This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Error Prediction Based Redundancy Control for Robust Transmission of Video over Wireless Links

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

***Abstract*—The loss of UDP/IP packets has a high impact on the quality of the low-rate video streaming, since one UDP/IP packet typically represents a considerable part of a picture, which is discarded and concealed if an error was detected by a UDP checksum. In this work we propose a method that allows for the utilization of error-free parts of such UDP packets. The detection of errors is facilitated by the use of CRC (Cyclic Redundancy Check) information of the smaller link layer packets. Resynchronization of the variable length code at the beginning of the link layer packets is provided by using side information. To keep the additional rate small and to have low distortion at the same time, we propose sending the side information depending on the prediction of the channel quality. Experimental results show a considerable improvement in video quality at the receiver.**

I. Introduction

The 3rd Generation Partnership Project (3GPP) standardizing the Universal Mobile Telecommunications System (UMTS) has approved the inclusion of H.264/AVC (Advanced Video Coding) as an optional feature in release 6 of its packet oriented mobile multimedia telephony [1] and streaming services [2] specifications. For the transmission of multimedia services over IP (Internet Protocol) networks, the video stream is encoded and packetized independently of other media types to separate Real Time Protocol (RTP) packets. These packets are then transported inside the User Datagram Protocol (UDP) units. Due to the real-time requirements of the services UDP protocol is used, which does not provide any retransmission mechanism. Despite of error control mechanisms in the lower layers, transmission errors cannot be avoided completely. To facilitate error detection at the receiving entity, each UDP packet is provided with a simple 16 bit checksum. Although the correct parts of the UDP packets could be used, the complete packets are typically discarded if they contain at least one erroneous link layer packet. The missing part of the video stream is subsequently concealed. The reason for this handling is mainly the variable length coding (VLC). After a bit error, VLC may easily 'desynchronize', making the correct distinguishing between the neighboring codewords impossible. The decoding of such desynchronized stream may result in considerable visual impairments, or may become even impossible due to non-existing code-words or too many/few bits left for decoding.

To overcome this problem several solutions have been proposed. In [3], a reversible VLC is proposed, the drawback of which is its lower compression gain resulting in higher total rate. A more scalable alternative to this approach is the insertion of synchronization marks – sequences of bits that cannot occur in an encoded VLC stream (e.g. analyzed in [4]). To cope with the errors that can affect also the 'in-stream' synchronization marks, in [5] an 'out-of-stream' resynchronization is proposed. There, the new codeword start positions are signalized in packets separated from the actual video stream. Moreover, the utilization of link layer CRC (Cyclic Redundancy Check) information is investigated to detect the errors in resynchronized segments. The side information necessary for resynchronization requires a 'preventive' rate increase even in the time intervals with low error probability. Another class of error resilience methods employs an unequal error protection (UEP) [6] paradigm by protecting the more important parts of video stream stronger. Alternatively, content aware scheduling is proposed in [7]. Both methods require the knowledge of the packet content at the scheduler. This information is typically not available (e.g., in UMTS).

In this work we make use of the predictability of link errors (analyzed in [10]). The proposed system transmits side information over the wireless interface according to the probability of the link error. No additional knowledge of the packet content is necessary. We analyze the quality improvement depending on the resulting rate and compare the proposed method to other common error resilience approaches. In our experiments we focus on UMTS and use H.264/AVC for video compression. Nevertheless, the proposed method applies equally to other wireless systems with predictable link quality and to other video codecs using VLC.

This document is organized as follows. In Section II the framework for a prediction of link errors is introduced and the prediction performance is evaluated. Section III presents error resilience techniques reducing the impact of errors on the quality. In Section IV a link prediction based adaptive error resilience method is proposed. The quality enhancements achieved by the proposed algorithm are demonstrated in Section V for video streams encoded by a H.264/AVC video codec. Section VI provides a summary and conclusions.

II. Link Error Prediction

In wireless systems (e.g. UMTS), power control (PC) based on physical link quality is typically used to reduce the effect of

1-4244-0353-7/07/$25.00 ©2007 IEEE

fading and to keep the interference level small. In UMTS, the outer loop PC sets the signal to interference ratio (SIR) target for the inner loop PC. Within the inner loop PC, the transmitter adjusts its output power to keep the SIR target according to the feedback commands (power up/down) received [9]. Such mechanism results in correlated transport block (TB) error statistics at the radio link control (RLC) layer, which allows for predictability of errors (for more details see [10]).

In this work we make use of the link-error predictability. We take the 'Karner' UMTS RLC error model presented in [11], since it fits well the UMTS channel characteristic in static scenarios (motionless receiver, few scatterers' movement) unlike other typically used error models (e.g. Gilbert [12], Fritchman [13]). In the 'Karner' model, the TB errors are modeled by a two-state alternating Weibull renewal process. The model describes the lengths of bursts (sequence of erroneous TBs) and gaps (sequence of error-free TBs). It distinguishes between 'short' gaps occurring in the 'bad channel' state of the model and 'long' gaps occurring between the 'bad channel' states.

In this document the following notation is used for describing the error process. An erroneously received TB is indicated by '1' while '0' means error-free transmission. A positive integer in the exponent determines the number of consecutive erroneous or error-free TBs (e.g. the sequence '000001' can be written as '$0^5 1$'). A gap with length m is defined as the number of 0's between two 1's and

$$f(m) = P(0^m 1|1) \tag{1}$$

for all positive integers m is the PMF (Probability Mass Function) of the gaplengths. The conditional probability $P(B|A)$ means the probability of sequence B, following sequence A. By definition,

$$\sum_{m=0}^{\infty} P(0^m 1|1) = 1 \tag{2}$$

and $P(0^0 1|1) = 0$, as gaps with length zero are not considered for being gaplengths. The CDF of the gaplengths is then defined as

$$F(m) = \mathrm{P}\{X \le m\} = \sum_{k=0}^{m} P(0^k 1|1). \tag{3}$$

The conditional link error probability $P(1|10^m)$ (the error probability conditioned to the number of error-free TBs since the last error) corresponds to the expected failure rate [14] and can be expressed by

$$P(1|10^m) = \frac{f(m)}{1 - F(m)} = \frac{P(0^m 1|1)}{P(0^m|1)}, \tag{4}$$

where $P(0^m 1|1)$ is the PMF (Probability Mass Function) of the gaplengths and $P(0^m|1)$ is their CCDF (Complementary Cumulative Distribution Function), which can also be written as

$$P(0^m|1) = 1 - F(m) = \sum_{k=m}^{\infty} P(0^k 1|1). \tag{5}$$

$P(0^m|1)$ denotes the probability of having a gaplength of at least length m.

We estimate the expected failure rate $P(1|10^m)$ based on the feedback information (e.g. RLC acknowledgements) at the transmitter. However, the feedback information is not available immediately, but after a certain delay d_{FB} after the transmission of each TB. Since the size of the short gaps in the 'Karner' model is about the size of d_{FB}, we can only use the statistic of the long gaps (Weibull distribution) for the error prediction. After inserting the Weibull PDF for $P(0^m 1|1)$ and the Weibull CCDF for $P(0^m|1)$ in (4), we obtain the expected failure rate

$$P(1|10^m)_{Weibull} = \frac{b a^{-b} m^{b-1} e^{-(\frac{m}{a})^b}}{e^{-(\frac{m}{a})^b}} = b a^{-b} m^{b-1}. \tag{6}$$

In this work we estimate $P(1|10^m)$ by using $b = 2.0176$, corresponding to the statistics of the long gaps measured in [11]. With $b \approx 2$, Weibull distribution becomes Rayleigh and the estimated expected failure ratio can be expressed by the much simpler term

$$\widehat{P}(1|10^m) = \frac{2m}{a^2}, \tag{7}$$

with $a = 3350$. Thus, it is linear increasing with the error-free run length m (the number of error-free TBs received so far after the last error). The performance of such an error estimator can be seen in Fig. 1 for different decision thresholds γ. This figure was obtained by applying following 'hard' decision: if $\widehat{P}(1|10^m) \ge \gamma$ an error is predicted. The probability of false error P_f is the conditional probability that $\widehat{P}(1|10^m) \ge \gamma$ if no error occurred. The probability of missed error P_m is the conditional probability that $\widehat{P}(1|10^m) < \gamma$ if an error occurred. The probability of a correct decision $P_c = 1 - [(1 - P_{err})P_f + P_{err}P_m]$ is the probability that an error was predicted and occurred plus the probability that an error was not predicted and did not occur; P_{err} denotes the probability of a link error. As a test channel, traces from static scenario measurements in a live UMTS network were used. The measurement setup is described in [11] in more details. To keep the quality high, low P_m is needed, whereas to keep the rate low, low P_f is desired. Thus, the thresholds around the crossing of P_m and P_f will be relevant for our investigation.

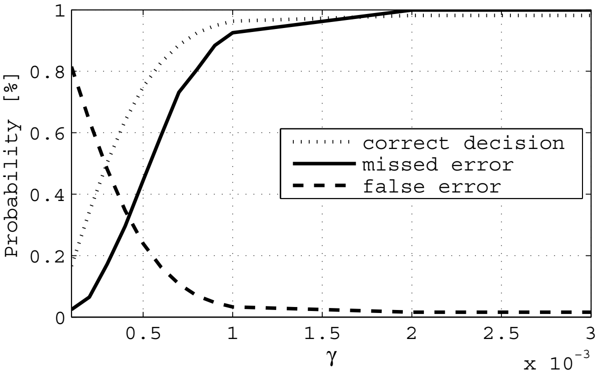

Fig. 1. Prediction performance.

III. Redundancy for Improved Error Detection

A. *Error Detection Mechanism*

In [5] it was proposed to use the CRC information present at the UMTS RLC layer for an improvement of the error detection within a video stream at the application layer. The RLC layer CRC is calculated over each TB. (The TBs are typically much smaller than one IP packet over which the UDP checksum is generated.) The RLC layer CRCs can be used in two ways. The first one utilizes the correctly received TBs until the first error occurs and decodes the information contained there. The rest of the packet has to be discarded due to the possible desynchronization of VLC after an error. The second approach enhances the first method by signaling the side information necessary for the resynchronization of the VLC for each TB. This side information allows for decoding of all correct TBs within the UDP/IP packet even after the error in previous TBs.

Fig. 2 shows an Empirical Cumulative Distribution Function (ECDF) of the frame Y-PSNR for different error handling methods for a 'soccer match' video sequence compressed with quantization parameter (QP) of 26. The two error detection methods (with and without resynchronization, using TB CRC) are compared to the error-free decoding and to the approach where the whole UDP packet is discarded if there was an error detected on the UDP layer. To introduce errors, we took a TB error trace obtained from measurements in the live network. The overall probability of a TB error was 0.266% and the resulting UDP/IP packet error rate was 0.888%. This low error rate is also the reason for the predominantly present high Y-PSNR values. The gain achievable by the additional resynchronization information compared to the decoding until the first error occurrence still is more than 5dB in Y-PSNR for some erroneous frames, which is significant from the user experienced quality point of view. On the other hand, the side information introduces redundancy causing a rate increase even in the cases where no errors occur. The amount of required redundancy and the possibility of reducing it by sending the side information only with higher predicted error probability are discussed in the following.

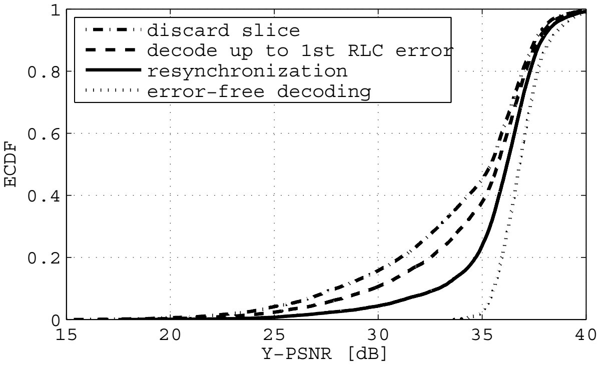

Fig. 2. Quality of video reconstruction at the receiver for different error handling mechanisms.

B. *Resynchronization Information Encoding*

The resynchronization information can be advantageously represented as the position of the start of the first MB contained in the RLC PDU. The maximum length $l_{\max}$ of such *MB Position Indicator* (MBPI) within a q bit large RLC PDU is

$$l_{\max} = \lceil \log_2 q \rceil \,[\text{bits}]. \tag{8}$$

For $q = 320$ bits long RLC PDU and fixed length encoding, $l = l_{\max} = 9$ bits per MBPI codeword will be required.

For an optimal Huffman binary prefix-free code, the average codeword length $\overline{l}$ (for a source s) is bounded by

$$H(s) \leq \overline{l} \leq H(s) + 1, \tag{9}$$

with $H(s) = -\sum_i p(x_i) \log_2 p(x_i)$ being the *entropy* of the source s [18] and $p(x_i)$ is the probability of the occurrence of the ith codeword. Fig. 3 shows the Empirical Probability Mass Function (EPMF) of MBPI values for I and P macroblocks in bits (for QP = 30 and QP = 20), obtained by encoding the 'soccer match' video sequence. The EPMFs for QP values between 20 and 30 (used for mobile communication) would lay in-between. We are only interested on MBs that begin within the RLC PDU. If the length of MB is larger than q, we can signalize this by a chosen additional symbol. Thus, as $p(x_i)$ for the upper bound estimation according to (9), we take the $p(x_i)$ = EPMF(i) in interval $i \in [1, q]$ and an additional point $p(q+1) = \sum_{j=q+1}^{\infty}$ EPMF(j). Finally, we obtain the upper bounded average size $l_P \leq$ 5.5826 bits of MBPI for P MBs and $l_I \leq$ 6.4121 bits for I MBs using the EPMF for $QP = 30$. Similarly, we obtain $l_P \leq$ 6.1992 and $l_I \leq$ 5.2106 bits using the EPMF for $QP = 20$. Note that higher lossless compression gain for I MBPI than for P MBPI is caused by the high probability of the I MB size $> q$. If the probability of the source is not known/varying (e.g. for different contents of video sequences), universal codes are more suitable. By taking an universal code, e.g. Elias-γ code [15] and our EMPF, we obtain $\overline{l}_P = 8.4418$ and $\overline{l}_I = 11.1417$ for $QP = 30$; $\overline{l}_P = 10.4350$ and $\overline{l}_I = 8.4752$ for $QP = 20$. Since the universal coding does not perform better than fixed length coding and the Huffman bound is not easily achievable (the statistic of source varies with content and QP), we prefer

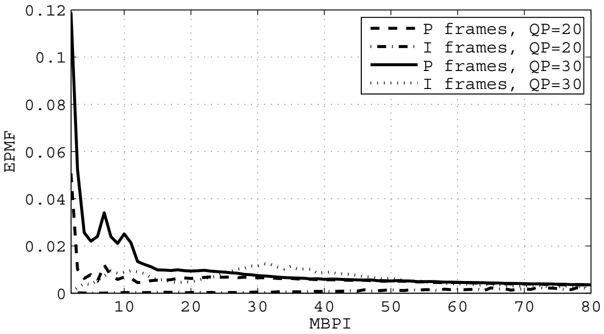

Fig. 3. Distribution of I and P macroblock sizes for QP= 20 and QP= 30.

using of fixed length coding, which also provides higher error resilience.

C. *Embedding of MBPI in Stream*

In H.264/AVC, the parts of encoded video (called slices) are encapsulated into network abstraction layer units (NALUs). Apart from the NALUs containing the video slices, the video stream contains also control NALUs containing parameter sets and other information [17]. To convey the MBPIs to the decoder, they could be appended to the NALUs containing the slices. However, that is not standard-compliant and might break compatibility with other receivers. A better way is to use a reserved NALU type, which will be ignored by receivers not knowing how to handle it. The total rate increase ΔR is given for a slice of size N bits by

$$\Delta R = \frac{\left\lfloor \frac{N}{q} \right\rfloor \cdot l + h}{N} \cdot 100[\%], \tag{10}$$

where h is the length of packet header in bits. In this work, we consider h be 41 bytes, corresponding to the UDP/IP and the NALU header.

IV. Redundancy Control Mechanism

As shown in Section II, a link layer error model can be used for the prediction of errors. Since the MBPI information is used by the decoder only if an error in the radio interface occurs, we can save rate by sending the side information packets only if a higher error probability is predicted. Fig. 4 illustrates the principle of the redundancy control mechanism if applied to UMTS DCH. At the encoder (located on the application server (AS) in the case of video streaming), the video stream is encoded with MBPI side information packets for each video frame and transmitted over the Core Network (CN) to the UMTS Terrestrial Radio Access Network (UTRAN). In UTRAN, scheduling for the radio interface is performed. Based on the link quality feedback, the error probability is estimated. If the probability of error is low, the side information packets are discarded in UTRAN, whereas if the probability of error is high, the side information packets are sent over the radio interface. Note that the side information can be optionally sent protected by additional encoding, or using acknowledged mode that allows retransmissions limited by discard timer (only possible for streaming scenario). The decision about low/high error probability can be based on a fixed threshold γ, or alternatively on an adaptive threshold (considering e.g. the expected distortion). In this work we focus on the fixed threshold approach.

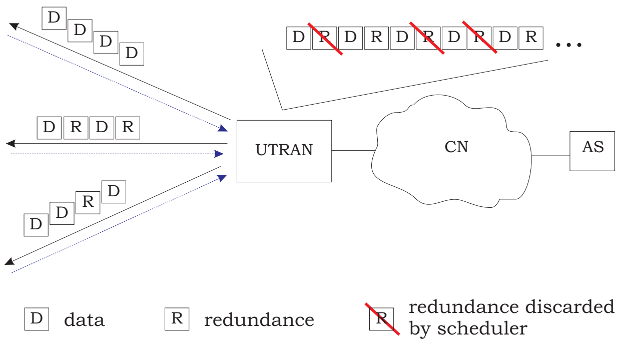

Fig. 4. Redundancy control mechanism for UMTS.

V. Performance Analysis

A. *Experimental Setup*

For our experiments we used H.264/AVC reference software Joint Model (JM) [16] encoder and decoder. The focus is given on the baseline profile (targeting the video conferencing, streaming and especially mobile applications) and thus, we work with Context Adaptive VLC (CAVLC) rather than with Context Adaptive Binary Arithmetic Coding (CABAC) designed mainly for storage applications. Missing parts of the video were concealed by copying the spatially corresponding areas from the previous frames. We implemented RLC segmentation and MBPI generation and adapted JM to accept link error traces as an input. We used the baseline profile encoder settings provided by the configuration file encoder_baseline.cfg included in JM with the following changes:

- We use slicing mode 2 with a limited maximum number of bytes per slice equal to 800. (We chose this value to keep acceptable RTP/UDP/IP overhead)
- The period of an I frame was set to 50.
- We experimented with QP = $\{20, 22, 24, 26, 28, 30\}$ set equally for both I and P frames. There are no B frames supported in the baseline profile.
- The output file mode was set to RTP and we switched off rate-distortion optimization (low-complexity mode) and rate control.

As a test sequence we encoded a soccer match, containing diverse scenes – wide angle shots, close-up shots to the audience, details of the players as well as several shots to the studio discussion. This video sequence was in QCIF resolution (176×144 pixels), 27000 frames long, with a frame rate of 10 fps, which resulted in a duration of 45 minutes.

To predict the errors we assumed a feedback delay $d_{FB} =$ 36 TBs, corresponding to 3 TTIs of 10ms each. We performed our experiments for thresholds $\gamma_1 = 1 \cdot 10^{-4}$, $\gamma_2 = 2 \cdot 10^{-4}$, $\gamma_3 = 3 \cdot 10^{-4}$, $\gamma_4 = 5 \cdot 10^{-4}$, $\gamma_5 = 7 \cdot 10^{-4}$ and $\gamma_6 = 1 \cdot 10^{-3}$.

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

$$\text{Y-PSNR}(n) = 10 \cdot \log_{10} \frac{255^2}{\text{MSE}(n)}, \tag{11}$$

$$\text{MSE}(n) = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} [\mathbf{Y}_n(i,j) - \mathbf{F}_n(i,j)]^2, \tag{12}$$

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

B. Results

Sending the side information necessary for the VLC resynchronization requires additional rate. The average rate increase for different QPs and γ can be seen in Fig. 5 for the MBPIs encoded using fixed length code and encapsulated into H.264/AVC NALU and UDP/IP protocol. The quality improvement achievable by resynchronization is illustrated in Fig. 6 for all investigated thresholds γ. We calculate the improvement as a difference between the frame Y-PSNR of the method using resynchronization with threshold γ and the frame Y-PSNR of the method decoding up to the first TB error. Note that this ECDF is derived by considering all investigated QPs (unlike Fig. 2). In this representation we can see that e.g. with γ_2, in more than 7% of the cases the improvement in quality is higher than 5 dB. Moreover, in 40% of the cases the quality improvement is more than 1.5 dB. The improved frames comprise the frames where error occurred as well as frames to which the error propagated due to temporal prediction. The improvement for the 'send all' strategy is only slightly higher, although it requires more rate. Of course, looking at the Y-PSNR is not fair without considering the additional rate. Therefore, Fig. 7 illustrates the average quality at the decoder normalized by required rate. As a rate we take the average number of bits per frame, that already comprises also the resynchronization information and its header. The averaging was performed over the entire video sequence, containing also the error-free packets. Clearly, the threshold γ_2 provides the best quality for the same rate. The improvement of approximately 0.25 dB can be achieved by using the error prediction when comparing to the full resynchronization (sending all MBPIs). This is a considerable improvement, since the error rate was small and we take into account the increase of the rate caused by signalling the side information. Furthermore, it can be seen, that the resynchronization alone brings approximately 1.5dB of average quality improvement for the same rate. It also demonstrates, that it is possible to reduce the rate at the radio interface significantly while keeping the same quality at the decoder when using the resynchronization information (with and without prediction).

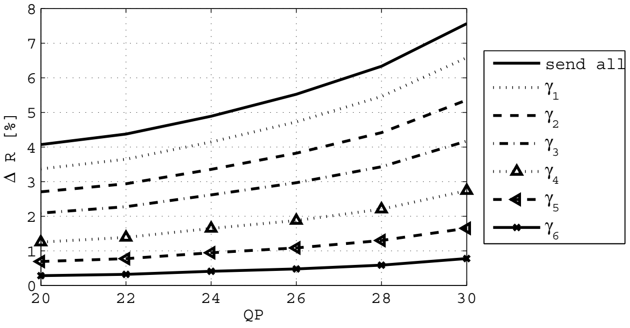

Fig. 5. Average rate increase per frame for chosen quantization parameters (QP) and error prediction thresholds γ_i.

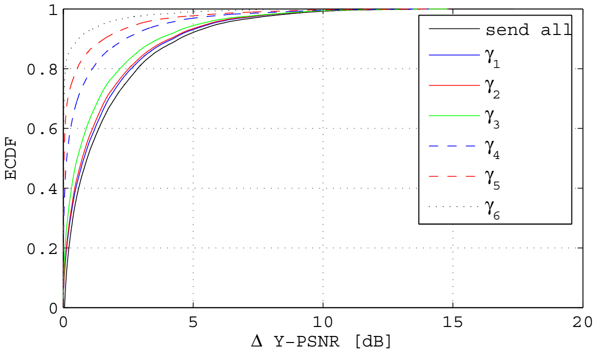

Fig. 6. Quality improvement resulting from resynchronization of VLC, compared to the method decoding up to the first error occurrence.

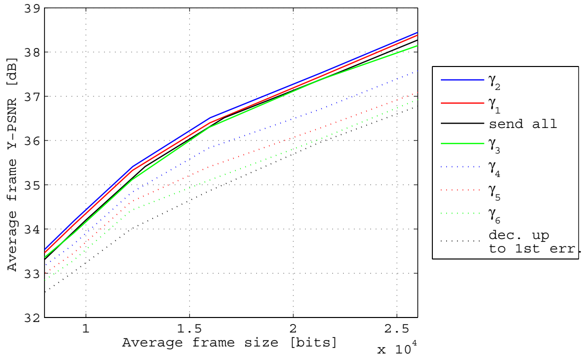

Fig. 7. Average Y-PSNR over the rate. The rate is expressed as average size of frame and side information in bits.

Fig. 8 illustrates the Y-PSNR per GOP (Group of Pictures; a GOP consists of an I frame and all P frames up to the next I frame). This representation provided an example of what quality the user does see over time (average per GOP $\sim$ 5 seconds of video in our case).

The difference between the full resynchronization and resynchronization using prediction (with γ_2, that provides the best quality for the given rate) is in most cases zero. Some small differences are caused by the wrong error predictor decision. It can be seen, that resynchronization improves the quality at the receiver considerably, even for a small packet loss rate in our case. The difference to the typically used slice discard method is up to 10 dB per GOP in average, a little smaller gain (up to 8 dB) can be seen compared to the decoding up to the first error. These differences are significant from the user perception point of view. In our experiments we used a simple temporal error concealment method. Better error concealment methods (e.g., edge aware spatial error concealment after scene change and motion compensated temporal

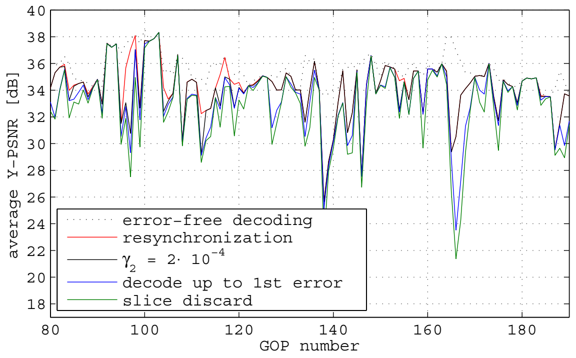

Fig. 8. Average Y-PSNR per GOP over time (GOP number).

error concealment otherwise as proposed in [19]) would surely improve the PSNR. However, even the performance of better error concealment mechanisms is highly dependent on the size of the area to be concealed and its spatial and temporal content. Thus, for the chosen video resolution/frame rate we expect similar gain of the proposed method if used with other error concealment.

The size of a picture area (in MBs) corresponding to one TB depends on the compression ratio of this area; the compression ratio depends on the content of the video sequence (amount and type of the movement, amount of edges) and on the QP. If we wanted to limit the erroneous picture area rather than number of bits, we would need to signal MBPIs on a per-M-macroblocks basis (MBPI sent for each Mth MB). However, additional error detection mechanisms would be needed. Such a mechanism working in connection with a link error predictor is an interesting point for further work.

VI. Conclusions

In this work we investigated the possibility to use correctly received information within the UDP/IP packets containing H.264/AVC encoded video. A method where decoding of erroneously received packets is performed until the first error occurrence is compared to the method where the whole packet is discarded. To detect the position of errors within a UDP/IP packet, link layer packet CRC information is used in the other investigated mechanism. Furthermore, a variable length code resynchronization mechanism is proposed that uses side information to signal the start of a new macroblock in each link layer packet. This allows for utilization of the link layer packets that were correctly received after an error. We analyze the rate increase caused by signaling of such side information. To save rate on the radio interface, we propose to send the side information only if the link error probability exceeds a certain threshold. We present an example how the error probability can be estimated in UMTS and study the possibility to use a low-complexity threshold-based error predictor for rate reduction at the radio interface. The concept can easily be extended to other systems.

Based on the results we can draw several conclusions: The quality at the receiver essentially depends on the error handling. Utilization of the lower layer information improves the quality significantly at no costs. Even higher gains can be obtained for the same rate if side information for resynchronization is provided (up to 10 dB in our scenario). Sending the side information only if the error probability is high, provides higher quality for the same rate compared to the 'sending all' strategy. Of course, the choice of the predictor threshold plays an important role.

Acknowledgements

The authors thank mobilkom austria AG 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.

References

[1] 3GPP, TSG Services and System Aspects, "Packet switched conversational multimedia applications; Default codecs (Rel. 6)," ver. 6.4.0.

[2] 3GPP, TSG Services and System Aspects, "Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Rel. 6)," ver. 6.8.0.

[3] Y. Takishima, M. Wada, H. Murakami, "Reversible Variable Length Codes,"IEEE Trans. on Communicatios, vol. 42, no. 2/3/4, 1994.

[4] A. Kiely, S. Dolinar, M. Klimesh, A. Matache, "Error Containment in Compressed Data Using Sync Markers," Proc. of Int. Conf. on Information Theory, June 2000.

[5] O. Nemethova, W. Karner, A. Al-Moghrabi, M. Rupp, "Cross-Layer Error Detection for H.264 Video over UMTS," in Proc. of Wireless Personal Multimedia Communications (WPMC), Aalborg, Denmark, Sept. 2005.

[6] Q. Qu, Y. Pei, J.W. Modestino, "An Adaptive Motion-Based Unequal Error Protection Approach for Real-Time Video Transport Over Wireless IP Networks," IEEE Trans. on Multimedia, vol. 8, no. 5, Oct. 2006.

[7] J. Chakareski, J. Apostolopoulos, B. Girod, "Low-Complexity Rate-Distortion Optimized Video Streaming," IEEE ICIP, 2004.

[8] W. Karner, O. Nemethova, M. Rupp, "Link Error Prediction Based Cross-Layer Scheduling for Video Streaming over UMTS," 15th IST Mobile & Wireless Communications Summit, Myconos, Greece, Jun. 2006.

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

[10] W. Karner, O. Nemethova, M. Rupp, "Link Error Prediction in Wireless Communication Systems with Quality Based Power Control," Proc. of IEEE Int. Conf. on Communications (ICC), Glasgow, UK, Jun. 2007.

[11] W. Karner, M. Rupp, "Measurement based Analysis and Modelling of UMTS DCH Error Characteristics for Static Scenarios," in Proc. of DSPCS 2005, Sunshine Coast, Australia, Dec. 2005.

[12] E.N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253–1265, Sep. 1960.

[13] B.D. Fritchman, "A Binary Channel Characterization Using Partitioned Markov Chains," IEEE Trans. Information Theory, vol. 13, no. 2, pp. 221-227, Apr. 1967.

[14] A. Papoulis, P.S. Unnikrishna, *Probability, random variables, and stochastic processes,* McGraw-Hill, 2002.

[15] P. Elias,"Universal Codeword Sets and Representations of the Integers," IEEE Trans. on Inf. Theory, vol. 21, no. 2, March 1975.

[16] H.264/AVC Software Coordination, "Joint Model Software," version 11.0, available in http://iphome.hhi.de/suehring/tml/.

[17] I.E.G. Richardson, *H.264 and MPEG-4 Video Compression, Video Coding for the Next-Generation Multimedia,* John Wiley & Sons Ltd., Mar. 2005.

[18] K. Sayood, *Data Compression,* Second Edition, Morgan Kaufmann Publishers, 2000.

[19] O. Nemethova, A. Al-Moghrabi, M. Rupp, "An Adaptive Error Concealment Mechanism for H.264 Encoded Low-Resolution Video Streaming," Proc. of EUSIPCO, Florence, Italy, Sep. 2006.