Multimed Tools Appl (2011) 55:227–246
DOI 10.1007/s11042-010-0572-5

# Priority-based Media Delivery using SVC with RTP and HTTP streaming

**Thomas Schierl · Yago Sanchez de la Fuente · Ralf Globisch · Cornelius Hellge · Thomas Wiegand**

Published online: 18 September 2010


**Abstract** Media delivery, especially video delivery over mobile channels may be affected by transmission bitrate variations or temporary link interruptions caused by changes in the channel conditions or the wireless interface. In this paper, we present the use of Priority-based Media Delivery (PMD) for Scalable Video Coding (SVC) to overcome link interruptions and channel bitrate reductions in mobile networks by performing a transmission scheduling algorithm that prioritizes media data according to its importance. The proposed approach comprises a priority-based media pre-buffer to overcome periods under reduced connectivity. The PMD algorithm aims to use the same transmission bitrate and overall buffer size as the traditional streaming approach, yet is more likely to overcome interruptions and reduced bitrate periods. PMD achieves longer continuous playback than the traditional approach, avoiding disruptions in the video playout and therefore improving the video playback quality. We analyze the use of SVC with PMD in the traditional RTP streaming and in the adaptive HTTP streaming context. We show benefits of using SVC in terms of received quality during interruption and re-buffering time, i.e. the time required to fill a desired pre-buffer at the receiver. We present a quality optimization approach for PMD and show results for different interruption/bitrate-reduction scenarios.

**Keywords** H.264/AVC · SVC · Scalable video coding · Adaptive HTTP streaming · Mobile channels · Link interruptions · Transmission rate variation · 3GPP PSS

This work has been partially published by the authors in "Priority-based Transmission Scheduling for Delivery of Scalable Video Coding over Mobile Channels" at the 3rd European Symposium on Mobile Media Delivery (EUMOB), London, UK, 2009

T. Schierl (✉) · Y. Sanchez de la Fuente · R. Globisch · C. Hellge · T. Wiegand
Image Communication Group, Image Processing Department, Fraunhofer HHI, Einsteinufer 58, 10587 Berlin, Germany
e-mail: schierl@hhi.fhg.de

T. Schierl · Y. Sanchez de la Fuente · C. Hellge · T. Wiegand
Image Communication Group Department of Telecommunication Systems, Technische Universität Berlin, 10587 Berlin, Germany

## 1 Introduction

Video delivery over mobile channels may be affected by temporary link bitrate variations or link interruptions caused by the changes in the channel condition or the wireless interface. Providing a reliable video transmission over mobile channels requires the use of pre-buffers to overcome link interruptions and the media bitrate adaptation to cope with varying channel throughput.

Priority-based Media Delivery (PMD) for SVC media data is presented in this paper as a possible solution for link interruption. In the traditional streaming approach, each video access unit or part thereof, e.g., an H.264/AVC NAL unit, is transmitted according to its decoding time. This transmission approach is referred to as earliest deadline first (EDF) transmission scheduling. On the contrary, the proposed approach uses a priority scheduling algorithm that pre-buffers larger amounts of more important data than of data with less importance to overcome problems in the network, as illustrated in Fig. 1. In this figure the video units are classified into three groups according to their importance. Different durations of media are buffered in each group: $t_3$ of the least important data, $t_2$+$t_3$ of the data with medium importance and $t_1$+$t_2$+$t_3$ of the most important data is stored in the pre-buffer. This results in a pre-buffer with a longer playback time compared to the traditional approach as both scheduling algorithms use the same buffer size. The resulting video playback quality can thus be improved. In [9], the PMD was realized using the temporal scalability feature of H.264/AVC. The approach in [9] aims to reduce the video frame rate down to a slide show during interruptions and to enter a re-buffering phase after the interruption. The priority-based transmission requires some means of controlling client buffer occupation as discussed in [7, 9], which is only possible over a bidirectional connection with feedback about the buffer filling level or similar information. The 3GPP PSS standard [1] provides means for feedback-based transmission rate control [3] which may need to be extended to perform the priority-based scheduling. However, a transmission of the different priority classes in different RTP flows would already allow the use of the existing 3GPP PSS signaling.

In this paper, we focus on the use of the new scalability features of H.264 Scalable Video Coding (SVC) [6, 15]. SVC has the useful advantage to allow additionally bitrate reduction using SNR fidelity or spatial scalability instead of relying just on temporal scalability as possible with AVC. We compare the use of SVC with PMD against H.264/AVC using temporal scalability with PMD as presented in [9] and H.264/AVC without PMD, i.e. the traditional streaming approach. We also compare the results with a Bitstream Switching (BSS) approach in which the client buffer fullness also determines the playback quality [4]. We show the benefits of using SVC in terms of received quality during interruptions and re-buffering times, where we use PSNR and frame rate as metrics for video

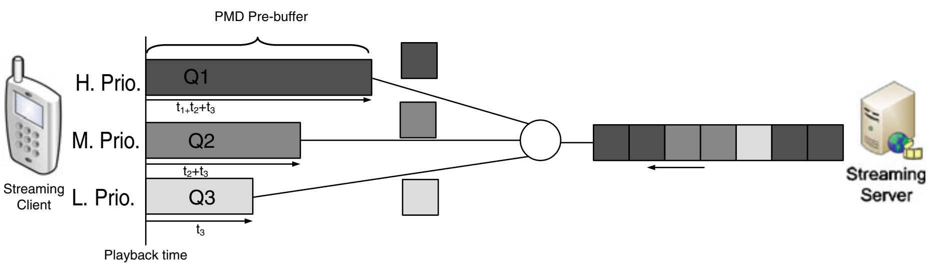



**Fig. 1** Priority-based Media Delivery (PMD) and streaming receiver buffer with three qualities

quality. We present a quality optimization approach for the PMD and show results for different interruption durations for the four different transmission methods.

We additionally extend the PMD approach to the adaptive HTTP streaming context. In adaptive HTTP streaming, media is typically downloaded by the client and the media data is not transported in single packets, but in small segments containing a set of access units. Studies have shown that HTTP/TCP is widely used to stream media to clients even though TCP imposes higher end-to-end delays on the communication [17]. HTTP is not affected by firewall and NAT traversal issues that exist in traditional streaming which typically uses RTP over UDP. Further, UDP communication and traffic using ports other than the default HTTP port (port 80) are often blocked by firewalls. As a result, many content providers have resorted to using HTTP streaming for media delivery, even though the end-to-end delay is increased by the use of TCP. We compare the use of SVC with PMD in the HTTP streaming context with results achieved using BSS.

The remainder of the paper is organized as follows. The next section provides an overview of adaptive streaming in both the traditional and the HTTP context. In Section 3, we present the Priority-based Media Delivery approach using SVC. This includes a novel adaptive HTTP streaming method using SVC, as well as a PMD approach applicable in the traditional streaming context. This is followed by an optimization for different buffer interruption scenarios. We present the results of our experiments in Section 4 and we conclude in Section 5.

## 2 State-of-the-art adaptive streaming

Adaptive streaming is a term used to describe a joint content and transmission bitrate adaptation in response to varying network conditions. Typically the content is pre-encoded at various bitrates or transcoded to a requested bitrate on the fly, and some feedback mechanism exists which allows the server or client to adapt to changing network conditions. To compensate for a transmission bitrate reduction, the video transmission rate is adjusted to match the estimated reduced network throughput. Such estimates are typically based on feedback information provided by the client. The server may then apply a graceful degradation of quality, which is preferable to playback disruptions. The latter occurs when the data transmission bitrate is less than the video consumption rate and as a result the client buffer is consumed and no video data remains to be played back. Once the bitrate reduction has ceased, streaming resumes at the higher bitrate.

### 2.1 Adaptive RTP streaming with BSS

Bitstream Switching (BSS) [16] is one adaptation approach in which the same content is encoded at multiple different bitrates using a single layer codec and stored at the server. The server is responsible for measuring the estimated transmission rate for each client which typically requires feedback from the client, e.g., RTCP Receiver Report (RR) packets. Based on this feedback, the server must deliver content of the appropriate bitrate. However due to the predictive nature of H.264/AVC encoding, BSS can only occur at specific switching or random access points in the bitstream, such as when Instantaneous Decoding Refresh (IDR), recovery point Intra, Switching Intra (SI) or Switching Predictive (SP) frames are encountered. An approach of combining BSS with H.264/AVC temporal scalability has been discussed in [10].

### 2.2 Adaptive HTTP streaming with BSS

In HTTP streaming, media content is subdivided into chunks, also known as segments, of a short duration resembling the real-time streaming of packets. The client is responsible for requesting and downloading each individual chunk also referred to as pulling data, where each chunk is self-contained and allows for immediate playback at the client.

The duration of the chunks affects the amount of HTTP requests required to retrieve the content and the initial delay. A short duration of chunks allows for quicker adaptation to network events and faster TV channel switching times.

This also makes the technology suitable for live media streaming, provided that the initial delay is acceptable. This delay is implicit, since a chunk only becomes available for download, e.g., 5 s after the actual events took place if the chunk size is considered to be 5 s. This delay is increased by the amount of pre-buffering at the client needed to adapt to transmission rate variations. However the server load can be decreased substantially in a live event in which all viewers wish to see the same content at approximately the same time. In this case existing HTTP caching infrastructures of the Internet can be reused by using HTTP streaming thus relieving the server load as well as reducing the overall link traffic. A further advantage of HTTP streaming is that TCP's rate control mechanism is automatically used allowing for fast transfer rates when the transmission rate is available. The maximum throughput rate however is limited since the individual chunk size may be relatively small.

To perform Adaptive HTTP Streaming the content is typically encoded into chunks at multiple bitrates. One chunk must exist for each of the desired bitrates and the client uses some adaptation model to retrieve chunks matching the estimated network throughput [4]. In this pull-based model the client is responsible for adaptation to network variations, eliminating the need for an additional feedback loop such as RTCP which is used for transmission rate control in traditional streaming. The bigger size of the HTTP streaming chunks in comparison to the size of the RTP packets allows for better network throughput estimation on the client side.

### 2.3 Adaptive streaming using SVC

Different solutions have been proposed in the last years where H.264/AVC-based Scalable Video Coding (SVC) has been analyzed in adaptive streaming service environments. In [19], an efficient adaptation framework is proposed using SVC. The benefits of SVC for adapting the video data in scenarios with heterogeneous device capabilities and network dynamic behavior are shown. Adaptation mechanisms to cope with packet loss in RTP streaming are presented in [8], where adaptation methods are introduced based on information contained in the SVC NAL unit headers [15]. [11] presents an approach which combines rateless forward error correction code and scalable video coding for distribution of layered video from different sources in an overlay network on top of a Mobile Ad-hoc Network (MANET). Rate allocation and adaptation of the scalable video stream are performed in a rate distortion optimized manner. These and many other works have already shown the potential of SVC in combination with adaptive streaming under real-time constraints.

## 3 Priority-based Media Delivery (PMD)

The priority-based transmission scheduling approach has been already discussed and evaluated for H.264/AVC over 3GPP Packetized Streaming Service (PSS) in [9] and [7].

The idea of Priority-based Media Delivery (PMD) for H.264/AVC in streaming scenarios is to pre-buffer a larger amount of more important media data and pre-buffer smaller amounts of less important media data in the streaming receiver's buffer. This allows for graceful quality degradation during interruption or transmission bitrate variation periods. Data in the buffer is consumed at the playback rate of the media. The pre-buffering level is maintained as long as the transmission rate matches the consumption rate, but during an interruption or transmission rate reduction period the pre-buffering level cannot be maintained as the data arrival rate is less than the data consumption rate. In the case where a buffer has been filled using a priority scheduling algorithm, the high quality data in the buffer runs out before the more important data of lower quality.

In Fig. 1, we show a streaming receiver's priority buffers for three different qualities as also used in [9]. The PMD achieves that at first the buffer of the lowest quality *Q1* (e.g., intra frames or base layer data) is built up to a data amount equivalent to a playback duration of $t_1$+$t_2$+$t_3$. The quality *Q1* corresponds to a constant bitrate $r_1$. After that a second buffer for *Q2* is built up which pre-buffers a playback duration of $t_2$+$t_3$, where the constant rate of *Q2* corresponds to $r_1$+$r_2$. The same procedure is applied to *Q3* media data, where the constant rate of *Q3* corresponds to $r_1$+$r_2$+$r_3$ and the buffer is filled up to a playback duration amount of $t_3$. In other words, $r_i$ would be the media rate for each of the buffered qualities $Q_i$, $t_3$ the duration of buffered data for *Q3* and $t_1$ and $t_2$ the additional duration buffered from *Q2* to *Q1* and *Q3* to *Q2* respectively.

Once the priority buffers have been filled to their minimum respective buffer levels, the pre-buffering phase is complete and the buffering phase begins. In the buffering phase the available transmission rate is used to ensure that each quality *Qi* remains filled up with its data rate $r_i$. If the available transmission rate exceeds the video rate, the additional transmission rate is used to fill the buffers in an EDF (earliest deadline first) manner, i.e. the data with the earliest playback time is transmitted first irrespective of priority.

The occurrence of an interruption or a bitrate reduction can result in the buffers not meeting the minimum required filling level $Q_i$ and the PMD scheduling algorithm must then re-enter the priority re-buffering phase (Prio/Reb in Fig. 2) until the minimum filling level requirements are once again satisfied. In the case where an interruption has occurred, the priority re-buffering phase can only begin once the connection has been restored as shown in Fig. 2.

## 3.1 Priority-based Media Delivery (PMD) using SVC

SVC has the advantage in media delivery to allow a layered representation using SNR fidelity or spatial scalability in addition to the pure temporal scalability as possible in H.264/AVC as shown in [9], which makes SVC a preferred candidate for the PMD approach. Individual layers can be identified based on SVC's T-D-Q values, which indicate the temporal (T), spatial (D) and quality (Q) resolution of a layer. These indicators are stored in the high level interface of SVC, the SVC NAL unit header [15], and indicate the levels of temporal, spatial and SNR fidelity resolution of a specific NAL unit. In the following, the term layer will loosely refer to a specific combination of T-D-Q values. Since


**Fig. 2** Priority-based Media Delivery (PMD) with priority pre-buffering (*Prio/Reb*) and buffering phases

temporal scalability is already a part of H.264/AVC, the level of temporal resolution can vary arbitrarily based on the selected prediction structure, the Group of Pictures (GOP).

In case of using SVC with PMD, a higher importance is assigned to the base layer and lower priorities are assigned to subsequent enhancement layers. Assuming a fixed client buffer size equivalent to the size of the buffer used in traditional streaming, the distribution of the media content within the buffer can be changed: By filling the buffer with larger amounts of the more important data, e.g., the base layer, and smaller amounts of less important one, e.g., enhancement layers. Therefore, the duration of data that can be played back in the event of an interruption or rate reduction is longer than the duration of data, had the same buffer been filled using EDF.

#### *3.1.1 PMD in real time streaming using SVC*

The idea of Priority-based Media Delivery (PMD) for single layer H.264/AVC in streaming scenarios is to pre-buffer a larger amount of more important NAL units and pre-buffer smaller amounts of less important NAL units in the streaming receiver's buffer. Compared to the use of earliest deadline first (EDF) transmission scheduling, in which data is sent in decoding order, the highest quality data is consumed faster when the PMD approach is used. Nevertheless, the priority-based scheduling allows for keeping the playback alive during longer interruptions than in the EDF case. However, as a result of the coding efficiency penalty incurred by SVC [14], the available transmission rate is 10% higher than the video rate in both H.264/AVC single layer cases with PMD and EDF when compared with SVC, i.e. this additional 10% rate is available to refill the buffer faster.

Since the application of PMD is based on the buffer filling level and traditional streaming is inherently push-based, the client needs to provide feedback regarding the buffer levels to the server. An extension to RTCP RRs is specified in [1] and could be used to provide this feedback. Additionally, the use of PMD requires that data is sent in an interleaved fashion, which is supported by both the RTP payload format for H.264/AVC as well as the RTP payload format for SVC [18].

#### *3.1.2 PMD for HTTP streaming using SVC*

To apply PMD in the case of HTTP streaming, an SVC bitstream is generated which is demultiplexed into separate streams determined by individual or sets of T-D-Q values of the NAL unit header. This approach is flexible with respect to the target transmission rates that can be achieved by applying different scalability features to subdivide the SVC bitstream. Each of these streams is again subdivided into chunks of $t_{chunk}$ length, where $t_{chunk}$ is kept relatively small to facilitate quicker response times. This subdivision must occur at the same point in all layers and may be based on containers such as the MPEG-2 Transport Stream extensions for SVC [12, 13] as well as the SVC File Format [2, 5].

In the pull-based HTTP streaming model the client adapts to network variations by selecting which chunks to get from the HTTP server and switching to a lower quality is accomplished by omitting HTTP GET requests for enhancement layer chunks. Since the client is aware of the buffer fullness state, the need for a feedback loop is eliminated, making HTTP streaming an obvious fit for the PMD approach. Once the client is ready to play back media data of a specific interval, all chunks belonging to the same time interval have to be multiplexed together before being delivered to the SVC decoder.

An added advantage of providing adaptive HTTP streaming using SVC is that content duplication on HTTP servers is avoided and that network caches such as HTTP proxies and the network itself can be used more efficiently by increasing the cache hit ratio for a chunk. Figure 3 illustrates the reduced network link usage and how the network cache is used more efficiently when using SVC for adaptive HTTP streaming in comparison to using BSS. This could be of interest to Internet Service and Content Delivery Network Providers that typically have to maintain large server and network infrastructures.

Figure 3 shows an exemplary network topology, where multiple clients request the same content with different bitrates and video qualities (*Q1*,*Q2*,*Q3* in the figure). The network employs some caching nodes (rectangles in the figure), which store transmitted content following a certain caching algorithm. The figure shows the throughput at each node within the network, as well as the content cached in the network. In the AVC case, three different quality representations are comprised of three separate streams alternative to each other and in case of SVC they are complementary to each other. This means that for AVC three complete video streams have to be encoded and must be transmitted and stored in the network caches when requested. Considering SVC, a single encoded video stream is subdivided into layers, each one corresponding to a given quality. Thus, when a representation is requested only the missing layers may be transmitted and stored in the network caches thereby reducing the load on the links and the allocated storage in the caches, as shown in the figure. Furthermore, all SVC clients requesting quality $Q_j$ must also request all chunks $Q_i$ where i<j. The number of clients requesting the same content increases improving the cache hit ratio, while in traditional HTTP streaming with AVC only the streaming clients requesting quality $Q_j$ expect to receive a chunk of this quality representation.

The idea of PMD using SVC in HTTP streaming scenarios is to pre-buffer a larger amount of more important chunks and to pre-buffer smaller amounts of less important chunks in the streaming receiver's buffer. The most important chunks contain the base layer media data. Less important, higher quality chunks can contain media information of higher spatial, quality or temporal resolution.

### 3.2 Optimizing PMD for SVC

The optimal prioritization for the PMD method depends on the constraints imposed by the network throughput and receiver buffer size. Two different scenarios have been considered

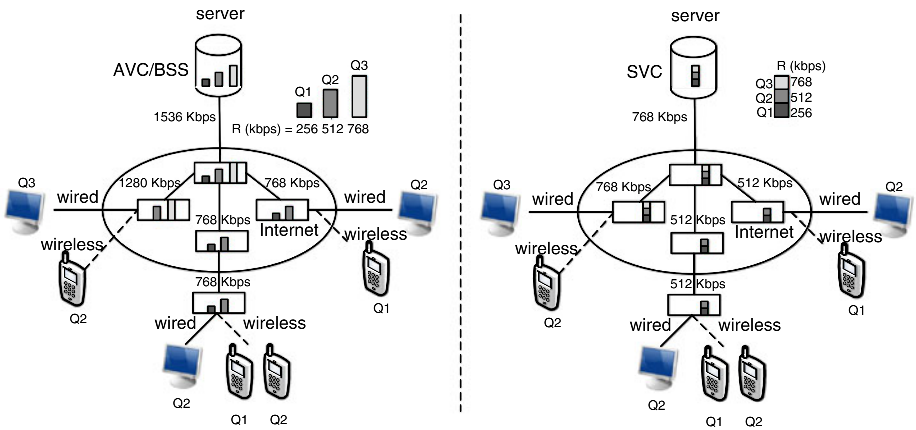



**Fig. 3** HTTP streaming, caching and usage of network resources

to optimize the PMD. In the first scenario the transmission rate is limited to the video rate and the receiver buffer is limited in size. In the second scenario, high transmission rates up to twice the video rate are possible, and an unlimited receiver buffer is assumed. In both scenarios a live streaming service is considered, in which the mobile connection is affected by channel interruptions and transmission rate variations.

For the optimization of the presented PMD method, each layer of an SVC stream is encoded at a constant bit rate resulting in a constant bitrate stream. We assume a PMD with three quality buffers, which are filled to the corresponding playback duration values for each buffer (*Q1:* $t_1$+$t_2$+$t_3$, *Q2:* $t_2$+$t_3$ and *Q3:* $t_3$) as is shown in Fig. 1, where *Q1* is represented with a dark grey color, *Q2* with medium grey color and *Q3* with a light grey color. The optimization entails finding the appropriate pre-buffer lengths that maximize the average quality over time.

### *3.2.1 Scenario 1: transmission rate limitation and memory limitation at the receiver*

In this scenario, the transmission rate $R_t$ is limited to the bitrate of the video $R = r_1 + r_2 + r_3$. Furthermore, the maximum buffer size is limited to a value corresponding to the amount of data played back during a time equal to the average interruption length. The interruption time $t_{interruption}$, which is the time interval with a transmission rate equal to zero, is assumed to always be less than the maximum interruption time $t_{max\,interruption} = t_1 + t_2 + t_3$, which at least allows for continuous playback at the lowest quality for the PMD cases. For comparison of EDF and PMD, the same total buffer size is used.

After an interruption, the PMD algorithm must enter the re-buffering phase $t_{rebuf}$. The duration of the interruption affects how the buffers are refilled:

A. During re-buffering, the quality is degraded to the lowest quality *Q1*
B. During re-buffering, the quality is degraded to a medium quality *Q2*

This is due to the constraint, that the available transmission rate for re-buffering and playback is limited to the video rate *R*. Therefore, at least one quality has to be dropped in order to obtain additional rate to re-buffer the lower qualities, while continuously playing back the lower qualities at the same time. Figures 4 and 5 show the two possibilities A and B.

Note that the re-buffering process in our model is carried out in the fastest possible way. Although it is possible to already increase the quality during the refill phase of the buffer to a medium quality, such a re-buffering phase would take more time and would therefore reduce the overall quality.

In Fig. 4, case A is depicted. The dashed boxes on the left side of the figure indicate the initial state of the buffer before the interruption happens. They correspond to the initial state of the pre-buffer, which should be equal to the buffer presented in Fig. 1. During the interruption (grey shaded box), the data in the buffer is played and thereafter the re-buffering process is started $t_{rebuf}$. PMD aims to ensure that at least data of video quality *Q1* is available. The dotted dark grey box represents additional data, necessary to keep on playing at *Q1* until the buffer is fully re-buffered.

If the interruption duration is short enough, so that the re-buffering completes before the *Q2* buffer runs out, case B occurs as depicted in Fig. 5. The dashed boxes for *Q1* and *Q2* in Fig. 5 show the initial state of the buffer. It is noticeable, that in case B less data has to be sent during the re-buffering phase.

In order to find the buffer structure, which maximizes the quality for a given interruption duration, the two possible cases A and B are studied in the remainder of this section.

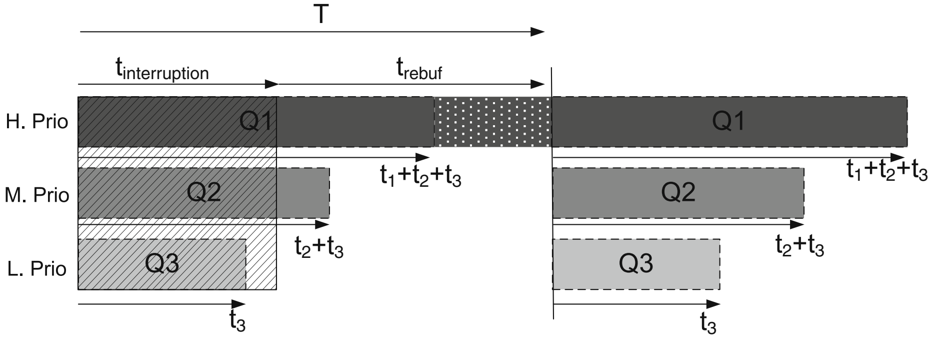



**Fig. 4** Impact of interruption which requires graceful quality degradation to the lowest quality: case A

Therefore, we first discuss the bounds for $t_{interruption}$ which differentiate between the two cases A and B, with the following definitions. The time $T = t_{interruption} + t_{rebuf}$ is the time it takes to get back to the highest quality after the beginning of the interruption. The condition for case B is fulfilled, if Eq. 1 is true.

$$T = t_{\text{interruption}} + t_{rebuf} \leq t_2 + t_3 \quad : case \quad B \tag{1}$$

That means that during the interruption and the re-buffering phase in case B, the medium quality *Q2* can still be played back. Taking further into account the video rate being $R = r_1 + r_2 + r_3$, the transmission rate $R_t$ and the amount of re-buffered data being $t_{rebuf}{}^*R_t = r_3{}^*t_3 + (r_2 + r_1)^*\left(t_{interruption} + t_{rebuf}\right)$, the aforementioned condition (Eq. 1), can be transformed to Eq. 2, which is the condition for being in case B. The amount of re-buffered data is equal to the sum of data played out at quality *Q1* + *Q2* during $t_{rebuf}$, the data to be re-buffered for *Q1* and *Q2* played during $t_{interruption}$ and the data to completely re-buffer *Q3*.

$$t_2 \geq \frac{R_t{}^*t_{\text{interruption}} + (r_1 + r_2 + r_3 - R_t)^*t_3}{R_t - r_2 - r_1} \quad : case \quad B \tag{2}$$

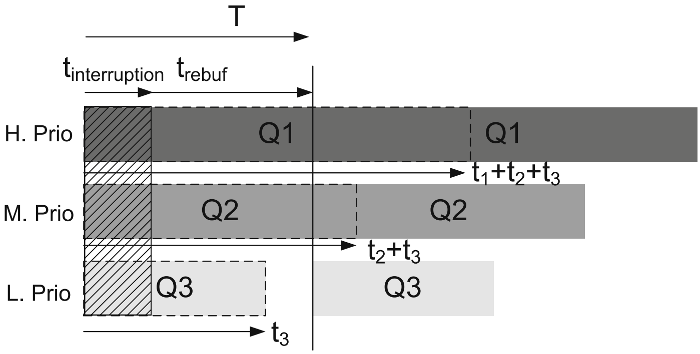



**Fig. 5** Impact of interruption which requires graceful quality degradation to the medium quality only: case B

In Eq. 3 the relation between the variables $t_2$ and $t_3$ is expressed through the fixed buffer size $B$, with $t_{maxinterruption} = t_1 + t_2 + t_3$ (compare Fig. 1).

$$B = t_{\mathrm{maxinterruption}}{}^{*}r_1 + (t_2 + t_3)^{*}r_2 + t_3{}^{*}r_3 \tag{3}$$

Besides, in Eqs. 4 and 5, the time for re-buffering $t_{rebuf}$ as well as the time of quality degradation $T$ are calculated, which may be not only dependent on $t_{interruption}$ but also on $t_2$ and $t_3$ as shown in the following.

$$t_{rebuf}(t_2, t_3) = \begin{cases} \dfrac{t_{\mathrm{interruption}}{}^{*}r_1 + B - t_{\mathrm{maxinterruption}}{}^{*}r_1}{R_t - r_1}, & t_2 \frac{R_t{}^{*}t_{\mathrm{interruption}} + (r_1 + r_2 + r_3 - R_t)^{*}t_3}{R_t - r_2 - r_1} \; (case \quad A) \\ \dfrac{t_{\mathrm{interruption}}{}^{*}(r_2 + r_1) + t_3{}^{*}r_3}{R_t - r_1 - r_2}, & t_2 \geq \frac{R_t{}^{*}t_{\mathrm{interruption}} + (r_1 + r_2 + r_3 - R_t)^{*}t_3}{R_t - r_2 - r_1} \; (case \quad B) \end{cases} \tag{4}$$

For case A (upper part) in Eq. 4, we summed first the amount of data played at lowest quality during the interruption (which needs to be re-buffered to $t_{maxinterruption}$) and the amount of data, which needs to be re-buffered for the two other qualities to fill up the priority buffer, and second divided the sum by the bitrate available for re-buffering (while playing quality *Q1*).

For case B (lower part) in Eq. 4, we first summed the amount of data played at lowest and medium quality during the interruption (which needs to be re-buffered to $t_{maxinterruption}$ and $t_2$+$t_3$ respectively) and the amount of data, which needs to be re-buffered for the highest quality *Q3* to fill up the priority buffer, and second divided the sum by the bitrate available for re-buffering (while playing quality *Q1* and *Q2*). From Eqs. 1 and 4, Eq. 5 is concluded as follows:

$$T(t_2, t_3) = \begin{cases} \dfrac{t_{\mathrm{interruption}}{}^{*}R_t + B - t_{\mathrm{maxinterruption}}{}^{*}r_1}{R_t - r_1}, & t_2 \frac{R_t{}^{*}t_{\mathrm{interruption}} + (r_1 + r_2 + r_3 - R_t)^{*}t_3}{R_t - r_2 - r_1} \; (case \quad A) \\ \dfrac{t_{\mathrm{interruption}}{}^{*}R_t + t_3{}^{*}r_3}{R_t - r_2 - r_1}, & t_2 \geq \frac{R_t{}^{*}t_{\mathrm{interruption}} + (r_1 + r_2 + r_3 - R_t)^{*}t_3}{R_t - r_2 - r_1} \; (case \quad B) \end{cases} \tag{5}$$

Since the aim is to ensure continuous playback during and after the interruption at least at minimum quality *Q1* and therefore ($t_1$+$t_2$+$t_3$) is fixed and equal to $t_{maxinterruption}$, the lengths $t_2$+$t_3$ and $t_3$ of the buffers for *Q2* and *Q3* are the variables for which the optimal solution is searched.

Whereas for case A $T(t_2,t_3)$ is constant, for case B the time $T(t_2,t_3)$ after which the playback is back at highest quality varies with $t_3$ and, therefore, with $t_2$, compare Eq. 5. For the optimizations, we use the largest possible value of $T(t_2,t_3)$ in case B, which corresponds to the smallest value $t_{2min}$ of $t_2$ and to the highest value $t_{3max}$ of $t_3$, and fulfills Eq. 2 with equality. For higher values of $t_2$, $T$ is smaller. Thus for the additional time $T(t_{2min,}t_{3max}) - T$ the highest quality *Q3* could be played back, in case B. Thus, the playback time with highest quality is $t_3 + \left(T\left(t_{2min,}t_{3max}\right) - T\right)$, compare Eq. 6. Since the quality of the video typically depends on a single rate-distortion function, *Q1*, *Q2* and *Q3* are derived as $q(r_1)$, $q(r_1+r_2)$ and $q(r_1+r_2+r_3)$ respectively in the following.

For case A, since $T(t_2,t_3)$ does not depend on $t_2$ *and* $t_3$, the average quality calculation is straight forward following Fig. 4. All qualities need to be re-buffered, but only quality $q(r_1)$ needs to be continuously played out.

The average quality for both cases A and B is shown in Eq. 6, where $T(t_2,t_3)$ is represented as $T$ for simplification.

$$Q(t_2,t_3)=\begin{cases}\frac{1}{T}*[(T-t_2-t_3)*q(r_1)+t_2*q(r_1+r_2)+t_3*q(r_1+r_2+r_3)] \\ \quad for \quad t_2<\frac{R_t*t_{\text{interruption}}+(r_1+r_2+r_3-R_t)*t_3}{R_t-r_2-r_1} \ (case \quad A) \\ \\ \frac{1}{T(t_{2\min},t_{3\max})}*[(T-t_3)*q(r_1+r_2)+(T(t_{2\min},t_{3\max})-T+t_3)*q(r_1+r_2+r_3)] \\ \quad for \quad t_2\geq\frac{R_t*t_{\text{interruption}}+(r_1+r_2+r_3-R_t)*t_3}{R_t-r_2-r_1} \ (case \quad B)\end{cases} \tag{6}$$

In order to calculate the maximum quality for each of the two cases, Eq. 6 has to be optimized with respect to $t_2$ and $t_3$ respectively as shown in Eq. 7.

$$\underset{t_2,t_3}{\arg\max}(Q(t_2,t_3)) \tag{7}$$

Note that to perform the optimization for bitrate reduction states, the following approach is used, which is to transpose the rate reduction phase to an equivalent interruption length $t_{eq\_interruption}$ following Eq. 8, where $R_{red}$ and $t_{rate\_reduction}$ correspond to the reduced rate and the duration of the rate reduction phase respectively.

$$t_{eq\_\text{interruption}}=\frac{(R_t-R_{red})*t_{rate\_reduction}}{R_t} \tag{8}$$

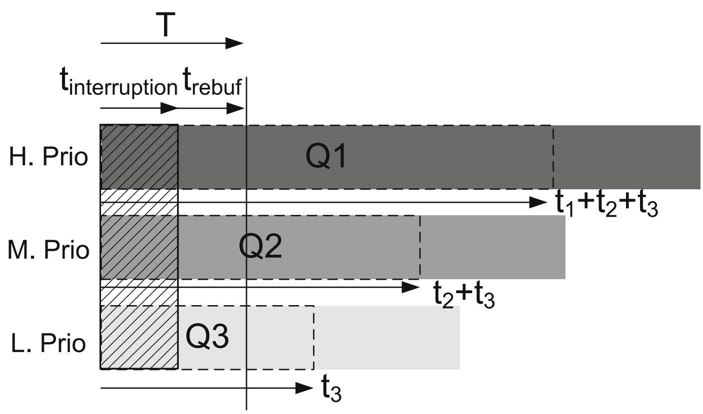



**Fig. 6** Example of an interruption shorter than the pre-buffer for the highest quality allowing continuous play back at highest quality

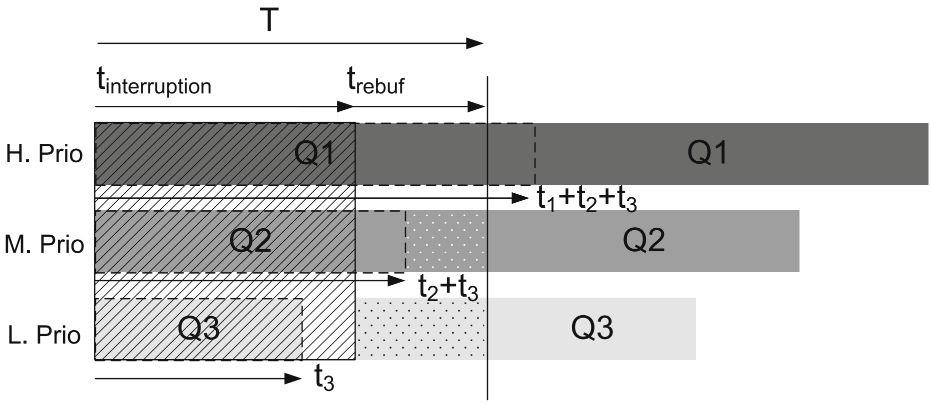



**Fig. 7** Example of an interruption longer than the pre-buffer for highest quality

*3.2.2 Scenario 2: high transmission rates and no memory limitation at the receiver*

In this scenario graceful degradation is not always necessary. In Scenario 1 the available transmission rate for re-buffering and playback after the interruption is limited to the video rate *R*. Therefore, at least one quality has to be dropped in order to obtain additional rate to re-buffer the lower qualities, while simultaneously playing back already buffered data. However in the scenario considered here, higher transmission rates up to twice the video rate are possible. Thus, re-buffering could occur while playing back the media at the maximum quality. Figures 6 and 7 show the two cases, where the interruption is not long enough to impose graceful degradation and where the interruption is longer than the pre-buffer for highest quality *Q3*, respectively.

Since in this scenario the graceful degradation is imposed by the length of the interruption and not by the limited transmission rate as before, the playback quality does not have to be kept at a lower quality during the re-buffering phase. Dropping higher quality media data is no longer necessary, provided that there is sufficient bandwidth to transmit the higher quality data as shown in Figs. 6 and 7. Furthermore, due to the fact that the buffer size is not limited, it is not necessary to pre-buffer less data of *Q3* to obtain additional space to pre-buffer larger amount of data of *Q2*.

Even though it is not necessary to stop the transmission of any quality to perform the re-buffering phase, PMD aims to restore the priority buffer requirements as fast as possible to overcome limitations in transmission rate. Depending on the selected quality ($Q_i$) and rate ($r_i$) values of the video streams different scheduling has to be performed. In the selected three quality layer case, the pre-buffering of *Q2* and *Q3* occurs in parallel using the maximum available transmission rate.

## 4 Selected simulation results

### 4.1 Real time streaming in mobile channels

Simulations were carried out to test the performance of PMD in mobile channels with a limited bitrate and buffer size. A buffer size of $B$=250 kB was selected, which amounted to approximately 5 s of SVC video at the selected bitrate of 410 kbps. Since SVC incurs a

10% efficiency penalty over H.264/AVC and AVC reaches the desired quality at a lower video rate, this amounts to a slightly longer duration of video that can be stored in the buffer when AVC video is used.

The selected streams are a concatenation of the ITU-T test sequences City, Crew, Foreman and Soccer at QVGA resolution and 30 fps with a length of about 40 s. The streams have a Group of Picture (GOP) size of 16 plus a preceding IDR picture for each GOP, i.e. the stream has a random access interval of 0.57 s. A rate control has been used to keep the bitrate within a ±2.5% window of the average value per IDR+GOP16 picture chunk. The bitrate of the H.264/AVC stream is 370 kbps and the bitrate of the SVC stream is 410 kbps. The SVC stream uses MGS fidelity scalability with one enhancement layer, where the base layer has about 160 kbps. The hierarchical prediction structure with GOP16 + a preceding IDR per chunk allows a temporal scalability with up to six levels for the H.264/AVC stream.

The following four methods were evaluated:

I. SVC-PMD—SVC with three quality levels and PMD, where the priority buffer is always re-buffered after an interruption.
II. AVC-PMD—H.264/AVC with three quality levels and PMD, where the priority buffer is always re-buffered after an interruption.
III. AVC-EDF—H.264/AVC with EDF, where re-buffering only occurs once the buffer is empty.
IV. BSS—H.264/AVC with Bitstream Switching, where the switching is performed based on the buffer fullness.

Figure 8 shows the three quality levels for the H.264/AVC and the SVC streams used for PMD, where the SVC rate-distortion points are derived by dropping MGS NAL units in the enhancement layer only and the H.264/AVC rate-distortion points are achieved using temporal scalability. Solving Eq. 7 results in $t_2=0$ for the selected qualities and rates of the test sequence. This means that the use of only two quality buffers makes sense for the selected rate points.

In Fig. 9, we show as an example the resulting video quality over time for two different interruptions of 3 and 8 s comparing PMD with both AVC and SVC codecs and the EDF and BSS approaches. In Fig. 9 a) the PSNR over the two interruption events is

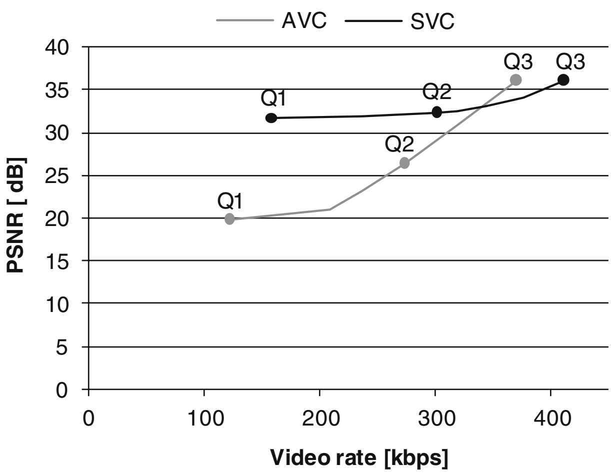



**Fig. 8** The test stream qualities—SVC using MGS scalability and H.264/AVC using temporal scalability

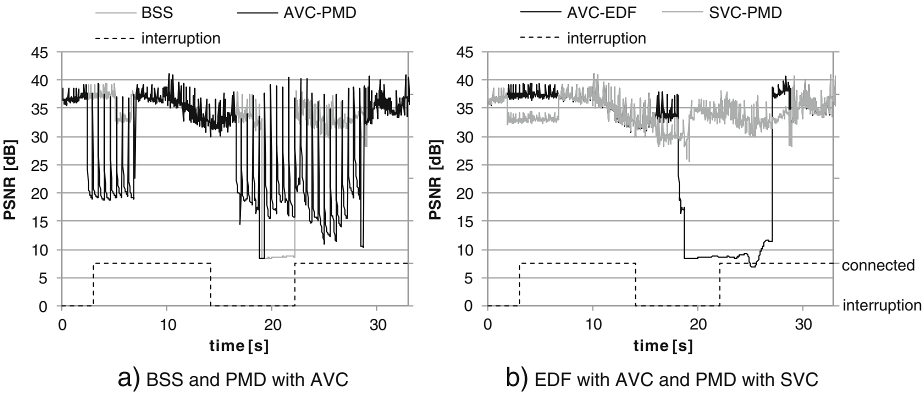



**Fig. 9** Two interruption examples over time with the resulting quality as PSNR

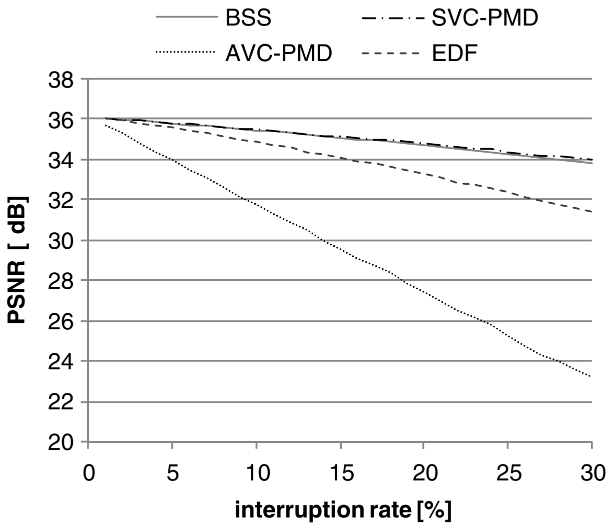
illustrated for the BSS and the PMD with AVC approaches, whereas in Fig. 9 b) the PSNR for AVC-EDF and SVC-PMD is shown. Note that the buffer is full at the beginning of the first interruption. Short interruptions, like the first interruption in the figure, are easily overcome by the traditional approach and, therefore, AVC-EDF may look better than PMD or BSS, since all approaches except for AVC-EDF enter the re-buffering phase immediately after the interruption receiving a lower video quality to fill up the incomplete buffer. Since the PMD prioritizes more important data and stores a bigger amount of it at the cost of storing a smaller amount of less important data, it runs out of the data corresponding to the highest quality faster than the traditional (EDF) and the BSS approaches. Therefore, the PMD client has to perform graceful degradation even for short interruptions, while the AVC-EDF client can keep on playing back the data without any problem. The BSS client may also perform graceful degradation and refill the buffer, which is under the threshold of minimum buffered amount of data. However, during longer interruptions, like the second one in Fig. 9, AVC-EDF and BSS clients cannot continue playing back data due to the lack of stored video in the buffer, while the client using the proposed PMD continues playing back the video at a lower quality representations, resulting in a longer continuous playback. However, after the end of the second interruption the BSS client can play back the video at low quality and at the same time refill the pre-buffer, while the AVC-EDF client has to stop the video until the re-buffering phase is completed.

For the BSS case, two AVC streams will be considered. The first one is the AVC stream at *Q3* shown in Fig. 8 and the second one corresponds to the base layer of the SVC stream, depicted as *Q1* in Fig. 8. For BSS the buffer threshold of 5 s was used to perform the switching. For lower values the lower quality stream is sent, whereas for higher pre-buffer fullness the high quality stream is sent.

In Figs. 10 and 11, the performance of the proposed SVC PMD is analyzed for different interruption rates and compared with the three other methods. Figure 10 shows the average PSNR quality over the whole sequence against different interruption rates. The interruption rate value is detailed in the text below. Figure 11 shows the average playable frame rate in frames per second against the interruption rates.

The simulation model is based on a Gilbert-Elliot model, with an interruption state and a good state and a simulation time step of 33 ms. The probability states for this model are selected so that the average interruption time is 5 s and the average time in good state

**Fig. 10** Average PSNR quality for different occurrences of interruptions (*interruption rate*) for the four methods

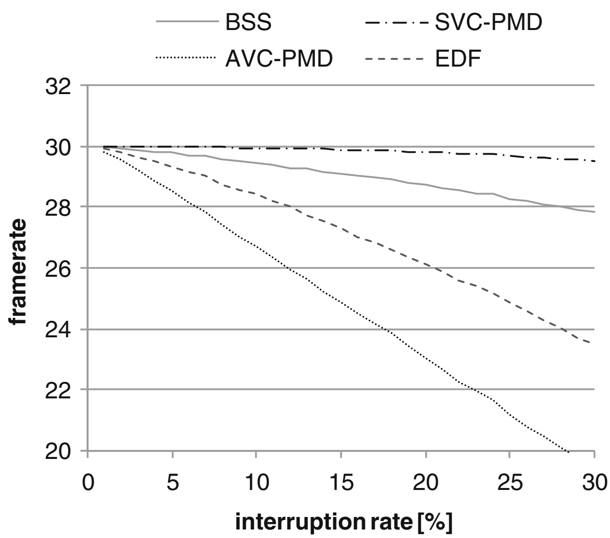



corresponds to a given interruption rate, i.e. the ratio of the average interruption time to the total simulation time as defined in Eq. 9.

$$\text{interruption\_rate} = \frac{E\left[t_{interruption}\right]}{E\left[t_{\text{interruption}}\right] + E\left[t_{good}\right]} \tag{9}$$

The results in Fig. 10 show that SVC outperforms the two H.264/AVC-based EDF and PMD methods. Although SVC imposes a penalty of an approximately 10% higher video rate to achieve the same video quality for the test sequence, this penalty can be reduced using encoder optimizations as proposed in [14], which have not been used in the test stream generation. When BSS is considered in combination with AVC, a noticeable improvement can clearly be noticed. In Fig. 10 there is a slight difference in quality between SVC-PMD and BSS which is negligible.

**Fig. 11** Average playable frame rate for different occurrences of interruptions (*interruption rate*) for the four methods



The results in Fig. 11 show the potential of SVC to overcome link interruptions with almost always the full video frame rate. All tested approaches using AVC introduce a reduction of the play out frame rate, where the AVC-EDF and BSS approaches result in complete playback disruptions depending on the interruption length. BSS however, has the advantage that the re-buffering phase is performed faster after an interruption when compared to AVC-EDF. The latter stops playback if the buffer runs out after an interruption has occured. Playback only resumes after the re-buffering phase, resulting in a long playback disruption which is not acceptable from a service point of view.

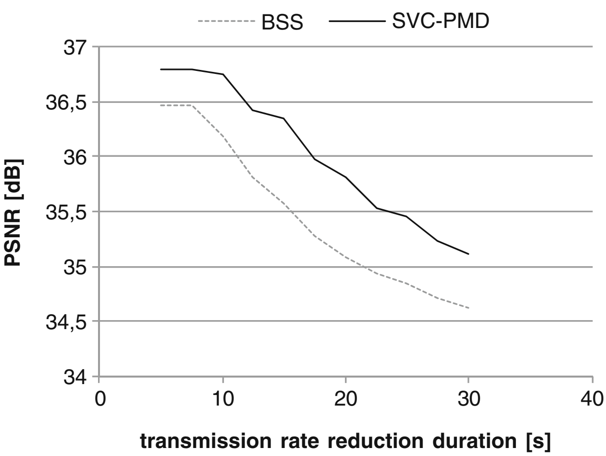
SVC using the PMD algorithm achieves much better results in terms of received video quality and playable frame rate when compared with the other methods over a wide range of operation points. The higher the probability of interruptions is, the more benefits are achieved by using SVC with PMD. There is, however, a price to pay as already mentioned before and shown in Fig. 9: short interruptions that would not affect the quality of the video when AVC-EDF is used would result in quality degradation with the proposed method. In the end, it is up to the service provider to decide whether he wants to be able to guarantee service continuity under all channel conditions or not.

The proposed approach can also be applied in scenarios, where playback is started with a minimum buffer that allows for playing the minimum quality. This would result in a streaming approach with nearly immediate playback while increasing the pre-buffer as well as the quality during playback. Thus the approach also facilitates reduced start-up times.

### 4.2 HTTP streaming

HTTP Streaming with PMD in combination with SVC does not only entail the aforementioned benefits related to HTTP caching efficiency or better usage of the network resources, but also enables a faster response to rate variations in the network. It allows for a media adaptation of buffered content that has not been played back yet, by requesting enhancement layers if there is sufficient time and transmission rate. When BSS is used, already buffered data cannot be modified and has to be played, even if there is enough time to download a higher quality representation of the content. This gain for SVC is shown in Fig. 12, where the quality in terms of PSNR is presented during the rate reduction phase.

**Fig. 12** PSNR quality for HTTP rate adaptation during rate reduction



In the simulations a video chunk duration of 5 s and a pre-buffer size of 15 s (3 chunks) were selected. The SVC stream had the same characteristics as illustrated in Fig. 8. We assume that transmission rate variations can be detected by measuring the time needed to download chunks. According to these estimates the switching to the appropriate quality can be performed. To simulate the BSS case, single layer AVC streams were encoded with the same rates as used for the SVC stream qualities. Due to the SVC efficiency penalty the AVC stream has approximately 0.8 dB higher quality in average. The results presented in Fig. 12 were averaged over different transmission rate reduction parameters: The transmission rate was reduced to values from 40% to 80% of the video rate using a 5% step size. For simulation purposes, the timeline was subdivided into 100 ms intervals. The starting time of the rate reduction phase was simulated for all the possible values within the chunk download interval. Afterwards, an average was performed over all the values for all the possible parameters. Furthermore, the duration of the rate reduction was varied from 5 to 30 s and the average PSNR was calculated as mentioned before.

The curves in Figure 12 show for each possible duration of a transmission rate reduction event, the average PSNR calculated as mentioned above, by averaging all the different values for the transmission rate reduction and start of the transmission rate reduction event. The results for HTTP streaming show the average PSNR value during phases of rate reduction. It can be seen that, independent of the rate reduction duration, SVC in combination with PMD outperforms AVC with the BSS approach, due to the faster response times and the possibility of adding enhancement layer qualities to already buffered chunks.

## 5 Conclusion

We presented Priority-based Media Delivery in combination with SVC as a method of optimizing the playback quality in both RTP-based and HTTP-based streaming scenarios where link interruptions and bitrate variations occur, as typical in mobile networks. Although SVC imposes a coding efficiency penalty, the results show that SVC with PMD allows overcoming of longer interruptions than traditional streaming, which makes it suitable for scenarios with constrained bitrates, potential link interruptions and transmission rate variations.

Furthermore, the application of SVC in HTTP streaming has additional advantages. With SVC, first the responsiveness to network variations can be increased and second the network and caching infrastructure is used more efficiently when compared to adaptive HTTP streaming based on bitstream switching. This could be of great interest to Internet Service and Content Delivery Network Providers that have to distribute many TV channels throughout their networks and typically have to maintain large server and network infrastructures. The combination of SVC with Priority-based Media Delivery enables a longer, smoother playback when interruptions and transmission rate variations affect the channel resulting in an improved user experience.

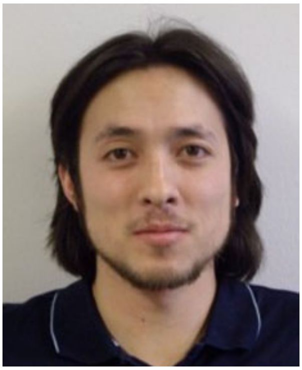
## References


1. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 9), V9.3.0, June 2010
2. Amon P, Rathgen T, Singer D (2007) File format for Scalable Video Coding (SVC). IEEE Trans Circuits Syst Video Technol TCSVT 17(9):1174–1185
3. Baldo N, Horn U, Kampmann M, Hartung F (2004) RTCP feedback based transmission rate control For 3G wireless multimedia streaming. In Proc. of Personal, Indoor and Mobile Radio Communications, PIMRC 1817–1821

4. Färber N, Döhla S, Issing J (2006) Adaptive progressive download based on the MPEG-4 file format. Journal of Zhejiang University Science A, JZUS-A. 7(1):164–111
5. ISO/IEC (2008) Information technology—Coding of audio-visual objects—Part 12: ISO base media file format. ISO/IEC 14496-12:2008 (3rd edition)
6. ITU-T, ISO/IEC, ITU-T recommendation H.264 (2010) Advanced video coding for generic audiovisual services. ISO/IEC 14496-10:2010 Information technology—Coding of audiovisual objects—part 10: Advanced Video Coding, 2010
7. Kritzner J, Horn U, Kampmann M, Sachs J (2004) Priority based packet scheduling with tunable reliability for wireless streaming. In Proc. of High Speed Networks and Multimedia Communications, HSNMC 707–717
8. Renzi D, Amon P, Battista S (2008) Video Content Adaptation Based on SVC and Associated RTP Packet Loss detection and Signaling. International In Proc. of Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 97–100
9. Schierl T, Kampmann M, Wiegand T (2005) H.264/AVC Interleaving for 3G wireless video streaming. In Proc. of IEEE International Conference on Multimedia and Expo, ICME 868–871
10. Schierl T, Wiegand T, Kampmann M (2005) 3GPP compliant adaptive wireless video streaming using H.264/AVC. In Proc. of IEEE International Conference Image Processing, ICIP 3:696–699
11. Schierl T, Johansen S, Hellge C, Stockhammer T, Wiegand T (2007) Distributed Rate-Distortion Optimization for Rateless Coded Scalable Video in Mobile Ad Hoc Networks. In Proc. of IEEE International Conference Image Processing, ICIP 6:497–500
12. Schierl T, Berthelot B, Wiegand T (eds) (2009) ISO/IEC 13818-1:2007/AMD3—Transport of SVC video over ITU-T Rec H.222.0 | ISO/IEC 13818-1
13. Schierl T, Grüneberg K, Wiegand T (2009) Scalable video coding over RTP and MPEG-2 transport stream in broadcast and IPTV channels. IEEE Wireless Communications Magazine 16(5):64–71
14. Schwarz H, Wiegand T (2007) Further results for an rd-optimized multi-loop SVC encoder. JVT-W071, JVT Meeting San Jose, CA, USA, ftp://avguest@ftp3.itu.int/jvt-site/2007_04_SanJose/JVT-W071.zip
15. Schwarz H, Marpe D, Wiegand T (2007) Overview of the scalable video coding extension of the H.264/AVC standard. In Proc. of IEEE Transactions on. Circuits and Systems for Video Technology, TCSVT. 17(9):1103–1120
16. Stockhammer T, Walter M, Liebl G (2005) Optimized H. 264-Based Bitstream Switching for Wireless. In Proc. of Video Streaming. Multimedia and Expo, ICME 1396–1399
17. Wang B, Kurose J, Shenoy P, Towsley D (2004) Multimedia streaming via TCP: an analytic performance study. In Proc. of ACM International Conference on Multimedia 908–915
18. Wenger S, Wang Y-K, Schierl T, Eleftheriadis A (eds) (2009) RTP payload format for SVC video. IETF AVT, http://tools.ietf.org/html/draft-ietf-avt-rtp-svc
19. Wien M, Cazoulat R, Graffunder A, Hutter A, Amon P (2007) Real-time system for adaptive video streaming based on SVC. IEEE Transactions on Circuits and Systems for Video Technology TCSVT 17(9):1227–1237

**Thomas Schierl** (Thomas.Schierl@hhi.fraunhofer.de) received his Dipl.-Ing. degree in computer engineering from Berlin University of Technology, Germany, in December 2003. He has been with Fraunhofer Institute for Telecommunications—HHI since 2004. As project manager he is responsible for HHI's multimedia networking activities, and leads various scientific and industry research projects. He has conducted research work on reliable realtime transmission of H.264/MPEG-4 AVC and SVC in mobile point-to-point, point-to-multipoint, and broadcast environments used by 3GPP PSS/MBMS and DVB-H/SH. He has submitted different inputs on real-time streaming to standardization organizations including 3GPP, JVT/MPEG, ISMA, and IETF. Besides the IETF RTP payload format for SVC and the IETF generic SDP signaling for layered codecs, he is co-authoring various IETF inputs. Furthermore, he is one of the editors of the Amendments for SVC and MVC over MPEG-2 TS. In 2007 he visited the group of Prof. Bernd Girod at Stanford University for research activities on application layer multicast using SVC over wired and wireless channels.

**Yago Sánchez De La Fuente** (yago.sanchez@hhi.fraunhofer.de) received his MSc.-Ing. degree in telecommunications engineering from Tecnun-Universidad de Navarra, Spain, in September 2009. In 2008–2009 he carried out his master-thesis on P2P application layer multicast using SVC at the Fraunhofer Institute for Telecommunications—HHI. He is currently working as a researcher at the University of Technology of Berlin in the Image Communication Group of Prof. Thomas Wiegand.



**Ralf Globisch** (Ralf.Globisch@hhi.fraunhofer.de) received his BSc, BSc (Hons) and MSc degrees in Computer Science from the University of Johannesburg, South Africa in 2001, 2002 and 2004 respectively. From 2005 to 2007 he worked in the industry as a software developer. Since 2007 he has been part of the Real-Time Video Coding group at the CSIR Meraka Institute where he worked on the design and implementation of a commercial grade video streaming system. In 2009 he was awarded a one year DAAD scholarship to begin his doctoral studies as a guest researcher at the Image Processing department of the Fraunhofer Institute for Telecommunications, HHI in Berlin, Germany.

**Cornelius Hellge** (cornelius.hellge@hhi.fraunhofer.de) received his Dipl.-Ing. degree in media technology from the Illmenau University of Technology, Germany, in July 2006. He has been with FRAUNHOFER HHI since 2006. Since 2008 he is affiliated with the Technische Universität Berlin and at the same time he is a guest researcher at FRAUNHOFER HHI. He contributed to different standardization activities (e.g., DVB, 3GPP) and has done different research works on reliable real-time transmission of H.264/MPEG-4 AVC and scalable video coding (SVC) in mobile point-to-point, multicast, broadcast and P2P environments. His main interest is the joint use of channel and source coding.

**Thomas Wiegand** (Thomas.Wiegand@hhi.fraunhofer.de) is a professor of electrical engineering at the Technical University of Berlin chairing the Image Communication group, and jointly heads the Image Processing department of the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany. Since 1995 he has been an active participant in standardization for multimedia with successful submissions to ITU-T VCEG, ISO/IEC MPEG, 3GPP, DVB, and IETF. In October 2000 he was appointed Associate Rapporteur of ITU-T VCEG. In December 2001 he was appointed Associate Rapporteur/Co-Chair of the JVT. In February 2002 he was appointed Editor of the H.264/AVC video coding standard and its extensions (FRExt and SVC). In January 2005 he was appointed Co-Chair of MPEG Video. In 1998 he received the SPIE VCIP Best Student Paper Award. In 2004 he received the Fraunhofer Award for outstanding scientific achievements in solving application related problems and the ITG Award of the German Society for Information Technology. Since January 2006 he has been an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology. In 2008 he received the Primetime Emmy Engineering Award that was awarded to the JVT standards committee by the Academy of Television Arts & Sciences for development of the High Profile of H.264/MPEG-4 AVC, for which he served as Associate Rapporteur/Co-Chair, Editor, and technical contributor.