Optimized IR-HARQ Schemes Based on Punctured LDPC Codes over the BEC

Iryna Andriyanova *Member, IEEE,* and Emina Soljanin *Senior Member, IEEE*

Abstract

We study incremental redundancy hybrid ARQ (IR-HARQ) schemes based on punctured, finite-length, LDPC codes. The transmission is assumed to take place over time varying binary erasure channels, such as mobile wireless channels at the applications layer. We analyze and optimize the throughput and delay performance of these IR-HARQ protocols under iterative, message-passing decoding. We derive bounds on the performance that are achievable by such schemes, and show that, with a simple extension, the iteratively decoded, punctured LDPC code based IR-HARQ protocol can be made rateless, and operating close to the general theoretical optimum for a wide range of channel erasure rates.

Index Terms

HARQ, incremental redundancy, throughput vs. delay tradeoff, LDPC codes, puncturing, BEC.

I. Andriyanova is with ENSEA/University of Cergy-Pontoise/CNRS, ETIS group, 95015 Cergy, France, e-mail: iryna.andriyanova@ensea.fr

E. Soljanin is with Math of Netw. & Comm. Research, Bell Labs, Alcatel-Lucent, Murray Hill, NJ 07974, e-mail: emina@alcatel-lucent.com

The paper was partly presented at International Symposium on Information Theory (ISIT'09) in Seoul and at Information Theory Workshop (ITW'09) in Taormina.

I. Introduction

In communications systems today, transmmission almost always takes place over a time varying channel, because of, for example, the channel's physical nature (e.g., wireless) or the length of a session (e.g., downloading a large file). Traditional channel coding schemes are ineadequate in these circumstances because they have fixed redundancy matching only a single channel condition. Similar problems arise in transmission to multiple users over non-varying but different channels. A number of recently proposed and/or implemented coding schemes address the time varying and multimuser communication scenarios, e.g., hybrid ARQ on the physical layer, rateless codes on the applications layer.

Hybrid ARQ transmission schemes combines the conventional ARQ with forward error correction. A scheme known as *incremental redundancy* hybrid ARQ (IR-HARQ) or *Type II* HARQ achieves higher throughput efficiency by adapting its error correcting code redundancy to varying channel conditions. Because of that, the scheme has been adopted by a number of standards for mobile phone networks. IR-HARQ is considered to be one of the most important features of the CDMA2000 1xEV-DO Release 0 and Revision A systems [1], [2]. A historic overview of HARQ schemes, up to 1998, can be found in [3]. For a survey of more recent developments, we direct the reader to [4] and references therein. In the third generation wireless standards, the IR-HARQ scheme resides in the physical layer and operates over time varying fading channels. The scheme is based on a turbo code dating back to the IS-95 standard. A possible replacement of this code by an LDPC or a fountain code was considered in [5].

Fountain codes are primarily designed to operate over erasure channels. They have superior performance in applications in which the channel variations are large and/or cannot be reliaably determined a priori. Because of this robustness, some classes of Fountain codes have been adopted into multiple standards, such as within the 3GPP MBMS standard for broadcast file delivery and streaming services, the DVB-H IPDC standard for delivering IP services over DVB networks, and DVB-IPTV for delivering commercial TV services over an IP network.

We here consider a hybrid ARQ scheme based on punctured LDPC codes over the BEC channel. The fundamental performances of the HARQ scheme are given by the achieved throughput and the delay between the beginning of the transmission and the moment when the useful information has been successfully decoded and is therefore available. Most often, one considers the average throughput and the average delay of a given retransmission scheme and one seeks an optimal tradeoff between these two quantities. To find such a tradeoff is one of the most relevant questions related to the HARQ setting. In this paper we investigate this issue in the particular case of IR-HARQ schemes based on finite-length, iteratively decoded LDPC codes and operating over time varying BECs. We proceed by following steps: (i) accurate approximation of the average throughout and delay, expressed through parameters of the used code ensemble and of the considered IR-HARQ scheme; (ii) optimization of the mentioned parameters subject to the optimal choice of both the throughput and the delay.

Note that the issue of determining the average throughput and the average delay has been intensively investigated, see for instance [5]–[7]. However, the obtained results only give bounds, obtained either under maximum-likelihood decoding assumption, either under (more practical) iterative decoding, but based on the bit error probability, which means that the bound is tight only for large code lengths. The approach taken in this paper is based on the *block* error performance under *iterative* decoding, which is the main novelty compared to existing results.

This paper is organized as follows: In Sec. II, we describe our IR-HARQ scheme and present expressions for its average throughput and delay. In Sec. III, we define finite-length rate-compatible LDPC codes, used further in the paper. Section III-C presents a model of the IR-HARQ scheme based on LDPC codes. In Sec. IV, we define the optimization problem to determine best code and protocol parameters. Section V presents a modification of the IR-HARQ scheme based on LDPC codes, enlarging its working region, and the comparison of the modified scheme with the HARQ scheme, based on LT codes. Finally, we then discuss our observations and future work in Sec VI.

II. Incremental Redundancy Hybrid ARQ Model

A. Multiple Transmissions Protocol and Channel Model

We analyze a particular retransmission protocol, called *Incremental Redundancy Hybrid ARQ (IR-HARQ)*, also known as Type-II HARQ. We adopt the following multiple transmission model of [8], [9]: at the transmitter, the user data bits are encoded by a low rate code, referred to as the *mother* code. Initially, only a selected number of encoded bits are transmitted, and the decoding is attempted at the receiving end. If the decoding fails, the transmitter, notified through the feedback, sends additional encoded bits, thus incrementing the redundancy. Besides the information about the success/failure of the transmission, the feedback may also carry the channel erasure rate information, to help the transmitter decide to which extent to increment the redundancy.

The new transmission may happen under different channel conditions. Decoding is again attempted at the receiving end, where the new bits are combined with those previously received. The procedure is repeated after each subsequent transmission request until all the parity bits of the mother code are transmitted. The channel is modeled as a time-varying BEC such that the channel erasure probability during the transmission of one block of encoded bits is constant and changes from one block transmission to another. We denote the channel erasure probability for block m as ϵ_m.

The main design parameters of the IR-HARQ scheme are [8], [9]: the maximum possible number M of transmissions for one block of user data and the fractions q_m, $m = \overline{1, M}$ of encoded bits assigned to transmission m. The maximum number of transmissions M is usually predefined by the protocol, while the fractions q_m's can be either predefined or calculated before each transmission, taking into account the feedback information about the previous channel erasure rates.

For further analysis of the IR-HARQ scheme, we adopt a probabilistic model in which q_m's are seen as probabilities, i.e., in which the transmitter assigns a bit to transmission m with probability q_m. Clearly, the transmitter has also the constraint (known as *rate compatible puncturing*) to assign to transmission m only the bits which have not been assigned to any of the previous transmissions. Even with this probabilistic model it is possible to make the scheme rate compatible as follows [8]:

`START`

`Before the IR HARQ protocol starts`

1) For each encoded bit, generate a number θ_v independently and uniformly at random over $[0, 1)$.
2) Determine M and q_1 (or all the q_m's if necessary)
3) Compute p_1 as $p_1 = 1 - q_1$.
4) For each node v s.t. $\theta_v \geq p_1$, assign bit corresponding to v to transmission 1.

`If transmission` $m-1$ `fails for` $2 \leq m < M-1$

1) Determine q_m (if not yet determined).
2) Compute p_m as $p_m = p_{m-1} - q_m$.
3) For each node v s.t. $p_m \leq \theta_v < p_{m-1}$, assign the bit corresponding to v to transmission m.

`If transmission` $M-1$ `fails`

transmit all remaining bits.

END

In Section IV we determine how q_m's are chosen. The criterion for such choice is to optimize the performance of the scheme, which are given by its throughput and delay.

B. The Performance Measures

Two standard measures of ARQ protocol efficiency are the *throughput* and the *delay*, defined as follows.

Definition 1: A throughput of a retransmission scheme is the number of user data bits accepted at the receiving end in the time required for transmission of a single bit.

Definition 2: A delay of a retransmission scheme is the number of bits needed to transmit in order to receive the useful information (user data bits).

In what follows, we are interested by the *average throughput* η and the *average delay* τ. We have the following lemma:

Lemma 2.1: Consider an IR-HARQ scheme with at most M transmissions and a set of fractions $q_1, \dots, q_M$. Let the underlying mother code be of length n and of rate R. Denote by ω_m the probability that it takes exactly m transmissions for the decoding to be successful. Then the average throughput and delay are determined by following expressions

$$\eta = \frac{R\sum_{m=1}^{M}\omega_m}{\sum_{m=1}^{M}\omega_m\Big(\sum_{j=1}^{m} q_j\Big)}; \tag{1}$$

$$\tau = \frac{n\sum_{m=1}^{M}\omega_m\Big(\sum_{j=1}^{m} q_j\Big)}{\sum_{m=1}^{M}\omega_m}. \tag{2}$$

Proof: The probability that one of the $m \le M$ transmissions is successful is $\sum_{m=1}^{M}\omega_m$. Because our protocol is limited to M transmissions, if none of these transmissions is successful, the throughput is equal to 0. When one of the $m \le M$ transmissions is successful, the number of user data bits communicated to the receiver is Rn. The number of the encoded bits sent to the receiver through the mth transmission is $n\sum_{j=1}^{m} q_j$. So, the average throughput η is given by (1). The calculation for τ is similar. ■

Note that the above expression for throughput becomes equal to its counterpart in [6] when $q_m = 1/M$. Moreover, the authors of [6] expressed the quantity ω_m in terms of the probability $P(m)$ that the asymptotic bit erasure rate P_b at the transmission m goes to 0, i.e., $P(m) \approx \text{Prob}[P_b^{(m)} \to 0]$. For LDPC codes, this probability has been computed with the help of density evolution. Clearly, $P(m)$ is a lower bound on the failure probability at transmission m, which thus gives an upper bound on η and a lower bound on τ. We next develop expressions of these bounds.

A randomly chosen code from an LDPC ensemble of length n has a successful iterative decoding with high probability when the channel erasure probability ϵ is smaller than $\epsilon^*_{(n)}$, where $\epsilon^*_{(n)}$ is known as a finite-length iterative decoding threshold. We will discuss $\epsilon^*_{(n)}$ in Section III.

Now we can state the following result:

Theorem 1: Consider an IR-HARQ scheme based on an LDPC code of rate R and having an iterative threshold $\epsilon^*_{(n)}$. The following bounds hold:

$$\eta \leq \begin{cases} R\frac{1-\epsilon}{1-\epsilon^*_{(n)}}, & 0 \leq \epsilon \leq \epsilon^*_{(n)}; \\ 0, & \text{otherwise}; \end{cases} \tag{3}$$

$$\tau \geq \begin{cases} n\frac{1-\epsilon^*_{(n)}}{1-\epsilon}, & 0 \leq \epsilon \leq \epsilon^*_{(n)}; \\ \infty, & \text{otherwise}. \end{cases} \tag{4}$$

Proof: Consider the limit case when $M = n$. Then the smallest number of received unerased bits, sufficient for successful decoding, is $1 - \epsilon^*_{(n)}$. The channel with some erasure probability $\epsilon < \epsilon^*_{(n)}$ passes $1 - \epsilon$ bits unerased. Hence, the smallest fraction γ of coded bits to be sent by the transmitter to receive $1 - \epsilon^*_{(n)}$ unerased bits at the receiver is

$$\gamma = \frac{1 - \epsilon^*_{(n)}}{1 - \epsilon}.$$

Note that $\eta \leq R/\gamma$, and (3) follows immediately.

Now consider the case when $M = 1$ and $n \to \infty$. At least γn bits should be sent to ensure the decoding success. Hence, $\tau \geq \gamma n$ and (4) follows. ■

Figure 1 illustrates the derived bounds. Unfortunately, the bounds were shown to be too optimistic for

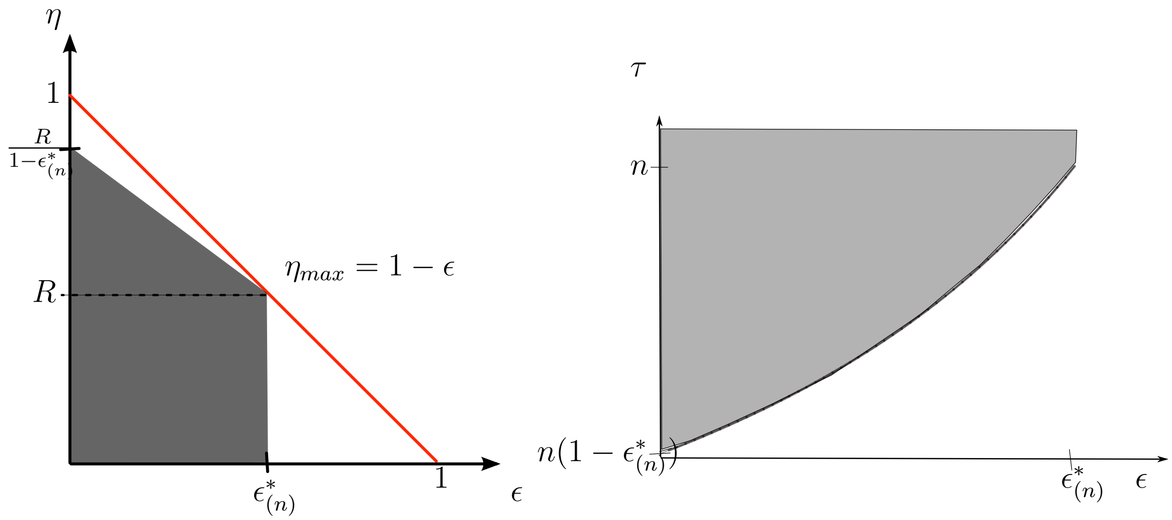

Fig. 1. Illustration of the upper bound on the region of attainable throughputs (on the left) and of the lower bound on the region of attainable delays (on the right) for an IR-HARQ scheme based on a mother LDPC code of code rate R and with the iterative threshold $\epsilon^*_{(n)}$. The red curve $\eta_{max} = 1 - \epsilon$ is the maximum attainable throughput.

finite-length codes for small and medium codelengths. In what follows, we are going to overcome this difficulty by proposing an accurate approximation for ω_m; it will be presented in Section III-C.

Remark 1: Note that the given expressions of η and τ implicitly assume that the feedback from the receiver to the transmitter is instantaneous. In practice the delay of the feedback transmission is positive, and we can introduce it in the above expressions as follows. Let the transmission time of one bit in the forward direction is t_{1bit}. As the feedback propagation delay, i.e. the time interval between two transmissions, is t, it is equivalent to time needed to transmit $n_{ACK} = t/t_{1bit}$ bits in the forward direction. It is easy to show that the expressions become

$$\eta = \frac{R\sum_{m=1}^{M}\omega_m}{\sum_{m=1}^{M}\omega_m\sum_{i,j}\lambda_i q_{ij}}, \tag{5}$$

$$\tau = n\frac{\sum_{m=1}^{M}\omega_m\sum_{i,j}\lambda_i q_{ij} + \frac{n_{ACK}}{n}\sum_{m=1}^{M} m\omega_m}{\sum_{m=1}^{M}\omega_m}, \tag{6}$$

where the term $\frac{n_{ACK}}{n}\sum_{m=1}^{M} m\omega_m$ is proportional to the average feedback transmission delay.

III. Performance of (Punctured) Finite-Length LDPC Codes over BEC

As we have seen above, the performance of the IR-HARQ scheme depends on the decoding performance after each transmission. In this paper we assume that the mother code is an LDPC code. We will see later on that the performance after each transmission in such case is related to the decoding performance of punctured mother code. First let us define the mother code and describe the puncturing technique.

A. Mother Code; Puncturing of the Mother Code

The mother code is assumed to be taken at random from an irregular LDPC code ensemble of length n, described by degree distributions $\lambda(x) = \sum_{i \in V} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i \in C} \rho_i x^{i-1}$. Given the parity matrix H of the code, λ_i (or ρ_i) represents the fraction of non-zero entries of H, located in columns (or rows) with exactly i non-zero entries. Also, the chosen LDPC code can be represented by its Tanner graph with variable and check nodes, where variable nodes correspond to columns of the matrix H, and check nodes corresponds to matrix rows. A variable node in the Tanner graph is connected to a check node iff the matrix H contains a non-zero entry in the intersection of the column and of the row, related to the variable node and to the check node. Hence, $\lambda_i n$ edges are connected to the variable nodes of degree i, $i \in V$, and $\rho_j n$ edges are connected to check nodes of j, $j \in C$.

The LDPC codes have a so called concentration property ([10]) - any given code takes at random from ensemble of (λ, ρ)-LDPC codes has the bit error probability close to the average bit error probability P_b of the ensemble. From this property, the concentration of the block error probability P_B for the region of channel parameters, where $P_B \propto P_b$, also follows. The concentration allows us to only consider the average performance of an LDPC ensemble instead of looking at the performance of one particular code, by using the ensemble analysis techniques, which are well developed in the coding theory.

Assuming the transmission over the BEC with erasure probability ϵ and n very large, the bit-erasure rate of given code vanishes for all ϵ up to some value ϵ^* and is non-zero for all $\epsilon \geq \epsilon^*$, with high probability. ϵ^* is the so called asymptotic iterative decoding threshold, and it is computed as follows

$$\epsilon^* = \min_{x \in (0,1)} \frac{x}{\lambda(1 - \rho(1 - x))}.$$

Clearly, ϵ^* is an upper bound on the channel erasure probability which can be supported by given LDPC code.

Puncturing is a technique to obtain a code of a higher rate from a given code of some rate R. It simply means non-transmitting (puncturing) a part of the encoded bits. The performance of the resulting code depends on the choice of the punctured bits. One way to make this choice is uniformly at random (e.g., by a coin toss of a certain bias for each variable node). This way of puncturing is often called just random puncturing. Another way to select the bits to puncture is to chose a degree of a punctured node according to certain (optimized) probability distribution, and then select a node to puncture uniformly at random from all nodes with the chosen degree. This way of puncturing is often referred to as intentional puncturing. It has been shown [11] that the intentional puncturing outperforms the random one, and, even more importantly, it can be designed to conserve the concentration property, whereas, the random one cannot. Therefore, in what follows we only consider the intentional puncturing.

A puncturing LDPC ensemble of some length n is described by three polynomials: degree distributions $\lambda(x)$ and $\rho(x)$ mentioned before and the puncturing degree distribution $p(x) = \sum_{i \in V} p_i x^{i-1}$, where p_i's are probabilities with which variable nodes of degree i are punctured. Note that, given p_i for some i, the fraction $p_i \lambda_i$ of edges of the bipartite graph is connected to the punctured variable nodes of degree i.

Notation 1: Let $\lambda_p(x) = \sum_{i \in V} p_i \lambda_i x^{i-1}$ and $\bar{\lambda}_p(x) = \sum_{i \in V} (1-p_i) \lambda_i x^{i-1}$.

It is easy to see that $\lambda(x) = \lambda_p(x) + \bar{\lambda}_p(x)$. Using this notation, the asymptotic iterative threshold of such punctured LDPC ensemble is

$$\epsilon^* = \min_{x \in (0,1)} \frac{x - \lambda_p(1-\rho(1-x))}{\bar{\lambda}_p(1-\rho(1-x))}, \tag{7}$$

and its design rate is given by

$$R_p = \frac{R}{1 - \frac{\sum_i p_i \lambda_i / i}{\sum_i \lambda_i / i}}, \tag{8}$$

where R is the code rate of the mother ensemble.

B. Finite-Length Performance

The finite-length of LDPC codes, punctured and unpunctured ones, is presented in this subsection. Before to proceed further, we present some useful notation.

Notation 2: Let us denote by $\Lambda(x) = \sum_{i \in V} \Lambda_i x^i$ the variable node degree distribution from the node perspective, where Λ_i represent the fraction of variable nodes of degree i, $i \in V$. Moreover, $\Lambda_i = i\lambda_i / \sum_j j\lambda_j$. Also, given the puncturing degree distribution $p_(x)$, let $\bar{\Lambda}_p(x) = \sum_{i \in V} (1-p_i) \Lambda_i x^{i-1}$.

Notation 3: Finally, we introduce the following notation:

$$y(x) = 1 - \rho(\bar{x}), \quad \pi(y) = \epsilon\bar{\lambda}_p(y) + \lambda_p(y),$$
$$\xi(x) = (\pi'(y))^2(\bar{y})(\rho'(1) - \rho'(\bar{x})), \qquad \mu(x) = \pi'(y)\rho'(\bar{x}),$$

where $\bar{x} = 1 - x$ and $\bar{y} = 1 - y$.

Conjecture 1: [12] Assume transmission took place over the BEC with erasure probability ϵ using a code chosen at random from a punctured LDPC ensemble with length n. Then, with high probability, its block erasure rate is given by the following expression

$$P_B = Q\left(\frac{\sqrt{n}(\epsilon^* - \epsilon - \beta n^{-2/3})}{\alpha}\right), \tag{9}$$

where α and β are respectively the scaling and the shift parameter, given by

$$\alpha_m = \sqrt{\frac{\xi(x^*)}{\Lambda'(1)}}\left(\frac{1}{\bar{\lambda}_p(y^*)} - \frac{2\bar{\lambda}'_p(y^*)\rho'(1-x^*)(1-\mu(x^*))}{\bar{\lambda}_p(y^*)^2 \cdot \mu'(x^*)}\right), \tag{10}$$
$$\beta_m = \left(\frac{b}{\bar{\Lambda}'_p(y^*)x^*\rho'(1-x^*) \cdot \sqrt{-\bar{\lambda}_p(y^*)\mu'(x^*)}}\right)^{2/3}, \tag{11}$$

where x^* satisfies (7), $y^* = y(x^*)$ and

$$b = x^*\rho'(1-x^*)\frac{\lambda'(y^*)}{\lambda(y^*)}\frac{y^* - x^*\rho'(1-x^*)}{y^*} + \frac{(x^*\rho'(1-x^*))^2}{\pi(y^*)}\left(\pi''(y^*) + \frac{\pi'(y^*)}{y^*} - \frac{\pi(y^*)'^2}{\pi(y^*)}\right) + \left(\frac{x^*(1-\epsilon^*)\rho'(\bar{x}^*)}{y^*}\right)^2 \cdot \frac{\sum_l lp_l(1-p_l)\lambda_{l+1}(y^*)^{l-1}}{\pi(y^*)}.$$

As we can see, α_m and β_m only depend on ϵ^*, x^* and y^* and on polynomials λ, ρ and p_m. Moreover, they give a very accurate approximation as it is shown in [12].

Example 1 (Particular case of regular codes): For LDPC codes with parameters $\lambda(x) = x^c$ and $\rho(x) = x^d$, we have that $p(x) = px^c$, where $0 \leq p_m \leq 1$. Moreover, its performance parameters become

$$\epsilon^* = \frac{\epsilon_0^*}{1-p}, \qquad \alpha = \frac{\alpha_0}{1-p}, \qquad \beta = \frac{\beta_0}{1-p}, \tag{12}$$

where ϵ_0^*, α_0 and β_0 are the parameters of the corresponding unpunctured ensemble.

Remark 2: For an LDPC code ensemble of length n, the finite-length iterative threshold $\epsilon^*_{(n)}$, already mentioned in Section II, is

$$\epsilon^*_{(n)} = \epsilon^* - \beta n^{-2/3}.$$

Note that, even for moderate lengths n, $\epsilon^*_{(n)}$ lies close to the asymptotic threshold ϵ^*.

C. Equivalent Puncturing Model of the IR-HARQ Scheme Based on LDPC Codes

Consider an IR-HARQ scheme described in Section II. Its mother code is an LDPC code chosen at random from the ensemble of given length n, with degree distributions $\lambda(x)$ and $\rho(x)$. Since it is irregular, the IR-HARQ scheme is now parametrized by the maximum number of transmissions M and the sequence of q_{ij}'s for $i \in V$ and $j = \overline{1, M}$, where q_{ij} denotes the probability with which a bit of degree i is chosen for transmission j.

Recall that in Section II, only one value q_j has been assigned to transmission j. However, if the bits of an irregular code were chosen to be transmitted with probability q_j regardless of their degree, it corresponds to the random puncturing scheme and the concentration property would be lost [12]. By introducing q_{ij}, $i \in V$, for transmission j, we obtain the intentional puncturing scheme and conserve the concentration of code performances around the average performance.

The IR-HARQ protocol can be described with the help of the following equivalent punctured code model: the bits that the transmitter chooses to sent up to the m-th transmission, can be equivalently seen as obtained by implementing a puncturing device which punctures a bit corresponding to a variable node of degree i with probability p_{im}, where $p_{im} = 1 - \sum_{j=1}^{m} q_{ij}$, or, as shown within the protocol described in Section II-A,

$$p_{i1} = 1 - q_{i1} \text{ and } p_{ij} = p_{i(j-1)} - q_{ij} \text{ for } 1 < j \leq M. \tag{13}$$

Further, assume that a transmission j takes place over the BEC with probability ϵ_j. When a bit corresponding to a variable node of degree i is assigned to one of the first m transmissions, it, equivalently, passes through the channel with erasure rate $(\sum_{j=1}^{m} q_{ij}\epsilon_j)/(\sum_{k=1}^{m} q_{ij})$. So, we can model our IR-HARQ protocol through transmission m as transmission of the punctured mother code over a BEC with erasure rate

$$\varepsilon_m = \sum_i \lambda_i \frac{\sum_{j=1}^{m} q_{ij}\epsilon_j}{\sum_{k=1}^{m} q_{ij}}, \tag{14}$$

where the considered bit is punctured with probability p_{im}.

The IR-HARQ protocol outlined below implements our model while conforming to rate compatible puncturing; it is based on the one in Section II-A.

The quantities q_{ij}'s being linked to p_{im}'s, now the IR-HARQ performance can be determined through the performance of punctured versions of the mother code. Let us determine the expected throughput and delay of our IR-HARQ scheme. Consider the expressions (1) and (2). First we switch to the irregular case by replacing q_m by $\sum_i \lambda_i q_{im}$. Then, we need to define how the ω_m's can be determined. Having defined the equivalent punctured model and the finite-length performance approximation of punctured LDPC codes, we define

$$\omega_m = P_B^{(m-1)} - P_B^{(m)}, \tag{15}$$

where $P_B^{(m)}$ is the finite-length average block erasure rate at transmission m, and it is given by (9).

To show that the proposed approximation of IR-HARQ performances, let us present here one figure from [9], that shows a good match of the approximation and the numerical results. In Figure 2 the average throughput of the IR-HARQ scheme with $M = 5$ and based on regular $(3, 6)$ LDPC codes of length 1024 is compared with its analytical approximation.

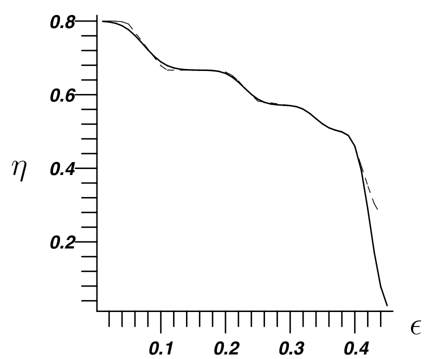

Fig. 2. Average throughput η versus equivalent channel erasure probability for $(3, 6)$ LDPC codes of length 1024. $M = 5$. Dashed line - numerical result, straight line - our estimation.

IV. Performance Optimization

Using the proposed puncturing representation, we aim to optimize the performance of the IR-HARQ transmission scheme based on LDPC codes, to be able to decide which bits and how many should be sent at each transmission. Note that, thanks to the concentration result for punctured LDPC codes, one has only choose the mother LDPC code and the puncturing degree distributions for each of transmissions, and one particular LDPC code ans particular puncturing patterns. The concentration of the punctured LDPC ensemble ensures that the performance of a particular punctured LDPC code, picked at random from the designed ensemble, is close to the average performance of this ensemble.

The parameters that we want to optimize are the average throughput η and the average delay τ. In previous sections, we have seen that η of finite-length schemes has a staircase behavior and thus can be optimized point-wise, i.e. knowing some particular working points on the ϵ-axe, one optimizes η to attain the maximum possible throughput for those points.

In the beginning, let us assume that the estimates of erasure probabilities $\epsilon_1, \dots, \epsilon_M$ are available at the transmitter. We also fix the acceptable block erasure probability $P_B^{(M)}$ after the maximum number of transmissions[1] and also the feedback propagation delay t.

In the following section we discuss the choice of other initial parameters, which should be fixed before the optimization, namely: a) the maximum number of transmissions M, b) the codelength n, c) fixed or maximum transmission block size K and d) the mother (λ, ρ) LDPC code ensemble. Further on, we investigate how to make the choice a puncturing degree distribution for each transmission m, $1 \leq m \leq M$, which leads us to design a rate-adaptable punctured LDPC ensemble, based on the initial (λ, ρ) ensemble and adapted to transmission conditions. Finally, in the last part we discuss how to get an estimate of erasure probabilities if they are not available at the transmitter.

A. Choice of Parameters (λ, ρ), n, M and K

The choice of parameters $\lambda(x)$, $\rho(x)$, n and M depends a lot from the considered application of the designed IR-HARQ scheme. In this section we only some insights on their choice.

The choice of degree distributions $\lambda(x)$ and $\rho(x)$ of the mother LDPC ensemble defines an upper bound on the region of attainable throughputs versus transmission erasure probability. The upper region is given by the iterative decoding threshold ϵ^* and the code rate R of the ensemble, as it was shown in Theorem 1. Note that the upper bound is achievable when $M \to n$ and n is quite large. Clearly, for practical schemes, i.e. for small values of M and finite n of order of several hundreds/thousands of bits, the average throughput is smaller. However, the degree distributions $\lambda(x)$ and $\rho(x)$ are chosen in such a way that the $\epsilon^* > \max(\epsilon_1, \dots, \epsilon_M)$ and the design rate R is sufficiently large, they can be good

[1] In practice, $P_B^{(M)}$ is dictated by the supported application, i.e. image or voice transmission, video streaming etc.

initial instances for the finite-length performance optimization. Note finally that, if the desired final block erasure probability $P_B^{(M)}$ is very low (e.g., 10^{-5} or lower, depending on the code), it induces additional constraints on the minimum distance of the code ensemble and, hence, on fractions λ's and ρ's. Concerning the constraints, see for example [13].

The choice of the codelength n is conditioned by the desired value of $P_B^{(M)}$, which should be attainable given ϵ and the chosen (λ, ρ)-pair. This can be verified by the finite-length analysis from [14].

The maximum number of retransmissions M should be chosen depending on a) the coherence time and b) the delay penalty. The coherence time T_C is the time, during which the channel conditions are the same, and it depends on the transmission environment. Note that in our model we assume the instantaneous erasure probability ϵ_m be constant during all the transmission m. Therefore, knowing d_{1bit}, we can transmit no more than $\frac{T_C}{d_{1bit}}$ bits at one transmission. From here we obtain that $M > \frac{d_{1bit} n}{T_C}$. As concerning the delay penalty, it is proportional to the total time of feedback transmissions, needed to spend in order to transit a packet of data. To keep the delay penalty low, one should choose M so that the time of one single transmission, proportional to $\frac{n}{M}$, is large compared to the feedback propagation delay t.

In practice, the number of sent bits during one transmission is usually a constant, dictated by the transmission protocol. However, some application may allow the variable length of the transmission block. Covering the both cases, we defined the constant transmission block of the first case or the maximum transmission block of the second case by K. The most often K is chosen to be $K = \frac{n}{M}$.

B. Optimization of Puncturing Degree Distributions When ϵ_m's Are Known

We consider the optimization of puncturing degree distributions when $\epsilon_1, \ldots, \epsilon_M$ are known at the transmitter.

C. Cost Function With the Penalty of Feedback

We start with defining a cost function, which needs to be optimized in order to increase the average throughput and to decrease the average delay. From (5) and (6), the average throughput and delay can be written as

$$\eta = \frac{R}{W_0}, \qquad \tau = nW,$$

where $W_0 = \frac{1}{n}\mathbb{E}[\#(\text{sent bits})|\text{ successful decoding, } n_{ACK} = 0]$ and $W = \frac{1}{n}\mathbb{E}[\#(\text{sent bits})|\text{ successful decoding, } n_{ACK}]$. Thus,

$$W_0 = \frac{\sum_{m=1}^{M} \omega_m \sum_{i,j} \lambda_i q_{ij}}{\sum_{m=1}^{M} \omega_m}$$

and

$$W = W_0 + \frac{\frac{n_{ACK}}{n} \sum_{m=1}^{M} m\omega_m}{\sum_{m=1}^{M} \omega_m}.$$

W is a natural choice of the cost function for our optimization problem, given that the minimum value of W gives the minimum τ, while η is also maximized. Using (15) and (13), W can be rewritten in terms of p_{ij}'s and $P_B^{(m)}$'s:

$$W = \frac{(1 - P_B^{(M)}) + \frac{n_{ACK}}{n} \sum_{m=0}^{M-1}(P_B^{(m)} - P_B^{(M)}) - \sum_i \lambda_i p_{i1} + \sum_i \lambda_i \sum_{m=1}^{M-1} P_B^{(m)}(p_{im} - p_{i(m+1)})}{1 - P_B^{(M)}}. \quad (16)$$

Let $\bar{p}_j = \sum_i \lambda_i p_{ij}$. Hence, finally,

$$W = \frac{(1 - P_B^{(M)}) + \frac{n_{ACK}}{n} \sum_{m=0}^{M-1}(P_B^{(m)} - P_B^{(M)}) - \bar{p}_1 + \sum_{m=1}^{M-1} P_B^{(m)}(\bar{p}_m - \bar{p}_{m+1})}{1 - P_B^{(M)}}, \quad (17)$$

with

$$P_B^{(m)} = \begin{cases} Q\left(\frac{\sqrt{n}(\epsilon_m^* - \varepsilon_m - \beta_m n^{-2/3})}{\alpha_m}\right), & \text{if } \varepsilon \le \epsilon_m^*; \\ 1, & \text{otherwise.} \end{cases}$$

Note that the expressions of ϵ^*, α and β have been already given in terms of p_{ij}'s. The average erasure probability ε_m at transmission m is also given by:

$$\varepsilon_m = \sum_i \lambda_i \frac{\sum_j (p_{i(j-1)} - p_{ij})\epsilon_m}{1 - p_{im}}. \quad (18)$$

D. Optimization of Puncturing Degree Distributions

The optimization problem thus reduces to the optimization of puncturing degree distributions $p_m(x) = \sum_{i\in V} p_{im} x^{i-1}$, $1 \le m \le M-1$, under the constraint of their rate-compatibility, i.e.

$$\boxed{\text{argmin}_{p_{ij}} W \quad \text{for } \forall i \in V \text{ and } \; 1 \le m \le M-1, \text{given } 1 \ge p_{i1} \ge p_{i2} \ge \ldots \ge p_{i(M-1)} \ge 0.}$$

In general, it is a non-linear optimization problem, given that $P_B^{(m)}$ depends on parameters ϵ_m^*, α_m and β_m, which themselves are dependent on p_{im}'s. We propose to use a gradient descent optimization to find a solution. This algorithm is described below.

`START`

`Initialization`

For m from 1 to $M-1$, find initial puncturing fractions $\tilde{p}_{im}$'s by assuming that the iterative threshold ϵ_m^*, given by (7), satisfies $\epsilon_m^* \geq \epsilon_m$. Moreover, p_{im}'s should satisfy the condition on K:

$$\sum_{i \in V} (\tilde{p}_{i(m-1)} - \tilde{p}_{im}) = \frac{K}{n} \text{ or } \sum_{i \in V} (\tilde{p}_{i(m-1)} - \tilde{p}_{im}) \leq \frac{K}{n} \tag{19}$$

for constant or variable transmission block size, respectively.

Choose the algorithm step Δ_{max}.

`Main part`

For m from 1 to $M-1$ **do** iterate until converged:

1) Compute $\tilde{W}_m$ given $p_{im} = \tilde{p}_{im}$, $\forall i$.
2) Find Δ_{im}'s minimizing

$$\Delta W = \sum_i \Delta_{im} \frac{\partial W}{\partial p_{im}} (\tilde{p}_{im}) \tag{20}$$

under the following conditions:

(Maximum changes) :	$\lvert\Delta_{im}\rvert \leq \Delta_{max}$
(Rate-compatibility) :	$0 \leq \tilde{p}_{im} + \Delta_{im} \leq \tilde{p}_{i(m-1)}, \;\; \forall i \in V$
(Number of sent bits) :	$\sum_{i \in V} (\tilde{p}_{i(m-1)} - \tilde{p}_{im} - \Delta_{im}) = \frac{K}{n}$
	or $\sum_{i \in V} (\tilde{p}_{i(m-1)} - \tilde{p}_{im} - \Delta_{im}) \leq \frac{K}{n}$,
	for constant and variable block size respectively

3) Set $\tilde{p}_{im} = \tilde{p}_{im} + \Delta_{im}$.

End of cycle over m

`Final part`

Set puncturing fractions to be equal to $\tilde{p}_{im}$, $\forall i, m$.

`END`

Below there are some details concerning the algorithm:

- *Initialization of p_{im}'s and choice of Δ_{max}*: The initial values of puncturing fractions are proposed to be set as if the used LDPC code were of infinite length. This is an optimistic choice of p_{im}'s as a finite-length code will behave worse than the infinite-length one with the same parameters. The fractions are to be found by linear programming, namely one chooses puncturing fractions to maximize the code rate of the punctured ensemble, under conditions (19). For more details on the optimization see for instance [11]. Note that, for small m and high values of ϵ, the solution may not exist. This means that the decoder will fail independently of chosen fractions. In this case any puncturing fractions can be chosen, assumed that they are rate-compatible with optimized puncturing fractions for for further transmissions.

  Such an optimized choice of p_{im}'s ensures a good convergence of the gradient descent because it already lies close to an optimal solution. Hence, the algorithm step Δ_{max} should be chosen quite small, close to $\frac{1}{n}$.
- *Minimization of (20)*: $\frac{\partial W}{\partial p_{im}}$ equals to

$$\frac{\partial W}{\partial p_{im}} = \begin{cases} -c\lambda_i \left(2 - P_B^{(1)}\right) + c\bar{p}_1 \frac{\partial P_B^{(1)}}{\partial p_{i1}}, & m = 1, \\ -c\lambda_i P_B^{(M-1)}, & m = M, \\ -c\lambda_i (P_B^{(m-1)} - P_B^{(m)}) + c\bar{p}_m \frac{\partial P_B^{(m)}}{\partial p_{im}}, & 1 < m < M, \end{cases} \tag{21}$$

where $c = (1 - P_B(M))^{-1}$ is a constant and

$$\frac{\partial P_B^{(m)}}{\partial p_{im}} = -\frac{\sqrt{n} \cdot exp\{\frac{n}{2}(\epsilon_m^* - \epsilon_m - \beta_m n^{-2/3})^2\}}{\sqrt{2\pi}\alpha_m^2} \left[\alpha_m \left(\frac{\partial \epsilon_m^*}{\partial p_{im}} - n^{-2/3}\frac{\partial \beta_m}{\partial p_{im}}\right) - \frac{\partial \alpha_m}{\partial p_{im}}(\epsilon_m^* - \epsilon_m - n^{-2/3}\beta_m)\right], \tag{22}$$

where $\frac{\partial \alpha_m}{\partial p_{im}}$ and $\frac{\partial \beta_m}{\partial p_{im}}$ can be easily found by taking the derivative of (10) and (11), and $\frac{\partial \epsilon_m^*}{\partial p_{im}}$ is obtained by implicit differentiation of the density evolution equation,

$$\frac{\partial \epsilon_m^*}{\partial p_{im}} = \frac{\lambda_i y_m^{i-1}(x_m - \lambda(x_m))}{\bar{\lambda}_p(x_m)^2}. \tag{23}$$

E. An Example of Optimization

Let us consider a particular example of the optimization of an LDPC ensemble for a particular value of the channel erasure probability. Initial parameters are: $n = 2000$, $M = 5$, $P_B(M) = 0.01$ and $K = n/M = 400$, where K is the constant size. We are going to optimize the throughput at target erasure probability $\epsilon_{target} = 0.35$, given the maximum channel erasure probability $\epsilon_{max} = 0.55$.

The following degree distributions have been chosen; $\lambda(x) = 0.220813x + 0.353686x^3 + 0.425502x^{12}$ and $\rho(x) = 0.390753x^4 + 0.361589x^5 + 0.247658x^9$. This gives rise to an LDPC ensemble with $\epsilon^* = 0.608$ and $P_B(M = 5, n = 2000, \epsilon_{max}) \approx 0.009$. The optimized puncturing degree distributions at the initialization stage are

$$\tilde{p}_4(x) = 0.6x$$

$$\tilde{p}_3(x) = 0.60264x + 0.123057x^2 + 0.474303x^{12}$$

$$\tilde{p}_2(x) = 0.735093x + 0.415371x^3 + 0.649536x^{12}$$

$$\tilde{p}_1(x) = 0.867547x + 0.707686x^2 + 0824768x^{12}$$

Note that for $i \leq 2$ the block erasure probability at erasure probability ϵ equals to 1, i.e., the the decoder will fail with a very high probability, no matter what the puncturing degree distributions are. $\tilde{p}_3(x)$ and $\tilde{p}_4(x)$, however, are the best choice for given initial parameters. Therefore, one needs to do at least 3 transmissions before to start decoding. Knowing this, we can to send three first coded packets one after another, without spending time waiting for the feedback.

For the initial-stage $p(x)$'s, the cost function $W = 0.677$. After the finite-length optimization, we have $W = 0.646$ with new $\tilde{p}_3(x)$ and $\tilde{p}_4(x)$:

$$\tilde{p}_4(x) = 0.1351x + 0.4649x^{12}, \qquad \tilde{p}_3(x) = 0.7351x + 0.4649x^{12}.$$

The average throughput, obtained by the proposed optimization, is shown in red in Figure 3. The throughput with puncturing degree distributions, obtained at the initialization stage, is shown in blue. Also, the green line represents the average throughput, obtained without any optimization, but simply after equal partitioning bits of each degree i ($i \in V$) between the transmissions. As we can, the throughput at $\epsilon = 0.35$ has been indeed improved.

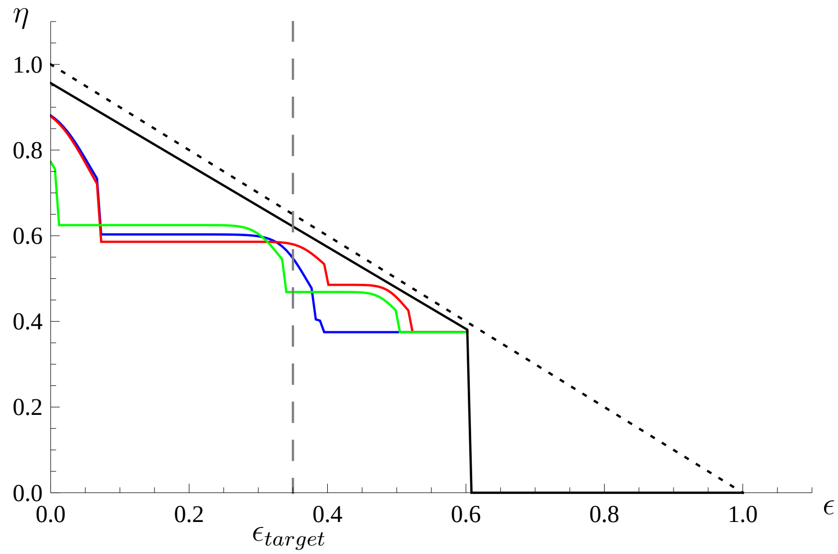

Fig. 3. Impact of the choice of $p_m(x)$'s on the average throughput. The dotted line represents the theoretical limit, the black line - the upper bound given the mother LDPC code, the green line - no optimization, the blue line - infinite-length optimization, the red line - finite-length optimization.

Note that the parameter K here operates as a regulator of the number of transmissions. If the number of sent packets per bit were unbounded, the optimizer would decide to send as many bits as to ensure the target P_B at ϵ, and if the first transmission were unsuccessful, would send the rest of bits at transmission 2.

F. Particular Case of Regular Codes

The particularity of regular LDPC codes is in the fact that their scaling and shift parameters do not depend on the puncturing fraction p_m. Indeed, based on Example 1, it is easy to see that, for punctured regular codes,

$$P_B^{(m)} = Q\left(\frac{\sqrt{n}(\epsilon^* - \epsilon(1-p_m) - \beta n^{-2/3})}{\alpha}\right), \tag{24}$$

where ϵ^*, α and β are parameters of the initial unpunctured regular ensemble. As $Q(x)$ is an increasing function, $P_B^{(m)}$ is a monotone increasing function in p_m and the cost function W takes its minimum at the smallest possible values of p_m, $1 \leq m < M$.

G. Estimation of ϵ_m's at the Transmitter

In general, the estimates $\epsilon_1, \ldots, \epsilon_M$ are not known at the transmitter and have to be found, before to perform the optimization of puncturing degree distributions. The quality of estimation depends on the knowledge of statistics of the transmission channel (mean, variance, probability distribution) and of the amount of feedback, obtained at the transmitter (1 bit representing ACK/NACK, previous channel erasure probability,...).

A lot of literature is available on the channel estimation. As examples, let us consider several case when of the channel estimation.

- *Known mean*: Let the mean $\bar{\epsilon}$ of the channel erasure probability is known at the transmitter. Then the puncturing degree distribution can be optimized as discussed before, assuming $\epsilon_m = \bar{\epsilon}$, $m = 1, \ldots, M-1$.
- *Known mean and previous ϵ's*: Let the mean $\bar{\epsilon}$ of the channel erasure probability is known and the receiver transmits to the transmitter the erasure probabilities of previous transmissions $\epsilon_1, \ldots, \epsilon_{m-1}$. In this case one can optimize the puncturing degree distributions in the online regime, i.e. just before to transmit. At transmission 1, the transmitter sends the fraction of coded bits, optimized for $\epsilon_1 = \bar{\epsilon}$, as it does not have any feedback information. At transmission $m > 1$, however, the estimation becomes
$$\epsilon_m = m\bar{\epsilon} - \sum_{i=1}^{m-1} \epsilon_i$$
and is used in the optimization.
- *Known probability distribution and 1-bit feedback*: Assume that the probability density function $p(\epsilon)$ is known and it has the support $[\epsilon_{min}, \epsilon_{max}]$. Then, for each transmission m,
$$\epsilon_m = \text{argmax}_{\epsilon \in [\epsilon_{min}, \epsilon_{max}]} \Pr(\epsilon = \epsilon_m | ACK/NACK_1, \ldots, ACK/NACK_{m-1})$$
Also note that, to ensure a good performance of the protocol, one should choose $\lambda(x)$ and $\rho(x)$ in such a way that $\epsilon^* \geq \epsilon_{max}$.

V. Rateless Incremental Redundancy Protocols

A. *Punctured Schemes with Repetitions*

As it can be seen in Fig.1, the IR-HARQ protocols based on punctured codes achieve high throughput only over a limited region of channel erasure rates. When they are based on iterative decoding and a mother LDPC code with the threshold ϵ^*, this region is from 0 to ϵ^*. Naturally, to cover a larger region, one can choose a mother LDPC code with $\tilde{\epsilon}^* > \epsilon^*$. However, such a code may have a lower rate $\tilde{R} > R$, and moreover $\frac{\tilde{R}}{1-\tilde{\epsilon}^*}$ may be lower than $\frac{R}{1-\epsilon^*}$ resulting in a lower throughput in the region $\epsilon < \epsilon^*$ (see

Fig.1). Compare, for example, the rate $1/2$ regular $(3,6)$ code with $\epsilon^* = 0.4293$ to the the rate $2/5$ regular $(3,5)$ code with $\epsilon^* = 0.5176$.

To extend the region of the high throughput for a given mother code, we propose to augment the HARQ protocol as follows. If after the transmission of all the bits in the codeword the decoding still fails, we further increment redundancy simply by repeating the same codeword, in the same manner as before (the same q_{im}). Hence, each coded bit might be transmitted twice through channels with erasure probabilities $\epsilon^{(1)}$ and subsequently with $\epsilon^{(2)}$. At the receiver side, both received values of the bit are combined together. So, after two transmissions the bit is erased with probability $\epsilon^{(1)} \cdot \epsilon^{(2)}$. One can continue the transmission in this manner, making the scheme essentially rateless.

The proposed protocol is therefore the incremental redundancy protocol with repetition. Let us denote it by IR-Rep-HARQ. Although repetition is in general not optimal, note that it takes place only when the channel conditions are bad ($\epsilon > \epsilon^*$), when it actually is a good strategy to follow.

Note that in repetition stage, we can either retransmit the same blocks as the first stage, or determine new blocks, according to the optimized fractions $\{q_{im}\}$. This translates in generating new θ values in the protocol of Section II-A. We next find expressions for the average throughput and the average delay for these two cases.

1) *Repetition of the same blocks*

Assume the IR-Rep-HARQ protocol with the repetition of the same block during the second transmission of the codeword. Denote the channel erasure probabilities of the first transmission stage by $\epsilon_1^{(1)}, \ldots, \epsilon_M^{(1)}$ and the ones of the repetition stage - by $\epsilon_1^{(2)}, \ldots, \epsilon_M^{(2)}$. Then, similarly to (1)-(2), the average throughput η_{IR-Rep} and the average delay τ_{IR-Rep} are given by

$$\eta_{IR-Rep} = \frac{R\sum_{r=1}^{2}\sum_{m=1}^{M}\omega_m^{(r)}}{\sum_{r=1}^{2}\sum_{m=1}^{M}\omega_m^{(r)}\Big(\sum_{j=1}^{m}\bar{q}_j\Big)}; \tag{25}$$

$$\tau_{IR-Rep} = \frac{n\sum_{r=1}^{2}\sum_{m=1}^{M}\omega_m^{(r)}\Big(\sum_{j=1}^{m}\bar{q}_j\Big)}{\sum_{r=1}^{2}\sum_{m=1}^{M}\omega_m^{(r)}}; \tag{26}$$

where $\bar{q}_j = \sum_{i\in V} \lambda_i q_{ij}$ and $\omega_m^{(r)} = P_B(\varepsilon_{m-1}^{(r)}) - P_B(\varepsilon_m^{(r)})$, with

$$\varepsilon_m^{(r)} = \begin{cases} \frac{\sum_{j=1}^{m} \bar{q}_j \epsilon_j^{(1)}}{\sum_{k=1}^{m} \bar{q}_k}, & r = 1; \\ \frac{\sum_{j=1}^{m} \bar{q}_j \epsilon_j^{(1)} \epsilon_j^{(2)}}{\sum_{k=1}^{m} \bar{q}_k} + \frac{\sum_{j=m+1}^{M} \bar{q}_j \epsilon_j^{(1)}}{\sum_{k=m+1}^{M} \bar{q}_k}, & r = 2. \end{cases} \tag{27}$$

Or, equivalently,

$$\eta_{IR-Rep} = \eta_{r=1}(1 - P_B(\varepsilon_M^{(1)})) + \eta_{r=1} P_B(\varepsilon_M^{(1)}); \tag{28}$$

where $\eta_{r=1} = \eta$, given by (1) and $\eta_{r=2} = \frac{R\sum_{m=1}^{M} \omega_m^{(2)}}{\sum_{m=1}^{M} \omega_m^{(2)} \left(\sum_{j=1}^{m} \bar{q}_j\right)}$. Similarly,

$$\tau_{IR-Rep} = \tau(1 - P_B(\varepsilon_M^{(1)})) + \frac{2n\sum_{m=1}^{M} \omega_m^{(2)} \left(\sum_{j=1}^{m} \bar{q}_j\right)}{\sum_{m=1}^{M} \omega_m^{(2)}} P_B(\varepsilon_M^{(1)}). \tag{29}$$

2) *Repetition with different blocks*

Assume the IR-Rep-HARQ protocol with the repetition of the block m such that it is chosen at random from the available, non-repeated bits, according to fractions $\{q_{im}\}_{i\in V}$. Then the expressions for η_{IR-Rep} and τ_{IR-Rep} are the same in the previous case (see (25) and (26)), except that the equivalent erasure probability $\varepsilon_m^{(r)}$ is computed as

$$\varepsilon_m^{(r)} = \begin{cases} \frac{\sum_{j=1}^{m} \bar{q}_j \epsilon_j^{(1)}}{\sum_{k=1}^{m} \bar{q}_k}, & r = 1; \\ \sum_{j=1}^{m} \bar{q}_j \epsilon_j^{(2)} (\sum_{k=1}^{M} \bar{q}_k \epsilon_j^{(1)}) + \sum_{j=m+1}^{M} (1 - \bar{q}_j)(\sum_{k=1}^{M} \bar{q}_k \epsilon_j^{(1)}), & r = 2. \end{cases} \tag{30}$$

Let us develop bounds on η_{IR-Rep} and τ_{IR-Rep} for those two IR-Rep-HARQ schemes.

Theorem 2: The average throughput η_{IR-Rep} for IR-Rep-HARQ schemes with similar or different blocks is bounded by

$$\eta_{IR-Rep} \le \begin{cases} \frac{R(1-\epsilon)}{1-\epsilon_{(n)}^*}, & \epsilon \le \epsilon_{(n)}^*; \\ \frac{R}{1+\frac{\epsilon-\epsilon_{(n)}^*}{\epsilon-\epsilon^2}}, & \epsilon_{(n)}^* < \epsilon \le \sqrt{\epsilon_{(n)}^*}; \\ 0, & \sqrt{\epsilon_{(n)}^*} < \epsilon \le 1. \end{cases} \tag{31}$$

Proof: For $\epsilon \leq \epsilon^*_{(n)}$, the expression of η is already given by Theorem 1. Let us focus on the case $\epsilon^*_{(n)} < \epsilon \leq \sqrt{\epsilon^*_{(n)}}$. Similar to the reasoning, used to prove Theorem 1, let us consider the limit case of $M = n$. Then, at the repetition stage, some fraction γ of bits has been sent twice, which is equivalent to sending them to a BEC with erasure probability ϵ^2, and the rest - through a BEC with erasure probability ϵ. It is easy to see that

$$(1-\gamma)\epsilon + \gamma\epsilon^2 \leq \epsilon^*_{(n)}.$$

Hence

$$\gamma_{min} = \frac{\epsilon - \epsilon^*_{(n)}}{\epsilon - \epsilon^2}.$$

Now note that the total number of sent bits is $2\gamma n + (1-\gamma)n = (1+\gamma)n$. So,

$$\eta_{IR-Rep} = \frac{R}{1+\gamma} \leq \frac{R}{1 + \frac{\epsilon - \epsilon^*_{(n)}}{\epsilon - \epsilon^2}}.$$

■

Similarly, the lower bound on τ_{IR-Rep} can be proven:

Theorem 3: The average delay τ_{IR-Rep} for IR-Rep-HARQ schemes with similar or different blocks is bounded by

$$\tau_{IR-Rep} \geq \begin{cases} \frac{n(1-\epsilon^*_{(n)})}{1-\epsilon}, & \epsilon \leq \epsilon^*_{(n)}; \\ n\left(1 + \frac{\epsilon - \epsilon^*_{(n)}}{\epsilon - \epsilon^2}\right), & \epsilon^*_{(n)} < \epsilon \leq \sqrt{\epsilon^*_{(n)}}; \\ \infty, & \sqrt{\epsilon^*_{(n)}} < \epsilon \leq 1. \end{cases} \quad (32)$$

As an example, the upper bound on the throughput for the scheme, based on (x^2, x^5) LDPC codes, is shown. For simplicity of presentation, we assume a large codelength n and $\epsilon^*_{(n)} \approx \epsilon^*$. We can see that in the region of erasure probabilities from 0.43 to 0.63 the repetition of the same codeword gives an almost linear throughput behavior.

Note that the bounds can be extended to the case of IR-Rep-HARQ schemes with any value R repetitions. We state it without proof in the following corollary.

Corollary 1: Consider an IR-Rep-HARQ scheme, based on LDPC codes, with R repetitions. Then the

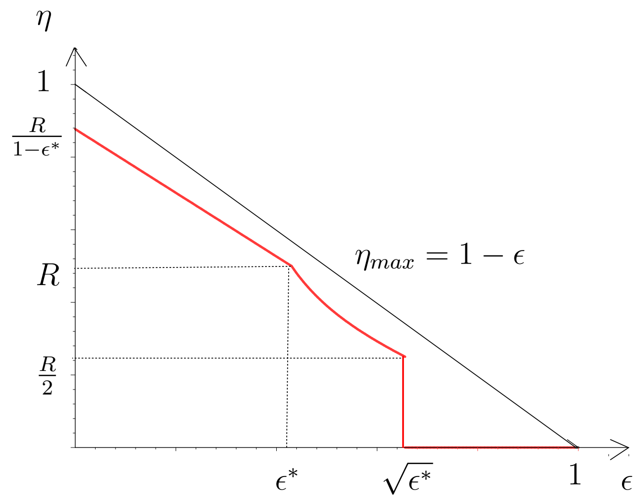

Fig. 4. Upper bound on the throughput for an IR-Rep-HARQ scheme based on (x^2, x^5) LDPC codes. The black curve is the maximum attainable throughput.

following bounds hold:

$$\eta_{IR-Rep} \leq \begin{cases} R\left(1+\frac{\epsilon^{r-1}-\epsilon^*_{(n)}}{\epsilon^{r-1}-\epsilon^r}\right)^{-1}, & \text{if } (\epsilon^*_{(n)})^{1/(r-1)} \leq \epsilon \leq (\epsilon^*_{(n)})^{1/r} \text{ for } r = 1, 2, \ldots, R; \\ 0, & \text{if } \epsilon \geq (\epsilon^*_{(n)})^{1/R}; \end{cases}$$

$$\tau_{IR-Rep} \geq \begin{cases} n\left(1+\frac{\epsilon^{r-1}-\epsilon^*_{(n)}}{\epsilon^{r-1}-\epsilon^r}\right), & \text{if } (\epsilon^*_{(n)})^{1/(r-1)} \leq \epsilon \leq (\epsilon^*_{(n)})^{1/r} \text{ for } r = 1, 2, \ldots, R; \\ \infty, & \text{if } \epsilon \geq (\epsilon^*_{(n)})^{1/R}. \end{cases}$$

B. Comparison with LT Codes

A natural thing is be to compare the performances of IR-HARQ schemes, based on punctured LDPC codes, with those based on rateless codes. To do this, let us consider, for instance, LT codes. As IR-HARQ based on LT codes do not have such a parameter as the maximum number of transmissions M, let us assume that $M = n$ for IR-HARQ-LDPC schemes, which leads us naturally to comparing the upper bounds on throughputs of two schemes.

Let there are K information bits to transmit. From Section 5 of [15], the upper bound on the throughput of LT-based HARQ schemes is then

$$\eta_{FC-HARQ} \leq \frac{1-\epsilon}{1+\frac{\log^2 K}{\sqrt{K}}}.$$

For LDPC-based schemes, we need to choose the code rate and the code ensemble. Let us take two code ensembles, already considered in the paper: regular (x^2, x^5) LDPC codes of rate $1/2$ and irregular LDPC codes of rate 0.37, optimized in Section IV-E.

Figure 5 presents the comparison. Three dashed curves present upper bounds for throughputs of LT-based HARQ schemes with $K = 500$, 5000 and 50000 from bottom to top. The straight lines are the upper bounds for throughputs of LDPC-based IR-HARQs with one repetition, the lower one corresponds to regular codes and the upper one - to irregular codes. Note that for all three values of K the bounds are very close to each other. Finally, the dotted line correspond to the maximum theoretical throughput. As we can see from the picture, in the region of small values of ϵ, the IR-HARQ-LDPC schemes have

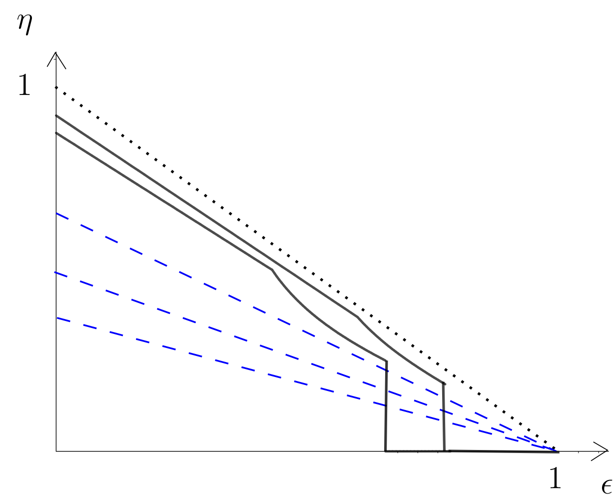

Fig. 5. Comparison of upper bounds on the throughput of 2 LDPC-based IR-Rep-HARQ schemes (two bold curves) and of LT-based HARQ codes with $K = 500; 5000; 50000$ (three dashed curves).

better throughput than LT-HARQ ones. Moreover, the throughput of IR-HARQ-LDPC schemes can be improved using repetition for $\epsilon > \epsilon^*$. In the region of very bad channels ($\epsilon \approx 1$) LT-based schemes offer a better throughput that LDPC-based ones.

VI. Discussion and Future Work

In this paper, we considered IR-HARQ schemes based on finite-length punctured LDPC codes, while the transmission was assumed to take place over the time-varying binary erasure channel, and investigated the tradeoff between the throughput and delay of such schemes, and also the optimization of those parameters. Our goal was achieved by using two approaches: 1) approximation of the block erasure performance of finite-length punctured LDPC codes and its use in computing the throughput and the delay; 2) computation of an upper/lower bound on the average throughput/delay. In the paper, we also proposed an optimization algorithm to improve the performances of an IR-HARQ protocol based on LDPC codes and defined another transmission protocol, called Incremental Redundancy HARQ with Repetition (IR-Rep-HARQ), which has a better throughput compared with the IR-HARQ scheme.

Three main *conclusions* can be made:

- We are able to define a cost optimization function which minimizes the delay and maximizes the throughput at the same time; the optimum of this function (if exists) is the best tradeoff between the throughput and the delay of the scheme. The proposed cost function can be optimized point-wise, i.e. for a set of target channel erasure probabilities.
- From the point of view of performance optimization, there is a principal difference between using regular an irregular LDPC codes in IR-HARQ schemes. The cost optimization function is monotone for regular LDPC codes, and hence the puncturing degree distribution for each of transmissions is trivial to calculate. For irregular LDPC codes this is not the case, and puncturing degree distributions should be thoughtfully optimized, in order to not to loose in the protocol performances.
- A repetition of the IR-HARQ round in the case, when the coded bits cannot be decoded correctly, improves the throughput and enlarges the region of working erasure probabilities of the protocol beyond the iterative threshold ϵ^* of the mother LDPC code. An IR-HARQ scheme with repetitions based on punctured codes outperforms the HARQ scheme based on LT codes in terms of throughput for quite a large region of channel erasure probabilities.

Also, the following *remarks* can be made:

- *Higher protocol layers :* Our approximation of P_B can also be combined with [16], to obtain an approximation of the failure probability of data packets at the application layer.
- *Other types of channels :* In principle, our results can be extended to other binary-input symmetric memoryless channels. In this case the estimation of approximation parameters, but still feasible. Otherwise they should be estimated numerically before the optimization part.
- *Flexibility of the optimization algorithm :* The optimization algorithm is quite general and can be easily adapted to many various cases, i.e. when the packet size is various or when the feedback is only sent from time to time and not obligatory after each transmission etc.
- *Tightness of the approximation :* Note that the average throughput and the average delay, obtained by our approximation for given $p_m(x)$, $1 \leq m \leq M$ is tight, owing to the tightness of the P_B's

approximation. Of course, this approximation is valid only for the so called waterfall region of the performance curve. However, as all the practical HARQ schemes work at high target block erasure probabilities, i.e. exactly on the waterfall region, the approximation is very well suitable for retransmission protocols.

Finally, the following **future work** is of interest:

- *Accurate channel state prediction* : It would be of interest to consider different cases of the channel state information (CSI) and to consider the IR-HARQ protocol performances in each of cases.
- *Extension to other punctured codes* : The optimization is not limited to LDPC codes. It can be also extended to all code ensembles, for which we are able to write a finite-length approximation (there exists, for instance, the analysis for turbo-like codes). The further comparison of different code ensembles could give an interesting insight on the design of codes for particular retransmission protocols.
- *Finding the expressions of throughput and delay at higher network layers* : Given work can serve as a base to optimize network parameters at higher network layers, i.e. transport or application layer. Therefore, one can develop further the existing approximations to find expressions for some important network parameters.

REFERENCES

[1] Q. Wu and E. Esteves, "The cdma2000 high rate packet data system," in *Advances in 3G Enhanced Technologies for Wireless Communications*, J. Wang and T.-S. Ng, Eds. Artech House, 2002, pp. 149–226.

[2] N. Bhushan, C. Lott, P. Black, R. Attar, Y.-C. Jou, M. Fan, D. Ghosh, and J. Au, "CDMA2000 1xEV-DO revision A: a physical layer and mac layer overview," *IEEE Communications Magazine*, vol. 44, no. 2, pp. 37–49, Feb. 2004.

[3] D. J. Costello, J. H. Jr., H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. on Inform. Theory*, vol. 44, pp. 2531–2560, Oct. 1998.

[4] C. Lott, O. Milenkovic, and E. Soljanin, "Hybrid ARQ: theory, state of the art and future directions," in *Proc. 2007 IEEE Int. Workshop Inform. Theory (ITW'07)*, Bergen, Norway, July 2007.

[5] E. Soljanin, N. Varnica, and P. Whiting, "Incremental redundancy hybrid ARQ with LDPC and Raptor codes," http://ect.bell-labs.com/who/emina/, 2005, submitted to IEEE Trans. on Inform. Theory.

[6] S. Sesia, G. Caire, and G. Vivier, "Incremental redundancy hybrid ARQ schemes based on Low-Density Parity-Check codes," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1311–1320, August 2004.

[7] M. Levorato and M. Zorzi, "Performance analysis of type ii hybrid ARQ with Low-Density Parity-Check codes," in *ISCCSP'06*, Malta, March 2008, pp. 804–809.

[8] E. Soljanin, R. Liu, and P. Spasojević, "Hybrid ARQ with random transmission assignments," in *Advances in Network Information Theory*, P. Gupta, G. Kramer, and A. Wijngarden, Eds. DIMACS Series in Discrete Mathematics and Thepretical Computer Science, American Mathematical Society, Aug. 2004, vol. 66, pp. 321–334.

[9] I. Andriyanova and E. Soljanin, "IR-HARQ schemes with finite-length punctured LDPC codes over the BEC," in *ITW'09*, Taormina, Sicilia, October 2009.

[10] T. Richardson and R. Urbanke, "Modern coding theory," Cambridge University Press, 2008.

[11] J. Ha, J. Kim, and S. McLaughlin, "Rate-compatible puncturing of Low-Density Parity-Check codes," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2824–2836, November 2004.

[12] I. Andriyanova and R. Urbanke, "Waterfall region performance of punctured LDPC codes over the BEC," in *IEEE Int. Symp. Inform. Theory (ISIT'09)*, Seoul, South Korea, June 2009.

[13] C. Di, T. Richardson, and R. Urbanke, "Weight distribution of low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 52, no. 11, pp. 4839–4855, November 2006.

[14] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Inform. Theory*, vol. 55, no. 2, pp. 473–498, February 2009.

[15] A. Shokrollahi, "Raptor codes," *IEEE Tran. Information Theory*, vol. 52, pp. 2551–2567, 2006.

[16] A. L. Duc, C. L. Martret, and P. Ciblat, "Packet error rate and efficiency closed-form expressions for cross-layer hybrid arq schemes," in *SPAWC'09*, Perugia, Italy, June 2009.

ACKNOWLEDGMENT

A significant part of this work was done in 2008 at Ecole Polytechnique Fédérale de Lausanne (EPFL). The authors would like to thank their EPFL hosts, professors A. Shokrollahi and R. Urbanke for their support.