Video Streaming over MBMS: A System Design Approach

Junaid Afzal, Thomas Stockhammer, Taigo Gasiba, Wen Xu
Email: {afzal, stockhammer, gasiba}@nomor.de, wen.xu@benq.com

Abstract— **Recently, Multimedia Broadcast Multicast Service (MBMS) has been specified by 3GPP as a Release 6 feature in order to meet the increasing demands of multimedia download and streaming applications in mobile scenarios. H.264, as the unique recommended video codec for MBMS, serves as an essential component because of its high compression efficiency and easy network integration capability. In this study, we introduce and analyze the main system design parameters that influence the performance of the H.264 encoded video streaming over EGPRS and UMTS bearers. Effective design methodology including robustness against packet losses and efficient use of the scarce radio resources is presented. Care is taken on the processing power of mobiles, service delay constraints, and heterogeneous receiving conditions. Then, we investigate application of an advanced receiver concept, the so-called permeable layer receiver, in MBMS video broadcasting environments. Selected simulation results show the suitability of certain parameter selection as well as the benefits provided by the advanced receiver concept. Finally, a real-time test bed for MBMS called RealNeS-MBMS is presented. With this tool, a standard-compliant GERAN network can be simulated and the system design procedure including H.264 based video broadcast streaming can be evaluated in real-time.**

I. INTRODUCTION

Due to the explosive growth of the multimedia internet applications and dramatic increase in mobile wireless access, there is a significant demand for multimedia services over wireless networks. It is, therefore, foreseen that next generation wireless systems will have to support applications with increased complexity and tighter performance requirements, such as real-time video streaming. Furthermore, it is expected that popular content is streamed not just to a single user, but to multiple users attempting to access the same content at the same time. Thus, 3GPP has introduced a new point-to-multipoint (p-t-M) optional service under the acronym of Multimedia Broadcast Multicast Service (MBMS) in Release 6, targeting at simultaneous distribution of multimedia content to many mobiles within a serving area. The expected traffic is believed to be in the areas of weather information, traffic telematics, news broadcast, music streaming, video concert, sports replay, or file sharing.

To save costs of new infrastructure as well as to enable fast introduction of multimedia broadcast services, MBMS relies on the infrastructure and protocols of the already existing GSM and UMTS. Specifically, packet-based bearers within UMTS or EGPRS (Enhanced General Packet Radio Services) are reused to support these services. Details on requirements and recommendations for possible MBMS-specific extensions to the GSM and and UMTS are provided in [1]. Among others it is stated that point-to-multipoint (p-t-M) solutions should be adopted to increase radio efficiency compared to multiple point-to-point (p-t-p) connections. However, schemes should be favored which minimize the impact on the current RAN physical layer and maximize the reuse of existing protocols.

Despite the possible reuse benefits, existing wireless communication systems have been optimized for point-to-point (p-t-p) data transfer. Several modifications to standards and network infrastructures are therefore required to provide resource-efficient multicast services. Due to missing feedback links and broadcast distribution, adaptation to actual transmission conditions of individual users by the use of power control, retransmission protocols, or adaptive modulation and coding schemes is obviously infeasible. Consequently, at least some receiving entities will experience increased radio block loss rates. Conventional retransmission schemes based on Automatic Repeat Request (ARQ) mechanism could be applied to MBMS, but it would overload the uplink when many receivers attempt to send their feedback simultaneously, making the ARQ procedure hard to realize for MBMS. Therefore, the loss of single radio blocks is unavoidable and loss rates in the order of 10% or higher can be quite common. To overcome the problem of reduced reliability, several proposals of additional Forward Error Correction (FEC) schemes have been considered in the MBMS standardization. A summary with discussions of benefits and drawback of different solutions for video streaming applications is for example provided in [2]. As a suitable trade-off between easy dissemination and sufficient performance, 3GPP has decided to introduce application layer (AL) FEC using Raptor Codes [3], [4] as an additional means to provide reliability.

3GPP distinguishes two types of multimedia delivery services, *streaming* and *download*, each requiring a different system design. Whereas download services must be offered error-free but can tolerate higher delivery delays, streaming services inherently include stringent timing constraints but can tolerate losses at least to some extent. In this study, we focus on the streaming

This paper is partly based on "System Design Options for Video Broadcasting over Wireless Networks," by J. Afzal, T. Stockhammer, T. Gasiba and W. Xu, which was presented at IEEE Consumer Communication and Networking Conference, Las Vegas, NV, USA, January 2006.

delivery, especially on video broadcast applications based on H.264/AVC [5], the unique recommended video codec in MBMS.

In section II, we provide a brief overview of the H.264/AVC codec and its integration into the MBMS streaming framework. Section III discusses available system design options on different protocol layers. Selected representative simulation results based on a formalized simulation environment are given in section IV. A real-time simulation environment for MBMS is introduced in section V. Conclusions and open issues for future work are finally provided in section VI.

II. Multimedia Streaming Framework over MBMS

A. H.264/AVC Video Transmission over Mobile Packet-Networks

H.264/AVC [5] is an attractive candidate for wireless video application in the near future mainly due to its excellent compression efficiency and network friendly video representation. For MBMS video services, H.264/AVC baseline profile is the only recommended video codec. The recommended baseline profile allows the use of most error-resilience issues such as slice structured coding, Flexible Macroblock Ordering (FMO), multiple reference frames, as well as higher frequency of intra information. The later can be accomplished by intra coding of macroblocks or by introducing Instantaneous Decoding Refresh (IDR) pictures to allow random access and full recovery from errors. Although compression efficiency is the major attribute for a video codec to be successful in wireless transmission environments, it is also essential and welcome that H.264/AVC provides means to be easily integrated into existing and future networks as well as that it addresses the needs of different applications.

H.264/AVC [5] is an attractive candidate for wireless video application in the near future due to its excellent compression efficiency and network friendly video representation. For MBMS video services, H.264/AVC baseline profile is the unique recommended video codec. It allows the use of the error-resilience mechanisms such as slice structured coding, Flexible Macroblock Ordering (FMO), multiple reference frames, as well as higher frequency of intra information. The latter can be accomplished by intra-macroblock coding or by introducing Instantaneous Decoding Refresh (IDR) pictures to allow random access and full recovery from errors. H.264/AVC is capable not only to provide high compression efficiency which is essential for a codec used in wireless transmission environments, but also to be easily integrated into existing and future networks. It has also been selected to be used in various applications such as DVD, DVB, etc.

The elementary unit processed by an H.264/AVC codec is called Network Abstraction Layer (NAL), which can be directly encapsulated into different transport protocols and packet-based networks such as RTP/IP. Most commonly, a single slice data is encapsulated in a NAL unit. One NAL unit type is specifically dedicated to a slice in a picture indicating a random access point in the video stream. In general a single NAL unit is encapsulated in an RTP packet according to the RTP payload specification [6]. More advanced packetization modes allow aggregation of several NAL units into one RTP packet as well the fragmentation of a single NAL unit into several RTP packets. Especially the latter mode can be successfully used in the MBMS streaming framework as it will be shown later. It allows to fragment any NAL unit into an arbitrary number of fragments, each with arbitrary size. Each fragment is then transported within a single RTP packet. Despite the introduced overhead, fragmentation can be beneficial in the considered framework as the loss rate of shorter packets is lower and can be exploited in the FEC framework. Fig. 1 shows the protocol stack

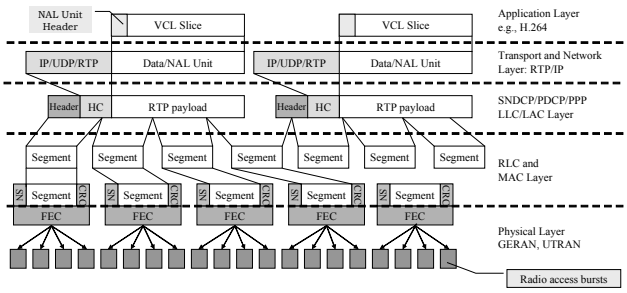


Figure 1. Processing of application layer packets in packet radio systems.

for the integration of RTP packets encapsulated in UDP and IP packets in UMTS and EGPRS. In the following the processing in UMTS is discussed, the corresponding layer acronyms for EGPRS are shown in Fig. 1. Robust Header Compression (RoHC) is applied to the generated RTP/UDP/IP packet resulting in a single Packet Data Convergence Protocol (PDCP)-Protocol Data Unit (PDU) that becomes a Radio Link Control (RLC)-Service Data Unit (SDU). The SDU is then segmented into smaller RLC-PDUs which serve as the basic units to be transmitted within the wireless system. The length of these segments depends on the selected bearer as well as the coding and modulation scheme in use. Typically, RLC-PDUs are in the range between 20 bytes, e.g. for GPRS CS1, and 1280 bytes, e.g. for some MBMS bearers within UMTS. The physical layer adds a block check sequence (BCS) for error detection and FEC to RLC-PDUs. This channel-coded block is further processed in the physical layer before it is sent to the far end receiver. The transmission time interval (TTI) between two consecutive RLC-PDUs determines the system delay and the bearer bitrate. As already mentioned, the RLC-PDU loss rates can be as high 10% and even more for some users at the edge of coverage.

The expected RTP/UDP/IP loss rates in wireless environments usually significantly exceed those experienced in wired Internet connection. Note that loss of a single RLC-PDU results in loss of one or more PDCP/RTP packets. Therefore, if the size of RLC-PDUs is small and/or the size of incoming RTP/UDP/IP packets is large,

the loss rate of RTP/UDP/IP packets will even be amplified compared to the RLC-PDU loss rates. Therefore, additional means of reliability are necessary.

B. Streaming Framework for MBMS

To support MBMS services and associated signaling, the existing packet-switched architecture is extended by a new functional entity, the Broadcast/Multicast Service Center (BMSC). The BMSC (see Fig. 2) provides a set of functions for MBMS User Services provisioning and delivery and serves as an entry point for multicast services to be transmitted over MBMS. For streaming applications the BMSC applies FEC, in the AL, to the incoming UDP flows according to [7]. The framework for streaming

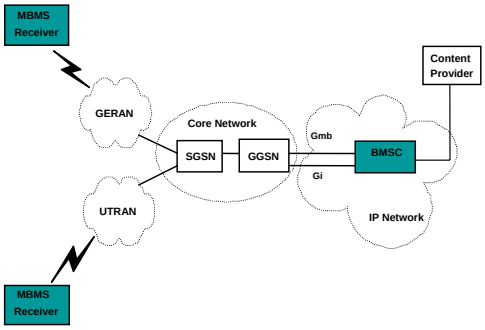


Figure 2. Broadcast-Multicast Service Centre.

delivery builds on the appropriate processing of compliant RTP packets or more precisely any UDP payloads, incoming at same or different UDP ports. These incoming source RTP packets and the UDP port information are used in order to generate FEC repair symbols. Here we will briefly present the packetization framework, more details can be found in [7].

Fig. 3 shows the processing of RTP packets and the interaction of the different protocol entities within sender and receiver. A generic application layer FEC layer, placed on top of the UDP layer, constructs FEC source packets by appending a 3 byte FEC source payload ID field at the end of each UDP payload[1]. These packets are then forwarded to UDP layer which, after UDP encapsulation, are transmitted to the receiver. Also a copy of these packets is forwarded to the FEC encoder and placed in a so-called source block, a virtual two-dimensional array of width T bytes, referred to as encoded symbol length. An inserted UDP payload is appended at the first empty row in the source block, the encoded symbol, and must start at the beginning of a new row. Each inserted UDP payload is preceded by a 3 byte field containing the UDP flow ID and the length field indicating the length of the inserted payload. The UDP flow ID (one byte) serves to distinguish between different streams and allows stream bundling. The two-byte length field indicates the length of the UDP payload. In typical cases the sum of the source UDP payload length length, length field and one byte of UDP flow ID is not an integer multiple of T. In this case, the remaining bytes in the last row are filled up with zero bytes. Note that these zeros are only virtual and are not transmitted. Each encoded symbol of length T has an associated encoded symbol ID (ESI). This ESI is placed in the FEC payload ID field together with a source block number (SBN) serving as a sequence number for the source block. Further source RTP packets are filled into the source block until the second dimension of the source block, the source block length (SBL) K determining the information length of the FEC code to be used, is reached. The SBL is flexible for each source block and might be varied to adapt the delay and the code strength.

After processing all source UDP payloads to be protected within one source block, the FEC encoder generates $N-K$ FEC repair symbols of size T by applying Raptor encoding. These FEC repair symbols can be transmitted individually or as blocks of G symbols as payload of a single UDP packet. Each FEC repair packet has a UDP payload header of 5 bytes, denoted as FEC repair payload ID, such that the receiver can insert correctly received source and repair UDP packets in its encoding block. Note that $G \times T$ determines the UDP payload size and also the resulting packet length. The repair payload ID contains the ESI of the first repair symbol, the SBN, and the SBL. If sufficient data for this specific source block is received, the decoder can recover all packets inserted in the encoding block, in particular the RTP packets and associated UDP flow. These packets are forwarded to the RTP layer which itself hands the recovered AL packets to the media decoder.

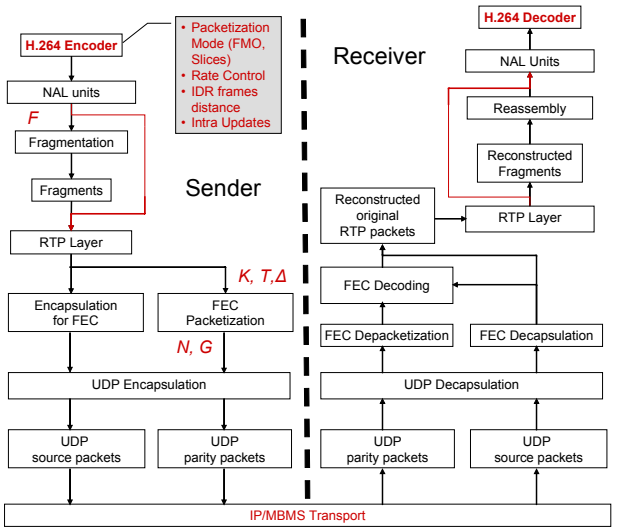


Figure 3. MBMS FEC Streaming Framework.

C. Raptor Code

Raptor codes were introduced in [3] and have recently been standardized by 3GPP [7]. In this section we will

[1]The FEC source payload ID field is appended at the end rather than as a payload header to allow RoHC work appropriately for RTP source packets.

briefly summarize the encoding and decoding algorithms of systematic Raptor codes as specified in [7, Annex B]. More details on the notations can be found in [4]. Raptor codes in general consist of an inner high-rate block code followed by a Luby transform (LT) encoder with some generation matrix $\mathbf{G}_{\text{LT}}$. The encoding and decoding procedures make use of the concept of a *code constraint processor* as shown in Fig. 4. The code constraint processor basically inverts a constraint matrix $\mathbf{A}(i_1, i_2, \cdots, i_r)$ to obtain L intermediate symbols $\mathbf{F}$ which serve as the input to an LT encoder. $\mathbf{A}(i_1, i_2, \cdots, i_r)$ contains the S constraints of the outer pre-code as well of the LT code taking into account ESIs $i_1, i_2, \cdots, i_r$. The construction of the code is such that the first K encoded symbols $\{E_i\}_{i=1,\ldots,K}$ are equivalent to the source symbols $\mathbf{C}$ such that the LT encoding process can also be used at the decoder to obtain the source symbols from the intermediate symbols. The code in [7, Annex B] is constructed such that $\mathbf{A}(1, \cdots, K)$ is invertible for any $K = 4, \cdots, 8192$.

At the decoder, only if an appropriate set of encoded symbols E_i with $i = i_1, i_2, \cdots, i_r$ is available such that the matrix $\mathbf{A}(i_1, i_2, \cdots, i_r)$ has full rank, is decoding successful. In average, the number of necessary encoded symbols, r, is only slightly more than K. The conventional receiver performs Raptor decoding on encoded symbols, i.e. symbols that have been received correctly from the channel. Note that if more than one symbol are contained in a UDP/IP packet, in case of the loss of an IP packet, all symbols in this packet are lost. At the receiver the correctly received symbols are processed by the *code constraints processor* that computes the original intermediate symbols. These are then forwarded to the LT encoder which, due to the systematic Raptor code, computes the source symbols.

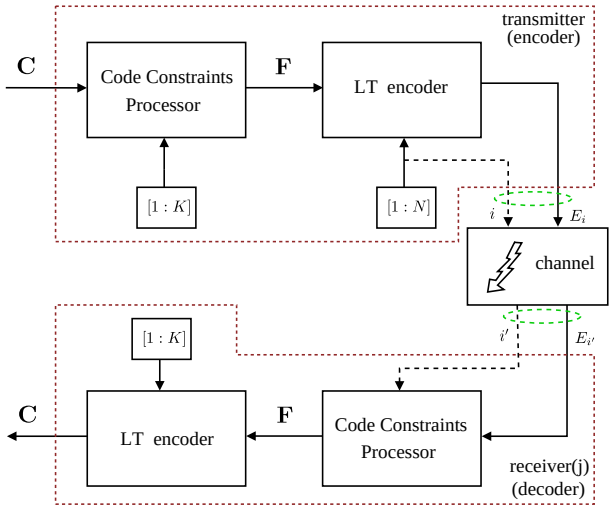


Figure 4. Practical implementation of a Raptor code system.

III. System Design and Optimization

A. H.264/AVC Video Coding and Packetization

Despite the good error resilience capabilities of H.264/AVC, solutions exclusively relying on reliability means in the video codec perform significantly worse than applying the necessary overhead for FEC on any of the lower layers [2], [8]. However, for users randomly accessing the stream, for stream switching purposes as well as for recovery in case of errors, especially for users with temporarily bad receiving conditions, IDR pictures should be inserted periodically. The *frequency of IDR pictures* is a parameter which needs to be considered very carefully. On one hand, too less IDR pictures result in bad random access property and, on the other hand, too frequent IDR pictures will reduce the compression efficiency.

Another important issue is that the encoded frames included in a single NAL unit are of arbitrary length and are typically much longer than RLC-PDUs. Therefore, alignment of IP packets with RLC-PDUs is virtually impossible and the loss rates are amplified as already elaborated. To provide smaller packets, slice structured coding and/or flexible macroblock ordering (FMO) might be used, but this does in general not provide packets of constant size, and in addition the compression efficiency is significantly reduced [2]. Therefore, it is proposed to rather apply NAL unit fragmentation for packet size adaption. This allows to generate packets of exactly the desired size in a very flexible manner. Although an NAL unit is lost even if only a single fragment in it is lost, this scheme is still very beneficial when combined with AL-FEC. The *fragment size* itself, which determines the resulting RTP packet size, is also a parameter which needs to be appropriately adjusted for the considered transmission.

Finally, the *bitrate* of the video application needs to be appropriately selected for given FEC parameters and specific MBMS bearer. If the video is encoded with too low bitrate, the channel may in general not be efficiently utilized. In contrast, if the bitrate is high, some congestion control in some intermediate buffers has to be applied to maintain a constant end-to-end delay.

The encoding of video is usually done offline such that the frequency of the IDR pictures as well as the bitrate are pre-determined and fixed for different transmission scenarios. This allows the distribution of one and the same stream in different environments, e.g. wired Internet, p-t-p wireless streaming, MBMS multicasting over UMTS, or video broadcasting over EGPRS. In contrast, the fragmentation can be applied specifically, for example in the BMSC, and therefore be adapted to the underlying network and transmission conditions. This will be elaborated in more detail in section IV.

B. FEC Options

The streaming framework including the AL-FEC increases the amount of adjustable parameters significantly.

Fig. 3 highlights several optimization parameters for AL-FEC. They should be adequately selected taking into account the application constraints and transmission conditions of the underlying bearers. The selection of FEC parameters is in general done in the BMSC and, in some favorable system design, can be adapted to the underlying transmission conditions.

Concretely, assume that a maximum end-to-end delay constraint Δ has to be maintained for the application and some overhead $(N-K)/K$, which can be equivalently expressed by the rate $r = K/N$, is to be targeted, e.g., to adapt to some expected worst-case receiving conditions. The symbol size T is fixed for the session and selected, as recommended, such that $K \geq 1000$ (for smaller K's the inefficiency of the Raptor code is more apparent) and $K \leq 8192$ (as this is the maximum K supported by the specification). In addition, it is advantageous in terms of complexity for the decoder if T is selected as 2^i with $i = 3, 4, \ldots$. We propose that K is obtained by inserting as many source packets into the source block such that the maximum end-to-end delay from the BMSC to the client, Δ, is not exceeded. As the packet sizes as well as the video bitrate fluctuates, K will also vary within certain range. For the total encoded symbols N it is proposed to select a certain target rate r such that $N \approx K/r$ taking into account all rounding effects. Finally, the number of code symbols per packet G should be appropriately selected such that it does not exceed some recommended maximum UDP payload size P.

C. Congestion and Rate Control Options

In MBMS, the application bitrate is usually fixed for each multicast stream disseminated to many clients, possibly clients being supported by different BMSC with even different radio technology. However, if the application bitrate exceeds the bitrate provided by an MBMS bearer, it is necessary to drop packets to maintain some constant end-to-end delay. Congestion control is used to solve the resource allocation problem, i.e. to allocate rates to users so as to maximize the sum of utilities subject to link capacity constraints. Details and insights for different congestion control schemes can for example be found in [9]. In general, simple queue-based congestion control is applied whereby incoming packets are dropped if they overflow a buffer or some delay is exceeded in some intermediate router. The incoming packets are dropped until some timeline is again fulfilled. More advanced buffer strategies as proposed and investigated in [9] might be used, but are not considered in this work.

Moreover, given a MBMS bearer with certain resources, congestion control can be applied to source and/or repair packets. Whereas the former case results in degraded quality for all users, the latter one only penalizes users with bad receiving conditions as the FEC overhead is reduced. Another important aspect is rate allocation for source and repair FEC packets. Congestion control and rate allocation, in general, is a rate constrained optimization problem but for MBMS due to missing feedback link it can be pre-decided based on worst case user assumption, on long-term measurements of quality-of-experience parameters, or based on network planning data.

D. Permeable–Layer Receiver

The advantages of applying the FEC on application layer in terms of reuse of existing protocols and infrastructure are in fact penalized by degradation in terms of efficiency. One issue comes from the processing of UDP/IP packets in the protocol stack according to Fig. 1: Conventional receivers ignore a significant amount of correctly received data as AL packets are discarded if any segment, i.e. any RLC-PDU, within the IP packet is corrupted. In general, this is reasonable for applications which cannot make use of incomplete and only partly correct AL packets. Note also that no standardized means exist to indicate missing parts of AL packets through network protocols. Wireless receivers, however, might be modified such that they allow to pass partially corrupted information from lower layers into the AL decoder. With the introduction of error correction on the AL, the propagation of such information to the FEC decoder is beneficial. In [10], it has been shown that by applying the so-called *Permeable–Layer Receiver* (PLR) the decoder for AL FEC can be modified to exploit this additional information in the decoding process. Specifically different decoding strategies, so-called simple PLR (s-PLR) and advanced PLR (a-PLR), have been presented for MBMS by considering specific encoding and decoding properties of Raptor codes [11]. With the PLR, no modifications at transmitter side are necessary. If the decoder applies the PLR strategy, it operates as if the AL packets do have the size of RLC-PDUs, but avoids the header overhead of the short packets. The work in [10] mainly focused on conceptual issues. In this study, we address a real practical implementation and evaluation of the s-PLR for H.264 encoded streams as presented in [2], [12], including the influence of header losses, the streaming framework, etc.

IV. Experimental Results

In order to verify the performance bounds for MBMS under different operating conditions, a series of simulations have been carried out which can be sub-divided into two categories: i) Worst Case User Simulations, and ii) Multiple-User System Level Simulations. The application layer simulations will elaborate the impact of different parameter settings on application layer performance whereas in system level simulations, the system design parameters will be optimized for better overall QoS.

A. Worst Case User Simulations

This section covers the simulations carried out to investigate the effects of different parameters variation on the application layer throughput (AL-TP) and the Mean Time between Failure (MTBF) for multiple-users in EGPRS and UMTS. The MTBF refers to the average time

between FEC block losses and serves as a good criterion of application layer (AL) performance. The MTBF of 3600 sec≅ 1 hour is at least desired for sufficient QoS in streaming applications [13]. H.264 encoded streams at different bitrates are used in channel rate adaptation mode so that the simulation results only highlight the variation in performance of AL-FEC and no system aspects, e.g. congestion, decoder distortion etc. are considered. In all the simulations, we assume random burst losses on the channel. The end-to-end application delay is fixed to $\Delta = 5$ sec in all experiments unless explicitly mentioned. The $\Delta = 5$ sec corresponds to practical end-to-end delay imposed by streaming applications. The number of repair symbols per packet, G, is chosen to satisfy the fragment size F. In the following, we will discuss influence of parameter variations given in section III on the application layer FEC performance based on experimental results.

*1) **Fragmentation (F)**:* Fig. 5 shows the impact on performance for fragmentation size F = {400, 500, 600, 700} bytes with $T = 20$ bytes, $TTI = 80$ms, $p = 10\%$ and bearer rates of 64 kbps in UTRAN. The general tendency of all the curves is to achieve better MTBF with the decrease in AL-TP. For the conventional receiver, better performance can be observed as long as F is smaller than RLC-PDU size. Due to the large RLC-PDU size of 640 bytes in UMTS, the PLR does not show significant gains, as in case of GERAN, because the probability that header is hit increases significantly. Fig. 6

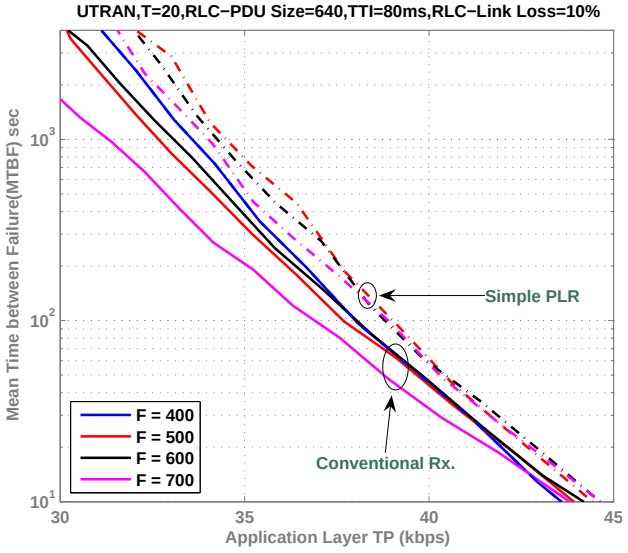


Figure 5. Impact of fragment size (F) variation in UTRAN, Bearer Rate:64 kbps.

shows a similar comparison for GERAN but here smaller fragment size shows the best result as loss of one RLC-PDU of 74 bytes leads to the loss of whole fragment, thus smaller F performs better. However, reducing F further would result in degraded performance due to the outrageous header overheads. Note that the fragmentation size F is less important for the s-PLR for desired MTBF of 3600 sec. Also gains for s-PLR as much as 10 kbps in

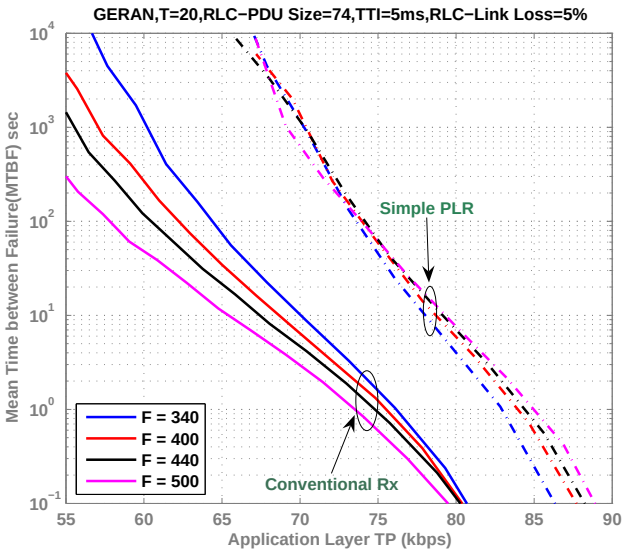


Figure 6. Impact of fragment size (F) variation in GERAN, Bearer Rate: 118.4 kbps.

AL-TP are obvious.

*2) **Symbol Length (T)**:* Fig. 7 highlights the effects on performance with symbol length T = {20, 60, 200, 300, 512} bytes, RLC packet loss rate $p = 10\%$, RLC-PDU size of 640 bytes and $F = 600$ bytes for UTRAN. The number of repair symbols per packet G are selected such that the repair UDP packet payload size satisfies the packet size $P \leq 600$ bytes. All the results show almost similar behavior with different slopes. The code rate r is varied to achieve the targeted $MTBF = 3600$ sec resulting in certain application layer TP. As the bearer rate is fixed to 64 kbps and an end-to-end delay constraint of $\Delta = 5$ sec is imposed, smaller T results in larger k. This would certainly affect the performance of the Raptor code since the Raptor code becomes more efficient for large k. However, it should be noted that this performance gain comes at additional cost of complexity due to larger encoding and decoding matrix. As such, appropriate T needs to be selected to meet the constraints of mobile processing power and required QoS. With the above mentioned system configurations and timing constraints, $T \leq 60$ would be a good compromise choice.

*3) **End-to-End Application Delay** (Δ):* The end-to-end delay Δ is another important constraint in video streaming as it corresponds to the playout waiting time observed by the clients. Longer end-to-end delays are usually annoying for the users and should be avoided. Experimental results for GERAN are shown in Fig. 8 with Δ = {5,20} sec and RLC-PDU size of 74 bytes. Simulations are performed for GERAN as usually high data rates are not available so enhanced end-to-end delay of 20 sec allows a larger source block construction (larger k) and hence better code performance is expected. It is obvious that more gains are possible for streams having higher packet losses as the channel code becomes stronger due to increased channel statistics available for enhanced

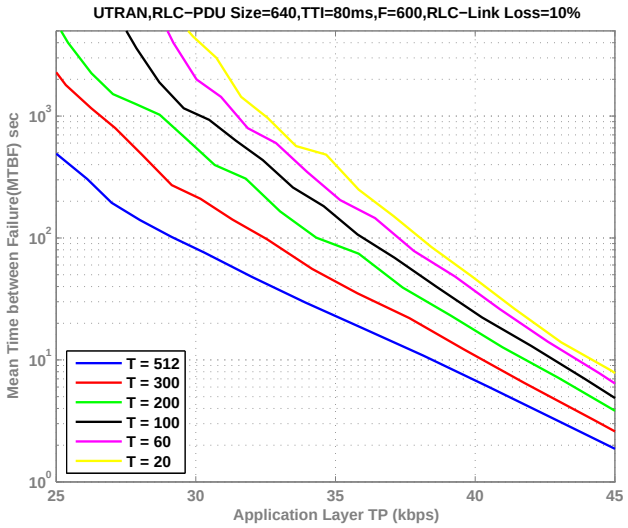


Figure 7. Impact of symbol length (T) variation.

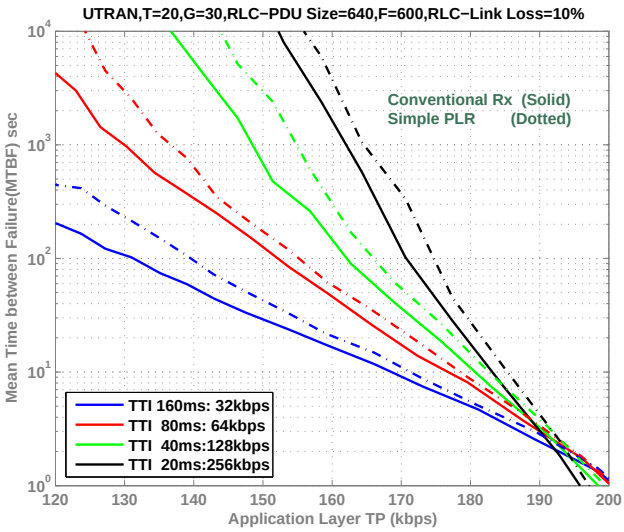


Figure 9. Impact of multimedia stream bundling on AL performance.

end-to-end delay of 20 sec. Therefore, more losses can be recovered, resulting in higher application layer TP. Note that each pair of curve also diverges with a decrease in AL-TP, i.e. decrease in code rate r. This results from systematic Raptor code as the increase in number of repair symbols results in more inter-symbol dependency, which improves decoder performance.

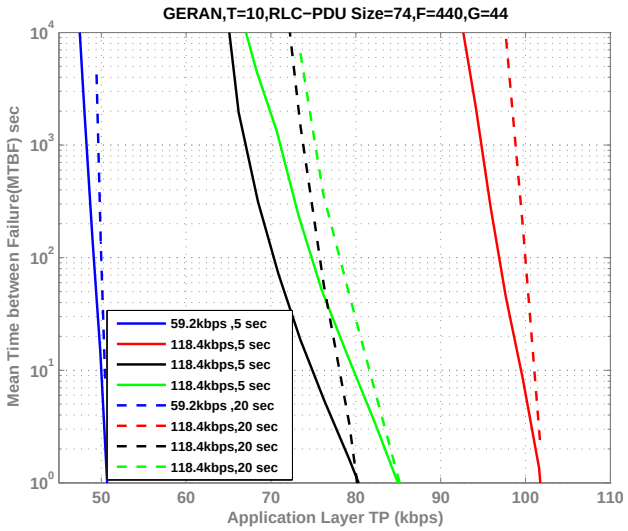


Figure 8. Impact of end-to-end delay (Δ) variation.

*4) **Stream Bundling**:* Here we compare and summarize the results of bundling independent multiple multimedia streams. The simulations are performed to exhibit the behavior of several streams of 32 kbps with and without multiplexing. Two such streams are multiplexed to get an aggregate rate of 64 kbps which is further multiplexed to get 128 and 256 kbps streams. Fig. 9 demonstrates the effects of multiplexing various streams for joint FEC protection. Clearly, the 64 kbps stream outperforms two separately FEC protected 32 kbps streams. It can be observed that at low code rates r (high application layer TP), the FEC protection is not enough to provide error free transmission and hence the $\{32, 64\}$ kbps results are similar. However, with the decrease in code rate the two pair of curves spread out. This results from the better code performance due to multiplexed streams. A similar increase in slope can be observed for further multiplexing two such streams of 64 kbps. This gain results partly from large k. Here the same parameter settings are used for different receivers. The s-PLR does not show much gains due to the RLC-PDU size of 640 bytes, as expected. Therefore the PLR strategy is not further considered for UTRAN.

B. Multiple-User System Level Simulations

The previous section investigates the application layer performance for different parameter choices. The impact of these parameter settings is not obvious due to p-t-M streaming and system effects like congestion control, multi-user effects on video quality, H.264 decoder parameters and distortion, delayed deadline etc. Therefore, the influence of different system design parameters, as explained in section III, is investigated.

To investigate these effects, an alternating news and sports video sequence of 90 sec duration, comprising of 2698 frames, in QCIF format is encoded using H.264/AVC. The baseline profile with 30 fps is used with an IDR refresh rate of 2 sec. This would result in an IDR frame distance of 60 frames which is a good compromise between random access and compression efficiency. RLC-PDU size of 74 and 640 bytes is used with corresponding TTI of 5 and 40 ms resulting in bearer rates of 118.4 and 128 kbps for GERAN and UTRAN, respectively. We assume statistically independent RLC-PDUs losses with probability p which varies for different MBMS clients. The choice of stream bitrate is not obvious due to variable channel losses observed by the users and hence different amount of error protection required. Therefore

we have generated constant bit rate (CBR) video streams at average bitrates of $\{52, 62, 72, 82, 92\}$ kbps. The same streams are used for transmission in both EGPRS and UMTS, i.e. one multicast content server is used.

Initial maximum playout delay constraint of $\Delta_p = 10$ sec is assumed for the application and any frames arriving later are discarded by the receiver. The fragment size $F = \{600, 440\}$ bytes, $T = \{32, 16\}$ with packet size of {600, 440} bytes is used for UTRAN and GERAN, respectively. The parameter G is chosen appropriately such that $P = G \times T$. A protection time window scheme is used i.e. all source RTP packets of variable size arriving in the given time window will be protected in a single source block. The window size is adequately chosen to be 5 sec such that this just results in the recommended source block length $k \geq 1000$ with given T. The encoding block length n varies with source block length k according to some target rate r as $n = k/r$. To avoid system overloading, simple congestion control is applied to the incoming source RTP packets, where packets are dropped by the BMSC if buffers at the lower layer already contain 2 sec of data in the queue. This value is chosen such that the end-to-end delay $\Delta_p = 10$ sec is maintained with the chosen protection period of 5 sec. The performance is evaluated in terms of average PSNR vs code rate r, where the video sequence is looped continuously 24 hours for each experiment. In these simulations, we consider a PSNR of 32 dB to be required for sufficient QoS which meets the practical streaming requirements.

Fig. 10 shows the simulation results for UTRAN for several clients experiencing different channel loss rates p. The system performance is measured in PSNR and the code rate r is varied to demonstrate different system effects. At high code rate r the FEC protection overhead

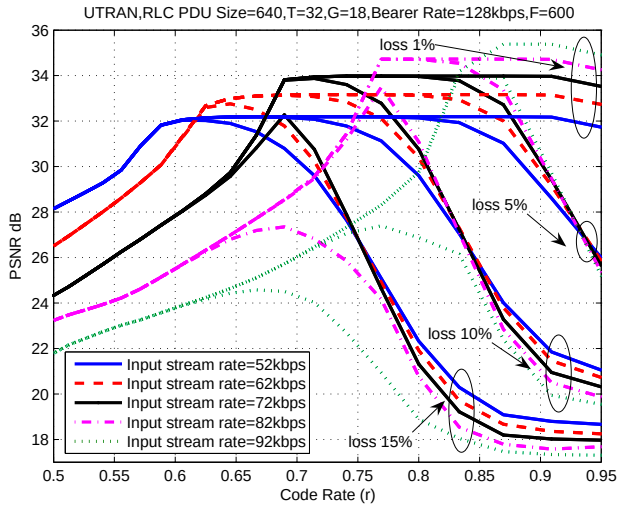


Figure 10. Average PSNR over code rate r for different channel loss rates in UMTS.

is not sufficient to recover the channel losses resulting in low PSNR. As the code rate decreases, PSNR improves gradually and at certain point the FEC is sufficient to receive error-free video and the PSNR saturates due to the encoder distortion. Note that for different users experiencing different packet losses p the choice of code rate r is not trivial. The MBMS clients receive video streams with different PSNR for a given code rate r. Increasing further the FEC overhead does not help but rather overloads the system and causes congestion losses which lowers the PSNR again. The saturation point varies with the input stream rate and packet loss which is obvious due to fixed bearer rate. In general, the packet losses are not constant and the clients are likely to experience loss rates as high as 15%. For example, it can be observed that $r = 0.69$ offers the best PSNR $\approx$ 34 dB for 72 kbps stream as shown in Fig. 10. This operation point is also able to support users with $p = 15\%$ with a PSNR $\approx$ 32.2 dB.

Fig. 11 shows similar behavior for EGPRS with the above mentioned system parameters for users experiencing loss rates of $p = \{0.5, 1, 2, 5\}\%$ which corresponds to typical packet losses observed by MBMS clients. If the system is designed with 72 kbps, a suitable operating point would be $r = 0.77$ resulting in PSNR $\approx$ 34 dB. Note, however, that users suffering from severe channel loss ($p = 5\%$) are not supported. On the other hand, if we sacrifice the PSNR to support users with such high loss rates, the operating point with $r = 0.67$ resulting in PSNR $\approx$ 33 dB for a 62 kbps stream might be more suitable. In this case, sufficient QoS can even be achieved for users with packet loss of $p = 5\%$. However, this strategy implies that video streams with different bitrates have to be multicast for EGPRS and UMTS.

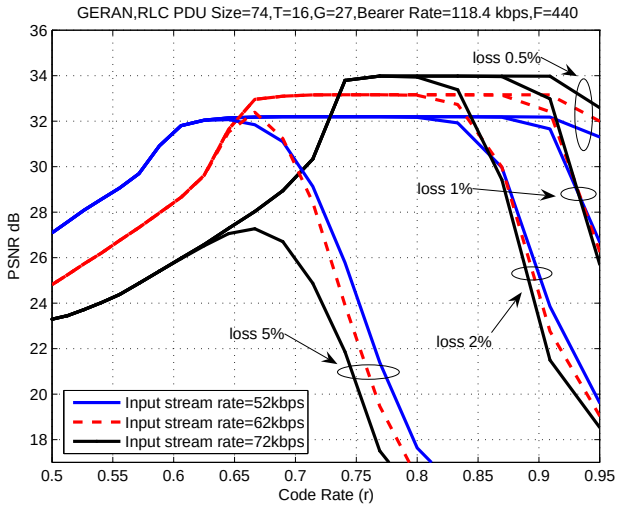


Figure 11. Average PSNR over r for different channel loss rates in EGPRS.

For the same EGPRS parameters, simulation results for PLR are shown in Fig. 12 compared to the conventional receiver. Especially for the higher loss rates, the PLR clearly out performs the conventional receiver and provides similar or better QoS at higher code rates r. It can be observed that the best system operating point, for link rate $p \leq 5\%$ is $r = 0.74$ with a stream of 72 kbps. Note that all the simulated users are now supported with sufficient QoS and also that the PLR provides 1 dB PSNR gain

for each user. Notice that these gains are achievable only if all receivers have PLR implemented and the BMSC is implemented such that the receivers support the PLR. In case that the PLR is optional and the BMSC is not aware of receivers operating the PLR and a system for 2% loss rates at $r = 0.74$ is designed, there is no performance. A client with the PLR can participate anyway in the session despite experiencing 5% loss rates.

Another consequence of the PLR gain is that we can use the same stream of 72 kbps from the content server to the BMSC with optimal performance in EGPRS and UMTS thus saving bandwidth between content server and BMSC. This would also result in less processing power at the BMSC and content server.

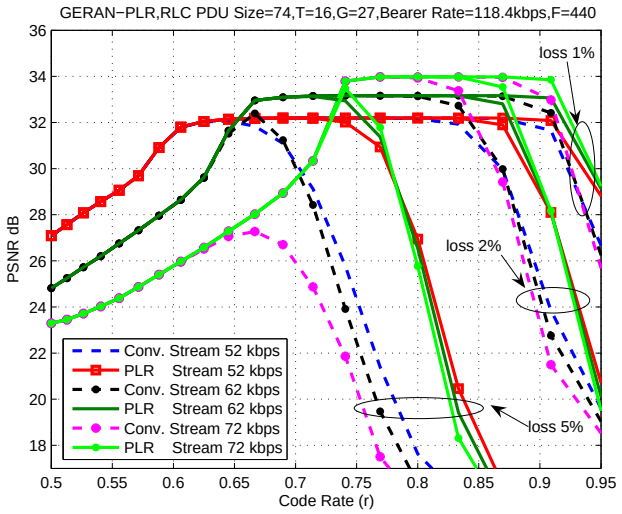


Figure 12. PLR performance in EGPRS with different loss rates.

V. Real Time Video Streaming over MBMS

During the 3GPP MBMS standardization phase, a significant effort has been spent in identifying the cooperation of content delivery protocols and applications with underlying MBMS bearers for GERAN and UTRAN. The huge design, implementation, and realization flexibility on each single layer within the system protocol stack requires a comprehensive treatment of the system to understand its full potentials. Specifically, an optimized design of emerging and future radio systems, transport protocols, or multimedia applications, as well as the combination of those is without any doubt a challenging task. Especially considering the viewpoint of application developers, comprehensive treatment, but also easy usage of such a simulation environment is of major importance.

Conventional offline system simulators are not well suited for investigating real-time services and multimedia applications on dynamic shared wireless networks. Certain effects such as end-user perception, end-to-end QoS and Quality-of-Experience (QoE) cannot be fully understood with offline simulations and/or analytical derivations. Therefore, we have attempted to emulate the performance of MBMS video streaming over GERAN and UTRAN using a standard video streaming system, namely H.264/AVC encoded and decoded video with accompanied audio using QuicktimePlayer7 as well as the Darwin Streaming Server.

The server and the client PC are connected through a gateway which emulates the MBMS streaming protocol stack over GERAN (see Fig. 13). The simulation model

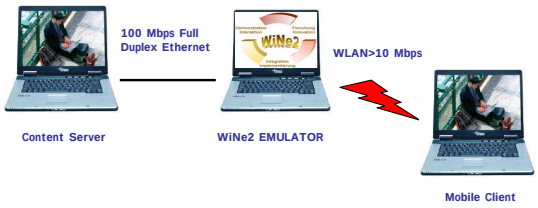


Figure 13. Demonstration Setup: Server, Emulator, Mobile Terminals.

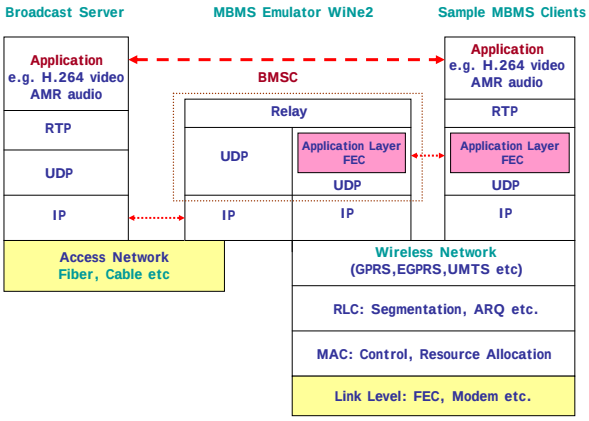


Figure 14. Protocol processing at different PCs for MBMS.

is divided into two parts: an offline link-level simulator that collects statistics about the loss characteristics at the physical layer for a huge set of different parameters which can be loaded as input link configuration file, and a real-time network-level simulator that allows injecting IP-traffic into a virtual wireless network. The basic protocol processing at different PCs is shown in Fig. 14 where the shaded yellow boxes indicate offline generated models and the white area indicates the real-time processing.

The emulator incorporates maximum available features at different protocol layers. The most complex component, however, is the RLC entity, where segmentation/reassembly according to the requirements of a specific bearer service, as well as optional blind repetition of RLC-PDUs, takes place. Moreover, since most IP-traffic is in general assumed to be transported over so-called shared channels at the air interface, the throughput and delay are not only determined by the segment loss behaviour and the chosen bearer service, but also depend largely on the actual resource allocation strategy at the medium access (MAC) layer. Hence, no abstraction is used for the protocol stack below the network layer at the air interface. All higher protocol layers, as well as

the backbone network of the provider, are assumed error-free and over-provisioned which constitutes reasonable assumption for practical systems. Roaming and handover issues are not dealt, thus assuming a certain number of MBMS clients attached to one base station within single cell.

Specifically, the MBMS Emulator, which is called RealNeS-MBMS, supports the following functionalities:

- Setup of several users with different receiving conditions (C/I in case of GERAN),
- Selection of the EGPRS modulation and coding scheme,
- Blind repetitions of RLC-PDUs on the GERAN layer,
- Streaming Framework as specified in 3GPP TS 26.346,
- PDAN (Packet Downlink ACK/NACK) retransmissions as specified in 3GPP TS 43.246,
- Application Layer FEC with different delay and overhead,
- Stream-Bundling,
- Congestion Control to avoid too significant end-to-end delay.

The implementation [14] [15] is kept flexible enough such that different parameters can be changed on-the-fly and the effects on the system and on the application performance can be monitored. Different packet loss profiles with constant and/or variable C/I can be loaded from offline simulation files, thus emulating several participants in the MBMS broadcast. The graphical user interface (GUI) at the emulator PC is also extended to demonstrate the effects of tuning of parameters on different layers throughput (TP) and delays as shown in Fig. 15. The client PC actually allows simultaneous presentation of the quality for MBMS users with different reception conditions using different UDP ports for each user (see Fig. 16). Many interesting and insightful effects of different parameter settings for the system and application performance can be demonstrated.

Figure 15. A snapshot of emulator GUI.

Figure 16. Received Quality for 6 clients with different reception conditions.

VI. Conclusions and Suggestions

This work investigates different system design options for MBMS video streaming over wireless networks such as EGPRS and UMTS using H.264 encoded streams. Goal is to evaluate and to clarify the impact of different parameters on the overall system performance. It has been observed that optimization is not a trivial task as different participants encounter different channel losses.

First we elaborated on what video encoding options should be restricted to the selection of the bitrates and the IDR frequency. Packet length adaptation should be done by the application of fragmentation units. Then, selected simulations provide insight into the MBMS streaming system design, especially the selection of overhead and video bit rates. For example, for UMTS clients with MBMS bearer rates of 128 kbit/s a PSNR of 34 dB for channel losses of 10% can be obtained. However, for EGPRS clients at 118 kbit/s to achieve similar performance, the channel loss rates should be at most 2%. With a permeable layer receiver, we have shown that video streaming with same quality (i.e. PSNR) is still achievable at channel loss rates of 5%, hence accommodating more users. Gains of 1 dB in PSNR can be obtained in MBMS over EGPRS and UMTS, which saves bandwidth and processing power in the network. As an extension of the offline simulations, we have implemented an EGPRS based real-time MBMS emulator, in order to verify the system design options, especially for H.264 encoded video broadcast. Future work may consider the impact of intelligent congestion control techniques based on priority and dependency-information in the NAL header, possibly combined with advanced video encoding modes.

References

[1] 3GPP TR 25.992-140, *Multimedia Broadcast/Multicast Service (MBMS); UTRAN/GERAN requirements*, ETSI, 2003.

[2] T. Stockhammer, H. Jenkac, and W. Xu, "Cross–layer design for wireless multimedia broadcast," in *Signal Processing, vol 86, pp. 1933-1949*, 2006.

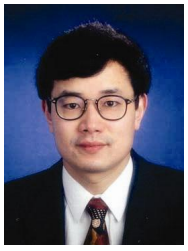
[3] A. Shokrollahi, "Raptor codes," Digital Fountain, Tech. Rep. DR2003-06-001, Jun. 2003.
[4] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu, "Raptor codes for reliable download delivery in wireless broadcast systems," in *Proc. of Consumer and Communications Networking Conference (CCNC)*, Las Vegas, NV, USA, Jan. 2006.
[5] *Advanced Video Coding for Generic Audiovisual Services*, ITU–T and ISO/IEC JTC 1, 2003.
[6] S. Wenger, T. Stockhammer, M. Hannuksela, M. Westerlund, and D. Singer, "RTP payload format for H.264 video," Internet Engineering Task Force (IETF)," RFC3984, Feb. 2005.
[7] 3GPP TS 26.346 V6.1.0, *Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs*, June 2005.
[8] H. Jenkac, T. Stockhammer, and W. Xu, "Cross-Layer Issues and Forward Error Correction for Wireless Video Broadcast," in *Proc. of 16th Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Berlin, Germany, Sept. 2005.
[9] H. Jenkac, G. Liebl, T. Stockhammer, and C. Buchner, "Joint Buffer Management and Scheduling for Wireless Video Streaming," in *Proc. 4th International Conference on Networking (ICN 2005)*, La Reunion, France, Apr. 2005.
[10] H. Jenkac, T. Stockhammer, and W. Xu, "Permeable-Layer Receiver for Reliable Multicast Transmission in Wireless Systems," in *Proc. WCNC 2005*, New Orleans, LA USA, Mar. 2005.
[11] J. Afzal, T. Gasiba, T. Stockhammer, and W. Xu, "System design and advanced receiver techniques for mbms broadcast services," in *IEEE International Conference on Communications (ICC)*, Istambul, Turkey., June 2006.
[12] TSG System Aspects, *Advanced Receiver for MBMS FEC*, Siemens, TSG WG4 # 34, Lisbon (Portugal), Feb. 2005.
[13] 3GPP TSG-SA WG4 S4-AHP252, *FEC Simulation Parameters and Assumtions for GERAN* , PSM SWG, Ad Hoc Meeting, Sophia Antipolis, France, Apr. 2005.
[14] T. Stockhammer, G. Liebl, H. Jenkac, P. Strasser, D. Pfeifer, and J. Hagenauer, "Wine2 wireless network demonstration platform for ip-based real-time multimedia transmission," in *International Packet Video Workshop*, Nantes, France., April 2003.
[15] T. Stockhammer, J. Afzal, C. Buchner, W. Xu, and A. Arnold, "Demonstration of mbms video streaming over geran," in *Proc. of Consumer and Communications Networking Conference (CCNC)*, Las Vegas, NV, USA., 2006.

Junaid Afzal was born in Karachi, Pakistan. He received his M.Sc. degree in telecommunication engineering from the Technical University of Munich (TUM) Germany in 2005, and his B.Sc. degree in electrical engineering from the University of Science and Technology Taxila, Pakistan in 2003.

He is currently working as Software Design Engineer at NoMoR Research GmbH, Munich, Germany. He is involved in the development of real-time simulation platforms for MBMS and HSPA. His research interests include wireless communication systems, forward error correction codes, multimedia streaming and video coding.

Thomas Stockhammer has been working at the Munich University of Technology, Germany and was visiting researcher at Rensselear Polytechnic Institute (RPI), Troy, NY and at the University of San Diego, California (UCSD). He has published more than 80 conference and journal papers, is member of different program committees and holds several patents. He regularly participates and contributes to different standardization activities, e.g. JVT, IETF, 3GPP, and DVB and has co-authored more than 100 technical contributions. He is acting chairman of the video adhoc group of 3GPP SA4. He is also co-founder and CEO of Novel Mobile Radio (NoMoR) Research, a company developing simulation and emulation of future mobile networks such as HSxPA, WiMaX, MBMS, and LTE. The company also provides consulting services in the respective areas. Between 2004 and June 2006, he was working as a research and development consultant for Siemens Mobile Devices, now BenQ mobile in Munich, Germany. Now he is consulting for Digital Fountain, Inc. His research interests include video transmission, cross-layer and system design, forward error correction, content delivery protocols, rate-distortion optimization, information theory, and mobile communications.

Tiago Gasiba was born in Oporto, Portugal. He received his M.Sc. degree in telecommunication engineering from the Technical University of Munich (TUM) Germany in 2004, and his Eng. degree in electrical engineering and computer science from the Faculdade de Engenharia da Universidade do Porto in 2002. He is currently working for Digital Fountain and NoMoR Research GmbH and working towards his PhD degree under the supervision of Prof. Hagenauer and Prof. Shokrollahi. In 2005 he was a visiting researcher at the Laboratoire d'Algorithmique et Laboratoire de Mathematiques Algorithmique (Algo+Lma) in Lausanne, Switzerland. His current research interests include forward error correction codes in particular fountain codes, wireless communications networks and video and data broadcast.



Wen Xu (wen.xu@ieee.org) received a B.Sc. degree in 1982 and an M.Sc. degree in 1985 from Dalian University of Technology (DUT), China, and a Dr.-Ing. (Ph.D.) degree in 1996 from Munich University of Technology (TUM), Germany, all in electrical engineering. Since 1995 he has been with the Siemens AG - Mobile Phones (now BenQ Mobile), Munich, where he is responsible for several RnD projects and has actively participated and contributed to standardization activities of ETSI and 3GPP. Since 2000 he is head of the Baseband Algorithms and Standardization Lab which is responsible for physical layer and multimedia signal processing, and protocol stack aspects. As a competence center, his lab has been actively involved in different standardization activities such as 3GPP and DVB for 2G, 3G, beyond 3G mobile systems as well as DVB-H system. His research interests include image/video/speech coding and processing, channel coding, equalization, cross-layer system design, and mobile communications. Dr. Xu is a senior member of IEEE and a member of the Verband der Elektrotechnik, Elektronik, Informationstechnik (VDE), Germany.