Evaluating RaptorQ FEC over 3GPP Multicast Services

Christos Bouras*†, Nikolaos Kanakis†, Vasileios Kokkinos*†, Andreas Papazois*†
*Computer Technology Institute and Press "Diophantus", Patras, Greece
†Computer Engineering and Informatics Department, University of Patras, Greece
bouras@cti.gr, kanakisn@ceid.upatras.gr, kokkinos@cti.gr, papazois@ceid.upatras.gr

***Abstract*—Third Generation Partnership Project (3GPP) introduced Multimedia Broadcast/Multicast Services (MBMS), in order to efficiently deliver the same content to multiple users over mobile cellular networks. To enhance the download and streaming delivery robustness against packet losses, 3GPP proposes the use of Forward Error Correction (FEC) on the application layer. The Application Layer FEC (AL-FEC) scheme that 3GPP standardized is based on the systematic, fountain Raptor code. In this work, in response to the emergence of an enhanced FEC scheme, we provide a performance evaluation of the existing standardized AL-FEC scheme with the newly introduced RaptorQ code. Introducing a variety of simulation experiments, we analyze the performance behavior of each AL-FEC scheme considering several FEC encoding parameters. Furthermore, we try to verify the improvements that have emerged in AL-FEC providing enhanced capabilities in multicast services over next generation mobile networks.**

***Index Terms*—forward error correction, raptor code, raptorq code, mobile networks, broadcast and multicast**

I. Introduction

Third Generation Partnership Project (3GPP) has firstly standardized Multimedia Broadcast/Multicast Services (MBMS) as a new feature in Release 6 specifications, in order to provide efficient delivery of data in a one-to-many way to multiple users located in a specific service area. The MBMS standard defines two delivery methods, the download and the streaming delivery method [1]. The purpose of MBMS download is to deliver content in files, while in MBMS streaming method continuous multimedia data are transmitted through a MBMS bearer.

Forward Error Correction (FEC) is an error control method where redundant data are transmitted in advance with the source information, in order to obtain the recipients the ability to overcome packet losses. The use of FEC provides particular advantages in one-to-many protocols such as the MBMS environment where common methods (e.g. ARQ) have a lot of limitations [2]. The FEC encoding eliminates the effect of independent losses at different receivers, while the dramatic reduction in the packet losses largely reduces the need of a back channel for retransmission requests.

In order to enhance the reliability of multicast services, 3GPP introduced the use of an Application Layer FEC (AL-FEC) scheme for both of the defined MBMS delivery methods. The chosen AL-FEC scheme is based on the systematic, fountain Raptor code [1], due to the Raptor FEC higher performance compared with existing AL-FEC codes.

Several works cover research on the application of AL-FEC over 3GPP MBMS environments. The authors of [3] provide an analytical investigation of the Raptor FEC performance, evaluating the tradeoffs between AL-FEC and physical layer FEC over MBMS download delivery for 3G systems. The work presented in [4] studies the Raptor FEC application both for download and streaming MBMS services over 3G mobile cellular networks considering the impacts of AL-FEC on the telecommunication cost. In [5], the authors evaluate the performance of Raptor codes over evolved MBMS (eMBMS) deployment, providing a novel proposal for the application of AL-FEC over the download delivery method. The work in [6] provides a performance evaluation of Raptor FEC for streaming services over Long Term Evolution (LTE) single-cell MBMS environments, examining several system and FEC encoding parameters.

A comparative analysis of the performance of Raptor FEC codes against Reed Solomon (RS) FEC codes is presented in [7]. The authors of this work provide a detailed comparison between the two FEC schemes considering several performance parameters over MBMS services. Furthermore, a comprehensive analysis of the processes behind the design and the performance of Raptor and RaptorQ FEC codes is provided in [8]. Apart from these works, during the standardization process of Raptor codes, 3GPP released several evaluation documents comparing candidates AL-FEC schemes. Indicatively, we cite [9] where presents simulation results that compare the performance of Raptor codes with that of RS codes over MBMS download and streaming services. Finally, the work in [10] contributes valuable results, comparing the performance of Raptor codes to that of RS codes and furthermore to that of an ideal FEC code over MBMS streaming environments.

Since the performance of Raptor FEC over 3GPP MBMS services seems to be a well investigated field considering several perspectives, our persuasion is to examine the improvements that have emerged on AL-FEC since the Raptor code was adopted from 3GPP. In this work, based on the occurrence of an enhanced FEC scheme of the Raptor codes family, named RaptorQ, we present an extensive comparison of the newly introduced RaptorQ FEC with that of the 3GPP standardized AL-FEC scheme, considering both functional aspects and investigating the benefits that the most promising version of Raptor codes can provide to the performance of mobile multicast services. To this direction, we provide an analytical investigation of the improved efficiency of the RaptorQ code with respect to the standardized Raptor FEC scheme over 3GPP MBMS services for beyond 3G mobile network standards.

The rest of this paper is organized as follows: in Section II we provide an overview of the 3GPP AL-FEC MBMS download framework and Section III presents a detailed description of the examined AL-FEC schemes. In Section IV we present the simulation environment and the conducted experimental results. Finally, in Section V we provide our conclusions and some possible future steps.

II. AL-FEC MBMS Delivery

MBMS was first specified in 3GPP Release 6 for the delivery of content in a point-to-multipoint (ptm) manner to mobile recipients. MBMS aims to provide an efficient mode of delivery for broadcast and multicast services allowing resources to be shared in the core and radio access network of 3GPP systems. 3GPP defines three distinct functional layers for the delivery of MBMS services: the user service, the delivery method and the bearers. The MBMS user service enables applications. Different applications impose different requirements when delivering content to MBMS subscribers and may use different MBMS delivery methods. Two delivery methods are defined, namely download and streaming, which make use of MBMS bearers in order to distribute an application to multiple receivers. Finally, the MBMS bearers provide the mechanism by which IP data are transported since are used to distribute multicast and broadcast traffic in an efficient one-to-many way.

MBMS download delivery method aims to distribute discrete objects (e.g. files) by means of a MBMS download session. Download method uses the FLUTE protocol when delivering content over MBMS bearers. FLUTE is built on top of the Asynchronous Layered Coding (ALC) protocol instantiation. ALC combines the Layered Coding Transport (LCT) building block and the FEC building block to provide reliable asynchronous delivery of content to an unlimited number of concurrent receivers from a single sender. A detailed description of the FLUTE building block structure can be found in [1]. Thereafter, FLUTE is carried over UDP/IP, and is independent of the IP version and the underlying link layers used. The MBMS download delivery protocol stack is illustrated in Fig. 1.

As mentioned above, a key aspect of both MBMS delivery methods is the provision of reliability control by means of a FEC technique. More precisely, 3GPP has standardized an AL-FEC scheme that is based on the systematic fountain Raptor code. On MBMS download delivery, the file is partitioned in one or several source blocks. Each source block consists of k source symbols, each of length T except from the last source symbol, which can be smaller. Through the Raptor encoding, for each source block, redundant repair symbols are generated according to the desired amount of protection. A unique ID is assigned on each resulting encoding symbol, which can be a source or a repair symbol, in order to identify the type of the symbol. Subsequently, one or more FEC encoding symbols are placed in each FLUTE packet payload and thereafter the resulting packets are encapsulated in UDP and distributed over the IP multicast MBMS bearer.

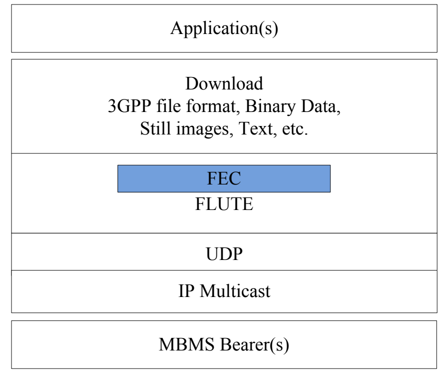

Fig. 1. 3GPP MBMS Protocol Stack

Furthermore, 3GPP defines a post-delivery procedure to enable file repair capabilities for the MBMS download delivery. A MBMS client is able to determine the missing symbols for each source block of each file and send a file repair request for unreceived symbols which will allow the receiver to recover the file. Thereafter, the MBMS client can receive the repair data through a point-to-point (ptp) or a point-to-multipoint (ptm) delivery.

III. AL-FEC Schemes

In this section we provide a detailed description of the 3GPP standardized Raptor code and the newly introduced RaptorQ code, focusing on the improvements that RaptorQ has emerged, in comparison with the older member of the Raptor codes family.

A. The standardized Raptor code

Raptor codes were introduced in [11]. The use of Raptor codes in the application layer of MBMS has been introduced to 3GPP by Digital Fountain aiming to improve service robustness against packet losses. Raptor codes are fountain codes, meaning that as many encoding symbols as desired can be generated by the encoder on-the-fly from the source symbols of a source block of data. Raptor codes are one of the first known classes of fountain codes with linear encoding and decoding time [11]. In preparation of the encoding, a certain amount of data is collected within a FEC source block. The data of a source block are further divided into k source symbols of a fixed symbol size. The decoder is able to recover the whole source block from any set of encoding symbols only slightly more in number than the source symbols. The Raptor code specified for MBMS is a systematic code producing n encoding symbols E from $k<n$ source symbols C, so as the original source symbols are within the stream of the transmitted symbols. This code can

be viewed as the concatenation of several codes. The most-inner code is a non-systematic Luby-Transform (LT) code [12] with l input symbols F, which provides the fountain property of the Raptor codes. This non-systematic Raptor code is not constructed by encoding the source symbols with the LT code, but by encoding the intermediate symbols generated by some outer high-rate block code. This means that the outer high-rate block code generates the F intermediate symbols using k input symbols D. Finally, a systematic realization of the code is obtained by applying some pre-processing to the k source symbols C such that the input symbols D to the non-systematic Raptor code are obtained. The description of each step and the details on specific parameters can be found in [1].

Considering the performance of Raptor codes the most typical comparison is that to an ideal fountain code. An ideal fountain code can produce from any number k of source symbols any number m of repair symbols with the property that any combination of k of the $k+m$ encoding symbols is sufficient for the recovery of the k source symbols. That is the point of the most important differentiation between an ideal fountain code and the standardized Raptor code. While an ideal code has zero reception overhead i.e., the number of received symbols needed to decode the source symbols is exactly the number of source symbols, the Raptor code has a performance close to that property. The performance of an AL-FEC code can be described by the decoding failure probability of the code. The study presented in [13] describes the decoding failure probability of Raptor code as a function of the source block size and the received symbols. In fact, the inefficiency of the Raptor code can accurately be modeled by (1):

$$p_{f_R}(n,k) = \begin{cases} 1, & \text{if } n < k \\ 0.85 \times 0.567^{n-k}, & \text{if } n \geq k \end{cases} \tag{1}$$

In (1), $p_{f_R}(n,k)$ denotes the decoding failure probability of the Raptor code if the source block size is k symbols and n encoding symbols have been received. It has been observed that for different k, the equation almost perfectly emulates the Raptor performance. While an ideal fountain code would decode the protected data with zero failure probability when n=k, the Raptor failure probability is still about 85%. However, the failure probability decreases exponentially when the number of received encoding symbols increases. Moreover, a crucial point for the robustness of an AL-FEC protected delivery session is the transmission overhead. The transmission overhead is defined as the amount of redundant information divided by the amount of source data and is equal to the fraction $(N-K)/K$ in terms of percentage. In this fraction, N denotes the number of transmitted packets and K denotes the number of the source packets.

B. The new RaptorQ code

In the interval since the systematic fountain Raptor code was adopted from 3GPP as the standardized AL-FEC scheme for MBMS, there has been significant progress in the design of erasure codes. An enhanced Raptor code has been emerged at Internet Engineering Task Force (IETF) [14] in order to address the drawbacks of the standardized Raptor code. This newer member in Raptor codes family is known as RaptorQ code. RaptorQ is a significantly more efficient FEC code than the older Raptor code, in terms of superior flexibility, support for larger source block sizes and better coding efficiency. RaptorQ is also a fountain and systematic FEC code. The encoding process of RaptorQ code is almost identical with that of Raptor code described in the previous subsection. A key differentiation between the two schemes is that the standardized Raptor code operates over Galois field GF(2) [15], while the enhanced RaptorQ code uses symbol operations over GF(256) [14] instead of over GF(2). Operating over larger finite fields allows RaptorQ to overcome the performance limitations of Raptor code since the operation over larger finite fields offers the potential of achieving recovery with lower reception overhead than the existing Raptor code. Moreover, additional important aspects of the enhanced properties of RaptorQ codes are the incremented number of possible source symbols and the incremented number of generated encoding symbols. More precisely, RaptorQ can encode up to 56403 source symbols into a source block in contrast to 8192 of the Raptor code and furthermore can generate up to 16777216 encoding symbols, 256 times more than the older Raptor code. Expanding the range of these two parameters simplifies the application of the AL-FEC protection. Based on the properties of RaptorQ code, it is obvious that can perform better and more flexible both for file delivery and streaming services. Since RaptorQ can deliver files up to 3.4 GB as a single source block, this property maximizes the decoding efficiency due to the spreading of protection across the whole file, particularly for very large files. On the delay-sensitive real-time applications, the flexible range of the block size parameter allows to determine a QoS trade-off between protection and latency considering the delay constraints of the transmitted application. At the same time RaptorQ achieves lower computational complexity [16] than the older Raptor code.

Concerning the performance of RaptorQ, the key property of a Raptor codes member is the probability of a successful decode as a function of the received symbols similar to that of the standardized Raptor code described above. The decoding failure probability of RaptorQ code can be modeled by (2) [16]:

$$p_{f_{RQ}}(n,k) = \begin{cases} 1, & \text{if } n < k \\ 0.01 \times 0.01^{n-k}, & \text{if } n \geq k \end{cases} \tag{2}$$

In (2), $p_{f_{RQ}}(n,k)$ denotes the probability of a failed decode of a RaptorQ protected block with k source symbols if n encoding symbols have been received. Comparing (2) with (1), the performance superiority of RaptorQ code is unambiguous.

C. Functional Comparison

The efficiency superiority of the newly introduced RaptorQ code derives from several design aspects. Although the majority of the basic encoding steps of RaptorQ are identical to those of Raptor code, there are several improvements and additions to the encoding and decoding operations. On RaptorQ before the intermediate symbol generation, for a given source block of k source symbols, the source block is augmented with additional padding symbols for encoding and decoding purposes. The reason for padding out a source block is to enable faster encoding and decoding and to minimize the amount of information that needs to be stored. The following step is the generation of the intermediate symbols from the source symbols where enhanced generator and pre-coding relationships (i.e., a two-stage pre-coding algorithm using LDPC and HDPC codes) are used, compared to the older Raptor code. Finally, in the second encoding step of RaptorQ, a modified, more efficient encoding process, than this of Raptor code, is applied in order to generate the encoding symbols. The number of encoding symbols RaptorQ can generate, is 256 times more than the foregoing Raptor code.

For the encoding procedure, Raptor code uses simple exclusive-or operations over the symbols, i.e. operations over GF(2). This selection limits the recovery properties of Raptor code, since the best recovery probability such a code can achieve is $1 - \frac{1}{2^{m+1}}$ if $k + m$ encoding symbols have been received. RaptorQ code introduces the use of arithmetic operations on octets. Mathematically, octets can be thought of as elements of a finite field, i.e., the finite field GF(256). Using symbol operations over GF(256) achieves recovery from the reception of $k + m$ encoding symbols with probability $1 - \frac{1}{256^{m+1}}$. In order to avoid increasing the computational complexity, RaptorQ uses a clever combination of GF(256) and the low-complexity GF(2) operations, so that the vast majority of the symbol operations are over GF(2) and only a small minority are over GF(256). All these functional choices of RaptorQ, in conjunction with an advanced decoding algorithm, result in the very close to an ideal fountain code performance described by (2).

IV. Experimental Evaluation

A. Simulation Environment

In order to evaluate the performance of the two examined AL-FEC schemes over a 3GPP MBMS environment we utilize an open-source simulation platform [17]. This platform allows simulating the performance of an evolved 3GPP access network, named evolved UMTS Terrestrial Radio Access Network (eUTRAN). For the purposes of our simulation procedure we apply only a few required modifications, in order to simulate an AL-FEC protected MBMS download delivery session. More precisely, as illustrated in Fig. 1 the simulator generates packets at the application layer, which are protected through a modeled AL-FEC framework. At the AL-FEC encoder the parameters that can be selected are the transmission overhead and the number of source symbols protected together within a source block. After the applied FEC protection, the resulting UDP flows are mapped on a MBMS IP multicast bearer and transmitted to the mobile users through a modeled broadcast channel.

For the conduction of the simulation process the single-cell MBMS mode is selected, where the MBMS download service is transmitted to a center MBMS cell with an adjacent ring of 6 cells acting as intercell interference. Moreover, concerning the AL-FEC encoding process the symbol size is fixed for both Raptor and RaptorQ FEC scheme and each resulting packet contains one single FEC encoding symbol. For the performance evaluation of each AL-FEC scheme, we examine the FEC decoding failure probability on each multicast receiver based on the previously described equations (1) and (2), according to the amount of the received encoding symbols. We assume that a sufficient failure probability threshold to consider that the FEC decoding process never fails is 10^{-4}. Moreover, the system parameters that are taken into account for the performed simulations are presented in Table I.

TABLE I
Simulation Settings

Parameter	Units	Value
Cell Layout		Hexagonal grid, 3 sectors per site
Inter Site Distance	m	1732
Carrier Frequency	MHz	2000
System Bandwidth	MHz	5/10
Channel Model		3GPP Typical Urban (TU)
Propagation Model		3GPP Macrocell - Urban Area
Path Loss	dB	$L = 128.1 + 37.6 log_{10} d$ *
# UE Rx Antennas		2
# BS Antennas		1
BS Transmit Power	dBm	43
BS Antenna Gain	dBi	14

*d: distance between eNB and UE in km

B. Decoding Performance vs. Reception Overhead

In this paragraph we draw one of the most important points featuring the efficiency of an AL-FEC scheme, i.e., the decoding performance compared to the reception overhead such a FEC code requires to successfully recover the protected data. Fig. 2 presents the probability the FEC decoding process to fail in function to the number of additional symbols received, i.e., the reception overhead, comparing the performance of the standardized Raptor FEC code with that of RaptorQ.

Comparing the two FEC schemes performance, although Raptor failure probability decreases exponentially with the growth of the number of additional FEC symbols, the RaptorQ decoding performance supremacy almost eliminates this behavior of Raptor code. Indicatively, while RaptorQ requires only two additional symbols to succeed a practically zero failure probability, Raptor code requires to receive more than 20 additional symbols as indicated in Fig. 2. Based on this, we can say that RaptorQ almost perfectly emulates an ideal fountain FEC code.

The minimum requirements of RaptorQ code over the number of additional symbols have a direct impact on some extremely important aspects of an AL-FEC scheme efficiency. Reception overhead characterizes the robustness of a FEC code against packet losses, meaning that RaptorQ FEC can operate successfully under poorer reception conditions than Raptor code, since, on condition that more symbols than the number of source symbols have been received, RaptorQ can tolerate higher packet losses than Raptor code can, because of the lower required reception overhead. A direct consequence of this property is that the protection scheme of RaptorQ FEC can be successfully applied requiring significantly lower amount of redundancy. This fact implies that RaptorQ can provide enhanced protection while achieving high reduction on the required transmission overhead and the encoding process overhead.

C. MBMS Coverage vs. AL-FEC Transmission Overhead

In this part of the provided simulations, we illustrate how the amount of the introduced AL-FEC transmission overhead affects the fraction of multicast users that can successfully receive the transmitted object based on the previously described assumptions. For this evaluation, we simulate 100 mobile multicast User Equipments (UEs) participating in a AL-FEC protected download session, randomly dropped in the MBMS service area and moving at 3 km/h corresponding to pedestrian mobility model. The presented results refers to the 3GPP standardized Raptor FEC code and the new RaptorQ scheme. Fig. 3a and Fig. 3b present the impact of the AL-FEC transmission overhead increase on the MBMS service coverage, simulating 5 and 10 MHz system bandwidth respectively.

From both figures, we can immediately remark that the proportion of UEs that can successfully decode the AL-FEC protected object, i.e., the MBMS service coverage, increase proportionally to the introduced transmission overhead in the multicast download session. This is not surprising, since increasing the amount of the transmitted FEC data leads to ever more UEs receiving successfully the MBMS service with respect to the particular reception conditions. Furthermore, another remark over the two figures is that increasing the system bandwidth from 5 to 10 MHz results in remarkable gains in the required AL-FEC transmission overhead. More precisely, in order to achieve complete MBMS service coverage considering the RaptorQ scheme the required overhead is 12% for the 5 MHz case, while selecting 10 MHz system bandwidth results in large reduction to the required transmission overhead

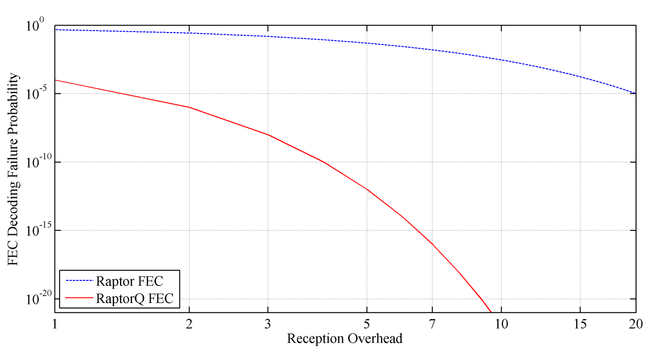

Fig. 2. FEC Decoding Failure Probability vs. Reception Overhead

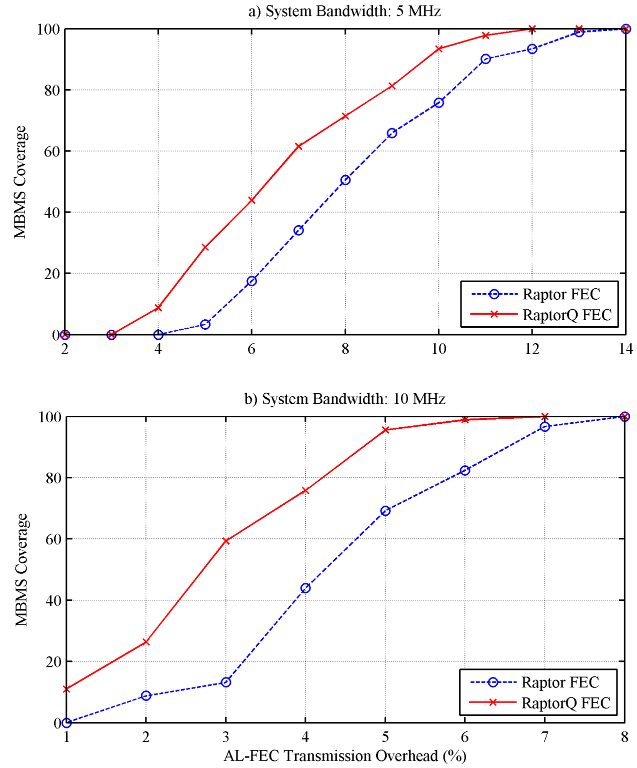

Fig. 3. MBMS Coverage vs. AL-FEC Transmission Overhead for a) 5 MHz and b) 10 MHz System Bandwidth

in half of the 5 MHz case amount. This reduction is reasonable and anticipated since increasing the system bandwidth leads to enhanced system throughput and consequently to reduced packet losses at each individual UE.

Furthermore, as regards the comparative evaluation of the two examined AL-FEC schemes the performance superiority of the RaptorQ FEC scheme against the Raptor FEC is clearly visible from both figures, although in our simulation setup we do not use the maximum values of encoding parameters that each FEC scheme supports but we select for both modeled AL-FEC codes the number of source symbols protected together within a source block fixed at 1024. It is obvious from the reception overhead performance of each code that RaptorQ is notably more efficient for small values of source symbols since as the value of k increase the supremacy of RaptorQ in the number of additional required symbols gradually eliminates. Finally, we observe that in Fig. 3b the dominance of RaptorQ becomes more pronounced since the wider system bandwidth results in lower packet losses and this fact in conjunction with the remarkable higher decoding efficiency of RaptorQ leads to even smaller amount of required transmission overhead in order a UE to recover the AL-FEC protected object.

D. Session Overhead vs. AL-FEC Transmission Overhead

In this last part of the simulation results, Fig. 4 presents the impacts of the two evaluated AL-FEC schemes on the overall session overhead i.e., the total number of additional

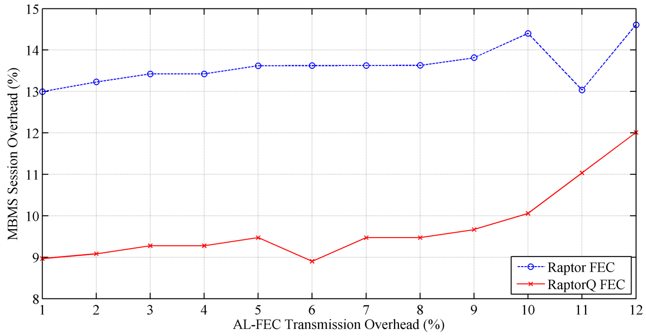

Fig. 4. Session Overhead vs. AL-FEC Transmission Overhead

transmitted packets divided by the number of the source packets. We examine the application of the Raptor and RaptorQ FEC scheme in conjunction with a ptm file repair procedure, as described in Section II, until all UEs can recover the transmitted object. For this evaluation we simulate 100 MBMS pedestrian UEs with the system bandwidth set at 5 MHz and the length of each FEC source block fixed at 1024 symbols.

The plotted results, immediately reveal the primacy of RaptorQ FEC since achieves significantly lower number of transmitted packets compared to Raptor FEC. This is anticipated considering that RaptorQ operates with lower reception overhead resulting in lower number of required additional packets, both at the FEC decoding procedure as well as at the file repair procedure.

Furthermore, an interesting remark is that each plotted curve reveals an optimal value of the introduced transmission overhead that minimizes the total number of transmitted packets. It is obvious that this point defines an optimal trade-off between the amount of introduced redundancy and the number of retransmitted packets since lower values of transmission overhead seems to have minimum effects on the FEC recovery and on the other hand further increase the transmission overhead results in increasing the amount of transmitted data without additional gains in the object recovery.

V. Conclusions & Future Work

In this work we have conducted a performance evaluation of the newest member in the Raptor codes family, named RaptorQ, compared to the 3GPP standardized Raptor FEC code. For the purposes of our investigation we have provided an extensive comparison of the two examined AL-FEC schemes considering functional aspects and moreover we have presented several simulation results comparing the impacts of the Raptor and RaptorQ code application over a 3GPP MBMS environment for beyond 3G mobile networks.

From the conducted results we have verified the enhanced efficiency of the new RaptorQ FEC scheme and we have examined how the theoretical superiority of RaptorQ is reflected in a 3GPP MBMS download delivery environment. The almost ideal behavior of RaptorQ concerning the reception overhead allows operating with significantly lower transmission redundancy with respect to the current reception conditions, compared to the standardized Raptor FEC. Concluding, it is clear that RaptorQ code can provide enhanced capabilities in multicast environments and the adoption of that new AL-FEC scheme is expected by several standards.

Regarding possible future steps, we could evaluate the performance of RaptorQ FEC over multicast streaming services considering the impacts of the new AL-FEC scheme on the specific constraints of a streaming environment. Furthermore, it is our belief that an adaptive mechanism for the selection of the introduced AL-FEC redundancy based on a feedback report scheme, would certainly be beneficial for the efficiency of AL-FEC over multicast services.

References

[1] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (Release 10)," 3rd Generation Partnership Project (3GPP), TS 26.346, 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/26346.htm

[2] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "The Use of Forward Error Correction (FEC) in Reliable Multicast," RFC 3453 (Informational), Internet Engineering Task Force, Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3453.txt

[3] M. Luby, M. Watson, T. Gasiba, and T. Stockhammer, "Mobile data broadcasting over MBMS tradeoffs in forward error correction," in *Proceedings of the 5th international conference on Mobile and ubiquitous multimedia (MUM)*, 2006.

[4] A. Alexiou, C. Bouras, and A. Papazois, "A study of forward error correction for mobile multicast," *Int. J. Commun. Syst.*, vol. 24, pp. 607–627, May 2011.

[5] A. Alexiou, C. Bouras, V. Kokkinos, A. Papazois, and G. Tseliou, "Enhancing FEC application in LTE cellular networks," in *Wireless Days (WD), 2010 IFIP*, Oct. 2010.

[6] C. Bouras, N. Kanakis, V. Kokkinos, and A. Papazois, "AL-FEC for Streaming Services over LTE Systems," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, Oct. 2011, pp. 1–5.

[7] U. Demir and O. Aktas, "Raptor versus Reed Solomon forward error correction codes," in *Computer Networks, 2006 International Symposium on*, 2006, pp. 264 –269.

[8] M. A. Shokrollahi and M. Luby, "Raptor Codes," *Foundations and Trends in Communications and Information Theory*, vol. 6, no. 3-4, pp. 213–322, 2011.

[9] 3GPP, "Simulation results for the performance and complexity of RS codes for MBMS FEC," 3rd Generation Partnership Project (3GPP), Tdoc S4-050107, 2005.

[10] 3GPP, "Streaming Simulations of FEC codes for MBMS," 3rd Generation Partnership Project (3GPP), Tdoc S4-040649, 2004.

[11] A. Shokrollahi, "Raptor codes," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[12] M. Luby, "LT codes," in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, 2002, pp. 271–280.

[13] T. Stockhammer, A. Shokrollahi, M. Watson, M. Luby, and T. Gasiba, "Application Layer Forward Error Correction for mobile multimedia broadcasting," *Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T and Media Flo, CRC Press*, pp. 239–280, 2008.

[14] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery," RFC 6330, Internet Engineering Task Force, Aug. 2011. [Online]. Available: http://tools.ietf.org/rfc/rfc6330.txt

[15] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery," RFC 5053 (Proposed Standard), Internet Engineering Task Force, Oct. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5053.txt

[16] 3GPP, "Rationale for MBMS AL-FEC Enhancements," 3rd Generation Partnership Project (3GPP), Tdoc S4-110449, 2011.

[17] G. Piro, L. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 2, pp. 498 –513, Feb. 2011.