CLOSE-TO-CAPACITY DECODING ON THE PACKET ERASURE CHANNEL (PEC)

B. Matuz, G. Garrammone
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
82234 Weßling, Germany

Overview

Research on packet-layer coding is driven by two main factors: keeping encoding/decoding complexity low, while finding suitable code designs to approach channel capacity. A class of codes that fulfills these requirements perfectly is the one of low-density parity-check (LDPC) codes. Recent works showed that nearly optimal performance can be achieved with high decoding speed, down to low error rates and even for small and moderate block sizes. This is thanks to maximum-likelihood (ML) aided decoding techniques in combination with a judicious code design. A review of the ML decoding approach will be first presented, followed by further insights on the code design. Performance and decoding speed assessments will be both provided in terms of simulation and field test results.

1. INTRODUCTION

Nowadays the use of upper layer forward error correction (ULFEC) is steadily gaining popularity [1]–[4]. Areas of application include wireless audio and video streaming (DVBH/SH standards) or file delivery as an alternative to automatic repeat request (ARQ) [5]–[7]. Moreover, there are ongoing discussions at Consultative Committee for Space Data Systems (CCSDS) to apply ULFEC also for space communications [8].

In principle upper layer (UL) codes operate on data units (referred to as packets in the sequel) that represent the symbols of the code. In order to perform the error correction, the decoder has to know which of the symbols/packets have been received correctly and which of them are erroneous. To this end, after physical layer decoding usually a cyclic redundancy check (CRC) check is performed on each data unit. Note that a packet is marked as erased if contains at least one bit error. Hence, from the point of view of the upper layer decoder the communication channel looks like a PEC with erasure probability ε: packets are either correctly received or lost.

Depending on the application the code is designed with different requirements, such as excellent erasure recovery capabilities, low encoding and decoding complexity or flexibility in the choice of the block length. A class of codes that matches these requirements quite well is the class of low-density parity-check (LDPC) codes [9] thanks to their sparse parity-check matrix and the belief propagation decoding algorithm. This decoding algorithm is also referred as iterative (IT) decoding, since iteration by iteration information is exchanged among variable nodes (VNs) and check nodes (CNs) along the edges of the Tanner graph of the code.

However, for high erasure probabilities in the waterfall region, a clear gap in performance with respect to the theoretical lower bound, the so-called Singleton bound, is observed. Moreover, at low erasure probabilities an error-floor might rise-up due to so-called stopping sets [10]. To mitigate these effects, a new hybrid IT/ML decoder was proposed by DLR [11], [12]. Simulations and on-field trials with the new decoder in the framework of the ESA sponsored project Ortigia and its follow-up J-Ortigia [13], showed that the obtained performance is closely approaching the Singleton bound down to low erasure probabilities. Moreover, a low complexity implementation of the algorithm allows speeds in the range of a few Gbps, via software.

In the following we will restrict our attention on binary LDPC block codes. First, the hybrid decoder will be described. Then, peculiarities on the code design will be pointed out in order achieve high decoding speeds at nearly optimal performance. Next, some simulation and field test results will be presented.

2. THE DECODING ALGORITHM

Usually, decoding of LDPC codes is based on belief propagation, where information is exchanged iteration by iteration among the nodes. Iterative decoding of LDPC codes on the erasure channel is extremely simple, since it does not need any manipulation of the code's parity-check matrix. The erased packets are, in fact, recovered by using the set of equations provided by the parity-check matrix **H**, in an iterative manner. The algorithm proceeds as follows.

1) Initialization. Mark all the variable nodes in the code's bipartite graph as correct/lost, depending on the state of the corresponding codeword packet.
2) Go through all the check nodes. If there exists a check node connected to only 1 lost packet, then recover the lost packet by bit-wise sum of the neighboring packets. Otherwise, stop the decoder.
3) Repeat step 2.

Thanks to the simplicity of the algorithm high decoding speeds can be achieved, while sacrificing decoding performance for short/moderate length codes (< few thousand symbols). To achieve close-to-optimal performance, even for short/moderate code lengths, the authors proposed new decoder architecture based on maximum-likelihood decoding [11].

In order to introduce the proposed decoding scheme, let us consider a binary (n,k) linear block code with parity-check matrix **H**, where a set of k information symbols is encoded into a set of n coded symbols. Over the PEC, ML decoding is equivalent to solving the linear equation

$$\mathbf{x}_{\overline{K}} \mathbf{H}^T_{\overline{K}} = \mathbf{x}_K \mathbf{H}^T_K.$$

Recall that $\mathbf{x}_{\overline{K}}$ and $\mathbf{x}_K$ denote the set of erased and correctly received encoded symbols, respectively. Analogously, $\mathbf{H}_{\overline{K}}$ and $\mathbf{H}_K$ denote the submatrices composed of the **H** columns corresponding to $\mathbf{x}_{\overline{K}}$ and $\mathbf{x}_K$, respectively. Note that the right hand side of the equation is fully known, whereas the left hand side contains the code symbols corresponding to the lost packets. This linear system of equations 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 however, large block lengths n can be easily managed by using IT decoding, which provides good erasure recovery capabilities with $O(n)$ complexity.

Even for moderate-to-large block lengths, ML decoding of LDPC codes over the PEC can be practically implemented thanks to a reduced complexity approach [14], which exploits the sparse nature of the LDPC parity-check matrix: provided that the number of erasures/unknowns e does not exceed the number of equations n-k, $\mathbf{H}_{\overline{K}}$ is first reduced to the approximate triangular matrix $\mathbf{H}'_{\overline{K}}$ through row/column permutations only. The matrix structure is depicted in FIGURE 1 a), where **T** is the obtained lower triangular (square) matrix. In a second step, **C** is made equal to the null matrix by row additions, leading to $\mathbf{H}''_{\overline{K}}$ in FIGURE 1 b). Finally, GE is applied to **R**, thus recovering the leftmost α unknowns (called *reference variables*). The remaining e-α unknowns are solved by simple back-substitution. Being the complexity dominated by the GE step applied to **R**, a small number of *reference variables* α is highly desirable. This can be obtained for LDPC codes by adopting a smart technique for the reference variables selection [14] together with an appropriate code design [16]. Note that a decoding failure takes place whenever the rank of **R** is smaller than α.

A simple enhancement in terms of complexity is given by the proposed hybrid IT/ML decoder (IML), in which a first decoding attempt is done iteratively. In case of IT decoding failure, a final attempt is done by ML decoding. This decoder assures the same performance of the ML one, but with a reduced complexity.

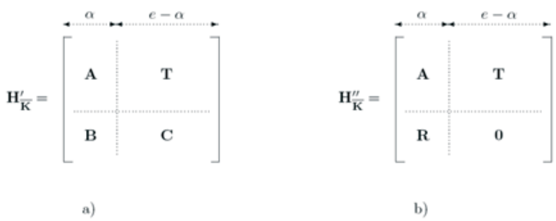

FIGURE 1. Maximum-likelihood decoding algorithm based on structured Gaussian elimination.

3. CODE DESIGN

The proposed hybrid IT/ML decoding scheme requires a revision of the conventional code design. The usual code design employed for LDPC codes over PEC deals with the selection of a proper variable node degree distribution λ(x) and a proper check node degree distribution ρ(x), achieving high iterative decoding thresholds ε_{IT} (as close as possible to the limit given by 1-R, where R is the code rate). A (*n,k*) LDPC code is then selected from the ensemble defined by the above-mentioned degree distributions. The selection may be performed following some girth optimization techniques [19]. Such iterative-decoding-based design criterion does not answer to the need of finding good codes for ML decoding. Namely, a different figure shall be put in the focus of the degree distribution optimization. In our code design, the corresponding feature of ε_{IT} under ML decoding, i.e. the ML decoding threshold ε_{ML}, is the subject of the figure driving the optimization. A method for deriving a tight upper bound on the ML threshold for an LDPC ensemble can be found in [17]. The upper bound on ε_{ML} can be derived as follows.

- Consider an *(n,k*) LDPC code and its corresponding IT decoder. The extrinsic information transfer (EXIT) curve of the code (under IT decoding) can be derived in terms of extrinsic erasure probability at the output of the decoder (p_E) as a function of the a priori erasure probability (input of the decoder, p_A). For the block length n going to infinity, the EXIT curve is a function of the variable node degree distribution λ(x) and the check node degree distribution ρ(x), and can be obtained in parametric form as

$$p_A = \frac{x}{\lambda(1-\rho(1-x))}$$

$$p_E = \Lambda(1-\rho(1-x))$$

  with $x \in [x_{IT}, 1]$, being x_{IT} the value of x for which $p_A = \varepsilon_{IT}$ and

$$\Lambda(x) = \sum_i \Lambda_i x^i$$

  being Λ_i the fraction of variable nodes with degree i.

  EXIT functions of regular LDPC ensembles are displayed in FIGURE 2 (dashed lines).

- Due to the Area Theorem [18], the area below the EXIT function of the code, under ML decoding, must equal the code rate R. Note that the EXIT function defined above is IT-decoder-based. Hence, the area below the EXIT curve might be larger than the code rate.
- Consider the extrinsic erasure probability at the output of an ML and of an IT decoder. Obviously, $p_E^{ML} \le p_E^{IT}$.
- Therefore, by drawing a vertical line on the EXIT function plot of the ensemble, in correspondence with $p_A = p^*_A$, such that

$$\int_{p_A^*}^{1} p_E(p_A)\,dp_A = R,$$

we obtain an upper bound on the ML threshold, i.e. $\varepsilon_{ML} \le p_A^*$. For ($d_v$,$d_c$) regular LDPC ensembles, see the example in FIGURE 2. Here, d_v (d_c) specifies the constant variable (check) node degree.

In [17] it was shown that the upper bound for the ML threshold is extremely tight for regular LDPC ensembles, and for many irregular ensembles. Extensions to the above-mentioned techniques can be applied to other code ensembles, once the IT EXIT curve is provided. A rule of thumb for the design of capacity-approaching LDPC codes under ML consists in the selection of sufficiently dense parity-check matrices, by keeping for instance a relatively large average check node degree. To given an idea, we found that for rate 1/2 LDPC ensembles, an average check node degree $d_c \ge 9$ is sufficient to provide ML thresholds close to the Shannon limit. This heuristic rule seems to work for both regular and irregular ensembles.

From the discussion above, it would appear quite clear that, by adopting an ML decoder, a good design is given by a code with a high ML decoding threshold. Nevertheless, to keep the ML decoder complexity low, the iterative decoding threshold shall be high as well. In fact, in [11] it was shown that good iterative decoding thresholds are indeed highly desirable for ML decoding, since they allow having a low average number of reference variables. Recall that the number of reference variables dominates the complexity of the ML decoder, since it determines the size over which the $O(n^3)$ brute-force Gaussian elimination is performed. In general ensembles with high iterative decoding thresholds require an average lower number of reference variables than ensembles with low iterative decoding thresholds.

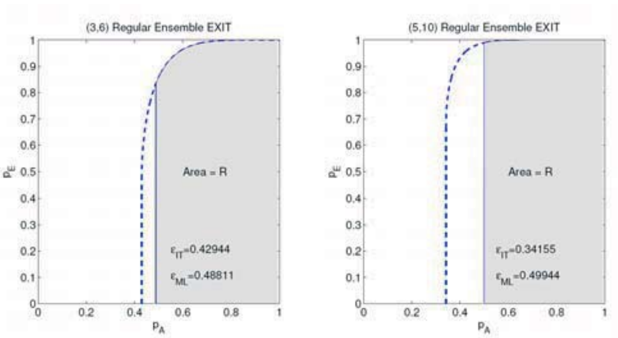

FIGURE 2. EXIT functions for the (d_v =3, d_c=6) and (d_v=5, d_c=10) regular LDPC ensembles. Dashed lines represent the IT decoder EXIT function. Solid lines are placed in correspondence of the ML thresholds upper bounds.

4. RESULTS

The results that are presented hereafter were obtained by numerical simulations and field tests. First, Monte Carlo simulations on an uncorrelated PEC channel were run collecting 50 errors per point (cf. FIGURE 3). A standard (n=2048,k=1024) irregular repeat accumulate (IRA) LDPC code was used. The non-constant degree distributions were obtained by differential evolution. Second, field trials were performed to evaluate the decoder performance on real communication channels. As measure for the service availability the so-called ESR5(20) criterion has been used [13]. It basically marks a 20 seconds time window as erroneous if more than 5 percent of the seconds in this window contain erroneous samples. The outcomes for IT, hybrid IT/ML and ideal code can be seen in TABLE 1.

It is worth to remark that the simulated curve (codeword error rate vs. erasure probability) in FIGURE 3 is very close to the theoretical lower bound applying the hybrid decoding algorithm. By contrast there is a non-significant gap when only applying plain IT decoding. Further enhancements on the code performance can be achieved by modifying the code structure. Generalized IRA LDPC codes turned out to approach the Singleton bound even closer [11]. Remarkably, the field trials results confirm the almost-ideal erasure correcting capabilities of the proposed scheme, which achieved the performance of an ideal maximum distance separable (MDS) code in all the trial sessions listed in TABLE 1. Large gains in service availability with respect to conventional IT-based decoding schemes can be observed.

In order to asses the decoder speed several simulations were run on a commercial Windows platform. FIGURE 4 shows the net data rate at the output of the IT and ML decoder versus the overhead (OH). The overhead is defined as the number of additional symbols, with respect to an ideal code, that are received at the decoder in order to solve the set of equations. As it can be seen, for very high erasure rates/low overheads both decoders are fast, since none of them is able to correct the erasures. Then, ML decoding starts working, approximately at an erasure rate of 0.495 (OH ~10) which slows down the decoding speed. Decreasing the erasure rate improves the speed for ML decoding, since the IT part is able to recover more and more erasures, and thus Gaussian elimination is performed less times. This can be verified by checking the IT decoding speed: it constantly decreases till an erasure rate of 0.44 (OH ~125), since the number of erased symbols that are restored increases. Lowering the erasure rate further improves the speed, since the number of unknown symbols decreases. Both decoders show same speeds here.

5. SUMMARY

By focusing on the improvement of a ML-based decoder design, high performance gains can be observed compared with conventional LDPC IT decoding schemes. It has been shown that ML decoding can be implemented in efficient ways over the erasure channel. A hybrid IT/ML scheme has been proposed in [11] which is able to guarantee high decoding speeds. Combined with an enhanced code design it permits to tightly approach the theoretical lower bound on the erasure channel. As a further matter of code design the decoder speed can be increased by reducing the number of reference variables (high IT decoding thresholds). Thus, high performance and fast decodable LDPC codes can be provided on the PEC for short, moderate, as well as for long block sizes.

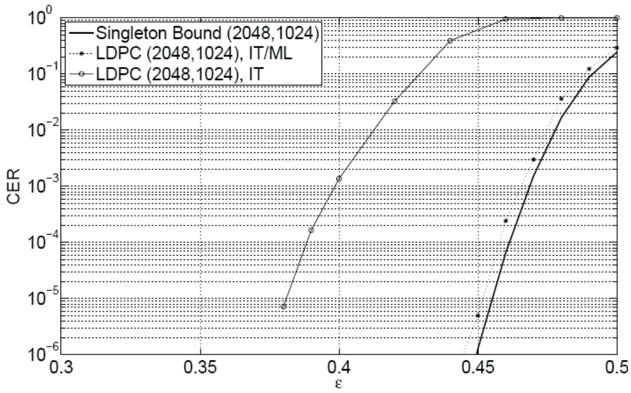

FIGURE 3. Performance curves of the IT and hybrid IT/ML decoder in comparison with the Singleton bound for a rate 0.5 (2048,1024) LDPC code.

Day	Take	IT ESR5(20)	IT/ML ESR5(20)	Opt. ESR5(20)
19. Apr	1	73,75	89,25	89,25
	2	63,29	78,93	78,93
20. Apr	1	78,05	88,23	88,23
	2	61,42	80,83	80,83
23. Apr	1	71,87	88,82	88,82
	2	56,31	71,84	71,84
24. Apr	1	72,63	92,07	92,07
	2	54,12	76,09	76,09

TABLE 1. Service availability for the IT and hybrid IT/ML decoding algorithm in terms of ESR5(20) and the theoretical bound; the measurement routes cover different environments.

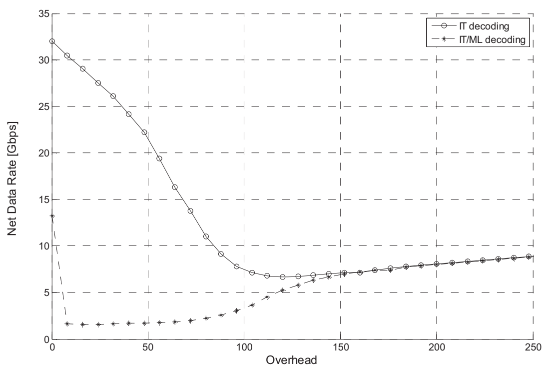

FIGURE 4. Decoding speed of a (2048,1024) IRA LDPC code for IT and hybrid IT/ML decoding.

[1] J. Byers, M. Luby, M. Mitzenmacher, and A.Rege, “A digital fountain approach to reliable distribution of bulk data,” in *Proc. of ACM SIGCOMM*, 1998.

[2] M. Luby, “LT codes,” in Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, Canada, Nov. 2002, pp. 271–282.

[3] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *Proc. 29th Symp. Theory Computing*, 1997, pp. 150–159.

[4] M. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[5] “Transmission System for Handheld Terminals (DVB-H),” Digital Video Broadcasting (DVB),” Blue Book, 2004.

[6] “Framing structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3GHz,” Digital Video Broadcasting (DVB),” Blue Book, 2007.

[7] “Multimedia broadcast/multicast service (mbms); protocols and codecs,” 3rd Generation Partnership Project (3GPP).

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

[9] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[10] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, “Finite-length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, 2002.

[11] E. Paolini, G. Liva, B. Matuz, and M. Chiani, “Generalized IRA erasure correcting codes for hybrid Iterative/Maximum Likelihood decoding,” *IEEE Communications Letters*, vol. 12, no. 6, pp. 450–452, June 2008.

[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] A. Heuberger, H. Stadali, B. Matuz, A. Del Bianco, R. De Gaudenzi, A. Bolea Alamanac, O. Smeyers, R. Hoppe, and O. Pulvirenti, “Experimental validation of advanced mobile broadcasting waveform in s-band.” ASMS, May 2008.

[14] D. Burshtein and G. Miller, “An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel,” *IEEE Transactions on Information Theory*, vol. 50, no. 11, nov 2004.

[15] “Digital Video Broadcasting (DVB); Trasnmission to Handheld Terminals (DVB-H); Validation Task Force Report,” ETSI TR 102 401 V1.1.1, Tech. Rep., 2005.

[16] E. Paolini, G. Liva, M. Balazs, and M. Chiani, “Generalized IRA codes for maximum-likelihood decoding,” submitted.

[17] Measson, C.; Montanari, A.; Richardson, T. and Urbanke, R. “Life Above Threshold: From List Decoding to Area Theorem and MSE” in Proc. IEEE Information Theory Workshop, October 2004.

[18] Ashikhmin, A.; Kramer, G. and ten Brink, S. “Extrinsic Information Transfer Functions: Model and Erasure Channel Properties”, IEEE Transactions on Information Theory, Volume 50, Pages 2657-2673. November 2004.

[19] X. Y. Hu, E. Eleftheriou and D.M. Arnold, “Progressive edge-growth Tanner graphs”, in *Proc. IEEE Globecomm*, San Antonio, Texas, Nov. 2001, pp. 995-1001.