT values mainly depend on the topology,

and a high RTT does not necessarily mean that the link is congested. On the occasion of exhausting on-path links, buffers start to fill and the RTT rises. However, at a first sight it is difficult to separate congestions-bound fillings from regular fluctuations in RTT that are characterized by the jitter. In fact, an abnormal condition causes RTT values to leave the regular jitter tolerances. This goes along with a jitter enhancement, indicated by a jump in its second derivative, the jitter variation. Such sudden discontinuities occur whenever the RTT heads off for a change in behaviour, i.e., an abnormal rise or fall. Thus an early congestion indicator is present, if the jitter variation jumps discontinuously and the RTT is increasing. Congestions start to resolve, if the delay shows a negative drift.

Collecting RTT and jitter at the sender site requires a feedback loop. This is commonly available through RTCP or TCP states, when HTTP progressive download is used. In our implementation, we use a light-weight reliable UDP extension called enet [7] for the video transmission. This protocol was developed for real-time applications and provides a reliable and in-order delivery of packets without head-of-line blocking. The enet protocol also gathers information about the network performance like the RTT, jitter, and provides information about internal protocol states like the reliable and unreliable packet buffer queue size. To do so, the enet protocol adds timestamps to the video stream, which allows us to measure the RTT in band without additional probing packets. This reduces the overhead and expedites the availability of results. Following this set-up, the requirements for monitoring real-time multimedia applications are met, since the jitter observation is a very lightweight and fast approach to determine the conditions on a link.

IV. Video Adaptation

SVC use multiple temporal, spatial and quality layers, that allow to reduce or increase the quality of an encoded video stream. In contrast to common codecs, SVC always encodes the video with the highest quality and scales the video stream for each participant individually by adding or removing enhancement layer on a packet base. However, external information are needed to steer the scaling. With jitter-based prediction, we are capable to detect a congested link and are able to adapt the video stream.

In our implementation, we use the DAVC codec [6] by Daviko [8] that is able to add temporal enhancement layers to the video stream in an H.264-compliant way. We also use the quantization factor of the codec to scale the quality of the video stream. With these two components, we implemented a fine-granular video adaptation. For testing purpose, we use three temporal layers for a rough video scaling, while the quantization is used for a more fine granular adaptation. In this scenario, we do not consider optimization of scaling from an QoE perspective. Instead, we focus on the bitrate of the video stream with the aim to exactly match the available bandwidth. A scaling that also provides a good QoE needs further research in the future.

In our approach, we do not know the available bandwidth on a link. This complicates scaling, since we cannot set the codec to a certain bit-rate. Instead, we have to adapt the video stream to runtime conditions that is measured by the jitter behaviour as previously discussed.

Jitter Ratio	Quality decrease
1.0 - 1.2	5%
1.2 - 1.4	5%
1.4 - 1.6	10%
1.6 - 1.8	10%
1.8 - 2.0	15%
2.0 - 2.5	15%
2.5 - ∞	25%

TABLE I
Codec quality adaptation with respect to the Jitter Ratio

The goal of congestion control is to determine significant jumps in jitter variation and react to them. We observe the jitter variation over a period of 8 frames and calculate the exponential moving average jitter variation. The current jitter variation is compared to the average jitter variation and the ratio is used to adapt the codec. Table I shows how the congestion control reacts to different ratios. Higher jumps in the jitter variation result in a higher ratio and trigger a higher quality reduction.

We also use a 5 ms jitter variation threshold, because the ratio is generally higher if we compare two low values. For example a 1 ms jump in the jitter from 1 ms to 2 ms would result in a 2.0 ratio, while the same jump from 5 ms to 6 ms results in a 1.2 ratio. At the moment, the 5 ms threshold is an experimentally determined fixed value, which could be improved by a variable value in the future.

The amount of temporal layers depends on the quality of the video stream. A temporal enhancement layer is added if the quality stays above 60% over a period of 8 frames. If the quality stays below 40% over a period of 8 frames a temporal enhancement layer is removed. This strategy is based on the quality and is only indirectly influenced by the jitter variation. Therefore the response time with the temporal enhancement layer is much slower, but since changes on the temporal layer inflict bitrate much stronger, we take more time to decide if it is beneficial.

Increasing quality is more complicated, since we do not have a reliable indicator of a free link. We can assume a resolving congestion if the delay shows a negative drift and then react to it, but whenever the video stream bitrate stays below the available bandwidth and additional bandwidth gets available, the jitter variation does not change. Therefore the quality increases by 5% - 15% (depending on the current quality) when the quality does not change over a variable period of frames, which is initialized with 8 frames. If the jitter variations stays stable, we lower the period of frames by one frame. On the other hand, when the jitter variation changes significantly the period of frames is reset to 8. This is an optimization to realize a faster quality increasing on a free link. Otherwise it takes too long until the quality is high

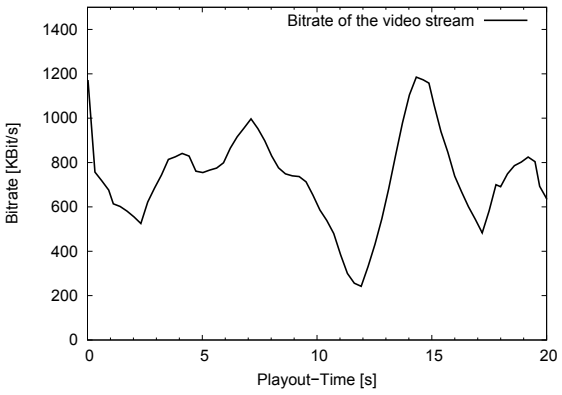

Fig. 1. Bitrate variation for test video

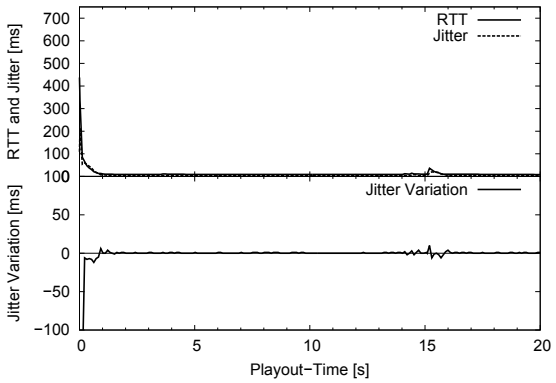

Fig. 2. RTT, Jitter and Jitter Variation on a free link

enough to use the available bandwidth optimally.

We tested our approach with a simple streaming application on a small testbed with a parametrizable, congested link. The test video has a 768x576 resolution and 20 fps. Figure 1 shows the bitrate variation of the video stream with highest quality. The available bandwidth is configured to 700 kbps and our objective in this scenario is to keep the video bit-rate below the 700 kbps and the RTT below 200 ms. A RTT higher than 200 ms (100 ms one-way delay on a symmetric link) is a noticeable congestion for the user and we will use this as an indicator for a not avoided congestion [9].

V. Results

We measured RTT, jitter and jitter variation on a free path without any congestion (both links had 100 Mbps up- and downstream). The results are shown in Figure 2. The RTT, the jitter and the jitter variation behave as expected on a free path. Sufficient bandwidth is available for our video stream without any competing traffic and therefore we have a constant

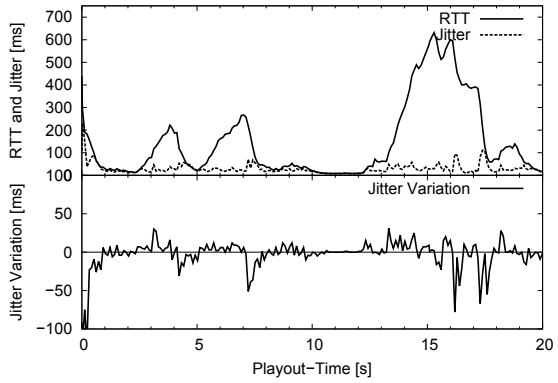

Fig. 3. RTT, Jitter and Jitter Variation on a link with 700 kbps available bandwidth

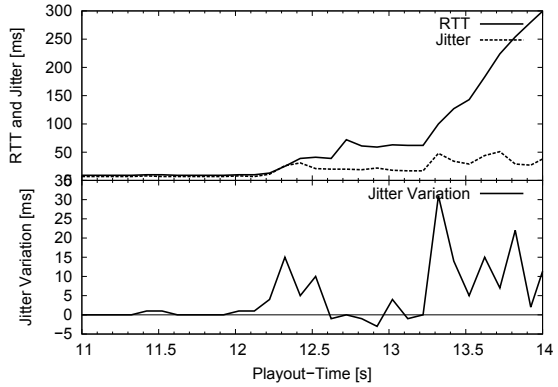

Fig. 4. RTT, Jitter and Jitter Variation on a link with 700 kbps available bandwidth between 11 s and 14 s

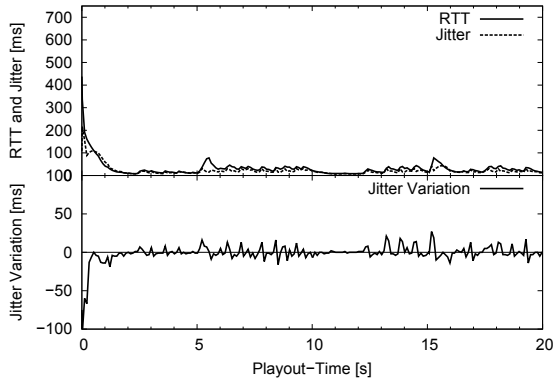

Fig. 5. RTT, Jitter and Jitter Variation on a link with 700 kbps available bandwidth and a scaled video stream

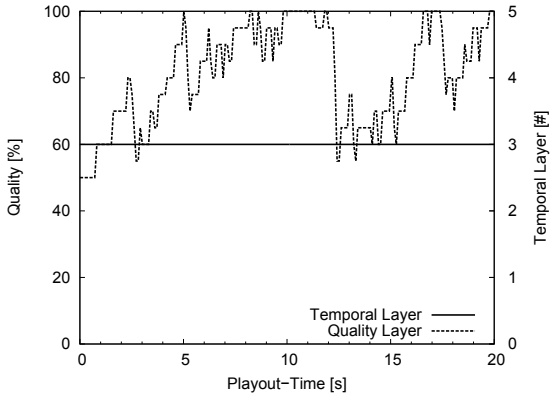

Fig. 6. Quantization and Temporal Layer on a link with 700 kbps available bandwidth

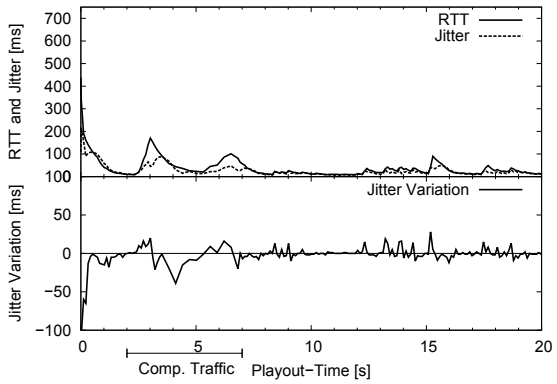

Fig. 8. RTT, Jitter and Jitter Variation on a link with 700 kbps available bandwidth and 100 kbps competing UDP traffic and an adaptive codec

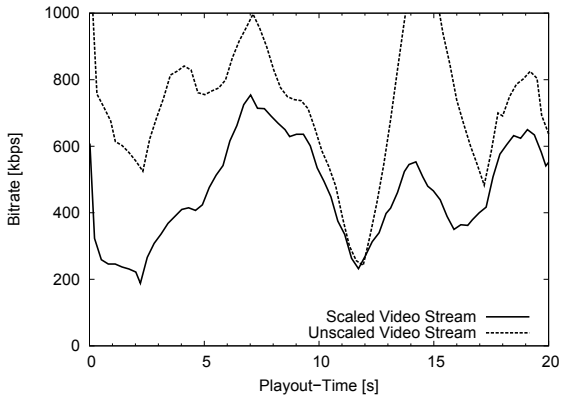

Fig. 7. Bitrate of the scaled and unscaled video stream on a link with 700 kbps available bandwidth

RTT, which is not inflected by any queuing delays. However, we have a little peak at the 15s mark. At this point in time, the video starts from the beginning and a new Intra frame is encoded, which is significant bigger than Inter frames.

We set the available bandwidth to 700 kbps, which is not enough for the video stream and the link will congest. The results are shown in Figure 3. The link congests at the 3, 5 and 15 second mark and the RTT rises above 200 ms. These congestions are noticeable by the user and therefore inflict the QoE. We try to predict and avoid these congestions and adapt the codec to lower the bandwidth demands of the video stream.

Figure 4 shows the beginning of the congestion at the 15 s mark more detailed. Before the 12 s mark, RTT and jitter variation are low and do not fluctuate much. At 12 s the RTT starts to rise, which goes along with a significant jump in the jitter variation. The RTT stays around 70 ms until it rises again shortly after the 13 s mark. This time the slope of the RTT is higher and therefore the jitter variation has an higher jump. Altogether, the Figure 4 shows that the jitter variation is significant higher when queuing delays occur and is an indicator for a congestion that we can use to scale the video stream.

Figure 5 shows the results for the scaled video stream. Compared to the unscaled video stream in Figure 3, the RTT remains nearly constant and stays below 200 ms all the time. No congestion occurs. Compared to the video stream on a free link in Figure 2, the RTT looks similar. The adaptation behavior is shown in Figure 6. It is sufficient to scale the video stream with quality factor and it is not necessary to remove temporal layers to stay below the available bandwidth. A comparison of the scaled video bitrate to the unscaled video bitrate is shown in Figure 7. The scaled video stream stays below the 700 kbps available bandwidth all the time in order to prevent a congestion. With further research, we hope to improve the congestion prediction especially in the regard that we sometimes underestimate the available bandwidth.

In this scenario we are able to avoid the congestion without removing any temporal layer. To stress the congestion prediction a bit more, we add competing 100 kbps UDP traffic to the network. The competing traffic starts at 2 s and ends at 7 s. The results for a congested link with 100 kbps UDP competing traffic are shown in Figure 8. The RTT stays below the 200 ms mark, but compared to the RTT in Figure 5 it is higher especially at the 2 s mark when the competing traffic starts and the RTT almost reaches 200 ms. The cause for this is the 8 frame waiting time before a temporal layer is removed, as described in IV. In this scenario the amount of frames seems suitable, but it might be too long or short in other scenarios and needs further research.

In this scenario it is not sufficient to scale the video only with the quality layer, but we also need to scale the temporal layer, which is shown in Figure 9. After the competing UDP traffic starts, the jitter variation increases heavily and

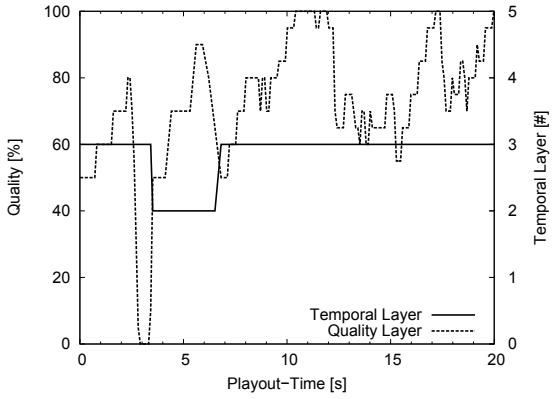

Fig. 9. Quantization and Temporal Layer on a link with 700 kbps available bandwidth and 100 kbps competing UDP traffic

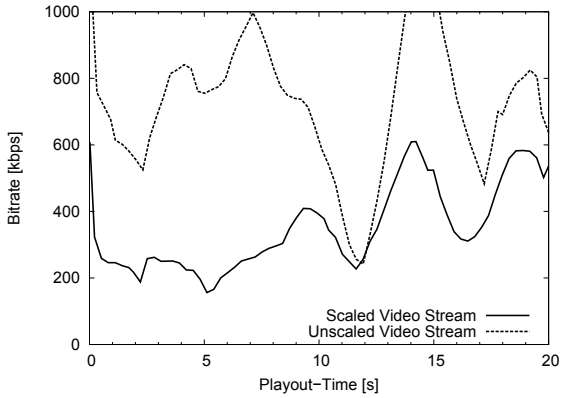

Fig. 10. Bitrate of the scaled and unscaled video stream on a link with 700 kbps available bandwidth and 100 kbps competing UDP traffic

the congestion control reduces the quality to a minimum. Nevertheless, the link will still suffer from congestion and we also have to remove an additional temporal enhancement layers. We avoid the congestion and after the 7 s mark and can readd a temporal enhancement layer. After this congestion it is not necessary to react with the temporal layer anymore to avoid a congestion and ensure a high QoE.

In Figure 10 the bit-rates of the scaled and the unscaled video stream are shown. Our objective in this scenario is to keep the video bit-rate below 700 kbps to ensure a good QoE for the user. The bit-rate of the unscaled video stream often exceeds this limit and the link will congest. The traversal time of the video frames increase due to the queuing delays and the video stream will stutter. The scaled video stream stays below the 700 kbps all the time and therefore no congestion occur. The video stream stays smooth and the user may only notices a reduction of the quality, which results in a higher QoE.

VI. Conclusion and Outlook

Multimedia application with high quality video streams need awareness of the network conditions to ensure a high QoE for each participant. We presented a sender-based, fast and lightweight video adaptation, which is capable to scale the video codec based on the network conditions without a complex and slow approximation of the available bandwidth. We are using the jitter variation instead to predict and avoid a congestion on the link. Since we do not approximate the exact bandwidth, we also had to find a new approach to adapt the video codec. At the moment we use the temporal layer and the quantization to scale the codec regardless of the best QoE. In the future we also consider the spatial and quality layer for scaling and also factor the QoE into the scaling decision.

Our test results showed, that this approach ensures a fast and accurate link observation and is capable to recognize a link congestion early enough to avoid it before it gets noticeable for the user. Especially on links with heavily changing traffic a quick reaction is important for real-time multimedia applications rather than a slow but more accurate measurement. In contrast to common approaches the jitter variation observation approach handles this tradeoff very well.

We conclude that this is a promising approach to adapt the video codec to the conditions of a link and improves the QoE for the participants. In our ongoing work we focus on utilizing the available bandwidth more efficiently and further optimize video scaling in regards of QoE. We will also evaluate our approach on the Internet to examine its performance on links with variable bandwidth and real competing traffic.

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[2] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), "Advanced Video Coding for Generic Audiovisual Services," ITU, Tech. Rep. 8, July 2007.

[3] H. L. Cycon, T. C. Schmidt, G. Hege, M. Wählisch, D. Marpe, and M. Palkow, "Peer-to-Peer Videoconferencing with H.264 Software Codec for Mobiles," in *WoWMoM08 – The 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks – Workshop on Mobile Video Delivery (MoViD)*, R. Jain and M. Kumar, Eds., IEEE. Piscataway, NJ, USA: IEEE Press, June 2008, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WOWMOM.2008.4594916

[4] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile Video Transmission Using Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1204–1217, September 2007.

[5] G. Urvoy-Keller, T. En-Najjary, and A. Sorniotti, "Operational comparison of available bandwidth estimation tools," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 1, pp. 39–42, January 2008.

[6] H. L. Cycon, T. C. Schmidt, M. Wählisch, D. Marpe, and M. Winken, "A Temporally Scalable Video Codec and its Applications to a Video Conferencing System with Dynamic Network Adaption for Mobiles," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1408–1415, August 2011. [Online]. Available: http://dx.doi.org/10.1109/TCE.2011.6018901

[7] Lee Salzman, "Enet," 2012. [Online]. Available: http://enet.bespin.org

[8] M. Palkow, "The daViKo homepage," 2012, http://www.daviko.com.

[9] ITU, "G.114 - One-way transmission time," ITU, Recommendation - Telecommunication Union Standardization Sector, 05 2003.