Packet Loss Recovery in the Telemetry Downlink via Maximum-Likelihood LDPC Decoding

Gianluigi Liva
Institute of Communication and Navigation
Deutsches Zentrum fur Luft- und Raumfahrt (DLR)
82234 Wessling, Germany
Gianluigi.Liva@dlr.de

Enrico Paolini and Marco Chiani
DEIS, WiLAB
University of Bologna
via Venezia 52, 47023 Cesena (FC), Italy
{e.paolini,marco.chiani}@unibo.it

***Abstract*—In this paper a novel framework for packet loss recovery in the Consultative Committee for Space Data Systems (CCSDS) telemetry downlink is presented. The framework relies on packet-level LDPC erasure correcting codes and on low-complexity maximum likelihood (ML) decoding. A code design tailored to the ML decoder is presented, which is based on Generalized Irregular Repeat-Accumulate (GeIRA) codes. An optional concatenation with outer BCH codes is proposed as a mean for lowering the error floors. Numerical results show that for short block lengths the outer concatenation gives rise to very low error floors, while already for moderate-length codes the concatenation can be skipped without sacrificing the erasure recovery performance. Remarkably, all the proposed schemes under ML decoding tightly approach the performance of ideal maximum distance separable (MDS) codes down to low error rates.**

***Index Terms*—LDPC codes, erasure channel, channel coding, satellite communications, CCSDS.**

I. Introduction

A recent flurry of research activities has been focused on the application of error control techniques to the protocol stack layers different from the physical one. Such coding techniques deal with the use of a (linear) block code applied to *encoding units* (symbols) that are usually packets with constant size. We will refer in the following to such coding techniques as *packet level coding*. The packet level encoder receives as input a set of k symbols (packets), and produces as output $n > k$ packets. Assuming systematic encoding, the final set of packets comprises the subset of k information packets together with a subset of $m = n - k$ checksum (parity) packets. On the receiver side, after the physical layer decoding followed by a validation test either provided by the channel code or performed through a CRC, the packets that have been validated are forwarded to the packet-layer decoder. The packets that are corrupted (due to decoding failures at physical layer) are marked as lost. This is the case of the CCSDS [1] standard, where the frames received at the physical layer are either successfully decoded or discarded. Therefore, the upper layers deal with packet erasures, whose characteristics may depend on the physical channel [2]. The role of the packet-level decoder is therefore to recover the erased packets by use of the redundant packets introduced on the transmitter side.

An appealing application for packet level coding deals with file-based transmission, where even a small (uncorrelated) erasure rate may result in an unacceptable file failure rate. It is clear that, in a context where retransmission of packets is not feasible (in the satellite communication link scenario this is often the case), packet level coding shall be used to increase the probability of success in file delivery applications. A further application may be represented by the upload of critical software update to a device operating on a satellite/space probe, where high reliability is required.

Within the CCSDS, the Long Erasure Correcting (LEC) Codes Birds Of Feather (BOF) is currently investigating the adoption of low-density parity-check (LDPC) codes for packet erasure recovery in deep-space communication systems [3]. In [4], a novel approach to the design of LDPC codes and decoders for the binary erasure channel (BEC) is proposed. In fact, while most of the research has been addressed to the design on capacity-approaching LDPC codes on the erasure channels for message-passing decoders, less effort was devoted to the construction of LDPC codes for ML decoders. If the communication channel is a BEC, ML decoding is equivalent to solving the linear equation system provided by the code's parity-check matrix $\mathbf{H}$, where the unknown variables correspond to the erased packets. Then, ML decoding can be implemented as a Gaussian elimination performed on the binary matrix composed by the columns of $\mathbf{H}$ placed in correspondence with the erased packes: its complexity is in general $O(n^3)$, where n is the codeword length. It is obvious that for an (n, k) linear block code with a large n ML decoding becomes impractical. However, for LDPC codes it is indeed possible to take advantage of both the ML and iterative (IT) approach. To keep the complexity low, a first decoding attempt may be done in an iterative manner. If not successful, the residual set of unknowns is processed by an ML decoder. Efficient ways of implementing ML decoders for LDPC codes can be found in [5], whose approach takes basically benefit from the sparse nature of the parity-check matrix of the code. In this paper, we will show how properly designed LDPC codes provide, under ML decoding, nearly-ideal MDS code performance on the erasure channel. The result is achieved with low decoding complexity (enough for real-time high-data-rate software decoding), keeping an efficient code structure which facilitates the encoding. The (n, k) code design fulfills the following requirements:

978-1-4244-2573-0/08/$25.00 ©2008 IEEE

- Almost capacity-achieving ML threshold (ϵ_{ML}).
- High IT threshold (ϵ_{IT}). Iterative decoding thresholds close to the capacity permit to reduce the complexity of the ML decoder. Moreover, they keep the door open for the adoption of IT decoding only.
- Low error floors.

Based on those requirements, a class of LDPC GeIRA codes tailored to both ML and IT decoding has been proposed within the LEC-BOF of the CCSDS [6]. Through this paper, we will provide some insights on the framework presented in [6].

The paper is organized as follows. In Section II, some basic concepts on ML decoding of LDPC codes on the BEC are recalled. Section III presents the GeIRA codes design for ML decoders. Numerical results are provided in Section IV. Conclusions follow in Section V.

II. Maximum-Likelihood Decoding of LDPC codes over the Erasure Channel

Let us consider an (n, k) binary linear block code with parity-check matrix $\mathbf{H}$. Over the BEC, ML decoding is equivalent to solving the linear equation

$$\mathbf{x}_{\overline{K}}\,\mathbf{H}_{\overline{K}}^{T} = \mathbf{x}_{K}\,\mathbf{H}_{K}^{T}\,. \tag{1}$$

In (1) $\mathbf{x}_{\overline{K}}$ and $\mathbf{x}_{K}$ denote the set of erased and correctly received encoded bits, respectively. Analogously, $\mathbf{H}_{\overline{K}}$ and $\mathbf{H}_{K}$ denote the submatrices composed of the $\mathbf{H}$ columns corresponding to $\mathbf{x}_{\overline{K}}$ and $\mathbf{x}_{K}$, respectively. The equation (1) can be solved by Gaussian elimination (GE) on $\mathbf{H}_{\overline{K}}$: its complexity is in general $O(n^3)$. As n increases ML decoding may become impractical. In the case of LDPC codes, large block lengths can be afforded by using IT decoding, which provides a good erasure recovery capability with $O(n)$ complexity.

However, even for moderate-to-large block lengths (in the order of several thousands symbols), ML decoding of LDPC codes over the BEC can be practically implemented thanks to a reduced complexity approach [5], [7]. Starting from that, a hybrid iterative-maximum likelihood decoder has been proposed in [4]. Provided that the number of erasures does not exceed $n - k$, a first decoding attempt is done iteratively. In case of IT decoder failure, the erasure pattern might be a reduced version of the original one, i.e., some erasures could have been solved by the iterative decoder. Let's denote by e the size of the so-called *residual erasure pattern* after iterative decoding. In a second phase, check nodes which are not connected to erased variable nodes are inactivated, since they will not be used in the next steps (see the example in Figure 1). The inactivation of the useless check nodes helps in the sense of reducing the matrix over which ML decoding is applied. Furthermore, it may happen that after the IT decoding step the number of residual check nodes is less than e, and the system is recognized to be unsolvable in an early stage.

Without loss of generality, we will denote by $\mathbf{H}$ the expurgated parity-check matrix (without the rows related to the inactivated check nodes, thus with $m' \leq m$ rows). Similarly, expurgated version will be considered for $\mathbf{H}_{\overline{K}}$ and $\mathbf{H}_{K}$. From the bipartite graph point of view, the splitting between known variables ($\mathbf{x}_{K}$) and unknown variable ($\mathbf{x}_{\overline{K}}$) is depicted in Figure 2. Here, the graph describes the expurgated $\mathbf{H}$, and is referred as the *residual graph* after IT decoding.

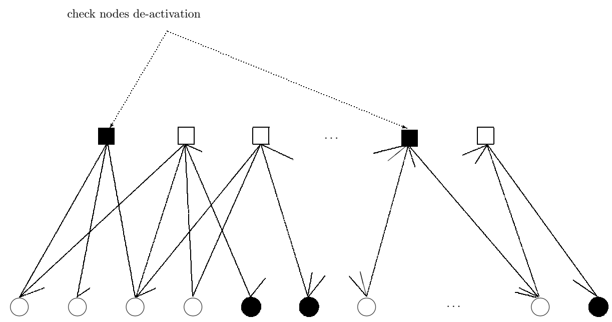

Fig. 1. Bipartite graph of the LDPC, after IT decoding. Dark circles represent the residual erased variables. The check nodes connected just to known variables are inactivated.

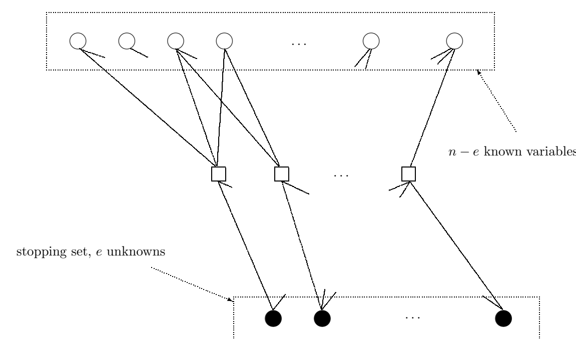

Fig. 2. Bipartite graph of the LDPC after check node inactivation. Known variables have been grouped on the upper part of the graph. The residual erasures (stopping set) lies in the lower part.

At this stage, a partial syndrome vector $\mathbf{s}$ can be built. The vector $\mathbf{s}$ possesses m' elements, one for each check node in the residual graph. The i-th element of $\mathbf{s}$ is given by the bit-wise sum of the known variables connected to the i-th check node (Figure 3). The computation of $\mathbf{s}$ is linear in n, and leads to the following relation:

$$\mathbf{x}_{\overline{K}}\,\mathbf{H}_{\overline{K}}^{T} = \mathbf{s}\,. \tag{2}$$

ML decoding proceeds as follows. $\mathbf{H}_{\overline{K}}$ is first reduced to the approximate triangular matrix $\mathbf{H}'_{\overline{K}}$ depicted in Fig. 4(a) (where $\mathbf{T}$ is an $((e-\alpha)\times(e-\alpha))$ lower triangular matrix) through row / column permutations only. The operation can be carried out following the Maximum Column Weight (MCW) pivoting algorithm described next (see Alg. 1 description).

The columns moved in the pivoting phase will constitute the α leftmost columns of $\mathbf{H}'_{\overline{K}}$. In the next step, $\mathbf{C}$ is made equal to the null matrix by row additions, leading to $\mathbf{H}''_{\overline{K}}$ in Fig. 4(b). In parallel, the corresponding elements of $\mathbf{s}$ are summed as well. Finally, GE is applied to $\mathbf{R}$, thus

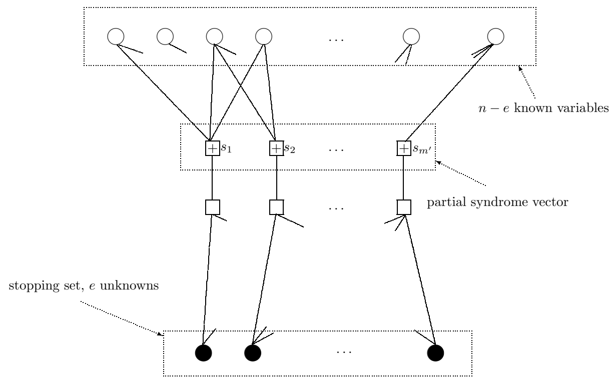

Fig. 3. From the known variables, the partial syndrome vector $\mathbf{s}$ can be computed. For the i-th check node ($i = 1 \dots m'$) in the residual graph, the partial syndrome value s_i is obtained by bit-wise sum of the known variables connected to the check node.

Algorithm 1 MCW Pivoting Algorithm

Set q=1.
for $j = 1 \dots e$ **do**
  for $i = q \dots m'$ **do**
    compute the weight of the i-th row of $\mathbf{H}_{\overline{K}}$, starting from the j-th position.
  end for
  if there is a r-th weight-1 row and the non null element belongs to the c-th column **then**
    procedure TRIANGULIZATION
      Insert the r-th row in the q-th position and the c-th column in the j-th position.
      Move the r-th element of $\mathbf{s}$ in the q-th position.
      Increment q by 1.
    end procedure
  else
    procedure PIVOTING
      Select the largest weight column with index greater or equal than j, and instert it in position 1.
    end procedure
  end if
end for

recovering the α leftmost unknowns (called the *reference variables* or *pivots*). The remaining $e-\alpha$ unknowns are solved by simple back-substitution. Note that a decoding failure takes place whenever the rank of $\mathbf{R}$ is smaller than α. Being the complexity dominated by the GE step applied to $\mathbf{R}$, a small α is highly desirable. This can be obtained for LDPC codes by adopting a smart technique for the reference variables selection [5] together with a judicious code design [4]. The above-proposed pivoting technique is rather simple compared to those proposed in [5], but provides already excellent results. Consider a (2048,1024) Irregular Repeat-Accumulate (IRA) code from [4] with degree distributions

$$\lambda(x) = 0.222x + 0.216x^2 + 0.031x^6 + 0.192x^8 + +0.047x^{17} + 0.075x^{18} + 0.217x^{53},$$
$$\rho(x) = x^8.$$

The code ensemble possesses the decoding thresholds $\epsilon^*_{IT} = 0.496$ and $\epsilon^*_{ML} = 0.499$. A test has been carried out by generating random erasure patterns with $e = 1016$ erasures, thus extremely close to the ultimate error correction capability of an MDS code. The average number of pivots obtained in 1000 tries was $\overline{\alpha} = 36.95$, while the peak observed value of α was 53. A throughput evaluation has been carried out by implementing the above-described decoding algorithm in software (ANSI C). For the test, symbols of 4000 bytes were considered. The (2048,1024) IRA code was used, and the erasure pattern size was fixed to $e = 1016$. In these conditions, on a Intel Core Duo E6850 (3.00 GHz) platform, the software decoder was providing a throughput of 1.2 Gbps. Note that this erasure pattern size was leading to the lowest decoding speed (i.e. for larger/lower erasure patterns, the decoder achieved larger throughputs).

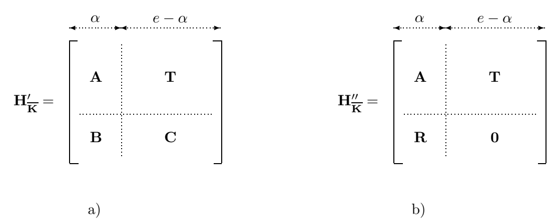

Fig. 4. Steps for reduced complexity ML decoding.

III. GENERALIZED IRA CODES FOR ML DECODING

The encoding structure investigated in this paper is depicted in Figure 5, and consist of a GeIRA encoder [8] with an (optional) outer (extended/shornetned) BCH code. These LDPC codes are characterized by a simple encoding and offer a large flexibility in terms degree distribution selection. Respect to IRA codes [9] they allow reducing the fraction of degree-2 variable nodes (VNs) which usually affects the performance in the error floor region. GeIRA codes are systematic LDPC codes generating the parity bits by a serial concatenation of an outer low-density generator matrix (LDGM) code with an inner rate-1 recursive convolutional code (RCC), as depicted in Figure 5. Decomposing the parity-check matrix as $\mathbf{H} = [\mathbf{H}_u|\mathbf{H}_p]$, where $\mathbf{H}_u$ corresponds to the k systematic bits and $\mathbf{H}_p$ to the $m = n - k$ parity bits, we have that $\mathbf{H}_u^T$ is the outer LDGM code generator matrix. Moreover, $\mathbf{H}_p$ is specified by the feedback polynomial $g(D) = \sum_{j=0}^{t} g_j D^j$ of the inner rate-1 RCC (where $g_j \in \{0, 1\}$ and $g_0 = g_t = 1$). The outer BCH code permits, for specific block lengths/code rates, to achieve low error floors. In particular, we will see in Section IV how medium-long, irregular GeIRA codes and short, nearly-regular ones do not need the outer concatenation for achieving low error floors, while short, irregular ones may need it. The serially-concatenated BCH-GeIRA encoder

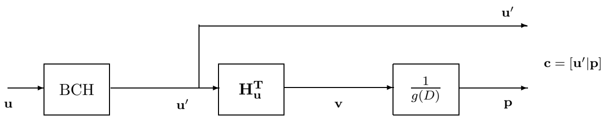

Fig. 5. Encoder structure for serially-concatenated BCH-GeIRA codes.

structure is depicted in Fig. 5, where the input word of a (n, k') GeIRA encoder is pre-coded by a (k', k) binary BCH code. In order to adapt the codeword length of the BCH code to the input block size of the GeIRA encoder, shortened or extended BCH codes may be adopted. The parity-check matrix of the serially-concatenated BCH-GeIRA code is in the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\text{BCH}} & \mathbf{0} \\ \mathbf{H}_u & \mathbf{H}_p \end{bmatrix}, \quad (3)$$

where $\mathbf{H}_{\text{BCH}}$ is the $((k' - k) \times k')$ parity-check matrix of the outer BCH code, $[\mathbf{H}_u|\mathbf{H}_p]$ is the $((n - k') \times n)$ parity-check matrix of the inner GeIRA code, and $\mathbf{0}$ is the $((k' - k) \times (n - k'))$ all-0 matrix.

The parity-check matrix (3) is less sparse due to the presence of the dense submatrix $\mathbf{H}_{\text{BCH}}$. This introduces a slight complexity increment in terms of reference variables number. We found that the number of additional reference variables respect to the non-concatenated case is on the order of $k' - k$: it is indeed possible to preserve a manageable ML decoding complexity by adopting high-rate outer codes.

IV. Numerical Results

We next compare the performance of GeIRA codes, with short-to-moderate block size, with some well-known bounds. In particular, the codes performance is compared with the Singleton lower bound on the codeword error rate (CER) (i.e., the CER of an ideal $d_{min} = n - k + 1$ MDS code) and with the Berlekamp (n, k) random coding bound [10].

A. Short Codes Designs

For short LDPC codes, several solutions have been considered. The first one deals with a design based on irregular degree distributions ensemble, characterized by high IT/ML decoding thresholds. Such choice follows the design guidelines proposed in [4] for achieving a good trade-off between error floor, waterfall performance and decoding complexity (i.e., number of pivots which are required in the average). Those rules can be summarized in the need of finding an ensemble with ML threshold as close as possible to the Shannon limit, good IT decoding threshold and a reduced fraction of degree-2 variable nodes. Nevertheless, while those rules are successful for moderate-to-large block sizes, they are less effective when applied to the design of short codes. Consider the case of a $(512, 256)$ code design. According to the guidelines of [4], a GeIRA code has been designed from the ensemble defined by the degree distributions

$$\begin{aligned} \lambda(x) =\ & 0.111x + 0.429x^2 + 0.284x^{13} + \\ & +0.122x^{52} + 0.054x^{53}, \\ \rho(x) =\ & x^8. \end{aligned}$$

The IT decoding threshold of the ensembles is $\epsilon_{IT} = 0.480$, while the ML one is $\epsilon_{ML} = 0.498$. The $(512, 256)$ GeIRA code has been constructed by selecting the feedback polynomial as $g(D) = 1 + D + D^{122}$, and by filling the remain part of the parity-check matrix according to the above-specified degree distributions and by exploiting girth-optimization techniques [11]. A second approach deals with the design of a near regular GeIRA code [4] with constant information variable node degree equal to 4 and with feedback polynomial $g(D) = 1 + D + D^4 + D^{10}$. Here, the IT decoding threshold is much lower, $\epsilon_{IT} = 0.383$, while the ML one is almost the same of the irregular LDPC ensemble. The much lower IT threshold may result in a larger number of reference variables. However, due to the short block size of the code, the larger α can be easily afforded. In Figure 6, the performance of the two codes is compared. The irregular code shows an error floor already at CER$\simeq 10^{-4}$, while the regular one does not down to CER$\simeq 10^{-6}$. Remarkably, according to the error floor estimation provided in [12], the regular code shall present its floor at CER$\simeq 10^{-20}$. Both codes, thanks to their excellent ML threshold, present a waterfall performance that is very close to the Singleton bound. Although the irregular code possesses a good IT threshold too, the gap between the IT decoding curve and the ML one is remarkable. A third approach considered here takes advantage of the concatenation between the (inner) irregular GeIRA code with an outer $(256, 247)$ extended BCH (Hamming) code. The resulting code $(512, 247)$ concatenated code possesses an effective code rate of $R \simeq 0.482$, thus slightly lower than $1/2$. In Figure 7, the performance of the irregular GeIRA code is compared with that of the concatenated BCH-GeIRA one. For having a fair comparison, the Singleton bounds for both codes are provided. Hence, one shall consider the actual gap between each code and its bound. A first remark deals with the huge gain in terms of error floor achieved thanks to the concatenation. The error floor for the concatenated code does not appear down to CER$\simeq 10^{-7}$. Furthermore, the gap between the performance curve and the Singleton bound is lower when the concatenation is used. A final remark on the complexity. A specific test has been carried out by measuring the average value of α $(\overline{\alpha})$ hen the decoder was operating with an overhead of 8 symbols (i.e., the number of correct symbols per codeword was set to $k + 8$) and with random erasure patterns. The irregular code presented $\overline{\alpha} = 11.65$, for the regular one $\overline{\alpha} = 49.13$, and for the concatenate code $\overline{\alpha} = 17.67$.

B. Rate-compatible GeIRA codes

In Figure 8, some numerical results are shown for families on GeIRA codes with different input block sizes. The codes performance is depicted in terms of CER vs. the BEC erasure probability ϵ. Two families of codes are considered, with input block sizes $k = 247$ and $k = 502$. The inner codes are obtained by puncturing the parity-bits of a $(512, 256)$ GeIRA code and of a $(1024, 512)$ GeIRA code. These two belong to the same ensemble, characterized by the degree distributions

The IT decoding threshold of the mother code ensembles

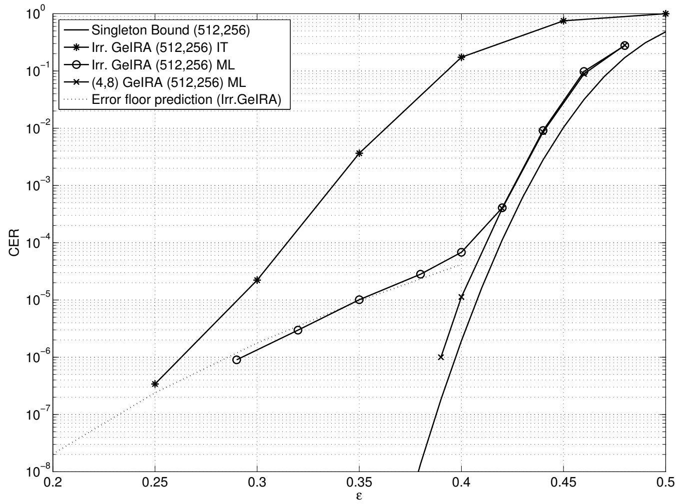

Fig. 6. CER vs. the BEC erasure probability ϵ for different (512,256) GeIRA codes.

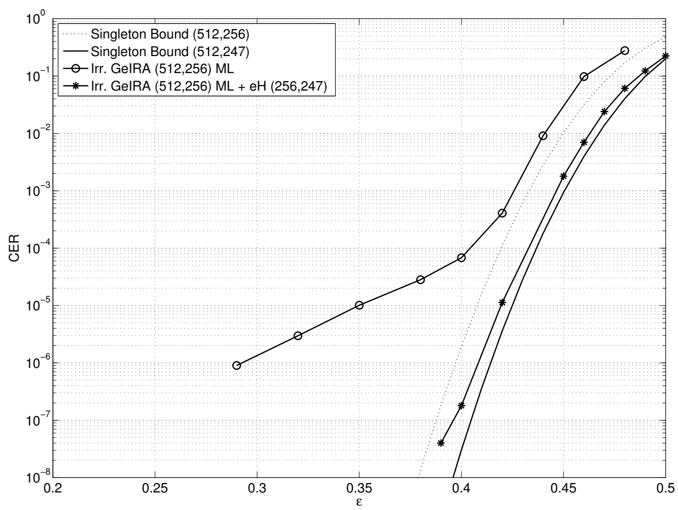

Fig. 7. CER vs. the BEC erasure probability ϵ for the irregular (512,256) GeIRA code, with and without the outer BCH code concatenation.

is $\epsilon_{IT} = 0.480$, while the ML one is $\epsilon_{ML} = 0.498$. The outer code adopted for the $(n, 256)$ GeIRA code family is a $(256, 247)$ extended Hamming code. For the $(n, 512)$ GeIRA code family the outer code is a $(512, 502)$ extended Hamming code. The actual code rates for the mother codes are thus $R = 247/512 \simeq 0.482$ and $R = 502/1024 \simeq 0.490$, respectively. Two higher code rate ($R \simeq 2/3$ and $R \simeq 4/5$) codes are obtained for each family by puncturing with periodic patterns the parity bits at the output of the generalized accumulator. For both families, the CER performance tightly follows the respective random coding bound, approaching the performance of an ideal MDS down to very low codeword error rates. For an $(2048, 1024)$ GeIRA code (designed with the same distributions adopted for the $k = 256$ and $k = 512$ codes), even without the outer BCH/Hamming code concatenation, it is possible to approach the Singleton bound down to low error rates.

C. *Performance vs. Overhead*

In Figure 9, the decoding failure probability (i.e., the CER) as a function of the overhead is depicted for some GeIRA codes, and for the Raptor codes specified in [13]. The overhead δ is here defined as the number of codeword symbols that are correctly received in excess respect to k (recall that k represents the minimum amount of correctly-received bits allowing successful decoding with an ideal MDS code). The comparison is carried out for various block sizes. There is basically no difference in performance between the MBMS Raptor codes and properly-designed LDPC codes (for both, under ML decoding). The performance of a $(512, 256)$ random binary linear block code is presented as well. In this case, the decoding failure probability can be expressed as

$$P_f(\delta, m) = 1 - \prod_{i=1}^{m-\delta} \left(1 - \frac{2^{i-1}}{2^m}\right), \qquad (4)$$

where $m = n - k$. One can note that the random code performance slightly depend on m, when m is large enough. Considering the case where $\delta = 0$, we have that

$$\lim_{m \to +\infty} P_f(0, m) \simeq 0.71121.$$

Already for $m = 10$, the failure probability would be $P_f(0, 10) \simeq 0.7109$. Respect to random binary linear block codes, properly-designed GeIRA codes suffer for roughly $1-2$ symbols of extra-overhead.

V. Conclusions

In this paper a design of packet-level LDPC codes for CCSDS space downlink telemetry has been proposed. The presented approach faces the problem of achieving near-optimum performance on the erasure channel by adopting low-complexity maximum-likelihood decoders and by designing the codes for ML decoding rather than just for IT decoding. The lowest decoding speed measured for the software implementation of the ML decoding algorithm, with an (2048,1024) code, was 1.2 Gbps. Low-complexity maximum-likelihood decoders, together with proper code designs, represent a straightforward solution for enhancing CCSDS space telemetry, even for high data-rate applications.

Acknowledgements

This work has been supported by the EC-IST projects SatNEx-II (IST-27393) and Optimix (IST-214625).

References

[1] "Telemetry Synchronization and Channel Coding," Blue Book, Issue 1, Consulative Committee for Space Data Systems (CCSDS), Newport Beach, CA, Sept. 2003.

[2] E. Paolini, G. Liva, M. Chiani, and G. Calzolari, "Tornado-like codes: a new appealing chance for space applications protocols?" in *3rd European Space Agency Workshop on Tracking, Telemetry and Command Systems for Space Applications, TTC 2004*, Sept. 2004.

[3] G. P. Calzolari, M. Chiani, F. Chiaraluce, R. Garello, and E. Paolini, "Channel coding for future space missions: New requirements and trends," *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2157–2170, Nov. 2007.

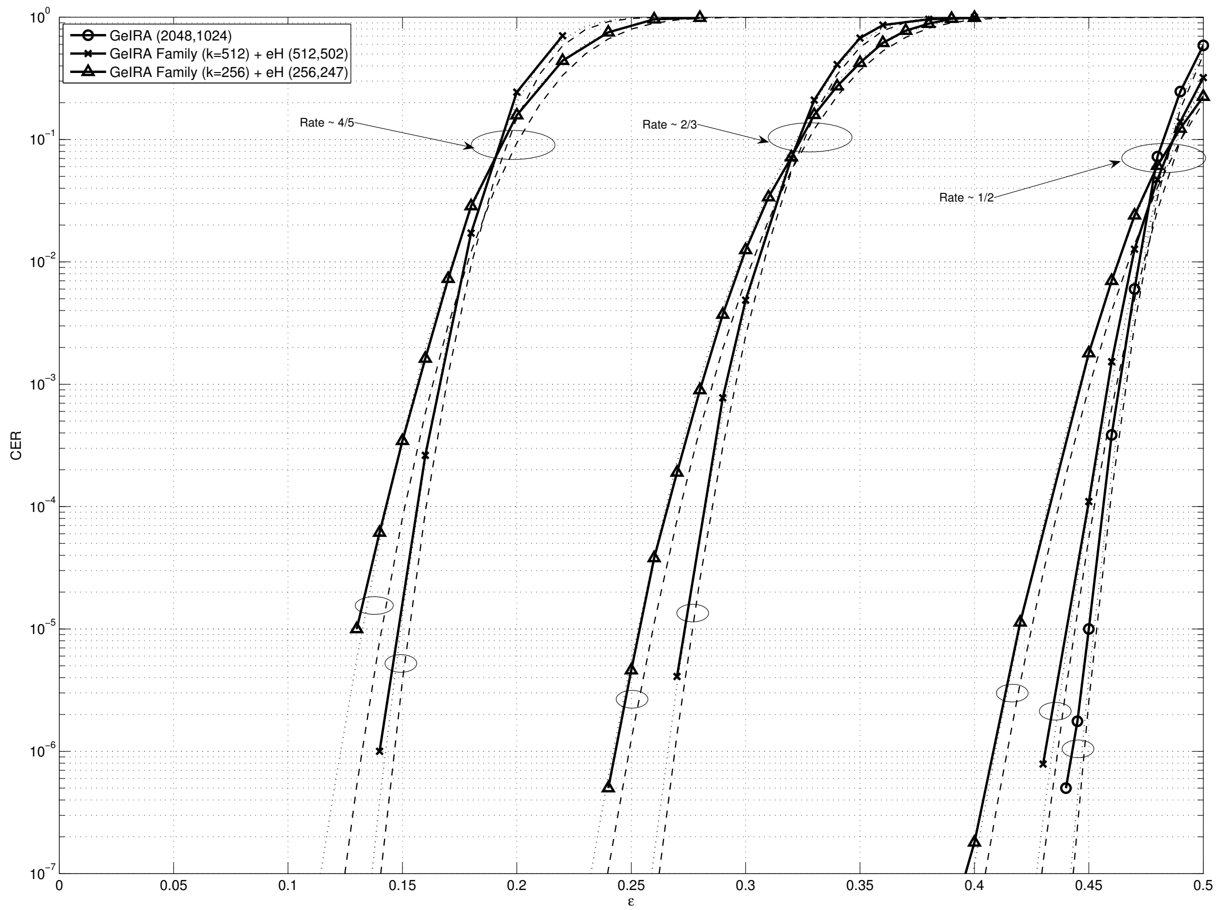

Fig. 8. CER vs. the BEC erasure probability ϵ for two families ($k = 247$ and $k = 502$) of concatenated extended Hamming - GeIRA codes, and for a $(2048, 1024)$ GeIRA code. Dashed lines represent the Singleton lower bound on the CER, dotted lines represent the Berlekamp upper bound on the average random codes CER.

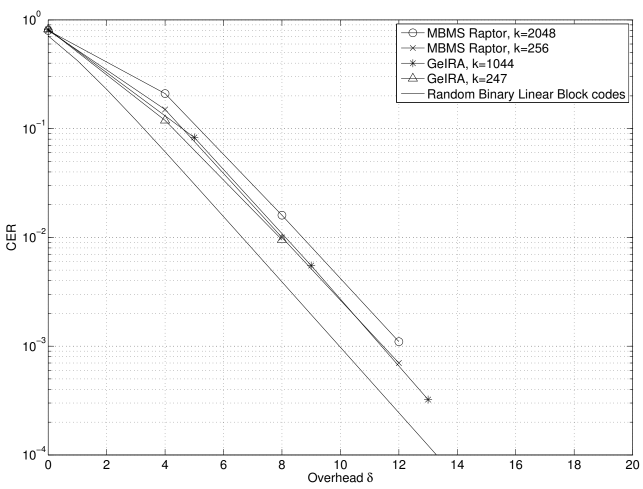

Fig. 9. Codeword Error Rate vs. overhead for Raptor and GeIRA codes vs. the performance achievable by random binary linear block codes.

[4] E. Paolini, G. Liva, B. Matuz, and M. Chiani, "Generalized IRA Erasure Correcting Codes for Hybrid Iterative / Maximum Likelihood Decoding," *IEEE Communications Letters*, June 2008.

[5] D. Burshtein and G. Miller, "An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, nov 2004.

[6] G. Liva, M. Chiani, E. Paolini, and G.P.Calzolari, "Achieving Nearly-Ideal Performance over the Binary Erasure Channel with LDPC codes," Consultative Committee for Space Data Systems, Tech. Rep., March 2008, CCSDS Spring meeting.

[7] B. A. LaMacchia and A. M. Odlyzko, "Solving large sparse linear systems over finite fields," *Lecture Notes in Computer Science*, vol. 537, pp. 109–133, 1991.

[8] G. Liva, E. Paolini, and M. Chiani, "Simple reconfigurable low-density parity-check codes," *IEEE Commun. Lett.*, vol. 9, no. 3, pp. 258–260, Mar. 2005.

[9] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. International Symposium on Turbo codes and Related Topics*, Sept. 2000, pp. 1–8.

[10] E. Berlekamp, "The technology of error-correcting codes," *IEEE Proc.*, vol. 68, pp. 564–593, 1980.

[11] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth Tanner graphs," in *GLOBECOM*, San Antonio, Texas, Nov. 2001, pp. 995–1001.

[12] E. Paolini, G. Liva, M. Varrella, B. Matuz, and M. Chiani, "Low-Complexity LDPC Codes with Near-Optimum Performance over the BEC," in *Proc. 4th Advanced Satellite Mobile Systems Conference*, Bologna, August 2008. [Online]. Available: http://aps.arxiv.org/abs/0804.2991

[13] 3GPP TS 26.346 V6.1.0, "Technical specification group services and system aspects; multimedia broadcast/multicast service; protocols and codecs," June 2005.