Multi-Relay Cooperative NB-LDPC Coding with Non-Binary Repetition Codes

David Declercq
ETIS ENSEA/UCP/CNRS UMR 8051
95000 Cergy-Pontoise, France
email: declercq@ensea.fr

Valentin Savin
CEA-LETI, MINATEC
38054 Grenoble, France
email: valentin.savin@cea.fr

Stephan Pfletschinger
Centre Tecnolgic de Telecom. de Catalunya
7, Av. Carl Friedrich Gauss, 08860 Castelldefels, Spain
email: stephan.pfletschinger@cttc.es

***Abstract*—In this paper, we propose a system based on non-binary Low-Density Parity-Check (LDPC) codes to communicate efficiently over the multiple-relay fading channels, with a simple joint decoding strategy at the receiver end. The particularity of our approach is to rely on non-binary LDPC codes at the source, coupled with multiplicative non-binary local codes at the relays, such that the joint decoding complexity is not increased compared to a system without relays, while preserving the coding gain brought by the re-encoding of the sequence at the relays. We show by simulations on simple configurations that this cooperative scheme is superior to other techniques proposed in the literature, and close to the Gaussian relay channel capacity, even at moderate codeword lengths.**

***Index Terms*—Non-binary LDPC; Coopertive coding; Relay networks.**

I. Introduction

In wireless communication systems, the spatial diversity brought by the existence of relays, which can broadcast modified re-encoded versions of the source streams, helps to improve greatly the global information throughput and its error rates. Those improvements are impacted by the use of *cooperative diversity* [1], [2], which has been proposed in the literature for wireless relay channels and their multi-terminal extensions. A relay channel is a multi-terminal network consisting of a source, a destination, and a collection of relays which could be of different nature, as depicted on Figure 1 for the case of two relays. The communication system acts as follows: the source broadcasts a message to both relays and destination, while the relays forward the message or modified versions of it to the destination. Subsequently, different authors have proposed cooperation protocols for the relay channel, which can be classified into three major categories, namely the amplify-and-forward (AF) relays, the compress-and-forward (CF) relays and finally the decode-and-forward (DF) relays [3]. In AF protocols, the relays simply amplify the received signals and forward them to the destination, while in CF protocols the received noisy signals at the relays are quantized and forwarded. The DF protocol allows each relay to decode the received signal, re-encode it, and forward it to the destination. The forwarded message can either be identical to, or part of the initial transmission (repetition coding), or it can be obtained by using a dedicated coding scheme at the relays (cooperative coding). In the repetition coding case the destination combines received signals from both source and relays, which results in an improved signal-to-noise ratio (SNR) on the received transmission. In the cooperative coding case, the receiver at the destination uses the global knowledge of the cooperative coding (namely all code structures corresponding to the source and the relays), to jointly decode the received signals from both source and relays.

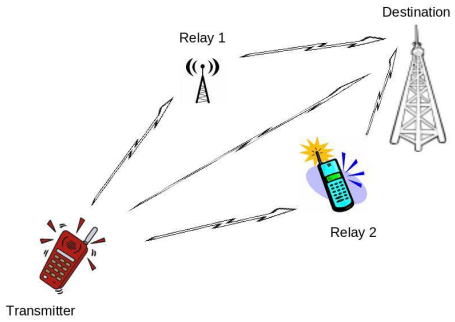

Fig. 1. Example of wireless relay channel with two relays.

In the simple case of repetition coding, there is no extra coding gain brought by the relay transmissions since the protocol does not change the Forward Error Correcting (FEC) code, and impacts only on the SNR improvement. On the other hand the receiver at the destination has the same low complexity as if no relays where used. In the case of distribution coding however, a proper design of the system aims at maximizing the coding gain brought by the relays to get closer to the relay channel capacity. This comes however at the cost of an extra decoding complexity at the receiver end, while joint decoding of the source and the relays are necessary to take advantage of the cooperative coding. In this paper, we propose a scheme which aims at having both advantages, namely an extra coding gain at no extra decoding cost. Distributed coding using parallel turbo-codes [4] or binary LDPC codes [5], [6], [7], [8], [9], has already been proposed in the literature. The existing approaches are either based on serial or parallel code concatenation, such that the graph of the LDPC code broadcasted from the source is a only subgraph of the destination decoding graph, or based on punctured rate-compatible LDPC codes. In a recent publication a cooperative LDPC code design which uses a turbo-like decoder at the receiver to jointly

 ISBN: 978-1-61208-183-0

decode the different sub-graphs of the source and the relays is proposed [10]. All these methods suffer from the large increase of decoding complexity at the receiver and the fact that they are not robust when the number of relays is larger than one, *i.e.* the coding gain using several relays is less and less important when the number of relays increases.

In this paper, we propose a new approach to the problem of distributed coding for cooperation in the case of multiple relays. The approach is based on non-binary LDPC (NB-LDPC) codes and the recently introduced technique of multiplicative non-binary coding [11], [12], which will be referred to as non-binary repetition coding. In our setting, the source transmits a codeword issued from a NB-LDPC code to the destination and the relays. When the relays successfully decode the received codeword, extra parity symbols are computed at the relays through optimized non-binary repetition codes, and are broadcasted to the destination. The receiver then collects the original received codeword from the source and the non-binary extra symbols from the relays and combines them before the iterative decoding. The iterative decoding complexity is the same in the presence or the absence of relays, while the combining of the codeword and the additional non-binary repetition symbols from the relays brings an effective coding gain. The paper is organized as follows. In Section II, we recall the basics about NB-LDPC codes and decoders and we present the concept of non-binary repetition coding coupled with NB-LDPC codes. In Section III, we describe our proposed cooperative system and discuss its advantages. We also propose in Section IV an optimization for the design of NB-LDPC codes and non-binary repetition codes for maximizing the coding gain, and finally, we present some simulation results on simple relay channels in Section V.

II. Non-Binary LDPC Codes and Decoding

A. Non-Binary LDPC codes

A Low Density Parity Check (LDPC) code is defined by a very sparse random parity check matrix H, which consists of $N-K$ rows and N columns, where K is the information block length and N is the codeword length; the code rate is defined by $R \leq \frac{K}{N}$. LDPC codes are nowadays used and proposed for a large number of communication and storage applications and standards, as their performance under low complexity iterative decoding approach the capacity for a large variety of channels. Binary LDPC codes can be generalized to non binary LDPC codes (NB-LDPC). The parity-check equations are written using symbols in a Galois field of order q, denoted GF(q), where $q = 2$ is the particular binary case. Throughout the paper, the Galois field elements will be denoted $\left\{0, \alpha^0, \alpha^1, \ldots, \alpha^{(q-2)}\right\}$, where α is a primitive element of the Galois field. The parity check matrix defining a NB-LDPC code has only a few nonzero coordinates h_{ij} which belong to GF(q), and a single parity equation involving d_c codeword symbols follows:

$$\sum_{j=1}^{d_c} h_{i,j}.c_j = 0 \qquad (1)$$

where $\{h_{i,j}\}$ are the nonzero values of the $i-th$ row of H, and $\mathbf{c} = \{c_1, \ldots, c_N\}$ is the notation used for the NB-LDPC codeword.

NB-LDPC codes are usually preferred to their binary counterparts when the blocklength is small to moderate [13], [14], or when the order of the symbols sent through channel are not binary [15], which is the case for high-order modulation (M-QAM) or for Multiple-antennas channels [16]. As a matter of fact, when the LDPC code is build in a field with order q equal or higher than the modulation order M, the non-binary LDPC decoder is initialized with uncorrelated vector messages, which helps the decoder to be closer to Maximum Likelihood Decoding than in the binary case. Recently, another advantage of NB-LDPC codes has been identified [17], [12]. The authors have shown in these papers that one can design flexible coding transmission in a very simple, though efficient way. The proposed approach is to concatenate non-binary multiplicative codes to a mother NB-LDPC, which leads to extra redundancy built from non-binary repetition symbols. When the repetition coding is properly designed, it results that the coding gain is greatly increased compared to binary repetition coding, expecially when the field order is sufficiently large GF(q), with $q \geq 64$. In this paper, we make use of the concatenation of NB-LDPC codes and non-binary repetition codes to design our distributed coding scheme, as presented in Section III.

B. Brief presentation of NB-LDPC decoders

The performance improvement of NB-LDPC codes is achieved at the expense of increased decoding complexity. As in all practical coding schemes, an important feature is the complexity/performance tradeoff, it is very important to try to reduce the decoding complexity of NB-LDPC codes, especially for high order fields GF(q) with $q \geq 64$. The base decoder of NB-LDPC codes is the Belief-Propagation (BP) decoder over the Tanner graph representation of the code [18]. The Tanner graph of an NB-LDPC code is drawn on Figure 2. The nonzero values of the parity-check matrix are put as *labels* for the edges connected to the non-binary parity check nodes. In this figure, we have represented all four parity-check nodes with the same labels $\{h_1, h_2, h_3, h_4, h_5\}$, and the information symbols are represented in red (left side of the codeword) while the redundancy symbols are drawn in blue (right side of the codeword). The number of edges connected to the nodes is constant throughout the Tanner graph, and furthermore the number of edges for the symbol nodes is minimum, equal to $d_v = 2$. This Tanner graph corresponds to a *regular and ultra-sparse* NB-LDPC code, with code rate $R = 1 - \frac{d_v}{d_c} = \frac{3}{5}$.

The main difference with the binary BP decoder is that for GF(q) LDPC codes, the messages from variable nodes to check nodes and from check nodes to variable nodes are defined by q probability weights, or $q-1$ log-density-ratios. As a result, the complexity of NB-LDPC decoders scales as $\mathcal{O}(q^2)$ per check node [19], which prohibits the use of codes build in high order fields.

 ISBN: 978-1-61208-183-0

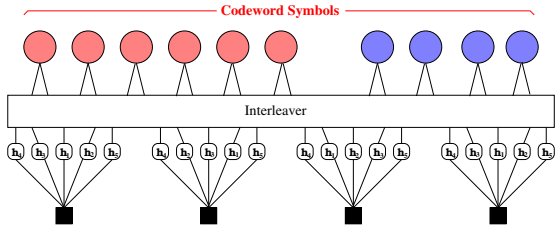

Fig. 2. Tanner graph of an ultra-sparse NB-LDPC code, with $(d_v, d_c) = (2, 5)$ and $R = 3/5$.

Sub-optimum decoders based on generalization of the min-sum decoder have been developed [19], [20], with the goal of reducing the decoding complexity at the check-node side. In particular, the EMS algorithm presented in [21] proposes the best complexity/performance tradeoff found in the literature, as the complexity scales as $\mathcal{O}(n_m.q)$ with $n_m << q$. We do not present with more details the EMS non-binary decoder in this paper and refer to the cited article for a complete description and analysis. Only the computation of the Log-Likelihood ratios (LLR) used for the initialization of the decoder are presented in the next section.

C. Computation of the LLR vectors

For transmission over a general wireless channel, the code symbols defined in GF(q) have to be mapped to $M-$QAM symbols, where M is a power of two, including $M \in \{2, 4\}$ for BPSK and QPSK. The information message $\mathbf{u} \in \mathrm{GF}(q)^K$ is encoded into a codeword $\mathbf{c} \in \mathrm{GF}(q)^N$, which is passed to the modulator and then transmitted over a continuous-valued fading channel. At the receiver, the soft demapper computes log-likelihood values (LLR-values), which constitute a sufficient statistic of the received signal $\mathbf{y}$ and form the initialization of the channel decoder.

To obtain a bijective mapping, we have to map m_1 code symbols to m_2 QAM symbols such that $q^{m_1} = M^{m_2}$. We denote the QAM alphabet by χ_M, and the mapping function by $\boldsymbol{\mu}$, i.e.

$$\boldsymbol{\mu} : \mathrm{GF}(q)^{m_1} \rightarrow \chi_M^{m_2} \tag{2}$$

The code symbols $\mathbf{b} = (b_1, b_2, \ldots, b_{m_1})$ belong to the same codeword $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ and are mapped to a vector of QAM symbols,

$$\mathbf{x} = [x_1, \ldots, x_{m_2}] = \boldsymbol{\mu}(\mathbf{b}) = [\mu_1(\mathbf{b}), \ldots, \mu_{m_2}(\mathbf{b})] \tag{3}$$

For binary codes (i.e. for q = 2), we always have $m_2 = 1$, while for codes in higher order Galois fields, for many modulations $m_1 = 1$, which is quite beneficial for the demapping, as we will see below. The soft demapper computes the LLR-vector $\mathbf{L}_i = [L_{i,0}, L_{i,1}, \ldots, L_{i,q-1}]^{\mathrm{T}}$, which corresponds to the code symbol b_i , and whose components are given by, for $i = 1, \ldots, m_1$ and $g \in \mathrm{GF}(q)$

$$L_{i,g} \triangleq \ln \frac{P[b_i = g|\mathbf{y}]}{P[b_i = 0|\mathbf{y}]}$$

where we identify, with a slight abuse of notation, the elements of GF(q) by their indices $g = 0, 1, \ldots, q-1$.

For a memoryless channel and assuming that all code symbols are equiprobable, we obtain

$$L_{i,g} = \ln \frac{\sum_{\mathbf{b} \in \mathcal{B}_i^g} \prod_{j=1}^{m_2} p\left(y_j|\mathbf{b}\right)}{\sum_{\mathbf{b} \in \mathcal{B}_i^0} \prod_{j=1}^{m_2} p\left(y_j|\mathbf{b}\right)}, \quad \begin{matrix} i = 1, \ldots, m_1 \\ g = 0, \ldots, q-1 \end{matrix} \tag{4}$$

where $\mathcal{B}_i^g \triangleq \{\mathbf{b} \in \mathrm{GF}(q)^{m_1} : b_i = g\}$ is the set of all code symbol vectors whose i-th component is fixed to g.

The mapping and in particular the demapping simplifies significantly for $m_1 = 1$, which means that exactly *one* code symbol $b \in \mathrm{GF}(q)$ is mapped to a vector of QAM symbols. In this case, we can drop the index i in (4), and since the sets $\mathcal{B}_i^g$ reduce to one element, we can write

$$L_g = \ln \frac{\prod_{j=1}^{m_2} p\left(y_j|b = g\right)}{\prod_{j=1}^{m_2} p\left(y_j|b = 0\right)} \tag{5}$$

For a flat fading channel given by $y_j = a_j \cdot x_j + w_j$ with $w_j \sim \mathcal{CN}(0, N_0)$, the conditional pdf is given by $p(y_j|b = g) = \frac{1}{\pi N_0} \exp\left(-\frac{|y_j - a_j \mu_j(g)|^2}{N_0}\right)$. With this, and noting that a BP decoder is typically insensitive to additive constants of the LLR vectors, we can further simplify (5) to

$$L_g = -\frac{1}{N_0} \sum_{j=1}^{m_2} |y_j - a_j \mu_j(g)|^2 + \ell_0 \tag{6}$$

where ℓ_0 is an arbitrary additive constant which does not depend on g.

As we can see from the LLR computations, using NB-LDPC codes with $m_1 = 1$ results in a significant complexity reduction of the demodulator (without any approximation) with respect to binary demappers, since no marginalization is required. This is the case if the number of bits per code symbol is a multiple of the number of bits per QAM symbol, i.e. $\log_2(q) = m_2 \cdot \log_2(M)$. For instance, codes in GF(64) allow a simple LLR computation for $\log_2(M) \in \{1, 2, 3, 6\}$, which corresponds to BPSK, QPSK, 8-QAM and 64-QAM. Note that more options are possible by mapping code symbols separately to the I or Q component, *i.e.* by considering real-valued PAM constellations. This does not incur any performance penalty, but allows e.g. to easily combine 16-QAM with a GF(64) code by mapping the 6 bits of one code symbol to three 4-PAM symbols.

III. Proposed Cooperative Transmission Scheme

A. Channel Model and System Description

Throughout the paper, we will assume that the source broadcasts a NB-LDPC codeword to the destination and a given number N_r of relays. All wireless channels in the system are either Rayleigh fading channels, or memoryless additive white Gaussian noise (AWGN) channels, depending on the type of relay (fixed or mobile). For sake of simplicity in the presentation, we restrict the model description to AWGN channels, but without loss of generality since the LLR computation presented in the preceding section does not change for AWGN or Rayleigh fading. The different channels will make use of the following notations:

 ISBN: 978-1-61208-183-0

- The link between source and destination uses $M_{SD}-QAM$ signalling with signal-to-noise ratio equals to γ_{SD},
- The link between source and the i-th relay uses $M_{SR_i}-QAM$ constellations with signal-to-noise ratio equals to γ_{SR_i},
- Finally, the link between the i-th relay and the destination uses $M_{R_iD}-QAM$ constellations with signal-to-noise ratio γ_{R_iD}.

Note that since relays and destination receive the same modulated signals, we have by construction $M_{SD}-QAM = M_{SR_i}-QAM$, although the SNRs could be different.

Now, let us discuss the channel and transmission protocol assumptions that we use in our work. First we assume that the direct link between the source and the destination is weak and that the relays links are stronger, both from source to relay and from relay to destination, which is a usual assumption in relay channels. It follows that $\gamma_{SR_i} \geq \gamma_{SD}$ and $\gamma_{R_iD} \geq \gamma_{SD}$, $\forall i$. The improved SNRs of the relay channels implies that either higher-order modulations would be used for the relay-to-destination channel, or higher rate cooperative coding. The optimization of the modulation order or of the cooperative coding rates requires that channel fading estimation and a link adaptation strategy is performed on the relay channel. We leave this issue for future research, and in this paper, we will assume fixed values for the modulation orders and coding rates, and measure the performance by the gap of error rates to the capacity of the relay channel.

Now, we must make an assumption regarding the non-propagation of errors through the relays. Since the relays are decode-and-forward nodes in the network, we can reasonably assume that the relay can detect if it decodes the received codeword from the source or not. We will then assume that when the relay fails to decode to a valid codeword, it does not transmit any information to the destination, which prevents unavoidable decoding failures at the destination.

Given this model and assumption, our proposed cooperative coding scheme can be described as follows:

- The source encodes the packet of information bits, generating a NB-LDPC codeword $\mathbf{c}$ of the parity check matrix H, with rate R. The source modulates $\mathbf{c}$ with the $M_{SD}-QAM$ constellation and broadcasts the modulated symbols $\mathbf{x}$ to both relays and destination.
- Each relay i decodes the received signal, correcting the transmission errors on $\mathbf{c}$. The relays then generate a new sequence $\mathbf{c}^{(i)}$ of non-binary symbols using the repetition coding, as depicted on Figure 3 in the case of 2 relays. Note that the size of the vectors $\mathbf{c}^{(i)}$ are not necessarily the same as the original codeword $\mathbf{c}$, since the coding rates for the links relay-destination are typically higher. The encoding procedure and the optimization of the repetition codes are presented in the next section. The vectors $\mathbf{c}^{(i)}$ of non-binary repetition symbols are then transmitted from the relays to the destination using $M_{R_iD}-QAM$ constellations.
- Thus, the destination receives noisy versions of $\mathbf{x}$ and $\mathbf{x}^{(i)}$ (from both the source and the relay), which can be jointly decoded using the only the matrix H, and the LLR computation presented in Section II-C.

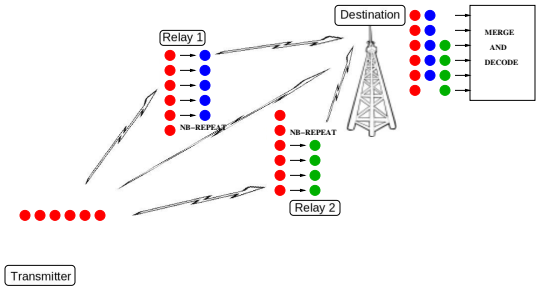

Fig. 3. Cooperative coding scheme using non-binary repetition coding.

The proposed distributed scheme relies mainly on the way the repetition symbols are generated and taken into account in the joint decoding at the destination. We explain in details in the next section why the non-binary repetition symbols bring a significant coding gain, at no extra encoding or decoding cost.

B. Non-Binary Repetition coding and Joint Decoding

As mentioned in the previous section, we assume in this paper that the parameters of the transmission system for each link are fixed, namely the constellation orders $\{M_{SD}, M_{R_iD}\}$ and the coding rates $\{R_{SD}, R_{R_iD}\}$ are fixed *a priori*. Now let us present how the non-binary repetition symbols are generated.

The $i-th$ relay is supposed to receive and decode correctly the broadcasted codeword $\mathbf{c}$ (otherwise, relay i does not transmit anything). From the N symbols in $\mathbf{c}$, the relay needs to build $N_i = N.\frac{R_{SD}}{R_{R_iD}}$ repetition symbols, which represents the coded block that relay i has to send to the destination. For example, if $R_{SD} = 1/2$ and $R_{R_iD} = 3/4$, $N_i = N.\frac{2}{3}$ repetition symbols have to be encoded at relay i. The repetition encoding is performed as follows for relay i:

- Select N_i non-binary symbols $\left\{c_{k_l^{(i)}}\right\}_{l=1...N_i}$ inside the codeword $\mathbf{c}$, the N_i symbols could be chosen arbitrarily, so either a random selection or a selection based on the knowledge of the transmitted NB-LDPC code are possible,
- For each selected symbol $c_{k_l^{(i)}} \in \mathrm{GF}(q)$, generate a new symbol $c_l^{(i)} = h_l^{(i)}.c_{k_l^{(i)}}$, with $h_l^{(i)} \in \mathrm{GF}(q)$ being the non-zero field value corresponding to the local repetition code. The vector $\mathbf{c}^{(i)}$ of size N_i is then sent from relay i to the destination.

We can easily see that this repetition encoding procedure is extremely simple, as it requires only N_i Galois field operations after a successful decoding at the relay. Note also that a sub-case of the proposed scheme corresponds to the particular choice of $h_l^{(i)} = 1, \forall l$, and which reduces to the usual decode-and-forward strategy, where the same codeword is sent both

 ISBN: 978-1-61208-183-0

from the source and the relays. In our case, with a very limited extra complexity, we allow the use of non-binary repetitions with $h_l^{(i)} \neq 1$, which provides a non-negligible coding gain, as explained in Section IV.

Now let us discuss how the collection of received symbols are jointly treated at the destination. For some particular code symbol $c \in \mathrm{GF}(q)$, we denote by $\mathbf{x}$ the QAM symbols build from c transmitted by the source and by $\mathbf{y}^{(0)}$ the corresponding received value at the destination. We also denote by $\mathbf{x}^{(i)}$ the QAM symbols transmitted by the relays corresponding to the same code symbol c, and $\mathbf{y}^{(i)}$ the corresponding channel outputs. Note that here we dropped the index of the symbol in the codewords to simplify the notations, and we just assume that the received values correspond indeed to the same symbol c. So, the symbol c receives the channel values $\left\{\mathbf{y}^{(0)}, \mathbf{y}_l^{(1)}, \ldots, \mathbf{y}_l^{(I)}\right\}$, from the source and I active relays according to one row in Figure 3 at the destination. The destination needs to compute the joint-LLR vector $\mathbf{L} = \{L_g\}_{g=1\ldots q}$, which merges the sufficient statistics of all active links. Like in Section II-C, we define the LLR vector up to an additive constant as

$$L_g \triangleq \ln P[c = g|\mathbf{y}^{(0)} \ldots \mathbf{y}^{(I)}] + \ell_1 \qquad \forall g \qquad (7)$$

Using Bayes' theorem and the fact that the source-destination and relay-destination channels are conditionally independent, we obtain

$$P[c = g|\mathbf{y}^{(0)} \ldots \mathbf{y}^{(I)}] \propto p\left(\mathbf{y}^{(0)}|c = g\right) \prod_{i=1}^{I} p\left(\mathbf{y}^{(i)}|c^{(i)} = h^{(i)}.g\right) \qquad (8)$$

where $h^{(i)}$ is the non-zero value used for the non-binary repetition encoding of symbol c at relay i.

We define then the LLR vectors corresponding to each separated channel, for the source ($i = 0$) and for the relay transmissions ($i > 0$) as

$$\lambda_g^{(i)} \triangleq \ln p\left(\mathbf{y}^{(i)}|c^{(i)} = g\right) + \ell_2 = -\frac{1}{N_0} \sum_{j=1}^{m_2} \left| y_j^{(i)} - a_j^{(i)} \mu_j(g) \right|^2 \qquad (9)$$

With $h^{(0)} = 1$ and (8), we finally obtain the joint-LLR vector components as the sum of the partial L-values:

$$L_g = \sum_{i=0}^{I} \lambda_{h^{(i)}.g}^{(i)} \qquad \forall g \qquad (10)$$

We thus combine the L-values of the main transmission and the relay transmissions into one joint-LLR vector per code symbol and feed the joint-LLR vectors to the decoder. In other words, the repetition scheme is transparent to the decoder and therefore does not affect the decoding complexity.

The process is depicted on Figure 4, which shows the factor graph used for the joint decoding of the NB-LDPC code from the source and the repetition codes from the relays. We considered on this figure the case of 3 relays, each of them sending $N_i = N.\frac{2}{3}$ extra repetition symbols. The repetition symbols are equally distributed among the codeword, *i.e.*, we have selected the repetition locations $\{c_{k_l}\}_{l=1\ldots N_i}$ at each relay, such that the destination receives 3 LLR measures for each coded symbol: one from the source, and two from the relays. This is a completely arbitrary choice and shows only the case of 3 relays with coding rates $R_{R_i D} = 3/4$. Our cooperative scheme is more general than the example of Figure 4 as the decoder can be initialized with joint-LLR vectors build from a different number of channel measurements for each coded symbol. We will discuss this issue in more details in the optimization Section IV.

Since the non-binary repetition codes impact only the the joint-LLR computation, it follows that the decoder complexity is the same, for any number of relays, which is a great feature of our cooperative coding scheme. Indeed, most of the cooperative coding schemes proposed in the literature require a joint decoding of the source and relays codes in an iterative turbo-decoding fashion [4], [10]. So the existing approaches firstly increase the receiver decoding complexity, but also prevents the use of multiple relays since the turbo-decoders with more than 2 component codes are very difficult to design so that they approach the capacity of the channel [22].

In our scheme, the decoding complexity does not depend on the number of relays — only the computation of joint-LLR vectors depends linearly on the number of relays — and more importantly, if the non-zero values for the repetition codes are well designed at the different relays, each relay brings an extra coding gain.

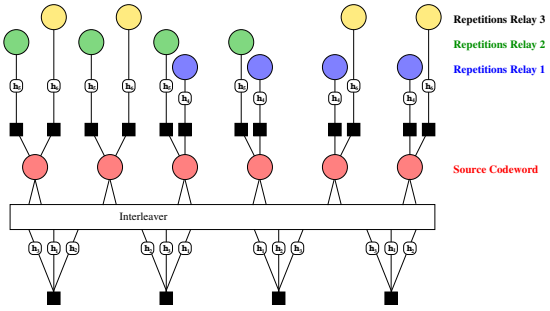

Fig. 4. Tanner graph of the joint-receiver at the destination. Case of 3 relays.

IV. Optimization of the Cooperative NB-LDPC Codes

In this section, we propose a fine optimization of the NB-LDPC cooperative coding scheme, which aims at having the best possible performance for the practical case of short to moderate codeword lengths. Both the NB-LDPC code used at the transmitter and the non-binary repetition codes at the relays have to be properly optimized. The source NB-LDPC code will be chosen so as to have the best performance in the waterfall region, to ensure that the successful decoding rates at the relays are large enough (remember that the relays are transmitting to the destination only if they successfully decode the word received from the source). Also, we propose a specific quasi-cyclic protograph construction of the Tanner graph of the NB-LDPC such that the relays can chose efficiently the locations at

 ISBN: 978-1-61208-183-0

which they should build the repetition symbols. As for the NB-repetition codes at the relays, we will propose the optimization of the non-zero field values such that the coding gain at the receiver is maximized.

A. NB-LDPC code Optimization at the Transmitter: Component Codes

For codes defined over GF(q), when addressing finite length design, it has been shown in [14] that selecting carefully the non binary entries of the parity-check matrix can improve the overall performance of the code when compared to randomly chosen coefficients. The selection of the non zero values can impact both on the waterfall and the on error floor. The observed performance gains are dependent of both the field order and the code rate.

In the waterfall region, selecting the edges label row-wise is critical. It is shown in [14] that *best* rows are selected according to their equivalent binary minimum distance and multiplicity of the minimum distance. The Binary Component Code of a non-binary parity check is build from the transpose of the companion matrices H_{ij} of the non-zero values h_{ij} composing the parity check. Using binary matrix images for the non-zero values of the check and binary vector images $\underline{c}_j$ for the codeword symbols, one get the following parity-check equation in a vector form, corresponding to the non-binary parity-check equation (1):

$$\sum_{j=1}^{d_c} H_{ij}.\underline{c}_j = \underline{0}_p \qquad \text{in GF}(2)^p$$

where $p = \log_2(q)$ is the number of bits per symbol of the Galois field.

The binary image of a non-binary parity-check in GF(q) for $q = 64$ is explained in Figure 5, and it can be easily seen that it acts as a binary component code of size $(N - K, N) = (p, p.d_c)$. The better is the component code in terms of minimum distance, the better will be the error decoding performance in the waterfall region.

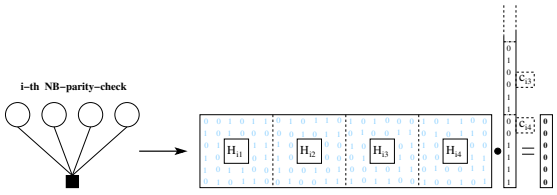

Fig. 5. Binary image of a non-binary parity-check equation in GF(64)

In this paper, we generalize the approach proposed in [14], and propose a new criterion selection for the non-zero values which compose a parity-check. In existing approaches, it is advised to maximize the strength of the component code, and then choose the non-zero field values such that the binary image has the maximum *minimum Hamming distance* (D_{min}), together with the minimum multiplicity of codewords with Hamming weight D_{min}. Although locally optimal, this strategy is not optimal when used in a message passing iterative decoder, where extrinsic vector messages are propagated along edges, *i.e.* between component codes. A better strategy, which is especially efficient when the code is a strictly regular ultra-sparse code with $d_v = 2$, is to optimize the balance between sub-codes of the component code.

Let us describe here this new idea. Since the message-passing decoder will propagate d_c extrinsic messages computed from the incoming message at each iteration, it is better to build extrinsic messages which statistically behave equally. In other words, the extrinsic messages should have their quantity of mutual information as close as possible to their average. Indeed, increasing the mutual information of one particular extrinsic output will result in decreasing the mutual information for other extrinsic messages, therefore propagating worse messages to the rest of the Tanner graph. Note that this property of *equal balance* between the extrinsic messages is verified on average if the non-zero values are taken uniformly at random. However, when one wants to optimize only a limited number of non-zero values to improve the performance, then the equal-balance property is lost, and we propose here a design technique to compensate for it.

The new optimization criterion for component code selection is described in the following algorithm.

Algorithm 1 Component Code Optimization

1) Let us a non-binary parity check of degree d_c with non-zero values $\{h_1 \dots h_{d_c}\}$
2) Consider the d_c binary subcodes $\mathcal{S}_{cc}(k)$ formed from the combination of the $d_c - 1$ values in $\{h_1 \dots h_{d_c}\}$ except h_k.
3) We choose for $\{h_1 \dots h_{d_c}\}$ the field values in $GF(q)$ such that:

$$\{h_1 \dots h_{d_c}\} = \max_{\{h_1 \dots h_{d_c}\}} \left(\sum_{k=1}^{d_c} \mathrm{D}_{min}(\mathcal{S}_{cc}(k)) \right)$$

$$\text{constrained to } \left| \mathrm{D}_{min}(\mathcal{S}_{cc}(k) - \mathrm{D}_{min}(\mathcal{S}_{cc}(k') \right| \leq 1$$

This criterion ensures that both the component code and all the sub-codes of the components codes have good and equally distributed error correction capability. This new optimization criterion is indeed interesting since we saw slight improvement in the waterfall region compared to codes that use existing sets of non-zero values. We give below the best sets of field coefficients for GF(64) and GF(256) that have been optimized with the new criterion, and that we can use for the source NB-LDPC code design. Recall that the notations used for the field elements are $\left\{0, \alpha^0, \alpha^1, \dots, \alpha^{(q-2)}\right\}$. For $d_c = 4$ and $d_c = 6$, four sets of values were found to have the exact same performance with respect to the criterion of the optimization algorithm.

- best rows for GF(64) and $d_c = 4$

$$(\alpha^0, \alpha^9, \alpha^{26}, \alpha^{46}) \quad (\alpha^0, \alpha^{17}, \alpha^{26}, \alpha^{43})$$
$$(\alpha^0, \alpha^{17}, \alpha^{37}, \alpha^{54}) \quad (\alpha^0, \alpha^{20}, \alpha^{37}, \alpha^{46})$$

- best rows for GF(256) and $d_c = 4$

$$(\alpha^0, \alpha^8, \alpha^{173}, \alpha^{183}) \quad (\alpha^0, \alpha^{10}, \alpha^{82}, \alpha^{90})$$
$$(\alpha^0, \alpha^{72}, \alpha^{80}, \alpha^{245}) \quad (\alpha^0, \alpha^{165}, \alpha^{175}, \alpha^{247})$$

 ISBN: 978-1-61208-183-0

B. NB-LDPC code Optimization at the Transmitter: Global Tanner Graph

In this section, we describe our NB-LDPC code design, based on protographs. First introduced by [23], a binary protograph is defined as a small bipartite graph from which a larger graph is obtained, by the so-called *lifting* technique. The protograph itself is generally described using its *adjacency matrix* H_B also called base matrix [24], where the coefficients $H_B(i,j)$ represent the number of edges between the $i-th$ check node C_i of the protograph and the $j-th$ variable node V_j. The base matrix H_B is then a small matrix containing small integer values. The lifting process is then to expand the base matrix by replacing each non-zero entry $H_B(i,j) > 0$ by the same number of non-overlapping circulant matrices. Circulant matrices are usually preferred for practical purposes since it reduces the descriptional complexity (ie. storage) of the parity check matrix in the hardware realizations of the LDPC encoder and decoder. If L is the size of the circulant matrices, we obtain — after lifting — a Tanner graph with L times more nodes and edges than the protograph. The last step for non-binary LDPC codes is then to assign non-zero values to the edges of the lifted Tanner graph. The nonzero values are randomly assigned from the optimized subsets presented in the previous section. Note that an additional optimization step could be performed with the objective of improving the performance in the error floor, as described in [14]. We have performed this optimization technique in our code design, but we do not present it in this paper, and refer to [14] for a complete description.

On Figure 6, we show the protograph which has been chosen for the coding rate $R = 1/2$. Similar protographs have been build for higher rates, but we limit the discussion to the rate $R = 1/2$ in this paper. The structure of the protograph has been chosen so as to maximize the number of *1-SR survivors* [25]. In [25], the authors show that under iterative decoding, all the codeword symbols do not have an equal protection, in the situation that some symbols in their direct neighborhood in the computational tree are erased or very noisy. They introduce the concept of $k-SR$ survivor symbols (k-steps recovery), which is defined as a symbol which can be recovered from the other symbols after k iterations of the message passing decoder — assuming that the other symbols are either correctly decoded, or have a large likelihood. The authors use this property to design puncturing patterns: $1-SR$ survivors can be preferably punctured, since they are less sensitive than other symbols and can be recovered easily under iterative decoding. Note that the $k-SR$ survivor property does not change if the code is a binary LDPC code or a NB-LDPC code.

Here we will use this concept to indicate to the relays where it is preferable to add non-binary repetition symbols in the codeword. The reasoning is the dual of the puncturing problem. If we know where the 1-SR symbols are in the codeword, then the relay will preferably build repetition symbols in the part of the codeword which does not contain the 1-SR symbols. This way, after merging the LLR vectors into a joint-LLR vector, the symbols which have the 1-SR property will receive less information — on average — than the other symbols. This is not a problem since the rest of the symbols, with better joint-LLR values, will be able to retrieve the 1-SR symbols. In the extreme case where the main link is so noisy that the received LLR corresponds almost to a complete erasure of the codeword, if we assume that the relays have transmitted only the symbols without the 1-SR property, then the receiver which uses the joint-LLR will successfully recover the entire codeword.

It is obvious, from this discussion, that the NB-LDPC code with the maximum of 1-SR symbols would be the best choice. This way, it helps the relays to concentrate only on the remaining part of the codeword to build the repetition symbols. This is the approach that we used in this paper for the design of the protograph. Figure 6 presents the obtained protograph for the case of a coding rate $R = 1/2$, which is the protograph with the maximum number of 1-SR symbols. In this structure, 4 checks are connected to the bottom 1-SR symbol (connected with a bold/strong link) while 2 checks are connected to 2 of the bottom symbols (indicated with a weak link). However, each one of the bottom symbol is connected exactly to one strong link and one weak link. The 1-SR condition is ensured when each and every symbol is connected to at least one strong link, i.e. at least one check node from which this symbol can be recovered in 1 iterative step. With this protograph we get 4 symbols with the 1-SR property, which is the maximum number for coding rate $R = 1/2$, and we would get $4\,L$ symbols with the 1-SR property after the lifting step. Note also that this protograph has girth 6 (size of the minimum cycle), which is also a good feature in order to get good Tanner graphs after the lifting step [26].

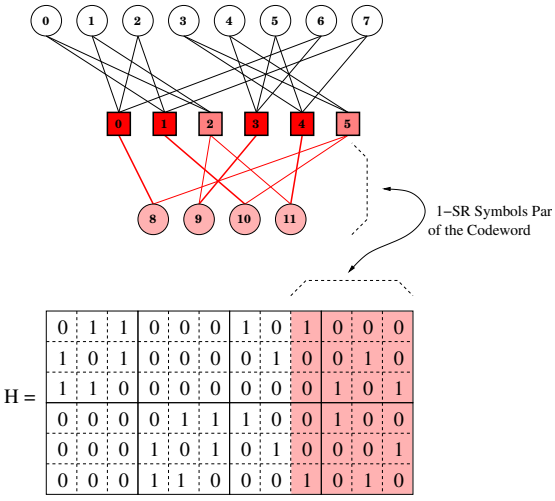

Fig. 6. Detailed protograph for the source NB-LDPC code design. This protograph has the property to maximize the number of symbols with the 1-SR property.

Let us have a look at the computational tree seen from one of the 1-SR symbol node, which is drawn on Figure 7. Symbol #8 is 1-SR from check node #0, and 2-SR from check node #5 since the symbol #10 will be recovered after the first decoding

 ISBN: 978-1-61208-183-0

iteration. Note that all the symbols in the 1-SR region have this property of being 1-SR from one of their edge, and 2-SR from the other edge, which indicates that the number of 1-SR symbols in maximum and there is no protograph with these dimensions having more 1-SR symbols.

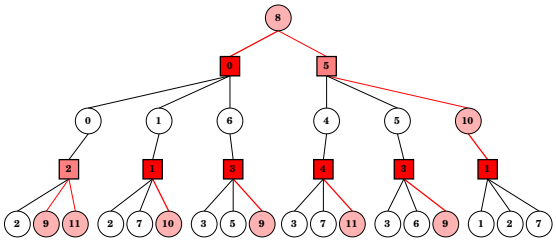

Fig. 7. Computational tree of the proposed protograph expanded from symbol node #8.

To conclude this section, we have designed source NB-LDPC codes based on a protograph approach, with connexion degrees $(d_v = 2, d_c)$. The considered protographs have both a good girth $g = 6$ and the property of localized (*a priori* known locations) maximum number of 1-SR symbols. The good girth of the protograph ensures that very large girths can be obtained for the lifted-graph, for example we obtained a girth of $g = 16$ for a NB-LDPC Tanner graph of length $N_s = 480$ coded symbols. The localized 1-SR symbols, known at the relays, are used to select the preferred locations of the non-binary repetition symbols.

C. Repetition code Optimization at the Relays

We now discuss the impact of the non-binary repetition symbols build by the relays and used in the joint-LLR computation at the destination. Usually, repetition coding is thought as having no real coding gain, but is employed to reduce the amount of noise in the received noisy symbols. Indeed, repetition coding is used in many transmission schemes, such as H-ARQ transmissions with *Chase combining* or in cooperative coding with DF or CF strategies. In the case of non-binary repetition codes however, it can be shown that the simple repetition of a symbol, weighted by a non-zero value $h^{(i)} \in \mathrm{GF}(q)$ with $h^{(i)} \neq 1$, results in a non-negligible coding gain [11].

Let us first concentrate on the case of a single repetition. We can simply explain the coding gain the following way: let c be the symbol to be repeated and $h^{(i)}.c$ being the repeated Galois field value. The receiver receives both noisy values on c and $h^{(i)}.c$, corresponding to the same codeword symbol. It follows that the demodulation actually acts as a maximum-a-posteriori decoder of the repetition code, which is build from the concatenation of the two Galois field values $[1, h^{(i)}]$. Now the coding gain is increasing with the minimum distance of the binary image of $[1, h^{(i)}]$. In the case of simple a copy — regular repetition with $h^{(i)} = 1$ — the binary minimum distance is $\mathrm{D}_{min} = 2$ and no coding gain can be achieved, while for non-binary repetitions, this minimum distance is typically larger $\mathrm{D}_{min} \geq 3$ when the field size q is sufficiently large. Additionally, the non-zero repetition values $h_l^{(i)}$ need to be optimized with the knowledge of the non-zero values which have been used in the source NB-LDPC code. Indeed, during the iterative decoding algorithm, the extrinsic vector messages will be computed using the joint-LLRs, that is, with the modified parity-check nodes, including the repetition nodes as well, as depicted on Figure 8. The modified parity-check nodes act then as the new *component codes* of the joint coding scheme. Following the discussion of Section IV-A, it is the minimum distance of this modified parity-check nodes that need to be optimized in order to have the best performance in the waterfall region.

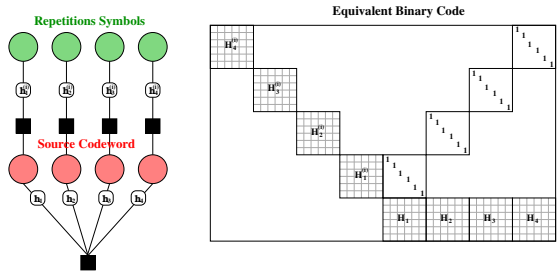

Fig. 8. NB-LDPC parity-check node with concatenated repetition codes.

We now present the optimization of non-binary repetition codes, with the objective of using only a small number of non-binary field values. We advice in particular to use the same non-zero value $h^{(i)}$ for all the repeated symbols at the relay i. By proper optimization, the coding gain is not reduced compared to a relay which would use different non-zero values $\{h_l^{(i)}\}_l$, and a single value per relay reduces greatly the complexity of re-encoding at the relay. We then use the following optimization procedure to optimize the values $h^{(i)}$, $\forall i = 1 \ldots I$.

We proceed as follows. As described in the Section IV-A, each and every check node of degree d_c will be labeled with the same set of non-zero values $\{h_1, h_2, \ldots, h_{d_c}\}$. So each location chosen by the relay in order to build a repetition symbol will *see* two of the non-zero values in this set (since $d_v = 2$). As a consequence, the non-zero values $h^{(i)}$ needs to be optimized jointly with all the values in $\{h_1, h_2, \ldots, h_{d_c}\}$. We have chosen to fix the set corresponding to the check-node values, and optimize the repetition values, conditionally to this set. The optimization is described by the following algorithm:

The optimization algorithm is stopped when the maximum number of potential relays I has been reached. We give as an example the optimized repetition values for the case of a $d_c = 4$ NB-LDPC code in GF(64) and GF(256) using the non-zero values sets presented in Section IV-A. The values that we obtained with our algorithm are indicated in table I. We have also indicated the minimum distance corresponding to the equivalent binary code (parity-check plus repetition codes). It can be seen that the minimum distance of equivalent codes grows linearly with the number of relays, which shows that the

 ISBN: 978-1-61208-183-0

Algorithm 2 Non-Binary Repetition Code Optimization

1) Let a parity-check equation have fixed non-zero values corresponding to the set $\{h_1, h_2, \ldots, h_{d_c}\}$. Let H_0 be the binary image of the equivalent code. Let $i = 1$.
2) Consider the modified binary code H_i, build from H_{i-1} and the repetition codes with the same $h^{(i)}$ on all the d_c symbols,
3) Choose $h^{(i)} \in$ GF(q) such that the the minimum distance of H_i is maximum. If several values $h^{(i)}$ have the same minimum distance, choose the one with minimum multiplicity,
4) $i = i + 1$, goto step 2).

Relay #	1	2	3	4	5	6	7	8
GF(64)	α^{26}	α^{41}	α^{52}	α^{6}	α^{56}	α^{17}	α^{50}	α^{11}
D_{min}	8	14	20	25	31	37	43	49
GF(256)	α^{15}	α^{165}	α^{71}	α^{150}	α^{128}	α^{122}	α^{113}	α^{104}
D_{min}	10	17	24	32	39	46	54	62

TABLE I
OPTIMUM NONZERO VALUES USED AT THE RELAYS FOR REPETITION CODING.

coding gain at the receiver increases as well with the number of relays.

In order to measure the performance gain in terms of frame error rate, brought by our optimized repetition scheme, we have performed Monte-Carlo simulations over a QPSK-AWGN channel for 2 different schemes using GF(256) NB-LDPC codes. The results are plotted on Figure 9. The direct link is indicated in black, and uses a rate $R = 1/2$ source NB-LDPC code. Then, we assume that the receiver receives gradually other channel values from the relays, with in this situation, the case of 4 relays. Each additional channel measurements lowers the overall coding rate, and in this figure, we assumed that 25% of the codeword length have been sent each time by the 4 relays. When the receiver has received the information from the 4 relays, the overall coding rate is indicated as $R = 1/4$. In our experiment, we have compared the simple repetition scheme with our optimized repetition scheme presented in this paper. One can see that the coding gain, when using the optimized repetition codes is non-negligible, between $0.3dB$ to $0.8dB$, with no extra decoding complexity at the receiver. It can also be seen on these curves, that the coding gain increases with the number of relays.

V. SIMULATION RESULTS IN A SIMPLE COOPERATIVE SITUATION

In order to evaluate the performance of our cooperative scheme and compare it to other works proposed in the literature, we have chosen to take the example of the simplest case of a single relay channel. The performance will be measured by the distance to the *capacity function*, inferred from the channel capacity. Capacities of various relaying strategies in case of a single relay have been computed in [27], [28], [29], and depend on the capacities of the three links. Since we assumed that source-to-relay transmission is error free, as the relay only propagates signals in case of successful decoding, we only consider the capacities of the two other links.

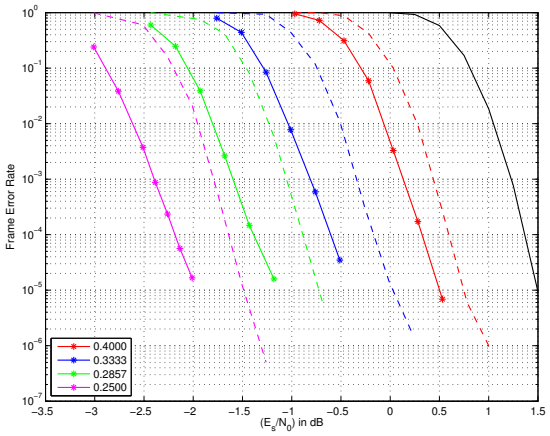

Fig. 9. Performance comparison of simple repetition scheme at the relays, and the optimized repetition scheme.

The meaning of the capacity function is the following. Assume that we want to transmit information with distributed rate (R_{SD}, R_{RD}) over a single relay channel. The rate R_{SD} is chosen according to the quality of the channel between source and relay, such that to ensure error free transmission between them. The rate R_{RD} is generally chosen according to the delay constraints of the cooperation system. Recall that γ_{SD} and γ_{RD} represent the signal-to-noise ratios from source-to-destination and respectively relay-to-destination. We define the relay *channel discrepancy* by the following quantity $\delta = \dfrac{\gamma_{\text{RD}}}{\gamma_{\text{SD}}} \geq 1$, which represents the relative quality of the relay link compared to the main link.

We have plotted on Figure 10 the SNR $(\gamma_{\text{RD}})_{\text{dB}}$ of the RD link with respect to the discrepancy $\Delta = 10\log_{10}(\delta)$ in the case of two scenarii, corresponding to $(R_{SD}, R_{RD}) = (0.5, 0.5)$ and $(R_{SD}, R_{RD}) = (0.5, 1.0)$. The second scenario is especially difficult since we assume that the relay transmit only K symbols to the destination. We have compared our cooperative scheme with the binary *split-and-extend* LDPC codes, which have been optimized for infinite length using density evolution techniques [10]. For the binary LDPC curves, we have plotted the minimum SNR $(\gamma_{\text{RD}})_{\text{dB}}$ for which the binary split-and-extend LDPC families converge to a zero error probability, when the codeword size grows to $+\infty$. As for our NB-LDPC cooperative coding scheme, we have indicated with symbols (circles and triangles) the SNR $(\gamma_{\text{RD}})_{\text{dB}}$ at which a Frame Error Rate of 10^{-5} has been reached with Monte-Carlo simulations. For our NB-LDPC coding scheme, GF(256) codes and repetition codes have been used, with a codeword length of $N = 720$ coded symbols. For the modulations, QPSK have been used for all links.

As we can see on these curves, the NB-LDPC cooperative scheme is close to the capacity curves in all cases, and shows especially a better robustness than the binary LDPC codes for the $(R_{SD}, R_{RD}) = (0.5, 1.0)$ scenario, when the discrepancy

 ISBN: 978-1-61208-183-0

becomes large. Additionnally, the capacity curves and the binary LDPC curve correspond to asymptotic performance, while our simulations are performed at relatively small lengths, corresponding to $N = 720$ coded symbols in GF(256). We then expect an axtra performance gain of our scheme by considering either longer codeword lengths or irregular mother NB-LDPC codes.

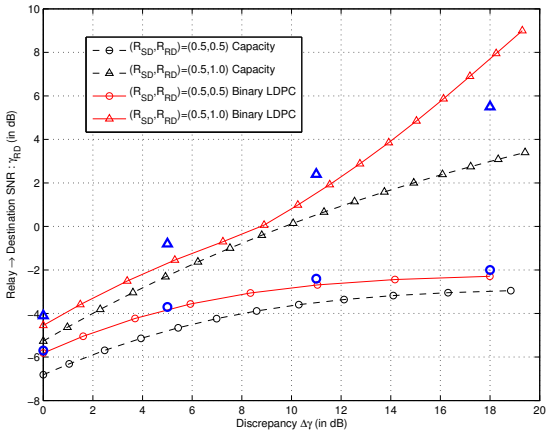

Fig. 10. Comparison of our cooperative scheme with existing binary LDPC cooperative scheme and the relay channel capacity. Our NB-LDPC scheme is indicated with the symbols (circles and triangles) in blue.

VI. Conclusions

In this paper, we have introduced and optimized a new cooperative coding scheme based on non-binary LDPC codes and the concept of non-binary repetition coding at the relays. We have shown that our scheme can reconciliate the problems usually encountered in decode-and-forward strategies, by ensuring a non-negligible coding gain at the receiver, while the joint-decoding complexity stays constant with the number of relays in the system. Additionnally, the cooperative coding scheme is independant on the channel model or on the order of the modulation used for each link in the network, which allows to keep all advantages shown in this paper with advanced link-adaptation and channel estimation techniques. This will be the purpose of a future work.

References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. Part I. System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. Part II. Implementation aspects and performance analysis," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1939–1948, 2003.

[3] J.N. Laneman, D.N. Tse, and G.W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behaviour," *IEEE Trans. on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[4] M. C. Valenti and B. Zhao, "Distributed turbo codes: towards the capacity of the relay channel," in *IEEE Vehicular Technology Conference (VTC)*, 2003, pp. 322–326.

[5] M. A. Khojastepour, N. Ahmed, and B. Aazhang, "Code design for the relay channel and factor graph decoding," in *Asilomar Conf. on Signals, Systems and Computers*, 2004, pp. 2000–2004.

[6] P. Razaghi and W. Yu, "Bilayer LDPC codes for the relay channel," in *IEEE Inter. Conf. on Communications (ICC)*, 2006, pp. 1574–1579.

[7] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Trans. on Information Theory*, vol. 53, no. 10, pp. 3723–3739, 2007.

[8] A. Chakrabarti, A. De Baynast, A. Sabharwal, and B. Aazhang, "Low-density parity-check codes for the relay channels," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 2, pp. 280–291, 2007.

[9] J. Hu and T. M. Duman, "Low density parity check codes over wireless relay channels," *IEEE Trans. on Wireless Communications*, vol. 6, no. 9, pp. 3384–3394, 2007.

[10] V. Savin, "Split-extended LDPC codes for coded cooperation," in *Information Theory and its Applications (ISITA), 2010 International Symposium*, October 2010.

[11] C. Poulliat K. Kasai, D. Declercq and K. Sakaniwa, "Multiplicatively repeated non-binary LDPC codes," *IEEE Trans. Infomation Theory*, vol. 57, no. 10, pp. 6788–6795, October 2011.

[12] D. Declercq K. Kasai and K. Sakaniwa, "Fountain coding via multiplicatively repeated non-binary LDPC codes," *to appear in IEEE Trans. Communications*, 2011.

[13] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[14] M. Fossorier C. Poulliat and D. Declercq, "Design of regular (2,dc)-LDPC codes over GF(q) using their binary images," *IEEE Trans. Communications*, vol. 56, no. 10, pp. 1626–1635, 2008.

[15] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary ldpc codes for arbitrary discrete-memoryless channels," *IEEE Trans. Information Theory*, vol. 52, no. 2, pp. 549–583, 2006.

[16] S. Pfletschinger and D. Declercq, "Non-binary coding for vector channels," in *SPAWC'11*, San Fransisco, CA, USA, June 2011.

[17] S. Pfletschinger and M. Navarro, "Link adaptation with retransmissions for non-binary ldpc codes," in *Future Network and Mobile Summit*, Florence, Italy, June 2010.

[18] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[19] H. Steendam H. Wymeersch and M. Moeneclaey, "Log-domain decoding of ldpc codes over GF(q)," in *ICC'04*, Paris, France, June 2004.

[20] D. Declercq and M. Fossorier, "Decoding algorithms for non-binary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, 2007.

[21] F. Verdier M. Fossorier A. Voicila, D. Declercq and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *IEEE Trans. Communications*, vol. 58, no. 5, pp. 1365–1375, May 2010.

[22] D. Divsalar and F. Pollara, "On the design of turbo codes," *The JPL TDA Progress Report*, pp. 42–123, November 1995.

[23] J. Thorpe, "Low-density parity-check (ldpc) codes constructed from protographs," *JPL INP, Tech. Rep.*, August 2003.

[24] Lan Lan Yifei Zhang Shu Lin William Ryan Gianluigi Liva, Shumei Song, "Design of ldpc codes: A survey and new results," *Journal of Communications Software and Systems*, 2006.

[25] D. Klinc J. Ha, J. Kim and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE, Trans. Inform. Theory*, vol. 52, pp. 728–738, 2006.

[26] D. Declercq A. Venkiah and C. Poulliat, "Design of cages with a randomized progressive edge growth algorithm," *IEEE Commun. Lett.*, vol. 12, no. 4, pp. 301–303, April 2008.

[27] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[28] M. A. Khojastepour, A. Sabharwal, and B. Aazhang, "On capacity of Gaussian 'cheap' relay channel," in *IEEE Global Telecom. Conference (GLOBCOM)*, 2003, vol. 3, pp. 1776–1780.

[29] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

 ISBN: 978-1-61208-183-0