Priority-based Transmission Scheduling for Delivery of Scalable Video Coding over Mobile Channels

Thomas Schierl[1], Yago Sanchez de la Fuente[1], Cornelius Hellge[2,1] and Thomas Wiegand[1,2]

[1]Fraunhofer HHI
Multimedia Networking
Einsteinufer 58
10587 Berlin, Germany
+49 30 31002 227

[2]Technische Universität Berlin
Image Communication Group
Department of Telecommunication Systems
10587 Berlin, Germany
+49 30 314 28777

Abstract

The media, especially the video delivery over mobile channels may be affected by link outages and transmission bit rate variations due to the actual channel condition and the used wireless interface. In this paper we present the use of Priority-based Transmission Scheduling of SVC media data to overcome link outages and reduction in channel bit rate in mobile networks. The Priority-based Transmission Scheduling framework aims to use the same transmission bit rate and the same overall buffer size as a traditionally streaming approach would require. In order to overcome outages and phases with reduced bit rate, a priority scheduling is used to pre-buffer larger amounts of more important data for longer playout than data with less importance for the resulting video playout quality. We compare the use of SVC with Priority-based Transmission Scheduling against H.264/AVC using temporal scalability with Priority-based Transmission Scheduling and H.264/AVC without Priority-based Transmission Scheduling. We show the benefits of using SVC in terms of received quality during outage and re-buffering time. We present a quality optimization approach for the Priority-based Transmission Scheduling and show results for different outage times.

Keywords

H.264/AVC, SVC, Scalable Video Coding, Priority based scheduling, Mobile channels, outages, transmission rate variation.

1. Introduction

The video delivery over cellular mobile channels may be affected by link outages and bit rate variations due to the changing channel condition. These effects mainly depend on the used wireless interface.

In this paper we present the use of H.264/AVC Priority-based Transmission Scheduling (PBTS) for SVC media data to overcome link outages and reduction in channel bit rate in mobile networks. This approach has been already discussed and evaluated for H.264/AVC over 3GPP Packetized Streaming Service (PSS) in [1] and [2].

The Priority-based Transmission Scheduling framework aims to use the same transmission bit rate and the same overall buffer size as a traditionally streaming approach would require. To a traditionally streaming approach, we refer to as earliest deadline first (EDF) transmission scheduling, where each NAL unit is transmitted depending on the decoding time of the video access unit. In order to overcome outages and phases with reduced bit rate, a priority scheduling algorithm is used to pre-buffer larger amounts of more important data for longer playout than data with less importance for the resulting video playout quality. In [1], this was achieved using the temporal scalability feature of H.264/AVC. The approach in [1] aimed to reduce the video frame rate down to a slide show during outage and re-buffering phase after the outage. The priority based transmission requires some means of client buffer occupation as discussed in [1], [2] which is only possible over a point-to-point connection with feedback about the buffer filling level. In the 3GPP PSS standard [4], there are means for a feedback-based transmission rate control [3] which may need to be extended to perform the priority based scheduling.

In this paper, we focus on the use of the new scalability features of H.264 Scalable Video Coding (SVC) [6][7]. Since there is currently an ongoing activity in 3GPP on improved video support including SVC [5], we show the benefits of using SVC instead of H.264/AVC temporal scalability for PBTS to overcome link outages and phases with reduced bit rate. SVC has the handy advantage to allow bit rate reduction using SNR fidelity or spatial scalability instead of relying on pure temporal scalability as shown in [1].

We compare the use of SVC with PBTS against H.264/AVC using temporal scalability with PBTS as shown in [1] and H.264/AVC without PBTS. We show the benefits of using SVC in terms of received quality during outage and re-buffering time, where we take PSNR as a measure for perceived quality. Here, we assume that the video in both cases, for single layer and multi layer, is transmitted at maximum rate equal to the video rate for the SVC case during pre-buffering and transmission phases. We present a quality optimization approach for the PBTS and show results for different outage times for the three different methods.

The remainder of the paper is organized as follows. In the next section we present the PBTS approach. In section 3, we present the model for the PBTS approach using SVC and present the optimization for different buffer outage scenarios. We present the results for our experiments in section 4 and we conclude in section 5.

Mobimedia'09, September 7-9, 2009, London, UK.

The work of Fraunhofer HHI has been funded by the European Commission under contract number FP7-ICT-214063, project SEA.

2. H.264/AVC Priority-based Transmission Scheduling (PBTS)

The idea of Priority-based Transmission Scheduling (PBTS) for H.264 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. This allows for graceful quality degradation during outage or bit rate variation phases. In such cases, the pre-buffering level cannot be kept due to the fact that more data is consumed than new data arrives in the buffer per time interval. In case of having a buffer filled with a priority scheduling algorithm, the high quality data in the buffer runs out earlier than lower qualities (most important data). Compared to the use of earliest deadline first (EDF) transmission scheduling, the highest quality runs out even faster with the PBTS approach. Nevertheless, the priority based scheduling allows for keeping the playout alive during longer outages than in the EDF case. In Figure 1, we show a streaming receiver's priority buffers for three different qualities as also used in [1]. The priority based transmission scheduling achieves, that a first buffer of the lowest quality *Q1* (e.g. intra frames) is built up to a data amount equivalent with a playout time of $t_1+t_2+t_3$. The quality *Q1* corresponds to a constant bit rate r_1. After that a second buffer for *Q2* is built up which pre-buffers a playout time of t_2+t_3, where the constant rate of *Q2* corresponds to r_1+r_2. The same procedure is applied to *Q3* NAL units, where the constant rate of *Q3* corresponds to $r_1+r_2+r_3$ and the buffer is filled up to a playout time 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 buffered data for *Q3* and t_1 and t_2 the additional time buffered from *Q2* to *Q1* and *Q3* to *Q2* respectively.

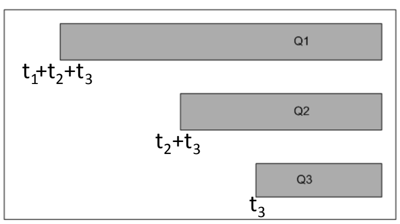

Figure 1 – Streaming receiver buffer for three qualities within the same media using Priority-based Transmission Scheduling (PBTS)

After reaching the playout time values for the quality *Qi* in the priority re-buffering phase, the transmission scheduling algorithm switches to the buffering phase, where each quality *Qi* is filled up with its playout rate r_i. This behavior is shown in Figure 2.

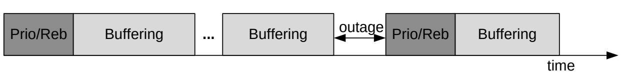

Figure 2 –Priority-based Transmission Scheduling (PBTS) with priority re-buffering (Prio/Reb) and buffering phases

After the occurrence of an outage, the PBTS scheduling algorithm refills the buffer once the connection is back. Note that we assume a mobile service with a maximum transmission bit rate equal to the constant bit rate of the video for SVC. Hence, during a re-buffering phase the quality still needs to be degraded. In the case of using EDF scheduling, during re-buffering the playout has to be stopped. This phase is only started if the buffer is completely empty. However, due to the coding efficiency penalty for SVC, both for H.264/AVC with the PBTS approach and H.264/AVC with the EDF approach the transmission rate is a 10% higher than the video rate, i.e. having this additional 10% rate available for refilling the buffer. Additionally, in the case of using EDF scheduling after an outage even if the re-buffering phase were not started the available 10% of the transmission rate could be used to refill the consumed data from the buffer.

For PBTS, it is required to send data 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 [8].

3. Optimized SVC Priority-based Transmission Scheduling

In this section, we discuss the model for PBTS using SVC with SNR fidelity scalability, H.264/AVC with temporal scalability as well as H.264/AVC EDF.

First we will discuss the re-buffering scenarios on which we focus in this paper. Therefore in our model, we assume that each quality of the media is encoded at a constant bit rate as denoted in Section 2. We further assume that the bit rate for transmission R_t is limited to the bit rate of the video $R= r_1+r_2+r_3$ for the SVC case. Furthermore, the outage time, which is the time interval without transmission rate, is always less than the maximum outage time $t_{outagemax} = t_1+t_2+t_3$, which allows for continuous playout at least at lowest quality for the PBTS cases. For comparison of EDF and PBTS, we use the same total buffer size.

We assume a PBTS with three quality buffers, which are filled to the corresponding playout time values for *Q1*, *Q2* and *Q3*: $t_1+t_2+t_3$, t_2+t_3, and t_3. Using PBTS for the receiver buffer two cases can be distinguished:

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

This is due to the assumption, that available transmission rate for re-buffering and playout after the outage is at maximum 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 out the lower qualities at the same time.

Figure 3 and Figure 4 show the two possibilities A and B. The length of the outage determines whether during the re-buffering phase t_{rebuf} the quality is degraded to the medium quality or the lowest quality.

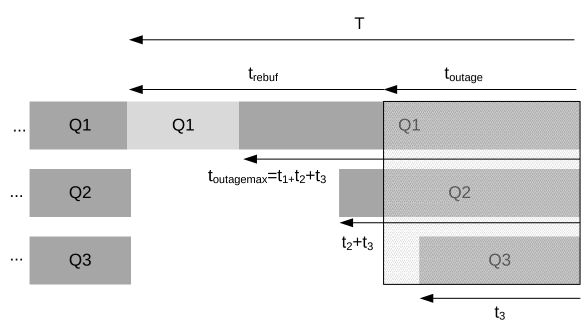

Figure 3 – Impact of outage which requires graceful quality degradation to the lowest quality – case A

Note that the re-buffering process in our model is carried out in the fastest possible way. Although it would be possible to increase the quality already 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 Figure 3, case A is depicted. The here dark grey bars on the right for *Q1*, *Q2* and *Q3* represent the state of the buffer when no outage happens. If an outage occurs (grey shaded box), the data of the buffer is played and thereafter the re-buffering process is started t_{rebuf}. PBTS aims to ensure that at least data for a video quality *Q1* is available. The light grey box represents additional data, necessary to keep on playing at *Q1* until the buffer is fully re-buffered.

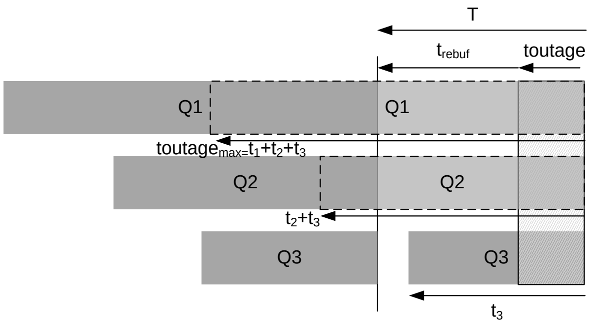

Figure 4 – Impact of outage which requires graceful quality degradation to the medium quality only – case B.

If the outage length is short enough, that the re-buffering could be finished before the *Q2* buffer runs out, case B occurs as depicted in Figure 4. The dashed boxes for *Q1* and *Q2* in Figure 4 show the initial state of the buffer. It can be noticed, 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 outage length, the two possible cases A and B are studied in the remainder of this section.

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

$$T = t_{outage} + t_{rebuf} \le t_2 + t_3 \quad : case \quad B \qquad (1)$$

That means during the outage and the re-buffering phase in case B, still the medium quality *Q2* can be played out. 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 $t_{rebuf} * R_t = r_3 * t_3 + (r_2+r_1) * (t_{outage}+t_{rebuf})$, the amount of re-buffered data can be calculated. Then 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_{outage} and the data to completely re-buffer *Q3*. With the aforementioned conditions, eq. (1) can be transposed to eq. (2), which is the condition for being in case B.

$$t_2 \ge \frac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \quad : case \quad B \qquad (2)$$

The fixed buffer size *B* is calculated in eq. (3), with $t_{outagemax} = t_1+t_2+t_3$ (compare Figure 4).

$$B(t_2, t_3) = t_{outagemax} * r_1 + (t_2 + t_3) * r_2 + t_3 * r_3 \qquad (3)$$

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. Besides

In eq. (4) and eq. (5), the time for re-buffering t_{rebuf} respectively the time of quality degradation *T* are calculated.

$$t_{rebuf}(t_2, t_3) = \begin{cases} \dfrac{t_{outage} * r_1 + B - t_{outagemax} * r_1}{R_t - r_1}, & t_2 < \dfrac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \\ \dfrac{t_{outage} * (r_2 + r_1) + t_3 * r_3}{R_t - r_1 - r_2}, & t_2 \ge \dfrac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \end{cases} \qquad (4)$$

For case A (upper part) in eq. (4), we summed first the amount of data played at lowest quality during the outage (which needs to be re-buffered to $t_{outagemax}$) and second the amount of data, which needs to be re-buffered for the two other qualities to fill up the priority buffer, divided by the bit rate available for re-buffering (while playing quality $q(r_1)$).

For case B (lower part) in eq. (4), we summed first the amount of data played at lowest+medium quality during the outage (which needs to be re-buffered to $t_{outagemax}$ and t_2+t_3 respectively) and second the amount of data, which needs to be re-buffered for the highest quality $q(r_1+r_2+r_3)$ to fill up the priority buffer, divided by the bit rate available for re-buffering (while playing quality $q(r_1)$ and $q(r_1+r_2)$).From eq. (1) and (4), eq. (5) is concluded as follows:

$$T(t_2, t_3) = \begin{cases} \dfrac{t_{outage} * R_t + B - t_{outagemax} * r_1}{R_t - r_1}, & t_2 < \dfrac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \\ \dfrac{t_{outage} * R_t + t_3 * r_3}{R_t - r_2 - r_1}, & t_2 \ge \dfrac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \end{cases} \qquad (5)$$

The aim is to ensure continuous playout during and after the outage, at least at minimum quality $q(r_1)$. Therefore, we set $(t_1+t_2+t_3)$ equal to a fixed $t_{outagemax}$. The lengths t_2+t_3 and t_3 of the buffers for $q(r_1+r_2)$ and $q(r_1+r_2+r_3)$ are the variables for which the optimal solution is searched, since $q(r_1)$ needs to be played continuously during *T*.

For case B, where the length of the outage is small enough to meet the condition in eq. (2), the time *T* after which the playout is back at highest quality depends on t_2 *and* t_3 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 respectively, which fulfill equation 2 with equality. For higher values of t_2, *T* is smaller, thus for the additional time $T(t_{2min}) - T$ the highest quality $q(r_1+r_2+r_3)$ could be play out, in case B. Thus, the playout time with highest quality is $t_3 + (T(t_{2min}) - T)$, compare eq. (6).

For case A, since *T* does not depend on t_2 *and* t_3, the average quality calculation is straight forward following Figure 3. 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).

$$Q = \begin{cases} \dfrac{1}{T} * \left[(T - t_2 - t_3) * q(r_1) + t_2 * q(r_1 + r_2) + t_3 * q(r_1 + r_2 + r_3) \right] \\ \qquad for \quad t_2 < \dfrac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \\ \dfrac{1}{T(t_{2min})} * \left[(T - t_3) * q(r_1 + r_2) + \left(T(t_{2min}) - T + t_3 \right) * q(r_1 + r_2 + r_3) \right] \\ \qquad for \quad t_2 \ge \dfrac{R_t * t_{outage} + (r_1 + r_2 + r_3 - R_t) * t_3}{R_t - r_2 - r_1} \end{cases} \qquad (6)$$

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).

$$\arg\max_{t_2,t_3}(Q(t_2,t_3)) \quad (7)$$

4. Results

For the simulations a buffer of 5 seconds at rate equal to the video rate for SVC was used, which is equivalent to B = 250 KB of the used H.264/AVC and SVC streams as shown in Figure 6, which are a concatenation of the ITU-T test sequences City, Crew, Foreman and Soccer. 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 0.57s. The resolution is QVGA at 30 frames per second. A rate control has been used to keep the bit rate in a +/- 2.5%-window of the average value per IDR+GOP16 picture chunk. The stream length is about 40 seconds. The bit rate for the H.264/AVC stream is 370 kbps and the bit rate for 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 GOP 16 + a preceding IDR per chunk allows a temporal scalability with up to six levels for the H.264/AVC stream.

The distribution function for the outages considered for the evaluations is a uniform distribution for outages between 0 and 10 seconds. In all evaluations, the three methods

I. SVC-PBTS – SVC with three quality levels and PBTS, where after an outage the priority buffer is always re-buffered.

II. AVC-PBTS – H.264/AVC with three quality levels and PBTS, where after an outage the priority buffer is always re-buffered.

III. AVC-EDF – H.264/AVC with EDF, where re-buffering is applied only in case of an empty buffer.

have been used.

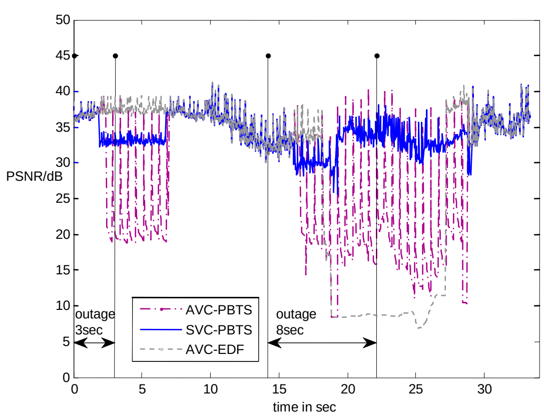

Figure 5 – Two outage examples over time with the resulting quality as PSNR

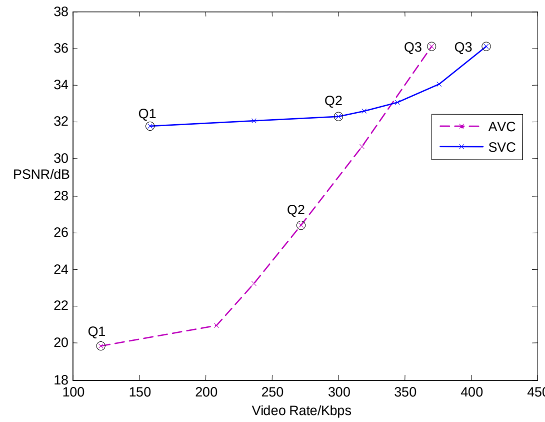

Figure 6 – The test stream qualities - SVC using MGS scalability and H.264/AVC using temporal scalability.

In Figure 5, we show as an example the resulting video quality over time for two different outages of 3 and 8 seconds using the three different methods. Note that at the beginning of the first outage the buffer is full.

Figure 6 shows the three quality levels for the H.264/AVC and the SVC streams used for PBTS, where the SVC rate-distortion points are derived 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 for the selected rate points the use of only two quality buffers does make sense.

In Figure 7 and Figure 8, the performance of the proposed SVC PBTS is analyzed for different outage fractions and compared with the two other methods. Figure 7 shows the average PSNR quality over the whole sequence against different outage fractions. The outage fraction value is detailed in the following. Figure 8 shows the average playable frame rate in frames per second against the outage fractions.

The selected distribution for the intervals for which no outage occurs is also a uniform distribution that fulfills that the relation between the average time in outage and the whole simulation time is equal to the given outage fraction. The results in Figure 7 show that SVC outperforms the two H.264/AVC based methods. Although SVC imposes a penalty of about a 10% higher video rate to achieve the same video quality for the test sequence, this penalty may be further reduced using encoder optimizations as proposed in [9], which have not been used for the test stream generation.

The results in Figure 8 show even better the potentiality of SVC to overcome link outages with the full video frame rate. Both AVC approaches introduce the reduction of frame rate, where the AVC-EDF approach results in complete playout interruptions depending on the outage length, which seems to be not acceptable from a service point of view.

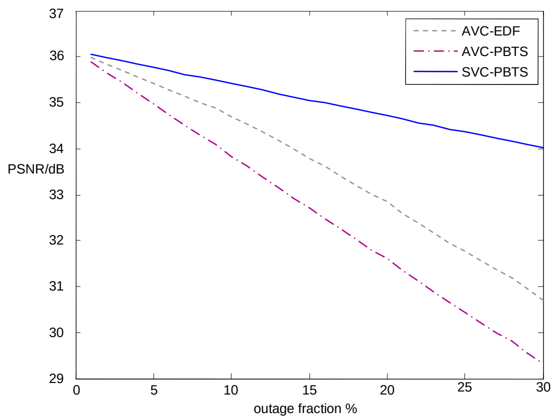

Figure 7 – Average PSNR quality for different occurrences of outages (outage fraction) for the three methods

SVC using the PBTS algorithm achieves much better results in terms of received video quality and playable frame rate than with the two other methods over a wide range of operation points. The higher the probability for outages is, the more benefits are achieved by using SVC with PBTS. In the end it is a question of being able to guarantee service continuity under all channel conditions or not being able.

The PBTS approach could be also applied to bit rate variation scenarios as well as to Internet streaming scenarios with transmission rates higher than the video rate. We keep evaluation of such scenarios for future work.

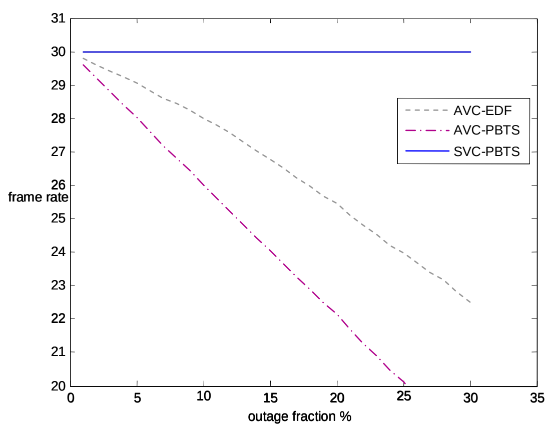

Figure 8 – Average playable frame rate for different occurrences of outages (outage fraction) for the three methods

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

5. Conclusion

We presented an optimization of Priority-based Transmission Scheduling (PBTS) for SVC and H.264/AVC scalability. We showed the benefits of using SVC in point-to-point streaming scenarios with occurrence of link outages as possible in cellular mobile networks. Although SVC imposes a penalty in maximum video quality, the results show that SVC allows overcoming much higher outages than the actual pre-buffer time, which makes it suitable for scenarios with constrained bit rate and a potentiality of link outages as well as transmission rate variations.

6. References

[1] T. Schierl, M. Kampmann, and T. Wiegand: H.264/AVC Interleaving for 3G Wireless Video Streaming, IEEE International Conference on Multimedia and Expo (ICME'05), Amsterdam, The Netherlands, July 2005.

[2] J. Kritzner, U. Horn, M. Kampmann, J. Sachs: Priority Based Packet Scheduling with Tunable Reliability for Wireless Streaming”, HSNMC 2004, Toulouse, France, July 2004.

[3] N. Baldo, U. Horn, M. Kampmann, and F. Hartung: RTCP Feedback Based Transmission Rate Control For 3G Wireless Multimedia Streaming, PIMRC, September 2004

[4] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 8), V8.3.0, June 2009

[5] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Improved Video Support for PSS and MBMS Services (Release 9), April 2009

[6] ITU-T, ISO/IEC, ITU-T recommendation H.264 (2008), Advanced video coding for generic audiovisual services. ISO/IEC 14496-10:2008 Information technology—Coding of audiovisual objects—part 10: Advanced Video Coding, fourth ed., 2008.

[7] H. Schwarz, D. Marpe, and T. Wiegand: Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Trans. Circuits Systems Video Technol., vol. 17, no. 9, Sep. 2007

[8] S. Wenger, Y.-K. Wang, T. Schierl, A. Eleftheriadis (edts.): RTP payload format for SVC video, IETF AVT, March 2009, http://tools.ietf.org/html/draft-ietf-avt-rtp-svc

[9] H. Schwarz and T. Wiegand: Further results for an rd-optimized multi-loop SVC encoder, JVT-W071, JVT Meeting San Jose, CA, USA, April 2007