Published in proc. of International Wireless Summit, 18 - 22 September 2005, Aalborg, Denmark

Cross-Layer Error Detection for H.264 Video over UMTS

Olivia Nemethova, Wolfgang Karner, Ameen Al-Moghrabi and Markus Rupp
Vienna University of Technology, Institute of Communications and RF Engineering
Gusshausstrasse 25/389, 1040 Wien, Austria
{onemeth, wkarner, mrupp}@nt.tuwien.ac.at

Abstract–**Third generation of mobile communication systems brought new packet switched real-time video services. To detect errors in a video packet, parity bits are used at different layers of the protocol stack. If an error occurs, the whole UDP packet is usually discarded. Still, it can happen that the major part of such a packet might have been correct and the detailed information about correctness of the packet segments is known at lower layers. In this article an error detection mechanism using information from lower layers at the decoder is proposed and its benefits and implementation costs are analyzed.**

Key words: RLC CRC, streaming video, error resilience.

1. INTRODUCTION

Universal Mobile Telecommunications System (UMTS) [1] is a 3rd generation mobile system supporting video streaming and conferencing services. H.264 [2] is the newest high compression digital video codec standard, well suited for transmission over bandwidth limited mobile systems. To match the screen of a mobile terminal, a 144×176 pixel resolution (also called QCIF) is used. Real-time video is usually transmitted via User Datagram Protocol (UDP) without any possibility for retransmissions at the transport layer. Each UDP packet contains Cyclic Redundancy Check (CRC) information bits allowing error detection. If the CRC fails, the whole UDP packet is discarded. A UDP packet usually represents a rather large part of the picture and its loss results in considerable visual perceptual quality distortion. However, the UMTS User Equipment (UE) protocol stack provides additional means for error detection at the lower protocol layers. Having the information from the lower layers at the decoder can help to locate the position of the errors and their size more exactly.

The intention of this article is to investigate the possibility of using the information from the lower layers to improve the error detection in the real-time video stream. In Section 2 the protocol stack for streaming video over packet switched (PS) domain of UMTS is introduced. Section 3 presents the proposed mechanisms. In Section 4 detection efficiency is analyzed while in Section 5 the results are shown. Section 6 contains conclusions and some final remarks.

2. PROTOCOL ARCHITECTURE

H.264 allows different types of slicing: slices containing whole frames, slices with constant number of macroblocks, slices with constant number of bytes, interleaved slices or Flexible Macroblock Ordering (FMO). For transmission over packet oriented mobile networks, slices with constant number of bytes are of major interest (e.g. efficiency of protocol mapping - padding). The video slices are then encapsulated into Real Time Protocol (RTP) packets and Figure 1 shows how the RTP packets are further processed by underlaying protocol layers [3, 4].

The RTP header is 12 bytes long. Each RTP packet is encapsulated into a UDP packet, which adds a header with 8 bytes to the RTP packet. If no segmentation is needed, a UDP packet is further encapsulated into an IP packet. The IP header has a size of 20 bytes for IPv4 and a size of 40 bytes for IPv6. Each IP packet is then segmented in the UMTS Terrestrial Radio Access Network (UTRAN) Radio Link Control (RLC) layer and mapped onto the transport channel by the Medium Access Control (MAC) protocol layer. Before the segmentation of an IP packet, the Packet Data Convergence Protocol (PDCP) may perform header compression. In case of a bearer with data rate below 384kbits/s, usually 320 bits (40 bytes) payload within the RLC packets is used; for higher data rates 640 bits long payload may be used. 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 de-

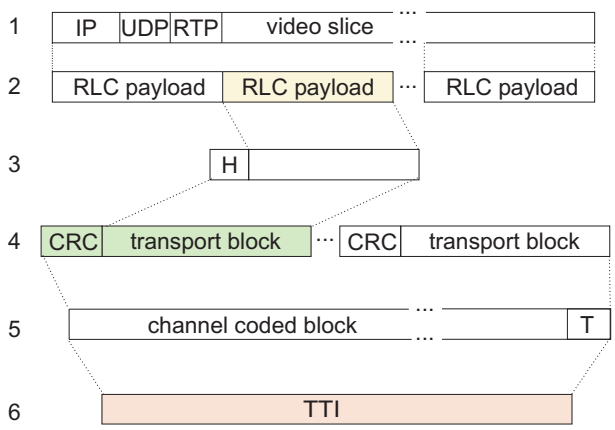

Figure 1: Packetization example of a video slice for transmission over UMTS radio interface: 1 - IP packet, 2 - RLC segmentation, 3 - RLC header addition, 4 - CRC addition, 5 - transport block concatenation, 6 - channel coding and tailing bits addition, 7 - after rate matching interleaving over TTI.

tection but no feedback. The header size for the RLC AM is 16 bits, whereas the RLC UM header contains 8 bits. By adding these RLC headers to the RLC packets each RLC packet becomes a transport block. Every tranport block gets some CRC bits attached where the size of the CRC is configurable and may be 0,8,12,16 or 24 bits [5]. The number of transport blocks which are interleaved over one Transmission Timing Interval (TTI) is defined by the transport format. After the segmentation/concatenation into the code blocks, the bit stream is encoded by a channel code. For packet oriented applications usually turbo code is used with a code rate of 1/3, which can further be punctured to match the rate with the physical resources. The next step is the multiplexing of the transport channels onto the physical channels.

3. DETECTION MECHANISMS

The smallest unit in which an error can be detected (assuming UMTS as an underlaying system), is the Radio Link Control (RLC) packet and not the whole UDP packet as usually assumed [6]. To enable the usage of RLC CRC information, this information needs to be passed from the RLC layer to the application layer (video codec). The change needed for that is implementation specific (does not violate standards) and can be achieved by means of software, depending on the design of the corresponding device (UE). Having the RLC CRC information at the video decoder, the position of the first erroneous RLC packet within the slice can be specified. Furthermore, all correctly received RLC packets before the first erroneous one can be decoded successfully. This method adds neither computational complexity nor data overhead. Due to the desynchronization of the Variable Length Code (VLC) after the first error within the slice, it is not possible to use successive RLC packets although they might have been received correctly. The start of the next VLC code word within the segment of the bitstream, corresponding to such RLC packet payload, is not known. Furthermore, the position of the picture part contained by the RLC packet is not specified. Each slice of the video contains information independent from the other slices and each slice contains a 24 bit long synchronization sequence at the beginning, which allows to stop error propagation caused by Variable Length Code (VLC) desynchronization. To use all correctly received RLC packets, adaptation of the slice size to the RLC packet size is needed. Such step is indeed connected with unacceptable overhead caused by the slice header and above all by the header of RTP/UDP/IP, which is then as long as such a slice itself.

Another possibility is to signal the information needed for resynchronization after every 320 (or 640) bits, which corresponds to the usual size of an RLC packet. In [7] different ways of inserting the resynchronization marks after a particular number of macroblocks were analyzed. However, the problem with resynchronization marks for the RLC packets is slightly different. While in [7] the detected erroneous areas were having the same limited size, a RLC packet will correspond to the areas with different sizes, depending on the compression efficiency of the present content. The resynchronization information can be for instance the position of the start of the first macroblock within each RLC packet.

Let RLC_size be the size of an RLC payload in bits. Then the maximum length m_p of the position indicator can be written as:

$$m_p = \lceil \log_2 RLC\_size \rceil [\text{bits}]. \quad (1)$$

The most efficient way to send the points of resynchronization is via an additional channel out of the actual VLC stream to avoid the errors of the synchronization information (i.e. by unequal error protection). The total overhead to the bitstream is given by:

$$O_{\text{RLC}} = \frac{\left\lfloor \frac{N}{RLC\_size} \right\rfloor \cdot m_p + Header}{N} \cdot 100\%, \quad (2)$$

where N is the size of a slice and $Header$ is the length of the header of the out-of-stream packet containing the position/length indicators. The header can introduce more overhead than the payload if we take again the (RTP)/UDP/IP protocol. This could be solved by using the data partitioning and adding

the position/length indicators to the data partition A. It would require a small addition to the H.264 standard - for instance using the 5 reserved bits to signalize the length of the position/length indicator bit sequence added at the end of the packet.

The overhead needed for the resynchronization marks for various slice sizes can be seen in Figure 2. The picture was obtained without considering the header, thus corresponding to the data partitioning solution. The header would cause a shift on the y-axis only.

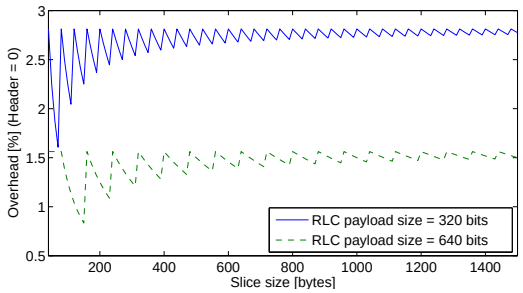

Figure 2: Overhead connected with synchronization marks in each RLC packet.

For the RLC payload size of 320 bits, the overhead is below 3%, for the size of 640 bits, the overhead remains about 1.5%, however also the size of the detectable area is twice as large.

4. DETECTION EFFICIENCY

The efficiency of these methods and their benefits are closely related to the character of errors at the UMTS RLC layer. The distribution of erroneous RLC packets belonging to one IP packet, assuming the video sequence encoded by H.264 with slicing mode 2 (slices with same size in bytes) and a slice size of 700 bytes is shown in Figure 3 for the static scenario and in Figure 4 for the dynamic scenario. The RLC layer error characteristics of the UMTS DCH of the static scenario were obtained by measuring the error rate in a situation, where the mobile terminal was lying on the table without any movement [8]. Changes of the channel conditions were only caused by the changing environment (change of the position of the scatterers, number of users in the cell - interference level). The error characteristics for the scenario with movement (dynamic scenario) was obtained by performing the measurements in a car and a tramway. We obtained the statistics from extensive measurements for static and dynamic scenarios in live UMTS networks in Vienna, Austria. The measurement setup is described in more detail in [8]. The results of the measurements (RLC trace carrying the information if the RLC packet is erroneous) were further mapped onto the IPv4 packets containing H.264 video stream (encoded with [9]) with various sequences, encapsulated into RTP/UDP.

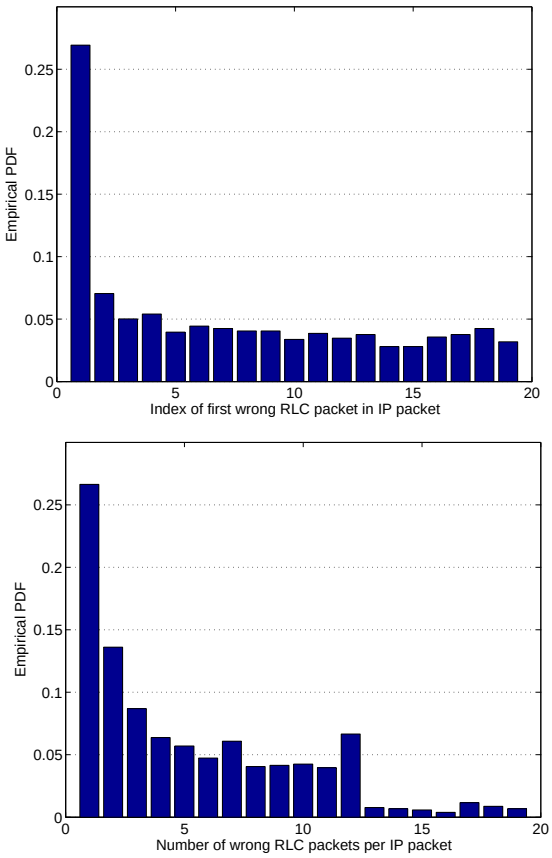

Figure 3: Static scenario: Distribution of (top) index of the first erroneous RLC packet within IP packet, (bottom) number of erroneous RLC packets per IP packet.

The upper diagrams of Figures 3 and 4 are relevant for the method without using any synchronization marks. The peak at the first position (approx. 25% for static and 45% for dynamic scenario) denotes the cases where this method cannot be applied. In the rest of the cases the method is beneficial without any data overhead. Even for the dynamic scenario we can still use this method in more than 50% of the situations.

The lower diagrams of of Figures 3 and 4 are relevant for the method using synchronization marks. The method cannot be applied only if all the RLC packets within an IP packet are erroneous, which corresponds to the last column of the histogram. The method gains in almost 99% of the situations for the static case. The situation is getting worse for the dynamic scenario, where the probability of all RLC packets within an IP packet being erroneous is growing to almost 12%. The peaks at 4, 8 and 12 pack-

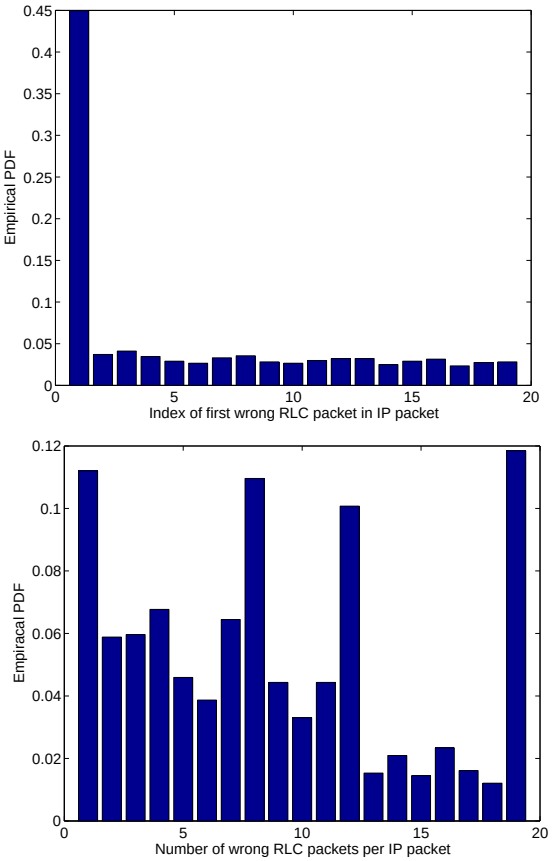

Figure 4: Dynamic scenario: Distribution of (top) index of the first erroneous RLC packet within IP packet, (bottom) number of erroneous RLC packets per IP packet.

ets, visible especially in the dynamic scenario denote the cases with 4, 8 and 12 RLC packets per TTI, which are all erroneous. The bearers used for the measurements were with a datarate of 384, 128 and 64kbits/s.

5. RESULTS

To further test the benefits of the proposed method, we used a Joint Model v7.3 H.264 codec implementation [9], in which we implemented temporal error concealment with boundary matching and weighted averaging spatial concealment [6]. The success of the error concealment depends on the size of the area to be concealed. We performed experiments using the RLC error mask obtained from measurements and mapped it on the foreman IP/UDP/RTP stream. By mapping onto the VLC bits in H.264 decoder we obtained the positions of the first error and decoded the stream up to it. The foreman sequence was encoded with following settings: quantization parameter 28, I frame frequency set to 20, without B frames, slices of 700 bytes.

In Figure 5 the resulting PSNR for the method without synchronization marks is shown depending on the index of the first erroneous RLC packet within an IP packet.

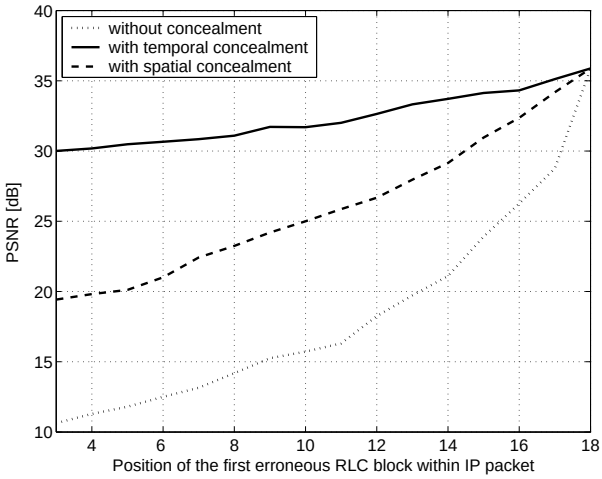

Figure 5: PSNR over the index of the first erroneous RLC packet in IP packet with and without error concealment.

The PSNR was calculated comparing the degraded/concealed sequence to the original non-compressed one. The PSNR curves in the figure are calculated for the detection method without overhead as an average over the erroneous frames and for different positions of the first erroneous RLC packet. The first RLC packet contains RTP/UDP/IP header and the second packet contains the slice header. Without them the slice cannot be detected correctly. The benefit of this method is considerable even if a good error concealment method is used (up to 5dB). In nowadays terminals usually weighted averaging or simple copying from the same position in the last frame is used for error concealment. With those techniques, cross-layer detection shows even higher benefit - i.e. for weighted averaging up to 15dB. The gain of these methods gets higher with complicated scenes and fast movement, where the error concealment looses its performance dramatically if applied to large areas.

Screenshots of concealed versions of video with and without the proposed method illustrate the visual quality improvement using an appropriate error detection mechanism (Figure 6).

Please, note that as the compression gain is content dependent, also the size of an erroneous area corresponding to a missing RLC packet will vary for different parts of a sequence. In the shown case the temporal error concealment works well (one can only see slight shift at the boundaries of an erroneous area). However, if the sequence contains fast move-

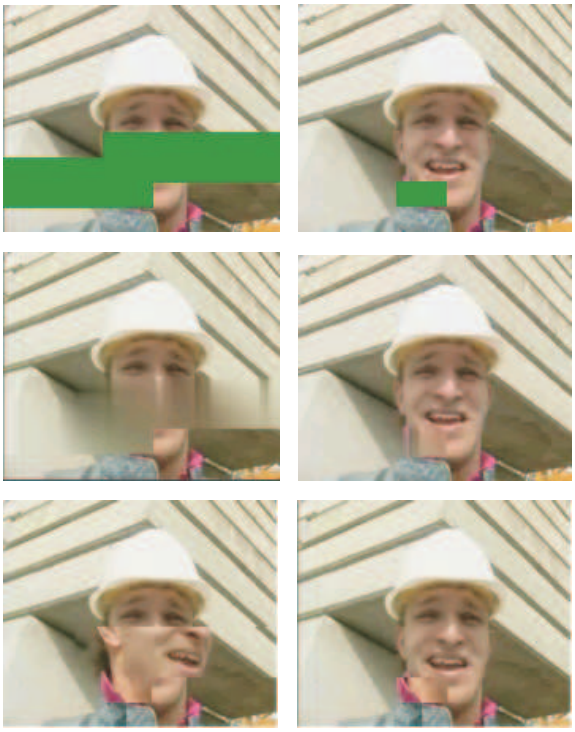

Figure 6: Screenshots from the foreman sequence without (left) and with the cross-layer detection (right); without and with spatial and temporal error concealment.

ment, performance of the error concealment will decrease rapidly (this is also true for sequences with reduced frame rate).

The main benefit of the method using synchronization marks is in its applicability as could be seen from the presented histograms. This method can be used also in the cases where the first payload RLC packets in an IP packet are wrong. The higher the slice size, the higher is also the benefit of both presented methods compared to the slicing itself.

Please note, that the presented cross-layer error detection can also be used with any other video codec (H.263, MPEG-4 etc.). We chose H.264 as it has the best performance and is suitable for video streaming over mobile communication networks.

6. CONCLUSIONS

In this article we introduced simple methods for the improvement of the detection in H.264 encoded video over UMTS. Presented methods use the information about the correctness of the RLC packets, present at the RLC layer of the radio access network. This information needs to be passed to the application layer, so that it can be used to refine the detection granularity and ease the error concealment. The first method uses the correctly received parts of an IP packet from its beginning to the first error. This results in a considerable improvement of the PSNR already after a very simple error concealment. Second method allows the usage of all correctly received RLC packets. To do so, synchronization marks for VLC are needed. However, the overhead below 3% for such solution could be still acceptable. To provide a realistic view on the presented methods, we analyzed the situations where these methods can be used. The analysis is based on measurements in live UMTS networks for static and dynamic scenarios. It showed, that both methods are applicable for static as well as for dynamic scenarios. Presented methods provide a significant improvement in the video quality at no/small additional costs (neither overhead, nor complexity). To deploy them no changes in the standards and only a slight change of the implementation is needed.

7. ACKNOWLEDGMENT

The authors would like to thank mobilkom austria AG&Co KG for supporting their research. The views expressed in this paper are those of the authors and do not necessarily reflect the views within mobilkom austria AG&Co KG.

References

[1] H. Holma, A. Toskala, "WCDMA for UMTS: Radio Access For Third Generation Mobile Communications," John Wiley & Sons, Ltd, UK, 2004.

[2] 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.

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

[4] 3GPP TSG TR 25.322, "Radio Link Control (RLC) protocol specification," v.4.12.0, June 2004.

[5] 3GPP TSG TR 25.212, "Multiplexing and channel coding (FDD)," v.4.6.0, October 2002.

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

[7] O. Nemethova, J. Canadas, M. Rupp, "Improved Detection for H.264 Encoded Video Sequences over Mobile Networks," Accepted for ISCTA 2005, Ambleside, UK, 2005.

[8] W. Karner, P. Svoboda, M. Rupp, "A UMTS DL DCH Error Model Based on Measurements in Live Networks," in Proc. 12th International Conference on Telecommunications 2005 (ICT 2005), Capetown, South Africa, May 2005.

[9] H.264/AVC Software Coordination, "JM Software," ver.7.3, available in http://iphome.hhi.de/suehring/tml/.