Link Error Prediction Based Cross-Layer Scheduling for Video Streaming over UMTS

Wolfgang Karner, Olivia Nemethova, Philipp Svoboda, Markus Rupp
Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology, Austria
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: {wkarner, onemeth, psvoboda, mrupp}@nt.tuwien.ac.at

Abstract— **In video streaming services the link error characteristics of the channel have a great impact on the perceived quality. Especially in wireless networks the influence of the channel is significant due to the higher error probability compared to wired links and due to the burstiness of the errors. In our work we take into account both the error characteristics and the characteristics of the video stream for optimizing the quality of the service. For improved scheduling of the individual layer 3 packets of the video stream we are using the predictability of the UMTS link errors. Our proposed scheduling algorithm reduces the error probability for the more important parts of the video data. To deploy our method, only implementation specific changes are needed for supporting the cross-layer communication. The analysis of the error characteristics and the performance evaluation of the proposed method is based on measurements in live UMTS networks. Our results show that significant improvements in the quality of the streamed video can be reached.**

I. INTRODUCTION

Video streaming services over wireless networks are becoming more and more popular. Especially the higher bandwidth and great flexibility offered by UMTS makes video streaming attractive. As video streaming services have been developed for wired links there is currently much effort in research to adapt the applications for the higher error probability and bursty error characteristics of the wireless links. On the other hand, the wireless networks are optimized in order to provide a better quality of e.g. the video service for the customer. For example in [1] a truncated power control is introduced for improving the video quality and for efficient transmission of the video stream. In [2] and [3] (opportunistic) scheduling algorithms are presented which make use of the characteristics of the streamed video data, where in [2] the more important parts of the video stream are transmitted prior to the less important ones in order to ensure more opportunities for retransmissions in case of error and in [3] the priority-based scheduling exploits the diversity gains embedded in the channel variations when having more than one stream. Another approach is shown in [4], where a prediction of the link errors is used in connection with call admission control and scheduling algorithms for avoiding the system of being overloaded and thus improving the quality of the services with higher priority.

In our work we are using the ability of predicting the UMTS link errors for improved scheduling of the individual layer 3 packets of the video stream. The prediction of the UMTS link errors is based on the analysis of the UMTS DCH (Dedicated Channel) error characteristics obtained from measurements in live UMTS networks in Vienna, Austria [5].

Unlike in [1], where transmission is stopped in times of bad channel quality, our scheduling algorithm makes use of all the available bandwidth but delays the packets with higher priority to a position where least error probability is predicted. Our method allows the improvement of the quality of the streamed video without reducing the quality of other services [4].

In a video stream, the I frames (not using the motion prediction) are more important than the P and B frames, since they refresh the stream. If an error occurs in an I frame, it propagates over the whole group of pictures (GOP) up to the next I frame in worst case. To prevent possible error propagation beyond an I frame, usually spatial error concealment is used for the I frames, having lower performance than the temporal error concealment which is used for P frames.

There have already been proposed several approaches for a stronger protection of the more important information in the video stream [6] by means of unequal error protection. Our proposal does not change the standard method of encoding the video stream, it only reduces the probability of an error of its more important parts. In this article we consider only the I frames as the more important information compared to the P frames and no B frames are used. The method can be enhanced to consider more priority levels for example in connection with data partitioning. To deploy our method, only implementation specific changes are needed for supporting the cross-layer communication - no changes in specifications are required.

In this work we analyze the benefits of the proposed scheduling method compared to the usual in-order scheduling based on traces from real UMTS measurements and by using the H.264 video codec [7]. Our results show that we can reach significant improvement in the quality of the streamed video.

This document is organized as follows. In Section II the protocol stack for streaming video over the packet switched (PS) domain of UMTS is introduced. Section III presents the proposed method for predicting the link errors. In Section IV the resulting cross-layer scheduling algorithm and the system requirements are explained. The quality enhancements achieved by the proposed algorithm are demonstrated in Section V for the video stream encoded by H.264 video codec.

Section VI provides a summary and conclusions.

II. Protocol Architecture

For transmission of a video stream over the UMTS network the following procedure of packetization has to be performed. Each frame of the video is first subdivided into smaller parts (slices) which then are encoded. Encoded video slices are encapsulated into RTP (Real Time Protocol) packets and Fig. 1 shows how the RTP packets are further processed by underlaying protocol layers [8], [9]. Each RTP packet is encapsulated

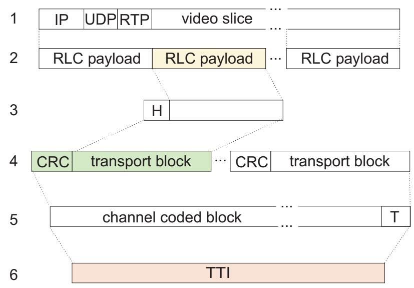

Fig. 1. Packetization example of a video slice for transmission over UMTS radio interface: 1 - IP packet, 2 - RLC segmentation, 3 - RLC and MAC header addition, 4 - CRC addition, 5 - transport block concatenation and channel coding (with tailing bits addition), 6 - after rate matching and interleaving over one TTI.

into a UDP and further on into an IP packet. Then the UTRAN (UMTS Terrestrial Radio Access Network) RLC (Radio Link Control) layer performs segmentation of the IP packets and adds an RLC header. After the mapping of the packets onto the transport channel, performed by the MAC (Medium Access Control) layer, the RLC payload in connection with the header becomes a transport block (trbk). For packet switched bearers the RLC of UTRAN can work in acknowledged mode (AM) allowing RLC packet retransmissions or in unacknowledged mode (UM) allowing only the error detection but no feedback.

After attaching CRC bits [10] to the transport blocks, these are segmented/concatenated into code blocks and the bitstream is encoded by a channel code. For packet oriented applications usually turbo coding is used with a coding rate of $1/3$, which can further be punctured to match the rate with the physical resources.

Before mapping the bitstream onto the physical channels for transmission, 1st interleaving (over one TTI), radio frame segmentation, transport channel multiplexing, physical channel segmentation and 2nd interleaving (over one radio frame) is performed.

The analysis for this work has been performed with a UMTS DCH in DL (Down Link) with 384kbit/s in RLC AM. We have been using IPv4, a RLC payload of 320 bits and 16 bits of CRC. The TTI was 10 ms and there have been transmitted 12 trbks within each TTI by using a spreading factor of 8.

III. Link Error Prediction

Errors in wireless links tend to have a certain bursty nature. That means errors are grouped together to so called bursts with longer error-free gaps in-between. Especially in the UTRAN, when the outer loop power control is built like proposed in [11] or [12], the error bursts occur with a certain periodicity. This property is quite distinctive, particularly in static scenarios (like shown in [5]) and can be used for predicting the errors, as shown later in this section. The analysis of the error characteristics of the UMTS link is based on measurements in live UMTS networks in the city center of Vienna, Austria. More details about the measurements are presented in [5].

In Fig. 2 the empirical CDFs of the gaplengths (the number of error free trbks between two erroneous trbks), measured at three different locations in the live UMTS network, are shown. We can observe two main regions of occurence of gaplengths: the short gaps with ≤ 12 trbks and the long gaps with > 500 trbks. Between these two main parts there is only a negligible small probability of having a gap. Thus, if we wait a time interval larger than the maximum of the short gaps after the last erroneous trbk, then the next possible gaplength is the minimum of the long gaps. Therefore, there is an interval of about 400 trbks with very small probability of containing an error.

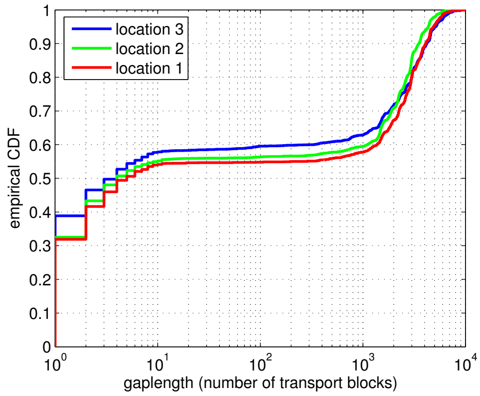

Fig. 2. Statistics of gaplengths based on measurements.

IV. Cross-Layer Scheduling

A. Proposed Algorithm

As shown in the previous section, there are predictable intervals with low transmission error probability. This property of the link errors can be used for a better protection of the more important information on the wireless link. The goal of our proposed scheduling algorithm is to transmit the highest priority packets at a time instant where the smallest error probability is predicted. As mentioned earlier in this document, we consider packets containing I frames as packets with high priority compared to packets containing P frames (no B frames are used in this work).

A schematic illustration of the function of the proposed scheduling algorithm is shown in Fig. 3 where in the upper part you can see the time series of the received trbks, with the erroneous trbks marked. The I packets in Fig. 3 are the layer 3 packets containing parts of I frames and the P packets contain parts of P frames. Due to the fact that we are able to predict intervals with lower error probability after the occurrence of an erroneous transport block, the scheduling algorithm tries to map the I packets onto trbks which are to be transmitted within such a predicted interval. This is done by delaying the I packets by a time d_{min} after the last erroneous trbk (if there is an erroneous trbk within d_{min} the counter for d_{min} is reset) and transmitting the P packets in the meantime. The delay d_{min} should include the maximum of the short gaplengths (as explained in Section III) and some additional time for the feedback delay, as the scheduler at the transmitting station must be aware of the error state of the trbks at the receiving terminal.

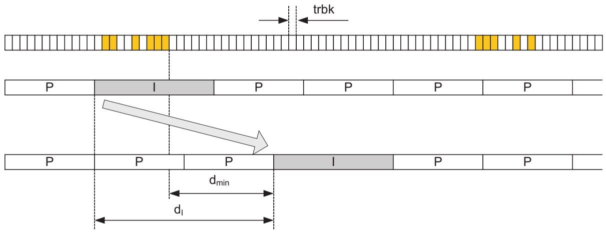

Fig. 3. Schematic illustration of the proposed scheduling algorithm

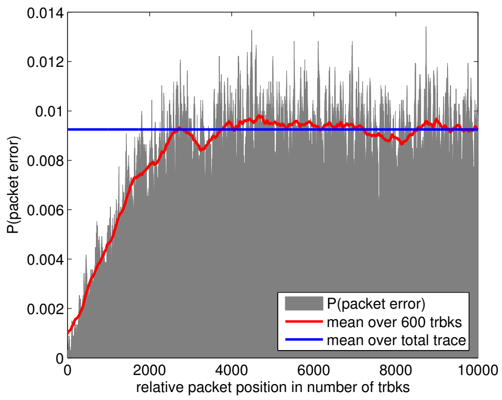

Fig. 4. Packet error probability vs. relative position to d_{min} (37 trbks) after the last erroneous trbk, two-layer model, location 1.

Fig. 4 shows the reduction in packet error probability when transmitting layer 3 packets with a size of 720 bytes (=18 trbks) with delays to the last erroneous trbks larger than a d_{min} of 37 trbks ($\geq$3TTIs=30ms). The packet size of 720 Bytes was selected as it is a typical size for layer 3 video streaming packets. We can observe that at d_{min} we are reaching the lowest packet error probability with $<$0.1%, going up to the total mean (over the whole trace) of 0.925% at a relative position to d_{min} of 3000 trbks (=2.5s). The simulations for Fig. 4 have been performed by using a trace, generated with the measurement based error model out of [5]. We have been using the model based trace instead of the original measured trace in this case in order to achieve better statistics.

Our method causes an additional transmission delay (d_I) for the I packets and thus also for the I frames. However, such delay would not cause any deterioration to the quality as long as it remains within the storage capacity of the playout buffer at the receiving terminal. In Fig. 5 the resulting I-frame transmission delay in number of trbks (12trbks=10ms) is shown for a real video streaming sequence (‘soccer’ as specified in the following section), simulated for the measured

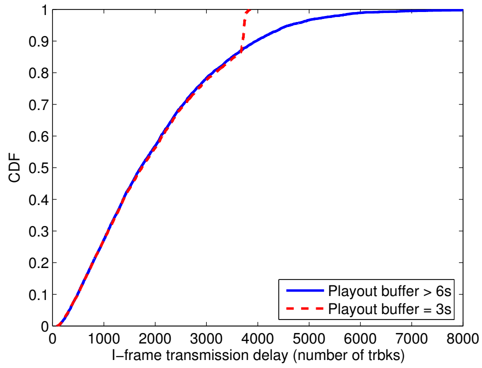

Fig. 5. Resulting I-frame transmission delay in number of transport blocks, ‘soccer’, location 1.

trace at location 1. We can see that without a limitation, the maximum resulting transmission delay for the I frames is about 7000 trbks ($\cong$6s). Thus a playout buffer of 6s - which is a common value for video streaming services - would be sufficient for a full utilization of the proposed method. Current video streaming applications usually use 5-20s playout buffer (pre-roll buffer) to cope with the channel fluctuations and the inherent variable-bit-rate nature of coded video sequences [2]. When limiting the playout buffer to 3s, our method still can be used successfully in 85% of the cases.

B. System Requirements

Our proposed scheduling algorithm can be applied in UMTS networks without changes to standards. What is needed at the scheduler (transmitting side) is the information about the error status of the trbks at the receiving terminal. This can be reached by using the RLC AM mode with the maximum number of retransmissions set to zero. Thus, in UMTS UL (Up-Link) all the necessary information for error prediction and scheduling is already available at the mobile terminal. The only required modification would be to forward both the content of the layer 3 packet (I or P) and the error status of the received trbks in the protocol stack. In the UMTS DL (Down-Link) the scheduler would be situated at the RNC (Radio Network Controller). As there is no IP layer available in the protocol stack of the RNC, the content information of the layer 3 packets would have to be extracted out of the payload, included within the header or sent on an extra link in the UMTS core network to the RNC.

The proposed method can also be applied when allowing a higher number of retransmissions in RLC AM. Of course, with more retransmissions in the RLC layer the gain of our scheduling algorithm becomes smaller as more retransmissions would result in a lower error probability in the link.

V. Transmission of H.264 Encoded Video

To test the efficiency of the proposed method we simulated the transmission of a video over a UMTS network. The UMTS network was emulated by means of an error trace obtained by measurements. We chose H.264 codec to encode the video as it is the newest video encoding standard, optimized for the transmission of video over error-prone environments.

A. Experimental Setup

For our experiments we used Joint Model H.264 encoder and decoder [13] which we adapted to our needs by introducing the following additions:

- H.264 encoder outputs the IP packet trace. The IP trace captures for each IP packet its size, the type of the encapsulated slice and the error flag set to zero (no error).
- H.264 decoder uses an IP packet trace with modified error flags as input. It decodes the error-free slices and conceals the slices corresponding to the erroneous IP packets (error flag set to one).
- We use temporal error concealment for the inter-predicted packets and spatial error concealment for the intra-predicted packets.

The simulation starts by encoding the video sequence. The sizes and types of the IP packets together with the IP error trace obtained from the measurements are fed into the cross-layer scheduler working as described in Section IV. The scheduler performs the rescheduling of the IP packets based on the probability of the error and constrained by the size of a playout buffer. It outputs two IP packet traces with error flag set to one in the case of an error. The first trace is the one corresponding to the in-order scheduling of IP packets; the second one to the cross-layer scheduling. The decoding is performed separately for these two traces to enable the comparison.

We chose two different video sequences with QCIF resolution (144×176 pixel): the well-known 'foreman' video sequence (400 frames) and the 'soccer' sequence (11000 frames) containing a soccer match with several scene cuts. We encoded both sequences using I and P frames only. We chose the slicing mode two with 700 bytes per slice without data partitioning and 15 frames per second. The quantization parameter was set to 25 so that we finally obtained a video stream with an average bit rate of 300 kbit/s.

The error trace was taken from the measurements performed in a live UMTS network in Vienna, Austria [5]. To obtain reliable results, the video was decoded several times reusing the whole measured trace (approx. 1 hour) for ten times resulting in almost 10 hours of video stream.

B. Results

To evaluate the improvement of the end-to-end video quality, we use the peak to signal-to-noise ratio of the luminance component (Y-PSNR) given for the nth luminance frame $\mathbf{Y}_n$ by

$$\text{Y-PSNR}(n) = 10 \cdot \log_{10} \frac{255^2}{\text{MSE}(n)}, \quad (1)$$

$$\text{MSE}(n) = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} [\mathbf{Y}_n(i,j) - \mathbf{F}_n(i,j)]^2, \quad (2)$$

where $\text{MSE}(n)$ denotes the mean square error of the nth luminance frame $\mathbf{Y}_n$ compared to the luminance frame $\mathbf{F}_n$ of the reference sequence. The resolution of the frame is $N \times M$, indexes i and j address particular luminance values within the

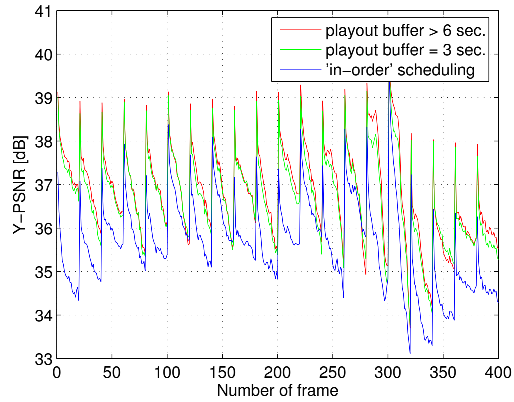

Fig. 6. Average Y-PSNR over the frame number of the 'foreman' sequence.

frame. As a reference sequence we used the *non-compressed* original (non-degraded) sequence.

In Fig. 6, the averaged Y-PSNR per frame of the 'foreman' sequence is shown. For the encoding of the 'foreman' sequence we set the I frame frequency of 20. Even with the small packet loss probability of 0.77% we obtain the average Y-PSNR improvement of more than 1 dB per frame without limiting the playout buffer. When interpreting Fig. 6 we have to be aware of the fact that all the correct frames are influencing the results. Thus, the effect of our cross-layer scheduler on the end-to-end quality is better visualized in Fig. 7 for the 'foreman' sequence. The histogram shows that the number of frames with lower Y-PSNRs is reduced and the number of frames with higher Y-PSNRs increases. Note that the figure only contains the lower Y-PSNR range. In the higher range, there is a peak for the Y-PSNRs corresponding to the error-free frames.

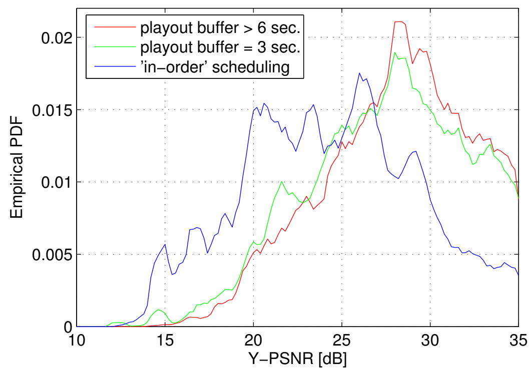

Fig. 7. Empirical PDF of lower Y-PSNRs per frame for the 'foreman' sequence.

The 'soccer' sequence was encoded with I-frame frequency of 75. The lower I frame frequency better matches to the link error periodicity. Thus, the lower amount of I frames lowers the probability of having an error in such I frame (after applied scheduling) due to the fact that efficient scheduling of all packets belonging to one I frame can be performed. In Fig. 8 the average Y-PSNR over the GOP can be seen for the erroneous GOPs. The error propagation is considerably suppressed already for the cross-layer scheduling with three

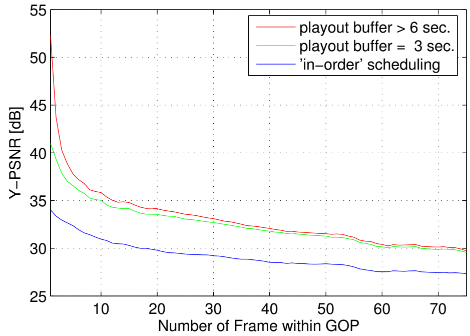

Fig. 8. Error propagation: Average Y-PSNR over the 75 frames long GOP of the 'soccer' sequence. (The averaging was only performed over the erroneous GOPs.)

second playout buffer. The Y-PSNR results would improve even more if the encoder was set to insert an I frame at every scene cut. At present the temporal error concealment used for the P frames in scene cut situations causes visible artifacts. Another possibility is to give also a higher priority to the large P frames as it is probable that they contain a new scene.

To see the benefit of the proposed scheduling algorithm without assuming a particular error concealment method, we define the frame error improvement R as

$$R = \frac{N_{\text{err}} - N_{\text{err}}^{(\text{new})}}{N_{\text{err}}} \cdot 100 \quad [\%], \tag{3}$$

where $N_{\text{err}}^{(\text{new})}$ is the number of erroneous frames after applying the improved scheduling and N_{err} is the number of erroneous frames for the case of 'in-order' scheduling. In Table I, the frame error improvement can be seen separately for I and P frames, for both the 'foreman' and the 'soccer' sequences. The difference between the 'foreman' and the 'soccer' sequence is caused by the lower key frame rate in the 'soccer' case which allows more efficient scheduling, as already explained. In case of a limited playout buffer to 3s, the I packets exceeding the maximum transmission delay have to be transmitted immediately and thus cannot gain of the predicted intervals and are experiencing the total mean (over the whole trace) error probability.

Video sequence	**Scheduling**	R_I [%]	R_P [%]
foreman	PB $\geq$ 6 sec	75.8	3.90
foreman	PB = 3 sec	39.5	2.43
soccer	PB $\geq$ 6 sec	83.4	4.20
soccer	PB = 5 sec	81.8	4.09
soccer	PB = 3 sec	54.0	2.62

TABLE I

FRAME ERROR IMPROVEMENT FOR THE 'FOREMAN' AND THE 'SOCCER' VIDEO SEQUENCES AND DIFFERENT PLAYOUT BUFFER (PB) SETTINGS.

VI. SUMMARY AND CONCLUSIONS

In this document we present the analysis of a scheduling algorithm for UMTS DCH which considers the error characteristics of the link layer as well as the application properties. Our proposed scheduling method makes use of the predictability of the link errors for transmitting layer 3 packets with higher priority at time intervals where the least error probability is expected. In this work we show that significant improvements can be reached in the quality of H.264 video streaming where the layer 3 packets containing parts of I frames are considered as higher priority compared to packets containing parts of P frames. Even with the small packet loss probability of 0.77% and applying spatial and temporal error concealment, our method obtains an average Y-PSNR improvement of more than 1 dB per frame with a playout buffer of 6s and only a little bit less by limiting the playout buffer to 3s. Furthermore, we have shown that our scheduling algorithm is capable of reducing the number of frames with very bad Y-PSNR significantly. In this work we have only considered I frames (high priority) and P frames (low priority) in the video stream. The achieved gain in the quality of the video stream could be even higher when considering more priority levels e.g. medium priority for motion vectors or P frames containing scene-cuts. Applying the proposed cross-layer scheduling algorithm does not require changes to the UMTS specifications.

ACKNOWLEDGEMENTS

We thank mobilkom austria AG&CoKG for technical and financial support of this work. The views expressed in this paper are those of the authors and do not necessarily reflect the views within mobilkom austria AG&CoKG.

REFERENCES

[1] E. Cianca, F.H.P. Fitzek, M. DeSanctis, M. Bonanno, R. Prasad, M. Ruggieri, "Improving Performance for Streaming Video Services over CDMA-Based Wireless Networks," in Proc. of International Symposium on Wireless Personal Multimedia Communications (WPMC 2004), Padova, Italy, Sep. 2004.

[2] S.H. Kang, A. Zakhor, "Packet Scheduling Algorithm for Wireless Video Streaming," in Proc. of 12th Intl. Packetvideo Workshop 2002 (PV 2002), Pittsburgh PA, USA, 2002.

[3] R.S. Tupelly, J. Zhang, E.K.P. Chong, "Opportunistic Scheduling for Streaming Video in Wireless Networks," in Proc. of the 2003 Conference on Information Sciences and Systems, Johns Hopkins University, Baltimore, MD, March 2003.

[4] P. Koutsakis, "Scheduling and Call Admission Control for Burst-Error Wireless Channels," 10th IEEE Symposium on Computers and Communications 2005 (ISCC 2005), 2005.

[5] W. Karner, M. Rupp, "Measurement based Analysis and Modelling of UMTS DCH Error Characteristics for Static Scenarios," in Proc. of 8th International Symposium on DSP and Communication Systems 2005 (DSPCS'2005), Sunshine Coast, Australia, December 2005.

[6] M.T. Sun, A.R. Reibman, "Compressed Video over Networks," Signal Processing and Communications Series, Marcel Dekker Inc., New York, 2001.

[7] T. Wiegand, G.J. Sullivan, G. Bjontegaard,A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560–576, July 2003.

[8] 3GPP TR 25.944, "Channel coding and multiplexing examples," v.4.1.0, June 2001.

[9] 3GPP TR 25.322, "Radio Link Control (RLC) protocol specification," v.4.12.0, June 2004.

[10] 3GPP TR 25.212, "Multiplexing and channel coding (FDD)," v.4.6.0, October 2002.

[11] H. Holma and A. Toskala, *WCDMA for UMTS, Radio Access For Third Generation Mobile Communications,* John Wiley & Sons, Ltd., 2004.

[12] A. Sampath, P.S. Kumar, J.M. Holtzman, "On Setting Reverse Link Target SIR in a CDMA System," 47th IEEE Vehicular Technology Conference, vol.2, pp. 929–933, 1997.

[13] H.264/AVC Software Coordination, "Joint Model Software," ver.10.1, available in `http://iphome.hhi.de/suehring/tml/`.