Enhancing Reliable Mobile Multicasting with RaptorQ FEC

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*—A crucial point on the successful deployment of multicast services is the enhanced reliability by means of advanced error control schemes. To this direction Third Generation Partnership Project (3GPP) has standardized exclusively for Multimedia Broadcast/Multicast Services (MBMS) the use of a Forward Error Correction (FEC) mechanism on the application layer based on Raptor codes. Since the standardized systematic, fountain Raptor code is now considered obsolete, the emergence of a new variant of Raptor codes, named RaptorQ, provides enhanced capabilities for mobile multicast services. In this paper, we provide a performance comparison of the standardized Raptor Application Layer FEC (AL-FEC) scheme with the very promising RaptorQ FEC code. We examine the enhancements that RaptorQ introduces on the AL-FEC protection robustness, providing a thorough performance analysis in comparison with the current Raptor FEC scheme. Furthermore, in order to verify the improved performance of RaptorQ, we provide several simulation results utilizing the ns-3 environment, considering the application of both the examined AL-FEC schemes over multicast services for next generation mobile networks.**

***Keywords*-forward error correction, raptor code, raptorq code, mobile telecommunications, reliable multicasting, network simulator 3**

I. Introduction

Broadcast and multicast are techniques specified to efficiently transmit datagrams from a single source entity to multiple destinations. For multiple mobile subscribers, a broadcast and multicast service allow to share radio and core network resources and therefore offer many advantages as far as resource utilization within the core and the radio access network. To this direction, Third Generation Partnership Project (3GPP) has firstly introduced Multimedia Broadcast/Multicast Services (MBMS) [1] as a new feature in Release 6 specifications, in order to efficiently deliver data from a single source to several destinations in a point-to-multipoint (ptm) way.

A crucial point on the provision of reliability over mobile multicast delivery is the use of a Forward Error Correction (FEC) scheme on the application layer. FEC, unlike the common methods for error control is not based on lost or corrupted packets retransmission, since the error correction is "forward" in the sense that redundant data are transmitted in advance with the source information, in order to obtain the receivers the ability to overcome packet losses. The application of FEC on ptm reliability protocols provides particular advantages. The most important property of FEC codes is the ability to use the same FEC packets to repair simultaneously different independent packet losses at multiple receivers, without the need of the costly or impossible procedure of packets retransmission. However, FEC has its own cost since it requires a higher "forward" channel bandwidth. Therefore, FEC protection must be carefully applied with respect to the current network conditions, in order to achieve an efficient and reliable multicast delivery.

In order to meet the error free transmission requirement of demanding applications, 3GPP recommends the use of the systematic, fountain Raptor code as an Application Layer FEC (AL-FEC) protection mechanism exclusively for MBMS [1]. However, since Raptor FEC was standardized there has been significant progress in the design of FEC codes. RaptorQ is the most recent member of Raptor codes family, providing exceptional protection performance and enhanced encoding parameters. To this direction, a general FEC framework is introduced in [2] describing the application of AL-FEC to arbitrary packet flows based on the Raptor and RaptorQ FEC codes.

Since the emergence and the 3GPP standardization of Raptor codes, several works have covered extensively the analysis and the evaluation of the systematic, fountain Raptor code as an AL-FEC protection scheme over mobile multicast environments. The authors of [3] provide an analytical investigation of the Raptor FEC performance, evaluating the trade-offs between AL-FEC and physical layer FEC over MBMS download delivery for UMTS systems. The work presented in [4] studies the Raptor application for MBMS services over 3G mobile cellular networks considering the impacts of AL-FEC on the telecommunication cost. The work in [5] provides a performance evaluation of the Raptor FEC scheme for streaming services over Long Term Evolution (LTE) single-cell MBMS environments considering several system and FEC encoding parameters. Moreover, a comprehensive analysis of the processes behind the design and the performance of Raptor and RaptorQ FEC codes is provided in [6]. Finally, the authors of [7] evaluate the application of RaptorQ compared to Raptor code focusing on the decoding complexity and energy consumption on embedded mobile systems.

In this work we provide a performance analysis of the newly introduced RaptorQ code in comparison with the 3GPP standardized Raptor FEC scheme. We analyze the differentiation points of the two Raptor codes family members and we highlight on the enhanced performance promised by the new RaptorQ code. Furthermore, through the ns-3 simulation environment [8], we evaluate the application of both the examined AL-FEC codes over mobile download and streaming delivery scenarios, investigating several mobile FEC encoding parameters.

The rest of this paper is organized as follows: in Section II we present the AL-FEC mechanism integration in the 3GPP MBMS standard. Furthermore, we provide a description of the two examined members of the Raptor codes family including a comparative analysis between them. Section III presents the mobile multicast simulation environment we utilized, in order to simulate AL-FEC protected download and streaming delivery sessions. In Section IV we present the conducted experimental results and finally, in Section V we draw our conclusions and we describe some possible future steps.

II. Raptor codes over Multicast Services

In this section we provide a brief description of the AL-FEC application over the 3GPP MBMS environment. Thereafter, we describe the performance modeling of the two examined AL-FEC schemes and a functional comparison between the two members of the Raptor codes family is finally provided.

A. 3GPP MBMS

The 3GPP multicast services standard, named MBMS [1], is a unidirectional ptm service in which data are transmitted from a single source to a group of multiple mobile endpoints in a specific service area. 3GPP defines two delivery methods namely, download and streaming. The MBMS user plane protocol stack of both delivery methods is illustrated in Fig. 1.

Download uses the FLUTE protocol which is carried over UDP/IP and is independent of the IP version and the underlying link layers used. In order to apply AL-FEC protection on the MBMS download delivery, the transmitted file is partitioned in one or several source blocks each consisting of k source symbols. For each source block, redundant repair symbols are generated through FEC encoding with a unique ID assigned on each resulting encoding symbol. Subsequently, one or more encoding symbols are placed in each FLUTE packet payload with the resulting packets encapsulated in UDP and distributed over the IP multicast flow. Furthermore, 3GPP defines a post-delivery procedure to provide file repair features for the MBMS download delivery. A MBMS client is able to determine which source symbols should have been received but have not. Therefore, each MBMS client is able to send a file repair request message to a file repair server for unreceived symbols, with the server responding through a point-to-point (ptp) or a ptm data delivery of the requested repair data.

On streaming delivery, RTP is the application layer protocol which provides means for sending real-time or streaming data over UDP transport layer. The MBMS AL-FEC streaming framework operates on RTP/UDP flows. A copy of the source packets is forwarded to the Raptor encoder and arranged in a source block with each packet occupying a new empty row of T bytes. The source block is filled up to k rows, where the

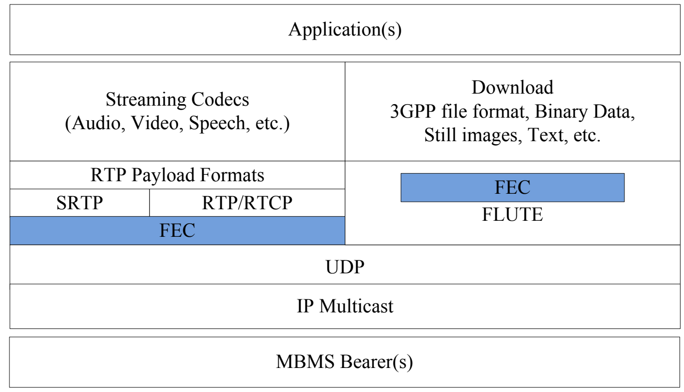

Fig. 1. 3GPP MBMS Protocol Stack

value of k can be different for each source block. After forming a FEC source block from the packets to be protected together, the Raptor encoder generates the desired repair symbols which are then sent using the FEC repair packet format.

B. Raptor codes

In general, AL-FEC codes can be considered as correcting codes for an erasure channel, where a symbol is sent with the receiver either receiving or not the transmitted symbol. AL-FEC aims to cope with these symbol erasures by adding some redundancy in the transmitted data. Raptor codes were firstly introduced as a FEC erasure code in [9]. 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 and are one of the first known classes of fountain codes with linear encoding and decoding time. 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. Subsequently, the Raptor encoder generates n encoding symbols from the $k<n$ source symbols, which are transmitted to the receiver. 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 performance of a Raptor AL-FEC code can be described by the decoding failure probability in function of the number of source and received symbols. Moreover, a crucial point for the robustness of an AL-FEC protected delivery is the transmission overhead, which is defined as the amount of the redundant information divided by the amount of source data and is equal to the fraction $(N-K)/K$ in terms of percentage where N denotes the number of transmitted encoding packets and K denotes the number of the source packets.

The 3GPP MBMS standardized Raptor code [1] is a systematic code i.e., the original source symbols are within the stream of the transmitted symbols. The decoding failure probability of the standardized Raptor code can accurately be modeled by (1) [10]:

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

Recently, an enhanced Raptor code has been emerged at Internet Engineering Task Force (IETF) [11] in order to address the performance drawbacks of the standardized Raptor code. This newer member of Raptor codes family is known as RaptorQ code. RaptorQ is also a systematic code with significantly more efficient performance than the older Raptor code, in terms of superior flexibility, support for larger source block sizes and better coding efficiency. The enhanced design of RaptorQ addresses the Raptor code recovery performance limitations, resulting in a very close to an ideal fountain code performance described by (2) [12]:

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

C. Comparative Analysis

In general, Raptor codes can be viewed as the concatenation of several codes. On the older systematic Raptor code, the most-inner code is a non-systematic Luby Transform (LT) code [13], 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 an outer high-rate block code. This means that the outer high-rate block code generates the intermediate symbols using the input symbols. Finally, a systematic realization of the code is obtained by applying some pre-processing to the source symbols such that the input symbols to the non-systematic Raptor code are obtained. On the newly introduced RaptorQ code, although most of the basic encoding steps are identical to those of Raptor code, there are several improvements and additions to the encoding and decoding processes.

On RaptorQ, before the intermediate symbol generation, each FEC source block is augmented with additional padding symbols for encoding and decoding purposes. Padding out a FEC source block enables faster encoding and decoding and minimizes the amount of information that needs to be stored. On the following step, for the generation of the intermediate symbols from the source symbols, RaptorQ introduces enhanced generator and pre-coding relationships (i.e., a two-stage pre-coding algorithm using LDPC and HDPC codes) in comparison with those of the existing Raptor code. Finally, in the second encoding step of RaptorQ, a modified, more efficient encoding process than this of Raptor, is applied in order to generate the encoding symbols.

However, the key differentiation between the two FEC schemes is that the standardized Raptor code operates over Galois field GF(2) [14], while the enhanced RaptorQ code introduces the use of arithmetic operations on octets, which mathematically can be thought of as elements of a finite field, i.e., the finite field GF(256) [11]. Operating over GF(256) allows RaptorQ to overcome the performance limitations of Raptor code, since the operation over larger finite fields offers the potential of achieving enhanced recovery with lower reception overhead. More precisely, the best recovery probability a code operating over GF(2) can achieve is $1 - \frac{1}{2^{m+1}}$ if $k+m$ encoding symbols have been received, while using symbol operations over GF(256) achieves recovery from the reception of $k+m$ encoding symbols with probability $1 - \frac{1}{256^{m+1}}$. Furthermore, 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 minority of the symbol operations are over GF(2) and only a small minority are over GF(256).

Except from the use of symbols over larger alphabets, another new technique improving the decoding performance of RaptorQ is the use of the permanent inactivation [6], which is an interesting extension of the LT code and of inactivation decoding. In brief, a limited number of the intermediate symbols are declared to be permanently inactive while the remaining majority of symbols are LT symbols. In the encoding and decoding procedure the permanent inactive symbols are treated differently from the LT symbols utilizing an innovative technique which enhances the recovery properties of the RaptorQ code.

Regarding functional capabilities, the number of encoding symbols RaptorQ can generate, is up to 16777216 symbols i.e., 256 times more than the foregoing Raptor code. Moreover, RaptorQ can encode up to 56403 source symbols into a single source block in contrast to 8192 of the Raptor code. Expanding the range of these two parameters simplifies the application of the AL-FEC protection. Obviously RaptorQ can perform better and more flexible than the standardized Raptor code both for file delivery and streaming services. RaptorQ can deliver files up to 3.4 GB as a single source block maximizing the protection efficiency due to the spreading of protection across the whole file, particularly for the delivery of very large files. Furthermore, on delay-sensitive real-time applications, the RaptorQ flexible range of the block size allows to determine a QoS trade-off between protection and latency considering the delay constraints of the transmitted application.

Finally, regarding the complexity of the presented Raptor FEC codes, in general both of them require linear encoding and decoding time i.e., the computation complexity of the FEC encoding or decoding process is proportional to the size of the source data. However, as illustrated in [7], RaptorQ code requires significantly higher decoding times than the existing Raptor code considering several block and symbol sizes. This is reasonable, since the tremendous improvement the GF(256) operation introduces on the decoding failure probability has a price, i.e., the higher decoding complexity of RaptorQ.

III. Simulation Framework

In order to simulate the application of the two examined AL-FEC schemes over 3GPP MBMS environments, we utilize the ns-3 network simulator [8]. Our simulation model is composed of a source entity which is responsible to introduce the modeled applications into the multicast gateway (GW) and furthermore to apply the AL-FEC protection concept on the transmitted data. Thereafter the multicast GW undertakes to forward the IP multicast flow to the simulated 3GPP radio access network, named evolved UMTS terrestrial Radio Access Network (eUTRAN). Finally, within eUTRAN, the base station, named evolved Node B (eNB), transmits the multicast traffic to multiple User Equipments (UE)s dropped in a specific cell area. The simulated network topology is illustrated in Fig. 2 and Table I presents further simulation settings we adopted during the conduction of the simulation experiments.

The modeled physical (PHY) channel covers several 3GPP requirements. More specifically, regarding the simulated channel coding scheme, PHY-FEC is applied to the data streams before the transport over the radio link. The applied coding scheme on the multicast channel is based on convolutional coding with fixed rate $1/3$. Moreover, a 24-bit CRC protection is attached to the transmitted bitstream. On the modeled PHY layer, the successful reception of each PHY-FEC block is calculated according to a Signal to Noise Ratio (SNR) estimation based on OFDM simulation traces. Finally, the blocks are concatenated in order to determine if the source burst can be reconstructed and forwarded to the upper layers.

The most important part of the simulation model, the AL-FEC protection is modeled on the application source before the transmitted data being forwarded to the multicast GW. According to the specified Source Block Length (SBL) the transmitted packets are organized in AL-FEC source blocks and thereafter the redundant AL-FEC symbols are produced for each source block. The number of the generated additional AL-FEC symbols is determined by the transmission overhead a multicast sender introduces to the transmission. Thereafter, the generated source and repair symbols, with the assumption of one FEC symbol per packet, are transmitted through an IP multicast flow to multiple recipients. At the receiver side, the probability of successful recovering each AL-FEC protected block derives from (1) and (2) when Raptor and RaptorQ code application is evaluated respectively.

IV. Performance Evaluation

This section presents the experimental results we extracted through the previously described ns-3 simulation environment,

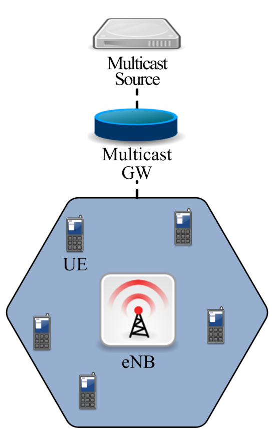

Fig. 2. Network Simulation Topology

TABLE I
Simulation Settings

Parameter	Units	Value
Cell Layout		Hexagonal grid
Cell Radius	m	1000
Carrier Frequency	MHz	2000
System Bandwidth	MHz	5
Environment		Macro Cell, Urban Area
Propagation Model		COST-Hata Model
BS Transmit Power	dBm	43
BS Antenna Gain	dBi	15
BS Antenna Height	m	30
# UEs		100
UE's Mobility Model		Random Walk @ 30 km/h

evaluating the application of the two AL-FEC schemes over both download and streaming delivery scenarios with respect to the performance of an ideal fountain code. While an ideal fountain 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, Raptor codes have a performance close to that property, meaning that the number of received symbols should be slightly higher than the number of source symbols per FEC block. Subsequently, the comparison of the two examined AL-FEC schemes with an ideal fountain code is typical for their performance.

A. MBMS Download Delivery

In the first part of the presented results we evaluate the application of the Raptor and RaptorQ AL-FEC schemes over a MBMS download delivery scenario. More precisely, at first we examine the impacts of the examined AL-FEC codes on the service coverage considering the exclusive use of the AL-FEC protection. Thereafter we provide simulation results considering the total number of retransmitted packets during the MBMS download session utilizing the AL-FEC protection in conjunction with a post-delivery ptp file repair phase as described in Section II-A.

Fig. 3 presents the impacts of the AL-FEC transmission overhead increment on the amount of UEs that can successfully recover the AL-FEC protected data i.e., the service coverage in terms of percentage. We assume that a UE can recover the protected object if the failure probability of the decoding process is 10^{-4} or less. For this evaluation the transmitted object consists of 1024 packets with each size fixed at 1024 B and the SBL fixed at 512 symbols.

Observing the plotted curves of Fig. 3 we can immediately remark the extremely close to ideal performance of the RaptorQ, since an ideal FEC code achieves less than 1% better service coverage than RaptorQ. On the other hand, Raptor code presents performance quite far from the ideal FEC code and only achieves a little closer performance to that of RaptorQ for high values of transmission overhead where AL-FEC has to confront UEs with high packet loss rates. This behavior is expected considering that RaptorQ requires only 2 additional symbols to meet the ideal FEC code performance according to the failure probability threshold, while Raptor code requires reception overhead equal to 24 additional symbols per source block. Moreover, we can observe that RaptorQ can operate almost ideal from the very first additional symbol, while the standardized Raptor code requires significantly more symbols to provide the possibility of successful recovering the protected data.

Furthermore, regarding the curves behavior, we can extract an efficient interval of transmission overhead selection between 5% and 20%. In more detail, we mean that increasing the transmission overhead in the specific interval results in a proportional increase on the amount of UEs successfully recovering the protected data, while beyond this transmission overhead zone the gains on the system coverage are minimized.

In Fig. 4 we present the total number of retransmitted packets during the ptp repair phase in function to the amount of the introduced AL-FEC transmission overhead. Each UE that fails to recover the protected data through the AL-FEC decoding procedure is able to request the retransmission of the lost data in order to reconstruct the source information through a ptp session.

As in the previous part of provided results, the plotted curves immediately reveal the primacy of RaptorQ since achieves significantly lower number of retransmitted packets compared to Raptor FEC. We can observe that RaptorQ performance is just a "step" behind the ideal fountain FEC code. At this point, we have to clarify that the total number of retransmitted packets is independent from each code reception overhead itself and only depends on the service coverage, because if a UE fails to decode the FEC protected block requests the retransmission of the exact number of lost source packets only. Therefore, the extremely close to the ideal RaptorQ performance but also the closest performance of Raptor to that of RaptorQ is justified. In addition, we can observe that Raptor curve presents an initial delay until it can

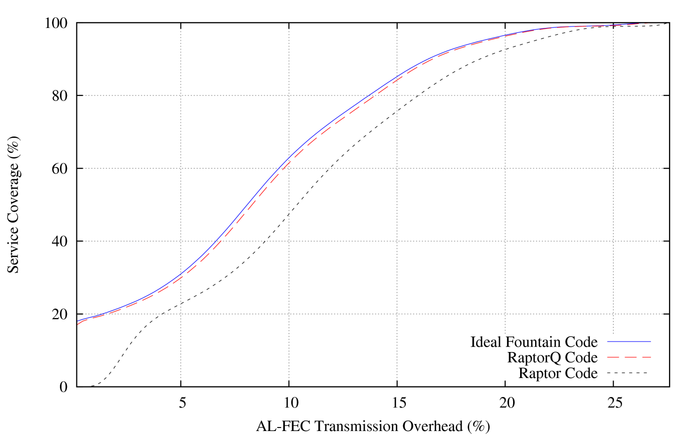

Fig. 3. Service Coverage vs. AL-FEC Transmission Overhead

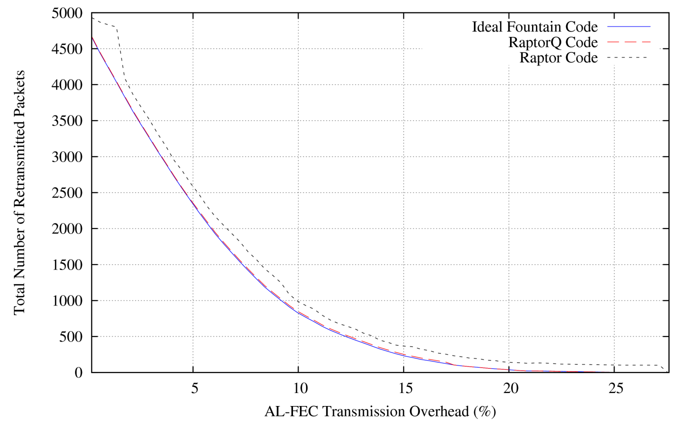

Fig. 4. Total Number of Retransmitted Packets vs. AL-FEC Transmission Overhead

reach the ideal curve form. This is a direct result of the conduct described previously in Fig. 3, since Raptor FEC inefficiency is more pronounced for low values of transmission overhead.

All of the above presented results make clear that RaptorQ FEC can operate far more effectively than Raptor, providing significant gains on the transmission efficiency since substantially reduces the required transmission redundancy with all the benefits this fact implies.

B. *MBMS Streaming Delivery*

In this subsection we provide a performance evaluation of the two examined AL-FEC schemes over MBMS streaming scenarios. We consider the application of the Raptor and RaptorQ FEC over a multicast video streaming transmission. For the conduction of the presented simulation results we utilize H.264 video traces examining the impacts of the AL-FEC protection on the tune-in delay with respect to the MBMS service coverage as defined in the previous subsection. Tune-in delay is defined as the time interval between the start of the packets reception until the start of correct decoding the received packets of each FEC source block. Tune-in delay is experienced by a user who joins the multicast streaming session and the first received packet is anywhere but at the very start of the FEC source block. On the tune-in process a receiver first synchronizes to the FEC block, waiting for the reception and successful processing of each FEC block, before attempting to decode the media. Subsequently, the tune-in delay is a function of the FEC protection period and the decoding delay, typically defined as $\textit{tune-in delay} = \textit{protection period} + \varepsilon$ [15]. It is obvious that tune-in delay strongly depends on the FEC encoding parameters and more specifically on the selected length of the FEC source block and the introduced AL-FEC transmission overhead.

In Fig. 5 we examine how the tune-in delay varies in function of the streaming service coverage with the FEC SBL fixed at 512 and variable length of transmitted packets.

As in the previous subsection, the RaptorQ superiority over the older Raptor code is directly perceived from the

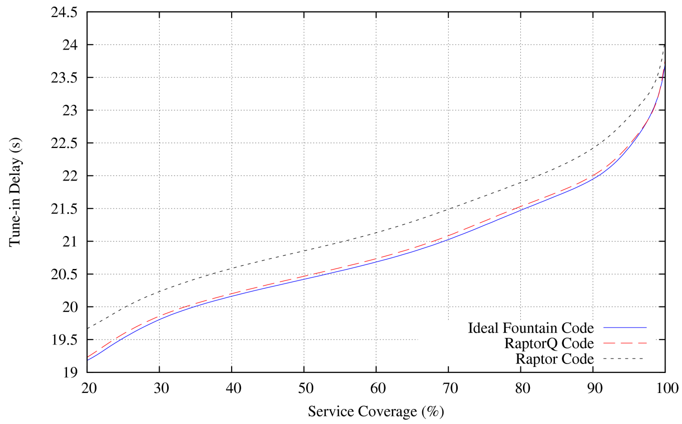

Fig. 5. Tune-in Delay vs. Service Coverage

plotted curves. Once again, RaptorQ almost perfectly emulates the performance of an ideal FEC fountain code. Examining the conducted curves, we can remark that RaptorQ requires consistently significant lower time for the tune-in process in comparison with Raptor until the service coverage reaches the value of about 95%. For higher values of service coverage we observe that the achieved reduction of RaptorQ on the tune-in delay is gradually reduced. This behavior, which is also denoted in the previous subsection results, is due to the fact that the AL-FEC transmission overhead has an optimal zone of efficient operation, always of course with respect to the system packet loss conditions. Therefore, for high values of service coverage, where both Raptor schemes has to confront a small fraction of UEs with extremely bad reception conditions, the close behavior of the two examined AL-FEC codes is expected, since as already mentioned the tune-in delay strongly depends on the transmission overhead while the examined SBL is fixed.

V. Conclusions & Future Work

In this work we have provided a performance evaluation of the most recent member of Raptor codes family, named RaptorQ code, in comparison with the 3GPP MBMS standardized Raptor FEC scheme. In order to verify the almost ideal theoretical performance of RaptorQ, we have introduced a MBMS simulation environment considering the application of the examined AL-FEC schemes on both download and streaming delivery scenarios over evolved 3GPP systems. From the conducted simulation results we have verified that RaptorQ almost emulates the performance of an ideal FEC code since the minimized required additional data enables RaptorQ operating with significantly lower transmission overhead in comparison with the standardized Raptor FEC. This property is beneficial for the mobile system efficiency since RaptorQ can effectively operate under poor reception conditions while achieving significant reduction in the required redundancy. Furthermore, we were able to examine the reflection of the enhanced RaptorQ properties on some specific constraints of both the MBMS delivery methods, verifying once again the universal supremacy against the existing Raptor FEC scheme. Concluding, based on the enhanced capabilities and performance of RaptorQ code, the adoption of that new powerful AL-FEC scheme is expected by several multicast standards.

Concerning some possible future steps that could follow and extend this work, providing a cross-layer design could be beneficial for the multicast transmission performance, since the interoperability between the AL-FEC with lower layers protection mechanisms could optimize the costly error protection framework in total. Furthermore, it is our belief that the introduction of an adaptive algorithm computing the optimal transmission overhead of the AL-FEC mechanism based on a sophisticated feedback-reporting scheme could further enhance the AL-FEC efficiency. Finally, since the almost ideal performance of the RaptorQ FEC addresses the shortcomings of Raptor FEC, we could evaluate the application of AL-FEC protection over different transmission environments and standards.

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. Watson, T. Stockhammer, and M. Luby, "Raptor FEC Schemes for FECFRAME," draft-ietf-fecframe-raptor-05, Internet Engineering Task Force, Sept. 2011. [Online]. Available: http://tools.ietf.org/id/draft-ietf-fecframe-raptor-05.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] 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.

[6] 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.

[7] T. Mladenov, K. Kim, and S. Nooshabadi, "Forward error correction with RaptorQ Code on embedded systems," in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*. IEEE, pp. 1–4.

[8] "The NS-3 Network Simulator." [Online]. Available: http://www.nsnam.org/

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

[10] 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.

[11] 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

[12] 3GPP, "Rationale for MBMS AL-FEC Enhancements," 3rd Generation Partnership Project (3GPP), Tdoc S4-110449, 2011.

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

[14] 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

[15] 3GPP, "Report of FEC selection for MBMS," 3rd Generation Partnership Project (3GPP), Tdoc S4-050250, 2005.