Applicability of UDP-Lite for Voice over IP in UMTS Networks

Frank Mertz, Ulrich Engelke, Peter Vary
RWTH Aachen University, Institute of Communication Systems and Data Processing (IND)
D-52056 Aachen, Germany
Email: {mertz, vary}@ind.rwth-aachen.de

Hervé Taddei, Imre Varga
Siemens AG Com MD
Haidenauplatz 1
D-81675 Munich, Germany
Email: {herve.taddei, imre.varga}@siemens.com

Abstract— **This paper examines the application of UDP-Lite for unequal error detection in packet-switched speech transmission via Internet protocols (Voice-over-IP) over UMTS radio channels. Traditionally, UDP is used as transport layer protocol, which contains a checksum that covers the complete packet. Thus, any packet with residual bit errors is discarded. Speech codecs like AMR, however, can tolerate bit errors in less sensitive parts of the bitstream. A more recent development, UDP-Lite, provides unequal error detection with a partial checksum that covers only the sensitive parts of a packet. Thus, only packets with errors in important bits are discarded. We compare the use of UDP-Lite for UMTS channels with convolutional and channels with Turbo coding. The results show that the achievable quality improvement by applying UDP-Lite depends on the residual bit error distribution of the chosen UMTS channel coding method. While we determined a quality improvement for channels with convolutional coding, we did not get an improvement for Turbo coding. Furthermore, when combined with header compression, the convolutional coder with use of UDP-Lite can reach the performance of the Turbo coder with use of UDP.**

I. Introduction

With the increase of broadband DSL Internet accesses in private homes and the development of new mobile communication systems, Voice over IP (VoIP) has become increasingly important over the last years. The deployment of 3rd generation systems like UMTS and cdma2000, which provide higher data rates and more flexibility than 2nd generation systems, will lead to various new services. Previous studies conducted within 3GPP have shown that the use of VoIP from mobile to mobile in UMTS is feasible at good quality [1],[2]. In these studies, extensive conversational tests have been performed with a real-time PC based simulation system as shown in Fig. 1, developed jointly by Siemens, RWTH Aachen University and France Telecom.

Speech transmission over UMTS packet-switched networks is realized using RTP/UDP/IP protocols. At the receiver, residual bit errors after channel decoding are detected by a *Cyclic Redundancy Check* (CRC) within the physical layer. Usually, all affected packets must be discarded to guarantee error-free transmission of the different headers. However, residual bit errors can be tolerated in some parts of the encoded speech frames, e.g. of the AMR (Adaptive Multi-Rate) codec, as the encoded bits have different impacts on the quality of the reconstructed speech. Therefore, on the speech encoder side, bits are usually sorted into different classes according to their significance. On circuit-switched channels, these classes

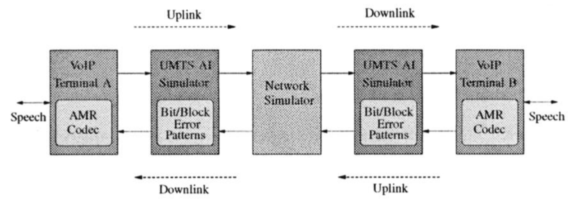


Fig. 1. Simulation System for VoIP over UMTS

are then protected differently against transmission errors. This *unequal error protection and detection* (UEP/UED) could also be beneficial for VoIP over UMTS.

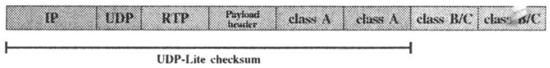
To provide the ability of protecting only important parts of IP packets, the *Lightweight User Datagram Protocol* (UDP-Lite, [3]) has been standardized by IETF (Internet Engineering Task Force). UDP-Lite is compatible with the User Datagram Protocol (UDP). It provides *unequal error detection* with a partial checksum coverage of only the important payload bits.

In UMTS, two channel coding schemes are available: convolutional coding and the advanced Turbo coding method. It has been shown in [4] that the application of UDP-Lite for VoIP over wireless links is beneficial for channels using convolutional coding. However, it has not yet been shown whether this is also true for channels using Turbo coding. This paper compares the application of UDP-Lite for Turbo and for convolutional coding channels in UMTS. We will show that while the quality can be improved by applying UDP-Lite for convolutional coding, it is not able to improve the quality for Turbo coding. Furthermore, we will show that when combined with header compression, the convolutional coder in UMTS with UDP-Lite is as effective as the Turbo coder with UDP, although the Turbo coder itself is generally more effective than the convolutional coder.

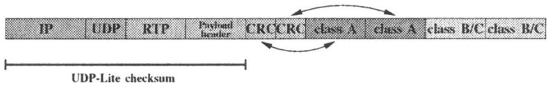
II. Simulation System: VoIP and UDP-Lite

A VoIP over UMTS simulation system developed at RWTH Aachen University is shown in Fig. 1. The VoIP terminal sends AMR encoded speech frames that are encapsulated using IPv6/UDP/RTP headers to the UMTS air interface simulator which simulates the air interface uplink (UL) transmission. The network simulator simulates the IP core network behavior and transmits packets to the second air interface simulator for downlink (DL) simulation. The receiving VoIP terminal decomposes the received IP packets and decodes the contained speech frames. In this study, we only consider a single UMTS

978-3-8007-2909-8/05/$20.00 ©2005 IEEE



(a) without internal frame CRCs



(b) with internal frame CRCs

Fig. 2. Packet structure with two AMR speech frames in the payload

downlink transmission of VoIP packets from the network simulator to the terminal. In the following, the VoIP transmission issues are described in detail, and Sec. III will explain the UMTS air interface simulation.

A. Voice over IP Terminal

In the VoIP terminal, the speech signal is encoded by the AMR [5] speech codec using 20 ms frames. The encoded bits are then RTP encapsulated (Real-time Transport Protocol, IETF RFC 3550). The RTP header contains timing information and a sequence number. The encoded speech frames are placed into the RTP payload according to the payload format defined by IETF in RFC 3267. It allows an arbitrary number of frames per packet and optional internal CRCs for each frame. When *robust sorting* is enabled, the most important bits (class A) of each frame are placed at the beginning of the payload. The payload format is designed to optionally support UEP and UED. Two basic approaches are defined and shown in Fig. 2:

a) Use of a partial checksum that covers the IP/UDP/RTP headers, the payload header and all class A bits of the AMR payload.
b) Use of a partial checksum that only covers the IP/UDP/RTP and payload headers; additional *Frame CRCs* for each speech frame that cover the class A bits.

With the first approach, a single bit error in one of the contained frames may cause all frames to be discarded. If using the second approach, only the frame which includes errors would be discarded, at the expense of two additional bytes for the CRCs. RTP packets are finally encapsulated with UDP or UDP-Lite and IPv6 headers.

B. UDP-Lite vs. UDP

The UDP-Lite protocol [3] has been developed to provide *unequal error detection* (UED) for IP based packet streams. The protocol is a modification of UDP and can be used as replacement. The UDP header contains source and destination ports to address the respective applications, a field specifying the length of the packet in bytes, and a checksum field. The checksum is calculated over the entire payload as well as over the UDP header and important parts of the packet's IP header. In the UDP-Lite header (see Fig. 3), the length field of the UDP header is replaced by a checksum coverage field that specifies which part of the payload is used in the checksum calculation, starting from the beginning of the packet. If used with full checksum coverage, UDP-Lite is semantically identical to UDP.

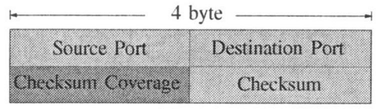


Fig. 3. UDP-Lite Header

To make this mechanism applicable, UDP-Lite assumes that the important bits of a packet are located at the beginning of the packet. Furthermore, it requires the hand-over of erroneous packets from the lower layer, i.e. in the considered UMTS system, radio frames with residual bit errors should not be discarded by the physical layer CRC.

C. Robust Header Compression (ROHC)

In packet switched speech transmission, the size of packet headers compared to the speech payload is very high. The RTP, UDP and IPv6 headers require 12+8+40=60 bytes per IP packet, while a single AMR compressed speech frame needs about 12-31 bytes, depending on the codec bit rate. Therefore, the IETF has standardized an algorithm to compress these headers for transmission over wireless links: Robust Header Compression (ROHC), IETF RFC 3095. Utilizing the significant redundancy between header fields of consecutive packets, this algorithm is able to compress an RTP/UDP/IPv6 header down to an average size of 3 bytes.

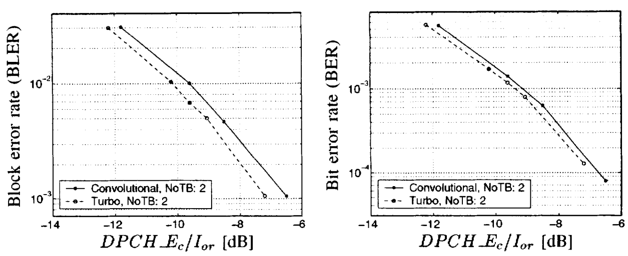
III. UMTS Air interface simulation

Our real-time UMTS simulator emulates the DL transmission of IPv6/UDP/RTP packets containing AMR speech frames over the UMTS air interface. Main part of the air interface simulator is an implementation of the Radio Link Control (RLC) protocol (3GPP TS 25.322) for assigning IP packets to radio frames. The underlying *physical layer* has previously been simulated offline for different radio channel qualities and channel coding schemes, and the resulting bit error patterns are inserted within the real-time simulation.

In UMTS, different channels can be chosen for the transfer of data between the user equipment and the base station. A channel is defined by a RAB (Radio Access Bearer). For the air interface simulations, we used different packet-switched (PS) RABs as defined in 3GPP TR 25.993. These RABs specify the use of a PDCP (Packet Data Convergence Protocol) header, of RLC in unacknowledged mode, of MAC (Medium Access Control) in transparent mode, different data rates by transmitting 1, 2 or 4 transport blocks per TTI (transmission time interval) of 40 ms, each with a size of 656 bits, and a 16 bit CRC. For comparison, both Turbo and convolutional coding have been used in the simulations.

A. Radio Link Control Simulation

In the data link layer, the PDCP sub-layer optionally performs header compression (ROHC, see Sec. II-C) on the IP packets, it adds a 1-byte PDCP header and forwards the resulting PDCP PDUs (Protocol Data Units) to the RLC sub-layer. Each TTI of 40 ms, available PDCP PDUs are segmented/concatenated and placed into one or more fixed size RLC PDUs (i.e. transport blocks). Appropriate RLC headers are added, containing a sequence number for detecting discarded PDUs and length indicators defining the boundaries



(a) Block error rates (b) Bit error rates

Fig. 4. Block and bit error rates of transport blocks for the performed channel simulations with NoTB=2

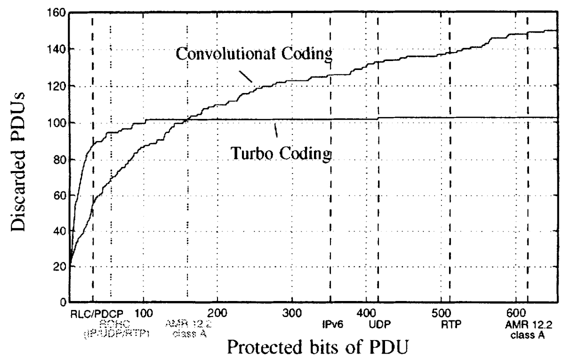


Fig. 5. Discarded PDUs depending on the checksum coverage

of contained packets. The RLC PDUs are then passed to the error insertion block that adds the pre-processed bit error patterns. Because of delay requirements for VoIP applications, the RLC is used in *unacknowledged mode*, i.e. there are no retransmissions of erroneous PDUs. Usually, any residual bit errors in an RLC PDU, detected by the physical layer CRC, would lead to the discarding of this PDU. However, when applying UDP-Lite, the PDUs with residual bit errors have to be handed over to the upper layers. The RLC and PDCP headers are not protected by any partial checksum, and the physical layer CRC may only indicate the possibility of bit errors in any part of the PDU. Therefore, while decomposing the PDU, the receiving RLC instance will have to check if sequence number and length indicators make sense for the received packet size and else discard the PDU.

B. UMTS Physical Layer Simulation

The UMTS downlink physical layer simulations for the generation of the bit error patterns were carried out following the parameters of the conformance tests for base stations (3GPP TS 25.141) and user equipment (3GPP TS 34.121). As test scenario, the "outdoor to indoor and pedestrian test environment" defined in [6] was selected.

In UMTS systems, two different channel coding schemes are specified (see 3GPP TS 25.212): a convolutional coder (rate 1/2 or 1/3) and a Turbo coder (rate 1/3). All simulations were carried out with both channel coding methods in order to compare their residual bit error distributions and the impact on the application of UDP-Lite. The rate-1/3 *Turbo coder* consists of two parallel concatenated convolutional codes (PCCC), coupled by the Turbo code internal interleaver. The two constituent codes are recursive and have a constraint length of 4. The iterative decoding scheme has been configured to use the Log-MAP algorithm with 4 iterations. The maximum usable block length for the Turbo coder is 5114 bits. The *convolutional coder* is also used with rate 1/3, has a constraint length of 9 and encodes blocks of maximal 504 bits.

IV. Simulation Results

Based on the simulated bit error patterns, theoretical considerations about the expected improvement to be gained by the application of UDP-Lite will be given. These considerations will be supported by simulation results of VoIP packet stream transmissions using the simulation system described in Sections II and III.

A. UMTS Channel Simulations

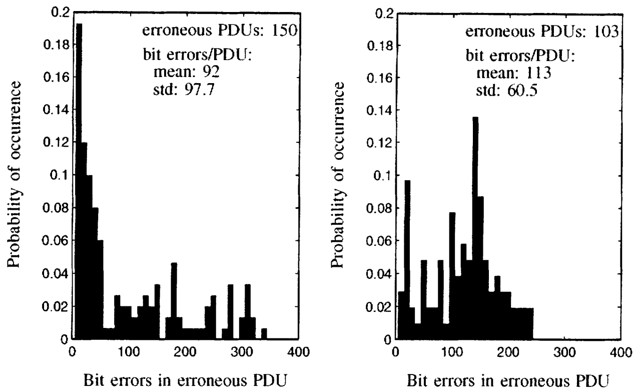
Channel simulations with the settings given in III-B have been performed for different channel coding types, i.e. convolutional and Turbo coding, and different channel qualities. The channel quality is given as E_c/I_{or}, i.e. the ratio of the transmit energy of the dedicated physical channel (DPCH) and the interference of other channels in the cell plus additional noise from other base stations. Different channel data rates were simulated using different numbers of transport blocks per TTI, in the following abbreviated as NoTB, and respective spreading factors. Data rates of 16, 32 and 64 kbit/s have been simulated by using 1, 2 and 4 transport blocks of size 656 bits. The length of each simulation was set to 5 minutes. With a TTI of 40 ms, you get, e.g., 15000 transport blocks if you transmit two blocks per TTI (NoTB=2). Each simulation produced a bit error and a block error pattern file. A block error is a transport block that contains an arbitrary number of bit errors. These error patterns have been used in the later simulations of packet based speech transmission.

The resulting bit error rates (BER) and block error rates (BLER) for different channel qualities are shown in Fig. 4 for the 32 kbit/s channel (NoTB=2). It can be noted that the Turbo channel coding scheme performs better than convolutional coding at a given channel quality.

B. First Considerations on UDP-Lite Potential

In a first consideration of the potential that unequal error detection by using UDP-Lite may offer, we assume that each transport block will contain a single IP packet starting at the beginning of the RLC PDU.

We want to determine how many RLC PDUs will have to be discarded because of bit errors in the sensitive part of the contained IP packet, i.e. the part being protected by the UDP-Lite checksum. Therefore, we determine a function of the number of PDUs that contain bit errors in the sensitive part of the PDU in dependence on the position of the last bit in the PDU to be covered by the checksum. The function is determined from the simulated bit error files and shown in Fig. 5 for both coding schemes, simulated with NoTB=2 at a channel quality of $E_c/I_{or} = -9.6$ dB. The respective bit positions of the different header and data parts of the contained RLC header and IP packet within a 656 bit PDU are marked by the vertical dashed lines. Black dashed lines indicate uncompressed headers, in this case the remaining codec bits



(a) Convolutional, NoTB=2

(b) Turbo, NoTB=2

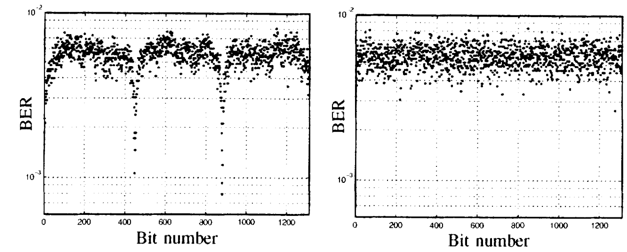
Fig. 6. Distributions of residual bit error number in erroneous PDUs

will be transmitted in the following PDU, and gray dash-dotted lines indicate the case of using header compression.

For Turbo coding, a total number of 103 PDUs (of 15000 transmitted PDUs) contain bit errors. This corresponds to the number of discarded PDUs if all 656 bits are protected by the checksum. Already 90 of these PDUs contain bit errors in the first 32 bits, i.e. in the important RLC and PDCP headers that describe how to decompose the PDU into the contained IP packets. The curve rises steeply and reaches close to its maximum already around the bit position 100, which not even covers the sensitive IPv6 header. Even if header compression is considered, the end of the sensitive class A bits lies in the saturation part of the curve. Therefore, the application of UDP-Lite with Turbo coding is not expected to reduce the number of discarded PDUs and thereby to improve the quality.

The curve for convolutional coding rises slower, therefore providing a reduction in discarded PDUs even for a greater length of the protected sensitive part. However, if you protect the complete PDU, this coding scheme leads to a higher overall loss rate than Turbo coding at the same E_c/I_{or}.

The different behavior can be explained by differences in the residual bit error distributions of both coding schemes. As shown in the histograms of Fig. 6, the convolutional coder produces many PDUs with a low number of residual bit errors, while for Turbo coding most erroneous PDUs have a higher number of residual bit errors. Furthermore, the probability for each single bit to be disturbed is partly different for the two coding systems. As shown in Fig. 7, the likeliness of a bit being disturbed is quite equally high for every bit within an erroneous PDU in the case of Turbo coding. Whereas, for convolutional coding, the bits at the beginning and ending of the coding blocks (here 3 coding blocks for the two transport blocks) have a lower residual bit error probability. This results from the Viterbi decoding algorithm. Start and end state are fixed, therefore the Viterbi decoder has fewer possibilities for state transitions in the Trellis diagram for these bits. These effects explain why among the disturbed PDUs it is more likely for convolutional than for Turbo coding to have some disturbed PDUs with no bit errors in the sensitive part at the beginning of the PDU.



(a) Convolutional, NoTB=2

(b) Turbo, NoTB=2

Fig. 7. Bit error distributions within the PDUs after channel decoding

C. VoIP Simulations

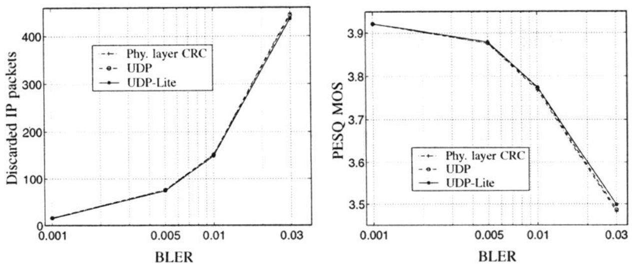
Different scenarios have been simulated with real VoIP packet streams to confirm the previous considerations and measure the possible quality improvement on the decoded speech. The speech frames have been encoded with the 12.2 kbit/s mode of the AMR codec and packed into RTP/UDP/IPv6 packets with the VoIP terminal software as described in Sec. II. First, no RTP internal frame CRCs are considered. Depending on the desired data rate of the channel, NoTB=1, 2 or 4 transport blocks (PDUs) were used per TTI to transmit the IP packets.

16 kbit/s channel, AMR 12.2 kbit/s, header compression

For this scenario the number of transport blocks was set to NoTB=1, resulting in a channel rate of 16 kbit/s. Each packet contained one speech frame, i.e. the VoIP simulator provided one IP packet every 20 ms. With a TTI of 40 ms on the UMTS channels, two IP packets had to be transmitted in each TTI. Considering the overhead of RTP/UDP/IPv6 packet headers, a header compression algorithm was required for the given channel. The ROHC algorithm has been used as described in Sec.II-C, resulting in an average packet data rate of 14.4 kbit/s.

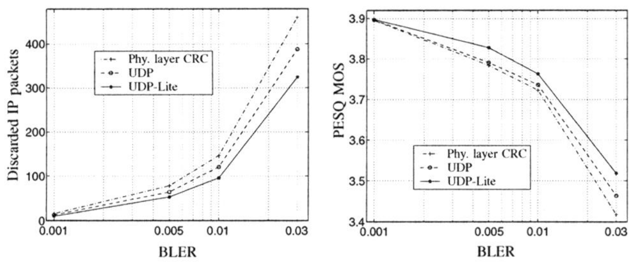
The simulation results in terms of discarded IP packets and resulting speech quality at different channel qualities (in terms of block error rates) are shown in Fig. 8 for Turbo and in Fig. 9 for convolutional coding. The speech quality is measured in PESQ MOS according to [7]. For Turbo coding, there is hardly any difference whether we apply the physical layer CRC, the full packet coverage by the UDP checksum, or a partial coverage by the UDP-Lite checksum. For the convolutional coder, however, the application of UDP-Lite yields a noticeable performance improvement, i.e. fewer discarded packets and better speech quality in spite of residual errors in the less sensitive payload bits. The quality improvement from the physical layer CRC to the UDP checksum usage can be explained by padding at the end of not fully filled PDUs. Sole bit errors in that part will only lead to a discarded packet when using the physical layer CRC, which does not consider where the data part of the PDU ends.

For a better comparison of both channel coding schemes, the numbers of discarded IP packets for an example channel quality of $E_c/I_{or} = -12.5$ dB are shown in Fig. 10(a). The Turbo coding scheme is more effective than the convolutional code, reflected in the lower number of discarded packets in case of physical layer CRC usage (noted as CRC in the figure).



(a) Discarded IP packets (b) PESQ evaluation

Fig. 8. AMR 12.2; NoTB=1; ROHC; Turbo coder



(a) Discarded IP packets (b) PESQ evaluation

Fig. 9. AMR 12.2; NoTB=1; ROHC; convolutional coder

Applying partial UDP-Lite checksum coverage does only lead to an improvement for convolutional coding, finally resulting in a comparable quality of convolutional coding with UDP-Lite and Turbo coding with UDP.

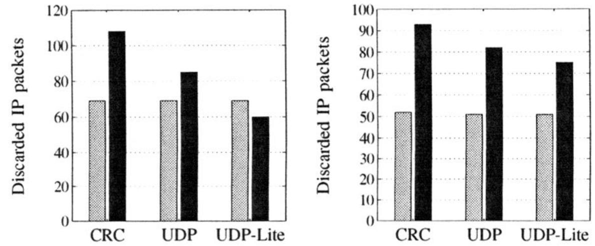
32 kbit/s channel, AMR 12.2 kbit/s, 2 frames per packet, no header compression

For this scenario, the number of transport blocks was set to NoTB=2, resulting in a channel rate of 32 kbit/s. Two speech frames encoded with the AMR 12.2 kbit/s mode were transmitted in each IP packet. No header compression was required, the IP packets were transmitted in their original size, resulting in a packet data rate of 25.0 kbit/s, which is not fully utilizing the available capacity.

The number of discarded IP packets is shown for an example channel quality of $E_c/I_{or} = -9.6$ dB for both channel coding schemes in Fig. 10(b). Again, an improvement by applying UDP-Lite can only be seen for convolutional coding. However, the quality remains significantly worse than for Turbo coding with UDP. The advantageous residual bit error distribution of the convolutional coder cannot be fully utilized in this scenario, because without header compression the sensitive part of the PDU is still quite large. Besides, in this scenario every discarded IP packet will lead to a loss of two consecutive speech frames that are transmitted in the packet.

Frame CRC within RTP packets

We conducted further studies on the transmission of two AMR encoded speech frames within each RTP packet. We compared the use of a partial UDP-Lite checksum coverage over the class A bits of both contained frames to the use of a partial checksum only over the headers and use of separate internal CRCs for both speech frames, as described in Sec. II-A. However, there was no improvement by the latter.



(a) AMR 12.2; ROHC; $E_c/I_{or} = -12.5$ dB; NoTB=1 (Sec. IV-C) (b) AMR 12.2; 2 frames/pkt $E_c/I_{or} = -9.6$ dB; NoTB=2 (Sec. IV-C)

Fig. 10. Number of discarded IP packets for turbo coding (gray) and convolutional coding (black)

V. Conclusion

This study examined the application of UDP-Lite for unequal error detection in Voice over IP transmission over UMTS. We compared both UMTS channel coding schemes to see whether it is as effective with Turbo coding as it is with convolutional coding. The simulation results show that the achievable performance gain (in terms of less discarded IP packets and therefore better speech quality) depends on the distribution of the residual bit errors after channel decoding and thus on the choice of the channel coding scheme. When convolutional coding is used, there is a significant number of PDUs with only few residual bit errors in the less sensitive part of the contained packets. Therefore, a reduction of discarded packets and an increase in speech quality can be achieved by utilizing UDP-Lite with convolutional coding (especially when header compression is added to reduce the sensitive header size). When Turbo coding is used, however, there is no significant reduction in discarded packets by using UDP-Lite. This is because with Turbo coding the erroneous PDUs contain a fairly high number of residual bit errors that are distributed fairly equally over the whole PDU. Therefore, the application of UDP-Lite together with Turbo coding does not lead to a better speech quality when compared to the conventional UDP.

We also compared the effectiveness in correcting bit errors for both channel coding schemes in UMTS. Under the same channel conditions, the Turbo coder is usually more effective than the convolutional coder. Despite this fact, we show that when combined with header compression and the unequal error protection of UDP-Lite, the convolutional coder can reach the performance of the Turbo coder and UDP.

References

[1] 3GPP, "TR 26.935: Packet switched conversational multimedia applications; Performance characterisation of default codecs (Release 6)," 2004.

[2] H. Taddei *et al.*, "Evaluation of AMR-NB and AMR-WB in Packet Switched Conversational Communications," in *IEEE Int. Conf. on Multimedia and Expo (ICME 2004)*, June 2004, Taipei, Taiwan.

[3] L.-A. Larzon, M. Degermark, *et al.*, "The Lightweight User Datagram Protocol (UDP-Lite)," IETF, RFC 3828, July 2004.

[4] L.-A. Larzon *et al.*, "Efficient Transport of Voice over IP over Cellular links," in *IEEE Global Telecom. Conf. (GLOBECOM)*, Nov. 2000.

[5] 3GPP, "TS 26.090: Adaptive Multi-Rate (AMR) Speech Transcoding."

[6] *Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03)*, ETSI TR 101 112, April 1998.

[7] ITU-T Rec. P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2001.