International Journal of Electronics and Communication Engineering (IJECE)
ISSN 2278-9901
Vol. 2, Issue 2, May 2013, 169-178
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

COMPARATIVE ANALYSIS OF AL-FEC RAPTOR AND RAPTORQ OVER 3GPP eMBMS NETWORK

AVANI U PANDYA[1], SAMEER D TRAPASIYA[2] & SANTHI S CHINNAM[3]

[1]PG Student, Department of Electronics & Communication Engg., G. H. Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India

[2]Assistant Professor, Department of Electronics & Communication Engg., G. H. Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India

[3]Telecom Engineer, 4G RAN & Device Validation, Rancore Technologies Pvt Ltd, Navi-Mumbai, India

ABSTRACT

Long Term Evolution (LTE) is the current standard for wireless mobile communication based on Third Generation Partnership Project (3GPP). LTE includes enhanced multimedia broadcast and multicast services (MBMS), also called as Evolved multimedia broadcast and multicast services (eMBMS) where the same content is transmitted to multiple users in one specific area. eMBMS is a new function defined in 3GPP Release 8 specification that supports content distribution and streaming to group users into LTE mobile networks. In LTE an important point of demanding multimedia services is to improve the robustness against packet losses. In this sense, in order to effectively support point-to-multipoint download and streaming delivery, 3GPP has included an Application Layer Forward Error Correction (AL-FEC) scheme in the standard eMBMS. The standard AL-FEC system is based on systematic, fountain Raptor codes. Raptor coding is very useful in case of packet loss during transmission as it recover all data back from insufficient data at receiver terminal However, the 3GPP standardized systematic fountain Raptor code is nowadays considered obsolete, since a new variation of the Raptor codes has emerged. This enhanced AL-FEC scheme, named RaptorQ, promises higher protection efficiency and superior flexibility on the provision of demanding mobile multicast services. In this work, we provide an extensive performance evaluation presenting at first a theoretical performance comparison of the newly introduced RaptorQ FEC scheme with its predecessor Raptor code, examining the enhancements that RaptorQ introduces on the AL FEC protection robustness. Thereafter, to verify the enhanced performance of RaptorQ, we present several simulation results considering the modeling of the AL-FEC protection over multicast services for next generation mobile networks.

KEYWORDS: Long Term Evolution, Multimedia Broadcast Multicast Services, Forward Error Correction, Raptor Codes, RaptorQ Codes

INTRODUCTION

Nowadays there is a significant demand for multimedia services over wireless networks due to the explosive growth of the multimedia internet applications and dramatic increase in mobile wireless access. It is, therefore, foreseen that the 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. This is addressed by standardization bodies through introduction of a point-to-multipoint service enhanced Multimedia Broadcast Multicast Service (eMBMS), a resource-efficient transmission scheme targeting simultaneous distribution of multimedia content to many user devices within a serving area, over a single set of Core Network and Radio resources. So, Multimedia Broadcast and Multicast Service (MBMS) has been standardized as a key feature in Third Generation Partnership Project(3GPP) systems to broadcast and multicast multimedia content to multiple mobile subscribers via MBMS radio bearer service.MBMS is a point-to-multipoint (PTM) Standard, whose further evolvement and enrichment attracts nowadays widespread interest.

Long Term Evolution (LTE) provides both the transmission mode single-cell MBMS, MBMS services which are transmitted in a single cell and multi-cellular evolved MBMS transmission mode, providing synchronous MBMS transmission from multiple cells , also known as multicast / broadcast single frequency network mode of transmission. To transmit the same data to multiple recipients allows network resources to be shared. In order to meet the increasing use of high bandwidth multicast services, 3GPP initially standardized MBMS in third generation mobile systems. MBMS is a unidirectionalptm service in which data are transmitted from a singlesource to a group of multiple mobile endpoints in a specificservice area. MBMS allow for multiple mobile subscribersto share radio and core network resources and as such offermany advantages regarding system resource utilization.

The MBMS provide two modes of operation, the broadcastand the multicast mode. 3GPP defines three distinct functionallayers for the delivery of MBMS services: the userservice, the delivery method and the bearers. MBMS userservices are built on top of the MBMS bearer service.3GPP defines a set of media codecs, formats and transport/ application protocols to enable the deployment of several MBMS user services with different requirements. Furthermore, 3GPP defines two delivery methods for the MBMS user services, namely download and streaming. The delivery of software upgrades is an example of application using the download delivery method, while the delivery of real-time video is an example of the streaming delivery. MBMS delivery methods make use of the MBMS bearer service in order to distribute an application to multiple subscribers. Finally, bearers provide the mechanism by which IP data is transported. A MBMS bearer is an IP-multicast packet flow between a multicast gateway and the mobile MBMS subscribers.

3GPP focuses on the provision of reliability control over the MBMS delivery. A crucial point in achieving this objective is the introduction of a Forward Error Correction (FEC) mechanism on the application layer for both MBMS delivery methods. FEC is a method used for ‘‘forward’’ error control in data transmission over unreliable channels, such as radio transmission channels. The ‘‘forward’’ concept of FEC is justified by the redundant data transmission in advance the source information, unlike the common methods for error control (i.e. ARQ, Carousel) that are based on lost or corrupted packets retransmission to obtain the recipients the ability to overcome packet losses.

The application of FEC on ptm reliability protocols, such as the MBMS environment, provides particular advantages since the redundancy introduced in the multicast transmission can overcome the common methods limitations [2]. The most important property of FEC codes is the ability to use the same FEC packets to simultaneously repair different independent packet losses at multiple receivers. However, FEC comes with its own cost since FEC protection must be carefully applied with respect to the current network conditions so as to avoid channel bandwidth wastage and achieve an efficient and reliable multicast delivery. 3GPP recommends the use of the systematic, fountain Raptor code as an Application Layer FEC (AL-FEC) protection mechanism exclusively for MBMS [1]. Raptor FEC [3] was selected due to the higher performance compared with existing AL-FEC codes. However, in the meantime a new very promising variation of Raptor codes has emerged, named RaptorQ [4]. RaptorQ is the most recent member of Raptor codes family, providing exceptional protection performance and enhanced encoding parameters

The rest of this paper is organized as follows: in Section 2 we provide an overview of the 3GPP AL-FEC eMBMS delivery framework and Section 3 presents a detailed description of the examined AL-FEC schemes. Furthermore, we provide a comparison between them concerning both functional and performance aspects. In Section 4 we present the simulation environment and the conducted experimental results. Finally, in Section 5 we draw our conclusions and we describe some possible future steps.

EMBMS PROTOCOL STACK

GPP AL-FEC eMBMS Delivery

3GPP defines two delivery methods namely, downloading and streaming. eMBMS user plane stack of these delivery methods is illustrated in Figure 1.

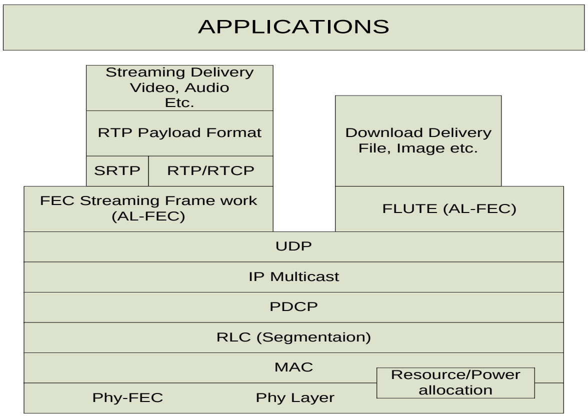

Figure 1: eMBMS Protocol Stack

MBMS Streaming Delivery Protocols Stack

The purpose of the MBMS streaming delivery method is to deliver continuous multimedia data (i.e. speech, audio, and video) over an MBMS bearer. MBMS makes use of the most advanced multimedia codecs such as H.264 for video applications and enhanced Advanced Audio Coding (AAC) for audio applications. Real-time transport protocol (RTP) is application layer transport protocol for MBMS streaming delivery. RTP provides means for sending real-time or streaming data over user datagram protocol (UDP), the resulting UDP flows are mapped to MBMS IP multicast bearers. Furthermore RTP provides RTP Control Protocol (RTCP) for feedback about the transmission quality.

3GPP recommends the use of an AL-FEC mechanism by the sender before RTP flows are mapped onto UDP. 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 row width T bytes with each packet occupying a new empty row. The source block is filled up to k rows, where the value of k can be different for each source block and depends on the variable streaming services constraints. After forming a FEC source block from the packets to be protected together, the Raptor encoder generates the desired repair symbols. These generated Raptor repair symbols are then sent using the FEC repair packet format.

MBMS Download (File) Delivery Protocols Stack

MBMS download delivery method aims to distribute discrete objects (e.g. files) by means of a MBMS download session. Download method uses the File deLivery over Unidirectional Transport (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 is forwarded to the Packet Data Convergence Protocol (PDCP) layer. The packets are then sent to the Radio Link Control (RLC) layer. The RLC Layer functions in unacknowledged mode. The RLC layer is responsible for mapping IP packets to RLC SDUs. The Media Access Control (MAC) Layer adds a 16 bit header to form a PDU, which is then sent in a transport block on the physical layer.

In order to apply AL-FEC protection on the MBMS download delivery, the transmitted file is partitioned in one or several source blocks. Each source block consists of k source symbols, each of length T except for the lastsource symbol, which can be smaller. Through the Raptorencoding, for each source block, redundant repair symbolsare generated according to the desired amount of protection.A unique ID is assigned on each resulting encodingsymbol, which can be a source or a repair symbol, in orderto identify the type of the symbol according to the assignedID. Subsequently, one or more FEC encoding symbols areplaced in each FLUTE packet payload and the resultingpackets are encapsulated in UDP and distributed over theIP multicast MBMS bearer.

Furthermore, 3GPP defines a post-delivery procedure toprovide file repair features for the MBMS downloaddelivery. The purpose of the file repair procedure is torepair lost or corrupted file fragments from the MBMSdownload data transmission. A MBMS client is able todetermine, for each source block of each file, which sourcesymbols should have been received but have not and is alsoable to determine the number of symbols it has received.Therefore, each MBMS client is able to determine thenumber of further symbols required and send a file repairrequest message to a file repair server for unreceivedsymbols which will allow the MBMS FEC decoder torecover each protected block of the file. Thereafter, theMBMS client can receive the requested repair data througha point-to-point (ptp) or a ptm repair data delivery.

AL-FEC SCHEMES

In general, AL-FEC codes can be considered as correcting codes for an erasure channel. In an erasure channel a transmitter sends a symbol i.e., a fragment of the source data, 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 [5]. In this section we provide an analytical description of the two examined members of the Raptor codes family. We focus on the improvements that the newer member of Raptor codes, named RaptorQ, has emerged and we further provide a theoretical performance

Raptor Codes (IETF RFC 5053)

The use of Raptor codes in the application layer of MBMS has been introduced to 3GPP by Digital Fountain [7] aiming to provide 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 [5]. The systematic Raptor Encoder is used to generate repair symbols from a source block that consists of K source symbols [3].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 [6] 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 [11]. 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 [11] 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) [11]

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

In (1*)*, p_{fR}(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 failure probability is still about 85 %. 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.

RaptorQ Code (IETF RFC 6330)

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. The outcome of this progress is the emergence of an enhanced Raptor code at Internet Engineering Task Force (IETF) [4] in order to address the drawbacks of the standardized Raptor code on the recovery properties described in A. This newer member in Raptor codes family is known as RaptorQ code. RaptorQ is also a fountain and systematic AL-FEC code. RaptorQ is a significantly more efficient AL-FEC code than the older Raptor code, in terms of superior flexibility and higher protection and coding efficiency. The encoding process of RaptorQ code is mostly identical with that of Raptor code described in the previous subsection. However, RaptorQ code introduces certain design selections, analyzed below, that ensure superior performance compared with that of Raptor code. A key differentiation between the two schemes is that the standardized Raptor code operates over Galois field GF (2) [3], while the enhanced RaptorQ code uses symbol operationsover GF(256) [4] instead of over GF(2). Operating over larger finite fields allows RaptorQ to overcome the performance limitations of Raptor code since utilizing 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 code are the increased number of possible source symbols and the increased number of generated encoding symbols. More precisely, RaptorQ can encode up to 56,403 source symbols into a source block in contrast to 8,192 of the Raptor code and furthermore can generate up to 16, 777,216 encoding symbols, 256 times more than the older Raptor code. The expanded range of these two parameters simplifies the application of the AL-FEC protection and offers higher flexibility to RaptorQ. 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 maximizes the decoding efficiency and protection 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 [12] than the older Raptor code. Concerning the performance of RaptorQ, as already mentioned, 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 (5) [12]:

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

In (2), p_{fRQ} (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 (1) with (5), the performance superiority of RaptorQ code is unambiguous.

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 symbols where enhanced generator and pre-coding source 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 $1-(1/2^{m+1})$ 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 $1-(1/256^{m+1})$. 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).

- 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.

- 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 [14], 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.

EXPERIMENTAL EVALUATION

The performance of Raptor & RaptorQ codes has been evaluated. The encoder and decoder are designed as specified in section III.

Comparison of Raptor and RaptorQ Code

Successful Decoding Vs Packet Loss

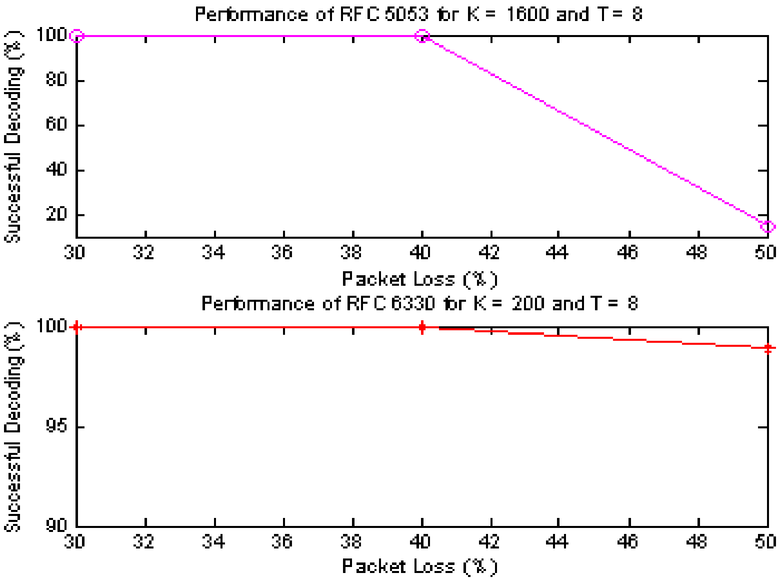

Figure 2: Comparison of RFC 5053 and 6330 for Same Bandwidth

Here we considered K=1600, T=8 for RFC 5053 and K=200, T=8 for RFC 6330 and applied packet loss 30% ,40% and 50% with 1000 channel realization to compare Probability of successfuldecoding for RFC 6330 and RFC 5053 for same Bandwidth allocation. The plots show that Raptor Q achieves the 98.9 % successful decoding rate with 50% packet loss while for same RFC 5053 raptor achieves 15.2% successful decoding. And also 100 % successful decoding rate for both RFC 5053 and RFC 6330 for 30% and 40% packet loss. This shows that for the same bandwidth utilization RFC 6330 performs better than RFC 5053.

Probaility of Failure Vs Different Values of K

Here we simulated RFC 6330 and RFC 5053 for four different values of K (372, 573, 1205, and 1600) with 1000 channel realization to compare the performance of RFC 6330 with theoretical Limit as per equation (2) with reception overhead zero.i.e it will receive exactly K number of encoded symbols and will try to decode. And RFC 5330 with theoretical Limit as per equation (1) with reception overhead zero with probability of failure.

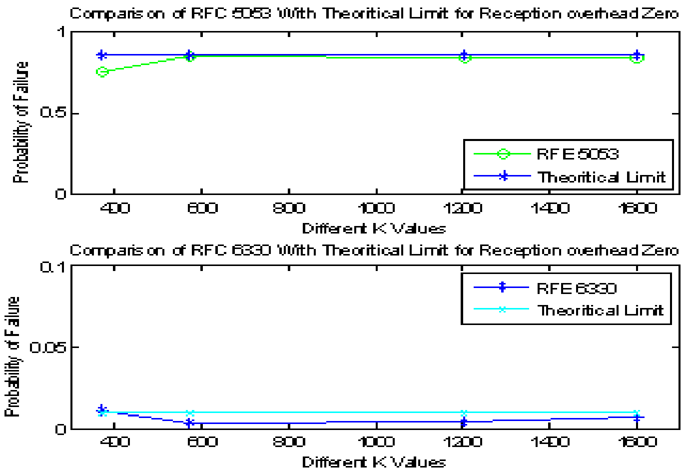

Figure 3: Comparison of RFC 5053 and RFC 6330 with Theoretical Limit for Reception Overhead Zero

Reception overhead zero means n-k equal to zero in equation (2), hence theoretical probability of failure equal to 0.01. From the graph we can say that RFC 6330 performs better than theoretical limit for many values of K. Similarly, Reception overhead zero means n-k equal to zero in equation (1), hence theoretical probability of failure equal to 0.85 From the graphi.e figure 3 we can say that RFC 5053.performs better than theoretical limit for many values of K.We observe one more thing that the performance of RFC 6330 and RFC 5053 are not dependent on K as probability of failure is not following any specific order with respect to K.

Decoding Performance Vs. Reception Overhead

In this paragraph we draw one of the most important pointsfeaturing 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.Here we simulated RFC 6330(RaptorQ) and RFC 5053(Raptor) for AL-FEC Overheads (0,1,2,3,5 and 7) with 1000 channel realization to compare the performance of RFC 6330 with RFC 5053 Figure 4 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 thestandardized Raptor FEC code with that of RaptorQ.

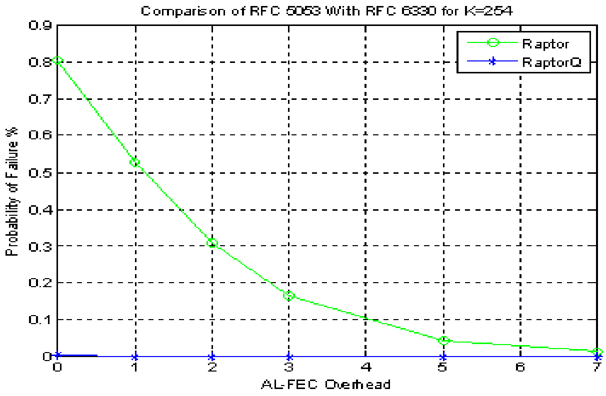

Figure 4: Comparison of RFC 5053 and RFC 6330 for K=254

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, theplot shows that RFC 6330 RaptorQ achieves the 0.042 failure probability rate with zero reception overhead, as opposed to the 0.8043 failure probability for RFC 5053 Raptor. The simulations also showed that for one receptionoverhead symbol RFC 6330 RaptorQ guarantees nearly the 0 failure probability rate for one reception overhead, as opposed to the 0.5284 failure probability for RFC 5053 Raptor. The graph also shows that RFC

6330(RaptorQ)achieves the 98.9% probability of successful decoding with zero overhead symbols as opposed to more than 7 overhead symbols required by RFC 5053 (Raptor). The simulations showed that 1 overhead symbolguarantees nearly 100% successful decoding for RFC 6330(RaptorQ), while RFC 5053 (Raptor) achieves nearly 50% successful decoding. The minimum requirements of RaptorQ code over thenumber of additional symbols have a direct impact on some extremely important aspects of 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 RaptorQFEC 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.

Consideringthe encoding properties of the two FEC codes. As described in chapter 4 Raptor can encode up to 8,192 symbols within a source block, so Raptor the transmitted object is divided intomultiple FEC source blocks, while using the RaptorQ encoder allows to send the whole object as a single source block. This property of RaptorQ maximizes the protection efficiency, since allows spreading the protection across the whole file in a FEC source block. Finally, regarding the complexity of the presented 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 [17], 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. All of the above presented results make clear that RaptorQFEC can operate far more effectively than Raptor, providing significant gains on the transmission efficiency since substantiallyreduces the required transmission redundancy with the less bandwidth utilization and all benefits this fact implies.

CONCLUSIONS & FUTURE WORK

In this work we have provided a performance evaluation and comparison of the most recent member of Raptor codes family, named RaptorQ code, with the 3GPP MBMS standardized Raptor FEC scheme. We have drawn the main functional improvements with enhanced efficiency regarding the achieved protection performance and the required transmission redundancy that the newest member of Raptor codes, named RaptorQ, has emerged compared to the 3GPP standardized Raptor code. To verify the superiority of RaptorQ against Raptor code we have provided a theoretical evaluation of the two examined AL-FEC schemes application through which we were able to detect the enhanced features that RaptorQ can provide on the field of reliable multicasting. The conclusions that are drawn from the previous discussions and simulation results are as Behaviour of RaptorQ concerning the required additional data (Reception Overhead) allows operating with significantly lower transmission overhead (redundancy) compared to the standardized Raptor FEC with respect to the evaluated AL-FEC encoding parameters. This property is beneficial for the mobile system efficiency since RaptorQ can effectively operate under poorer reception conditions while achieving significant reduction in the required redundancy and hence offering enhanced resource utilization. Concluding, it is clear that RaptorQ code can provide enhanced capabilities in eMBMS Networks and the adoption of this new AL-FEC scheme is expected by several standards.

On possible future steps we could design a cross-layer scheme, which could adapt the AL-FEC encoding parameters based on an interoperability scheme between the AL-FEC layer with other protection mechanisms deployed in lower layers, optimizing the costly error protection framework in total. Moreover, in order to avoid the individual constraints of a feedback based mechanism, we could introduce a probabilistic approach on the AL-FEC parameters selection. Finally, as the newly introduced RaptorQFEC scheme greatly addresses the shortcomings of the existing Raptor code, we could examine the possibility of utilizing AL-FEC protection over ptp environments where previously its utilization was considered inefficient. Also a next step we can analyze the performance of AL-FEC scheme called LDPC-Staircase and Triangle (Inria/STM)specified in IETF RFC 5170 and compare it with Raptor and RaptorQ AL-FEC schemes. LDPC-Staircase codes are included in the ISDB-TmmJapanese standard for mobile multimediaand are also considered as potential candidate for newer releases of 3GPP eMBMS specifications.

ACKNOWLEDGEMENTS

We would like to thank Yashesh Buch for providing such an interesting topic and also grateful to him for valuable discussions and friendly support.

REFERENCES

1. 3GPP TS 26.346 V10.4.0.," Technical Specification Group Services and System Aspects: MBMS, Protocols and codecs", (Release 10), 2012.
2. Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., &Crowcroft, J. (2002). The use of forward error correction (FEC) inreliable multicast. RFC 3453 (Informational). URL http://www.ietf.org/rfc/rfc3453.txt.
3. M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, "Raptor forward error correction scheme for object delivery," September 2007, Internet Engineering Task Force, RFC 5053. Available at http://tools.ietf.org/html/rfc5053.
4. 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. Available at: http://tools.ietf.org/rfc/rfc6330.txt
5. A. Shokrollahi, "Raptor codes," Information Theory, IEEE Transactions on, vol. 52, no. 6, pp. 2551–2567, June 2006.
6. M. Luby, "LT codes", In Proceedings of The43rd Annual IEEE Symposium on Foundations ofComputer Science, November 16-19 2002, pp. 271–282.
7. 3GPP. (2005). Specification text for systematic raptor forward error correction. TSG SA WG4 S4-AHP205, 3rd generation partnership project (3GPP).
8. Todor Mladenov, Saeid Nooshabadi, Kiseon Kim, "Efficient GF(256) Raptor Code Decoding for Multimedia Broadcast/Multicast Services and Consumer Terminals", *IEEE Transactions on Consumer Electronics*, Vol. 58, No. 2, May 2012
9. Luby M, Watson M, Gasiba T, Stockhammer T, Xu W." Raptor codes for reliable download delivery in wireless broadcast systems." In Proc. of CCNC, 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, ISDBT and Media Flo, CRC Press, pp. 239–280, 2008.
11. 3GPP."Rationale for MBMS AL-FEC enhancements."Tdoc S4-110449, 3rd generation partnership project (3GPP), (2011).
12. T. Mladenov, K. Kim, and S. Nooshabadi, "Forward error correctionwith RaptorQ Code on embedded systems," in Circuits and Systems(MWSCAS), 2011 IEEE 54th International Midwest Symposium on.IEEE, pp. 1–4.