

# A Survey on LDPC Codes for Cooperative Communications

<sup>1</sup>Phuong T. Tran

*Department of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam;*

[tranthanhphuong@tdt.edu.vn](mailto:tranthanhphuong@tdt.edu.vn)

## ABSTRACT

LDPC codes are being considered as the codes that can approach the theoretical capacity limit while are not so complicated to implement. In this survey paper, the application of LDPC codes in decode-and-forward cooperative communications is investigated. Most of current researches focus on how to implement the LDPC-coded cooperation effectively. Here we consider two approaches: the first one uses factor graph decoupling and the other uses bilayer LDPC codes. The performance analysis of these schemes is carried out by density evolution and by EXIT chart analysis. In addition, the methods to design the optimal LDPC codes for these systems are also introduced. The numerical results show that we can reach very close to the capacity limit of the relay channels. Finally, simulation results of a new iterative decoding technique for LDPC codes are also presented.

**Keywords:** LDPC; Cooperative communication; factor graph; density evolution; Decode-and-Forward.

## 1 Introduction

Idea about cooperative communications started from the work of Cover and El Gamal in 1979 [2], and then it is described more rigorously in some papers starting from 2003 ([3]– [6]). A concise tutorial about cooperative communications can be found in [1]. Briefly speaking, in cooperative communication systems, each wireless user is assumed to transmit data as well as act as a cooperative agent for another user. The data from each user can reach the base station (BTS) by at least two ways: direct transmission to BTS and relayed transmission via another user [1].

The capacities of cooperative networks has been studied rigorously in some works such as [2], [7]. However, how to design the coding schemes that can approach these capacities and are not complicated to implement is still a challenging problem and is attracting the interest of many scientists. LDPC codes are promising candidates for this application. Invented by Gallager in 1963 [8], they were almost forgotten for nearly 30 years before being rediscovered by Mac Kay in mid 90s and enhanced to irregular LDPC codes by Richardson et. al. in 2003 [9]. Since then, there has been a lot of studies about designing effective LDPC decoding methods, design optimal LDPC codes, as well as using LDPC codes in different communication systems.

Factor graph is a visualization technique used to model coding schemes [10]. In [11], an efficient implementation of LDPC codes for single-relay channel is proposed. In this scheme, the transmission of information from the source occurs in  $B$  blocks of equal length  $N$ . The factor graph of  $B$ -block transmission is then decoupled into the partial factor graphs, each of which is corresponding to a 2-block transmission. The LDPC codes for each partial factor graphs are then designed by methods proposed in [19]. By formulating the cooperative operations as equivalent SISO or MIMO systems, and using Gaussian approximation for AWGN channels, the author derive a joint relay and destination

optimization framework and develop the algorithms to solve these problems. The analysis and simulation of the performance of this LDPC-coded cooperative system will be demonstrated in this paper.

Another approach was developed from the Cover and Gamal's paper [2] is the concept of parity forwarding [18]. Based on the observation that the LDPC code designed for the relay systems is working at two different channel SNRs: a higher SNR at the relay and a lower SNR at the destination, a novel embedded LDPC code construction, namely, the bilayer LDPC code, is proposed [20]. More specifically, two new ensembles of LDPC codes, bilayer-expurgated codes and bilayer-lengthened LDPC codes, are proposed to simultaneously approach the capacities of two Gaussian channels at two different SNRs.

In this paper, the method for analyzing the performance of the design methodologies for the bilayer LDPC code, which is basically the generalization of the density evolution for standard LDPC codes, will be presented. The simulation results will show that these LDPC codes can approach in less-than-1dB distance to the theoretical capacity of the relay channels, for both single-relay case and multiple-relay case. Furthermore, this concept can be generalized to applied for more general networks.

The remaining of this paper is structured as follows: in Section II a short overview of the cooperative communication systems and LDPC codes will be given, followed by the performance analysis of the LDPC-coded cooperative systems which use the factor graph decoupling method together with the EXIT chart analysis [19]. The design methods for bilayer LDPC codes as well as the performance analysis of these codes is presented in Section III. Section VI is the conclusion of the paper.

## 2 LDPC-coded Cooperative System: Performance Analysis

### 2.1 Relay systems

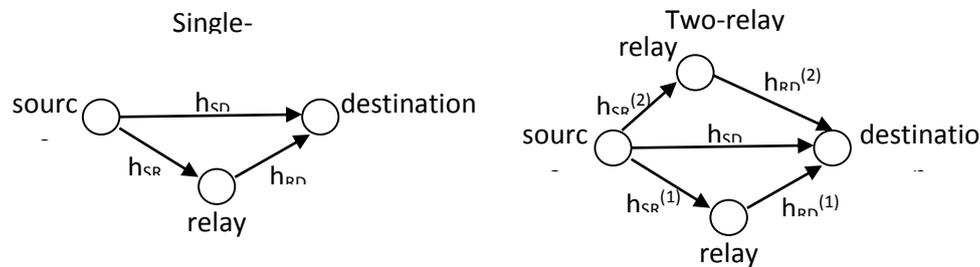


Figure 1. Diagram of single-relay system and multiple-relay system.

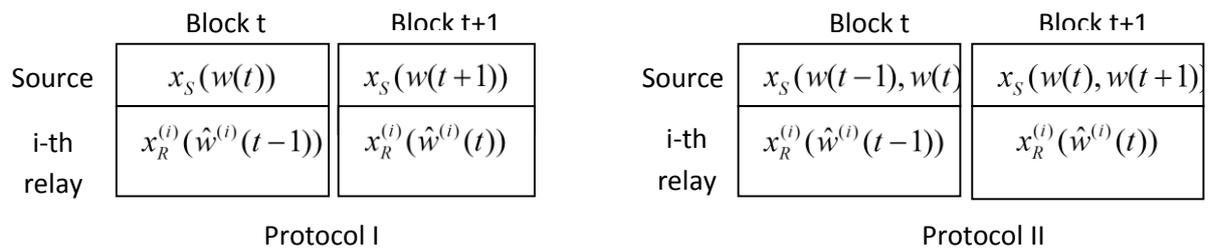
Basic relay systems are described by the diagrams in Figure 1. The left part is the diagram of a single-relay system and the right part is an example of multiple-relay system which two relays. These systems are modeled by the following equations:

$$y_R^{(i)} = h_{SR}^{(i)} x_S + \sum_{\substack{j=1 \\ j \neq i}}^K h_{jR} x_R^{(j)} + n_R^{(i)}, 1 \leq i \leq K \quad (1)$$

$$y_D = h_{SD} x_S + \sum_{i=1}^r h_{RD}^{(i)} x_R^{(i)} + n_D \quad (2)$$

where  $x_S$  and  $x_R^{(i)}$  are the signals transmitted from the source and from the  $i$ -th relay, respectively;  $y_R^{(i)}$  and  $y_D$  are the received signal at the  $i$ -th relay and at the destination, respectively;  $h_{SD}, h_{SR}^{(i)}, h_{RD}^{(i)}$ , and  $h_{j,i}$  are the channel gains between S and D, S and  $R_i$ ,  $R_i$  and D,  $R_j$  and  $R_i$ , respectively;  $K$  is the number of active relays;  $n_R^{(i)}$  and  $n_D$  are the AWGN noise at the  $i$ -th relay and at the destination, respectively. We assume the channel conditions is Rayleigh fading, i.e. the channel gains are zero-mean Gaussian distributed. Furthermore, we assume that the relays operate in full-duplex mode, and the channel distribution information (CDI) are known by the receivers, that is, the  $i$ -th relay has the knowledge of  $h_{SR}^{(i)}$  and the destination has the knowledge of  $h_{SD}$  and  $h_{RD}^{(i)}$ ,  $i = 1, 2, \dots, K$ .

The cooperation protocols are described by Figure 2. In [19], two protocols are considered, both of which transmit data in blocks. There are  $B-1$  information blocks are transmitted during  $B$  time slots.



**Figure 2. Cooperation protocols.**

where  $w(t)$  denotes the codeword transmitted by source node during time slot  $t$ , and  $\hat{w}^{(i)}(t)$  denotes the decoded codeword at the  $i$ -th relay node. For Protocol I,  $x_S(w(t)) = \sqrt{P_S} w(t)$  and  $x_R^{(i)}(\hat{w}^{(i)}(t)) = \sqrt{P_{R_i}} \hat{w}^{(i)}(t)$ ;  $P_S$  and  $P_{R_i}$  are the average transmit power per symbol at the source node and the  $i$ -th relay, respectively. For Protocol II,  $x_S(w(t-1), w(t)) = \sqrt{P_{S,1}} w(t) + \sqrt{P_{S,2}} w(t-1)$  and  $x_R^{(i)}(\hat{w}^{(i)}(t)) = \sqrt{P_{R_i}} \hat{w}^{(i)}(t)$ ;  $P_{S,1}$  and  $P_{S,2}$  are the average transmit power per symbol for  $w(t)$  and  $w(t-1)$  at the source node, respectively.

## 2.2 LPDC codes for relay systems

The performance analysis framework for the LDPC-coded cooperative systems in [19] are developed from the paper of S. ten Brink [15], in which the authors proposed a LDPC code design model for MIMO channels. In fact, the relay operations described above can be considered as a SISO system (for single-relay case) or MISO system (for multiple-relay case). Using the iterative decoding structure for MIMO systems in [15], we can set up the framework for analyzing the performance of two cooperation protocols mentioned in Section II-A. Figure 3 shows the diagram of the iterative receiver for the LDPC codes in the relay systems. The decoder consists of  $n/(M \cdot M_c)$  individual detector nodes, each of which is connected to  $M \cdot M_c$  variable nodes, where  $M$  is the number of inputs,  $M_c$  is the number of bits per symbol,  $n$  is the number of bits per codeword.

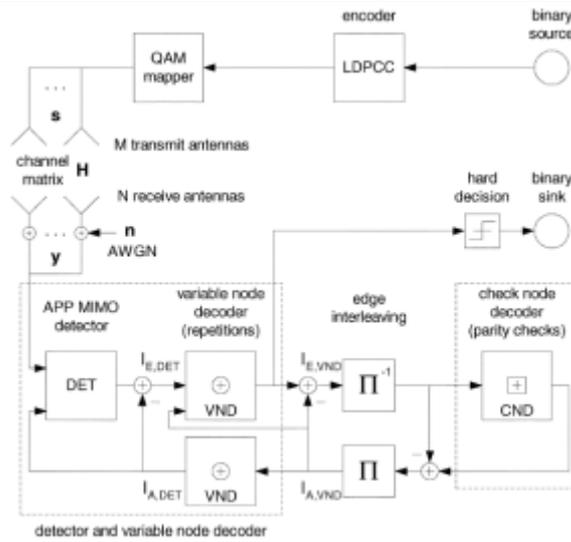


Figure 3. Iterative decoder for LDPC-coded MIMO systems

The EXIT curve formulas for this MIMO system are derived in [15]:

a. For check nodes:

$$I_{E,CND}(I_A, d_c) \approx 1 - J\left(\sqrt{d_c - 1} \cdot J^{-1}(1 - I_A)\right) \quad (3)$$

or

$$I_{A,CND}(I_E, d_c) \approx 1 - J\left(\frac{J^{-1}(1 - I_E)}{\sqrt{d_c - 1}}\right) \quad (4)$$

where  $J(\cdot)$  is a function that is defined in [15]:

$$J(\sigma) = \begin{cases} -0.0421\sigma^3 + 0.2093\sigma^2 - 0.0064\sigma & \text{if } 0 \leq \sigma \leq \sigma^* = 1.6363 \\ 1 - e^{0.0018\sigma^3 - 0.1427\sigma^2 - 0.0822\sigma + 0.0550} & \text{if } \sigma^* < \sigma < 10 \\ 1 & \text{if } \sigma \geq 10 \end{cases} \quad (5)$$

b. For variable nodes:

$$I_{E,VND}(I_{A,VND}, I_{E,DET}, d_v) = J\left(\sqrt{(d_v - 1) \cdot [J^{-1}(I_{A,VND})]^2 + [J^{-1}(I_{E,DET})]^2}\right) \quad (6)$$

where:  $I_{E,DET}(I_{A,DET}, \frac{E_b}{N_0}, R)$  can be found by Monte-Carlo simulation and

$$I_{A,DET}(I_{A,VND}, d_v) = J\left(\sqrt{d_v} \cdot J^{-1}(I_{A,VND})\right) \quad (7)$$

### 2.3 Factor graph decoupling

Any LDPC code is represented by its parity check matrix  $H$ , and also by its factor graph, which consists of the variable nodes (denoted by circles) and check nodes (denoted by squares). There is a connection between a check node  $i$  and a variable node  $j$  if and only if  $[H]_{i,j} = 1$ . Figure 4a shows the factor graph of a rate  $\frac{1}{2}$  regular (3,6) LDPC code. Because we're considering the LDPC code for relay channel, which consists of  $B$  blocks, it's more convenient to use the shorthand representation for each factor graph

structure similar to the one in Figure 4a. Figure 4b shows the equivalent shorthand representation of the LDPC code described by Figure 4a.

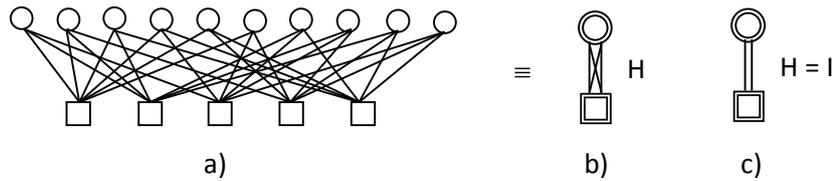


Figure 4. Factor graph of LDPC codes and its shorthand notation.

2.3.1 Relay factor graph

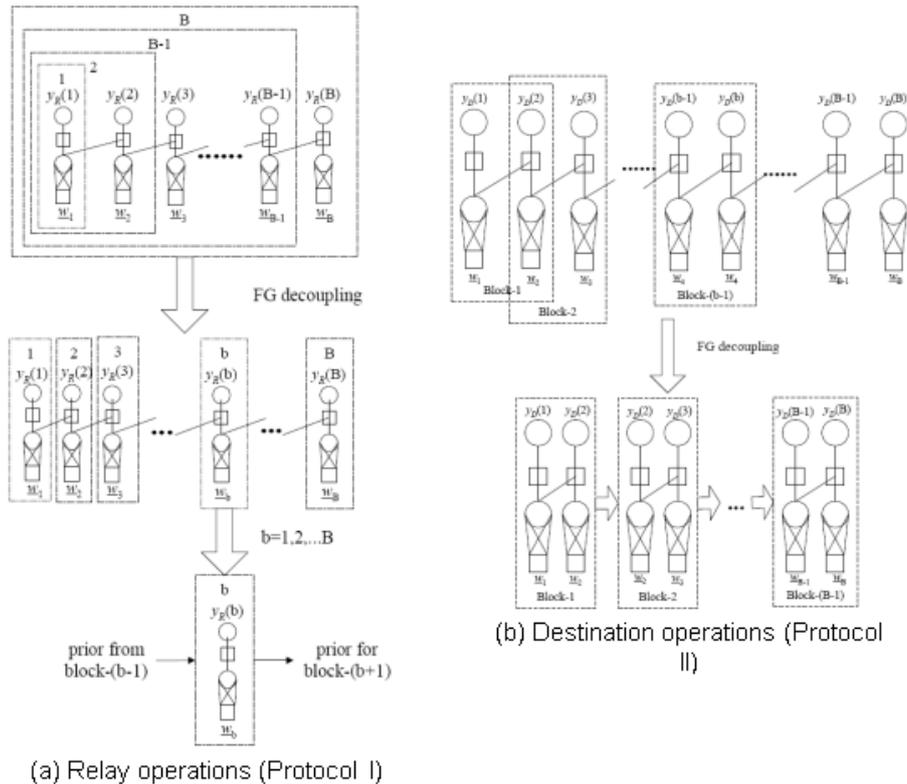


Figure 5. Factor graph for relay and destination

For Protocol I, the decoded message  $\hat{w}(t)$  depends only on  $w(t)$ , so the factor graph of the relay decoder consists of B independent, separated individual factor graphs, and we can consider each individual factor graph as a SISO model.

For Protocol II, there is a connection between the variable node  $w(t)$  with the variable node  $w(t+1)$ , and so the decoded codeword  $\hat{w}(t)$  depends on  $y_R(t')$  for  $t' = 1, 2, \dots, B$ . The factor graph for Protocol II is shown on the top part of Figure 5a. Decoding such a factor graph is impractical, therefore, a factor graph decoupling is proposed in [19] to overcome this problem. By decoupling the original factor graph as shown in the lower parts of Figure 5, we are now going to deal with B partial factor graphs. In particular, to decode the b-th factor graph, we need only the observation  $y_R(b)$  and the prior distribution of  $w(b-1)$ . Denote  $x_{R,j}^{DF}(t) \square [w_{t,j}, w_{t-1,j}]^T$  as the codeword transmitted by the source node during timeslot t, where the subscripts j indicate the j-th elements of the corresponding vector (for example,  $w_{t,j}$  indicates the j-th element of the vector  $w(t)$ ). Then:



### 2.4.1 Relay nodes

#### Algorithm 1: Performance analysis for single-relay operations under Protocol I (SISO model)

Inputs of the algorithm1 are: code ensemble edge distributions  $\left\{ \left\{ \lambda_j \right\}_{j=2}^{D_V}, \left\{ \rho_j \right\}_{j=2}^{D_C} \right\}$ ,  $\frac{E_b}{N_0}$ , coding rate

R and power allocation  $\alpha = \frac{P_R}{P_R + P_S}$ . Initialization: set  $I_{E,CND} = 0$ .

a. Calculate the average mutual information of detector output:  $I_{\text{det}}(h_{SR}, R, \frac{E_b}{N_0}, \alpha)$ . For fast fading channels,  $I_{\text{det}}(\cdot)$  is computed numerically using Monte-Carlo simulation. For AWGN channels, use

$$I_{\text{det}} = J \left( 2|h_{SR}| \sqrt{2(1-\alpha)R \frac{E_b}{N_0}} \right) \quad (9)$$

where  $J(\sigma) \square 1 - \int_{-\infty}^{+\infty} \frac{e^{-(x-\sigma^2/2)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}} \log_2(1 + e^{-x}) dx$ .

b. Set  $I_{A,VND} = I_{E,CND}$ ; then calculate the average extrinsic mutual information for the variable nodes:

$$I_{E,VND} = \sum_{i=2}^{D_V} \lambda_i J \left( \sqrt{\left[ J^{-1}(I_{\text{det}}) \right]^2 + (i-1) \left[ J^{-1}(I_{A,VND}) \right]^2} \right) \quad (10)$$

c. Set  $I_{A,CND} = I_{E,VND}$ ; then calculate the average extrinsic mutual information for the check nodes:

$$I_{E,CND} = \sum_{j=2}^{D_C} \rho_j \left[ 1 - J \left( \sqrt{(j-1) J^{-1}(1 - I_{A,CND})} \right) \right] \quad (11)$$

d. If maximum iteration number is reached, then stop and output the average extrinsic mutual information of the relay output as follows ; otherwise, go back to step (b).

$$I_R = \sum_{i=2}^{D_V} \lambda_i J \left( \sqrt{\left[ J^{-1}(I_{\text{det}}) \right]^2 + i \left[ J^{-1}(I_{A,VND}) \right]^2} \right) \quad (12)$$

#### Algorithm 2: Performance analysis for single-relay operations under Protocol II (Virtual MISO model)

During time slot t: the inputs are  $I_{A,VND}^{(t-1)}, \left\{ \left\{ \lambda_j \right\}_{j=2}^{D_V}, \left\{ \rho_j \right\}_{j=2}^{D_C} \right\}$ ,  $\frac{E_b}{N_0}$ , R,  $\alpha$ . Initialization: set  $I_{A,VND}^{(t)} = 0$ .

a. Calculate  $I_{E,Det}^{(k)}(i) = f_{R,Det} \left( I_{A,Det}^{(t)}(i), I_{A,Det}^{(t-1)}(i), \mathbf{h}_R^{DF}, \frac{E_b}{N_0}, R \right)$  for  $k = t, t-1$  and  $2 \leq i \leq D_V$  (13)

The function  $f_{R,Det}(\cdot)$  can be computed numerically by Monte-Carlo simulation.

b. Calculate

$$I_{E,VND}^{(k)} = \sum_{i=2}^{D_V} \lambda_i J \left( \sqrt{\left[ J^{-1}(I_{E,Det}^{(k)}(i)) \right]^2 + (i-1) \left[ J^{-1}(I_{A,VND}^{(k)}) \right]^2} \right) \quad (14)$$

and set  $I_{A,CND}^{(k)} = I_{E,VND}^{(k)}$  for  $k = t, t-1$ .

- c. Calculate  $I_{E,CND}^{(k)}$  using **Error! Reference source not found.** and set  $I_{A,VND}^{(k)} = I_{E,CND}^{(k)}$  for  $k = t, t-1$ .
- d. If maximum iteration number is reached, then stop and output the average extrinsic mutual information of the relay output as follows ; otherwise, go back to step (a).

$$I_R^{(k)} = \sum_{i=2}^{D_V} \lambda_i J \left( \sqrt{\left[ J^{-1}(I_{E,Det}^{(k)}(i)) \right]^2 + i \left[ J^{-1}(I_{A,VND}^{(k)}) \right]^2} \right), k = t, t-1 \quad (15)$$

### 2.4.2 Destination nodes

The decoding operations at the destination is equivalent to a virtual MIMO model. By the similar method which was applied for relay operations, we can derive the performance analysis algorithm for destination node. However, different from the previous MISO model, in this case, we have to deal with the imperfect relay decoding. To solve this problem, the author of [19] proposed a BSC modeling for the relay decoding as follows. Each bit  $\hat{w}_{t,j}$  of the decoded codeword from a certain relay is considered as the output of a BSC channel whose input is  $w_{t,j}$ . By using Gaussian approximation for the LLR of this BSC channel, its crossover probability can be estimated by

$$p_0 = Q \left( \frac{J^{-1}(I_R)}{2} \right) \quad (16)$$

$Q(x) = \int_x^{+\infty} \frac{e^{-t^2/2}}{\sqrt{2\pi}} dt$ . Now the Algorithm 3 for the destination node can be summarized below:

#### Algorithm 3: Performance analysis for joint relay-destination operations (Virtual MIMO model)

During time slot  $t$ : the inputs are  $\left\{ \left\{ \lambda_j \right\}_{j=2}^{D_V}, \left\{ \rho_j \right\}_{j=2}^{D_C} \right\}, \frac{E_b}{N_0}, R$ . Initialization: set  $I_{A,VND}^{(t)} = I_{A,VND}^{(t-1)} = 0$

- a. Compute the mutual information  $I_R$  of the relay output using Algorithm 1 (for Protocol I) or Algorithm 2 (for Protocol II).
- b. Compute the crossover probability  $p_0$  of the equivalent BSC channel for the relay output using **Error! Reference source not found.**

c. Analyze the destination performance as follows:

c1. Randomly generate  $w(k)$  and  $\hat{w}(k)$  using  $p_0$  above. Then compute the mutual information of the destination detector output:  $I_{E,Det}^{(k)}(i) = f_{D,Det} \left( I_{A,Det}^{(t)}(i), I_{A,Det}^{(t-1)}(i), \mathbf{H}, p_0, \frac{E_b}{N_0}, R \right)$ . The function  $f_{D,Det}(\cdot)$  can be computed numerically.

c2. Calculate  $I_{E,VND}^{(k)}$  using **Error! Reference source not found.** and set  $I_{A,CND}^{(k)} = I_{E,VND}^{(k)}$  for  $k = t, t-1$ .

c3. Calculate  $I_{E,CND}^{(k)}$  using **Error! Reference source not found.** and set  $I_{A,VND}^{(k)} = I_{E,CND}^{(k)}$  for  $k = t, t-1$ .

c4. If maximum iteration number is reached, then stop and output the average extrinsic mutual information of the relay output as follows ; otherwise, go back to step (c1).

$$I_D^{(k)} = \sum_{i=2}^{D_v} \lambda_i J \left( \sqrt{\left[ J^{-1}(I_{E,Det}^{(k)}(i)) \right]^2 + i \left[ J^{-1}(I_{A,VND}^{(k)}) \right]^2} \right), k = t, t-1 \quad (17)$$

### 2.4.3 LDPC code ensemble optimization

With the framework above for evaluating the performance of LDPC codes, the optimal code ensemble can be found by some searching strategies, for example, different evolution [12]. First, we generate a code ensemble, then evaluate its performance using the Algorithm 3. The criterion for assessment is the mutual information of the destination output. According to this value, we update the code ensemble to get better performance. Repeat this step until the mutual information converges to its optimum value, then we get the optimal code.

The remaining problem is the complexity of the optimization procedure. The bottle-neck of this procedure is computing  $f_{R,Det}(\cdot)$  and  $f_{D,Det}(\cdot)$  in Algorithm 3. In general, both these functions don't have closed-form expression, and must be computed numerically; therefore, it makes the optimization very time-consuming, or even infeasible. In [19], an efficient destination detector based on Gaussian approximation is introduced. The key idea is: based on the expression of the received signal at the destination, the signal component  $y_D(t)$  which corresponds to time slot  $t$ , has little effect on detecting  $w(t+1)$ . Therefore, the log-likelihood ratio  $L_{E,Det}^{(t+1)}$  can be approximated using  $y_D(t+1)$  only. Then, by using Gaussian approximation for  $L_{E,Det}^{(t+1)}$ , we can develop a sub-optimum detector  $L_{E,Det}^{(t+1)}$ , and can compute the average SNR of  $w(t+1)$ , which is denoted as  $\gamma_{E,Det}^{(t+1)}$ . Using this, we compute the extrinsic mutual information:  $I_{E,Det}^{(t+1)} = J \left( 2\sqrt{\gamma_{E,Det}^{(t+1)}} \right)$ . Now, the sub-optimum detector for  $w(t)$  can be performed by using  $y_D(t)$  and  $y_D(t+1)$  separately, which generate two LLR outputs. Then we sum up the results and get  $L_{E,Det}^{(t)}$ . This LLR is used to compute the average SNR ( $\gamma_{E,Det}^{(t)}$ ) for  $w(t)$ , and hence, the corresponding extrinsic mutual information  $I_{E,Det}^{(t)} = I(w(t), L_{E,Det}^{(t)}) = \sum_{X=-1,1} I(X, L_{E,Det}^{(t)})$ . Hence, step (c1) in Algorithm 3 can be performed semi-analytically, and the complexity is reduced [19].

## 2.5 Simulation results

In the simulation, we use the code ensemble of rate  $R = 1/2$ . The distance between source and destination is normalized to 1, and the distance from source to relay is  $d < 1$ . Under Protocol I, we consider two cases:  $d = 0.25$  and  $d = 0.5$ , and the corresponding power allocation are  $\alpha = 0.79$  and  $\alpha = 0.44$ , respectively. Under Protocol II, we choose the optimum power allocation for AWGN channels, which is expressed by the following formulas:

$$P_{S,1} = \frac{\left( |h_{SD}|^2 + |h_{RD}|^2 \right) P_{total}}{|h_{SR}|^2 + |h_{SD}|^2 + |h_{RD}|^2}, P_{S,2} = \frac{\left( |h_{SD}|^2 \cdot |h_{SR}|^2 \right) P_{total}}{\left( |h_{SD}|^2 + |h_{RD}|^2 \right) \left( |h_{SR}|^2 + |h_{SD}|^2 + |h_{RD}|^2 \right)}$$

$$P_R = \frac{\left( |h_{RD}|^2 \cdot |h_{SR}|^2 \right) P_{total}}{\left( |h_{SD}|^2 + |h_{RD}|^2 \right) \left( |h_{SR}|^2 + |h_{SD}|^2 + |h_{RD}|^2 \right)}$$

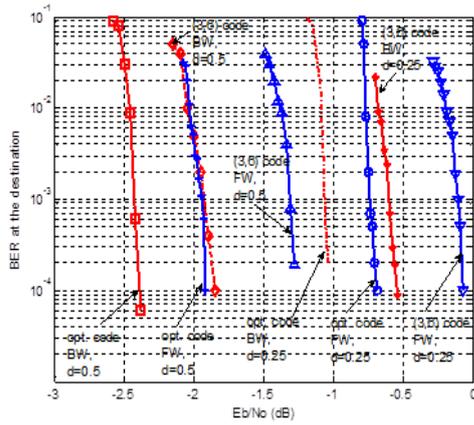


Figure 7. BER performance at the destination: AWGN channels, single-relay, Protocol I

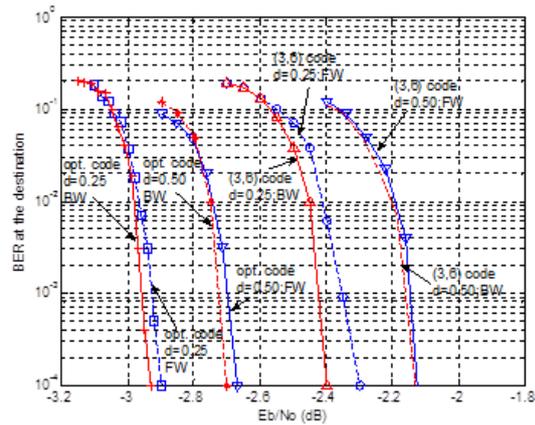


Figure 8. BER performance at the destination: AWGN channels, single-relay, Protocol II

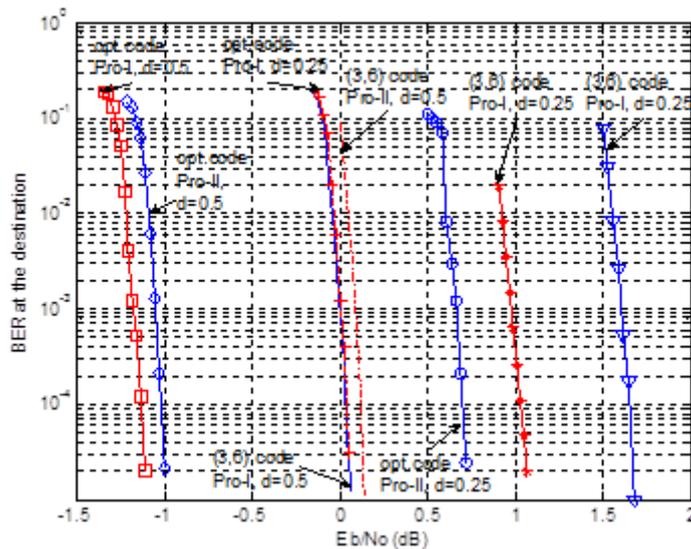


Figure 9. BER performance at the destination: fast fading channels, single-relay, backward-decoding

Figure 8 and Figure 9 show the BER performance at the destination for single-relay systems on AWGN channels, under Protocol I and Protocol II, respectively. Figure 10 shows the BER performance for single-relay system on fast fading channel.

The following are some important observation which drawn from the simulation results:

The optimized codes outperform the (3,6) codes with the gain about 1dB.

Backward-decoding outperforms forward-decoding. But under Protocol II, the gain is insignificant.

The optimized codes approach the capacity bound within 0.1dB gap for  $d = 0.25$ , and within 0.5dB gap for  $d = 0.5$ .

Protocol II is better than Protocol I in AWGN case, Protocol I is better than Protocol II in fast fading case. However, Protocol I is a special case of Protocol II when  $P_{s,2} = 0$ . We can conclude that the performance of Protocol II depends on how well the power is allocated.

### 3 Bilayer LDPC codes for Decode-and-Forward Cooperative Communication Systems

#### 3.1 Background

##### 3.1.1 Decode-and-Forward (DF) Strategy

In this section, we review the decode-and-forward cooperation strategy which is first introduced by Cover and Gamal in 1979 [2]. Recall that a Gaussian relay channel can be modeled as:

$$Y_1 = X + Z_1 \quad (18)$$

$$Y = X + X_1 + Z_2 \quad (19)$$

where  $Y_1$  and  $Y$  are the received signal at the relay node and the destination, respectively;  $X_1$  and  $X$  are the signals transmitted by the relay and the source, respectively.  $Z_1 \sim N(0, N_1)$  and  $Z_2 \sim N(0, N_2)$  are AWGN noises at the relay node and the destination, respectively. The decode-and-forward strategy is described as follows.

Let  $n$  be the number of bits in each codeword transmitted by the source. During block  $i$ , the source selects a message  $w_i \in \{1, 2, \dots, 2^{nR}\}$ , where  $R$  is the rate of the code. The set of source messages are randomly partitioned into  $2^{nR_1}$  bins, each of which has the size of  $2^{n(R-R_1)}$  ( $R_1 \leq R$ ). Let  $s_i$  denote the bin index of the message  $w_{i-1}$ . In block  $i$ , the source transmit a linear combination of the encoded version of the message  $w_i$  and the bin index of message  $w_{i-1}$ , while the relay transmits the encoded version of the bin index  $s_i$  of the previous message:

$$\underbrace{\mathbf{X}(w_i | s_i)}_{\text{transmitted by src}} = \underbrace{\tilde{\mathbf{X}}(w_i)}_{\text{enc. of } w_i} + \sqrt{\frac{(1-\alpha)P}{P_1}} \cdot \underbrace{\mathbf{X}(s_i)}_{\substack{\text{enc. of } s_i \\ \text{transmitted by relay}}} \quad (20)$$

where  $P$  and  $P_1$  are the maximum transmit powers of source and relay, respectively;  $\alpha$  is a fraction of power used for transmitting new message  $w_i$ .

The decoding process happens as follows: the relay node know  $\mathbf{X}(s_i)$ , so it can decode  $w_i$  based on  $\tilde{\mathbf{X}}(w_i)$  (plus Gaussian noise  $Z_1$ ). After decoding  $w_i$ , it can compute  $s_{i+1}$ , which is transmitted in the next block. At the destination,

$$\mathbf{Y} = \tilde{\mathbf{X}}(w_i) + \left(1 + \sqrt{\frac{(1-\alpha)P}{P_1}}\right) \mathbf{X}(s_i) + \mathbf{Z}_2 \quad (21)$$

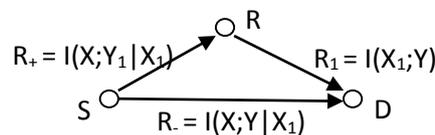


Figure 10. Code designing problem for DF relay systems

First, the destination considers  $\tilde{\mathbf{X}}(w_i)$  as noise and decode  $s_i$ . After that, the  $\mathbf{X}(s_i)$  is subtracted from  $\mathbf{Y}$ , and the remaining is used to decode  $w_i$ . For successful decoding in each step,  $R$ ,  $R_1$  and  $R - R_1$  must be upper-bounded by some constraints (Shannon's Theorem). Combining these constraints, the overall DF rate is upper-bounded by

$$R \leq \max_{\alpha} \min \left\{ \frac{1}{2} \log \left( 1 + \frac{\alpha P}{N_1} \right), \frac{1}{2} \log \left( 1 + \frac{P + P_1 + 2\sqrt{(1-\alpha)PP_1}}{N_1 + N_2} \right) \right\} \quad (22)$$

The goal of code designing for DF relay systems is summarized in Figure 11: want to construct a source codebook that simultaneously approach the rates  $R_+$  and  $R_-$ , and a relay codebook to approach the rate  $R_1 = R_+ - R_-$ .

### 3.1.2 Parity forwarding

Naturally, binning can be implemented by generating extra parity bits (or syndromes) on the codewords in the source codebook  $\tilde{X}$ . Codewords in each bin must satisfy a set of parity equation. Therefore, we can implement binning on the DF relay systems using LDPC code as follows.

- a. The source message is encoded using an  $(n, n-k_1)$  LDPC code and then transmitted during block  $i$ .
- b. The relay node decodes the transmitted codeword  $\tilde{X}(w_i)$ , generating  $k_2$  extra parity bits, encodes these  $k_2$  bits using its codebook  $\tilde{X}_1$  and send the result codeword to the destination in block  $i + 1$ .
- c. The destination first decodes the extra  $k_2$  parity bits, then decodes the source message over a bilayer code construction.

This strategy is called parity-forwarding strategy.

## 3.2 Designing Bilayer-Expurgated LDPC Codes

This code is proposed by P. Razaghi and W. Yu [20]. Its structure is shown in Figure 12. The lower layer of Figure 11 represents an  $(n, n - k_1)$  LDPC code for source-relay channel. The whole graph represent a  $(n, n - k_1 - k_2)$  subcode of the lower layer code, which is called bilayer-expurgated code. This subcode satisfies two set of parity equations:  $k_1$  equations by the source codebook and  $k_2$  equations by the relay codebook.

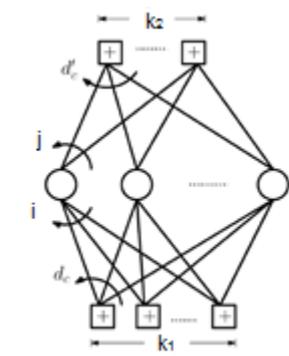


Figure 11. Bilayer-expurgated LDPC code structure

Our goal is to design the source and relay codes such that the lower code can approach the capacity  $R_+$  and the bilayer code can approach the capacity  $R_-$ .

### 3.2.1 BE-LDPC code ensemble

Like a standard LDPC code ensemble, an ensemble of bilayer-expurgated LDPC code is defined based on edge distributions. However, there is some modification in the definition of edge distributions for bilayer-expurgated LDPC codes. First notice that we have two sets of check nodes: the lower check

nodes corresponding to the  $k_1$  source parity checks and the upper check nodes corresponding to the  $k_2$  relay parity checks. Now we define:

- A *lower edge* is an edge connecting a variable node to a lower check node.
- An *upper edge* is an edge connecting a variable node to an upper check node.
- *Lower degree of a variable node* is the number of lower edges connected to it.
- *Upper degree of a variable node* is the number of upper edges connected to it.
- *Check node degree of a check node* is the number of edges connected to it.
- *Lower degree of an edge* is the lower degree of the variable node it is connected to.
- *Upper degree of an edge* is the upper degree of the variable node it is connected to.
- An *edge (or a variable node) has degree (i,j)* iff it has lower degree  $i$  and upper degree  $j$ .
- *Variable degree distribution*  $\lambda_{i,j}$  is the percentage of edges with degree  $(i,j)$  ( $i \geq 2, j \geq 0$ )
- $\eta$  is the percentage of *lower edges* in the bilayer graph.

A bilayer-expurgated LDPC code ensemble is characterized by  $\left\{ \left\{ \lambda_{i,j} \right\}_{\substack{i \geq 2 \\ j \geq 0}}, \eta \right\}$ .

### 3.2.2 Bilayer Density Evolution

Different from the density evolution for standard LDPC codes, the density evolution for bilayer LDPC codes employs two densities to be tracked: lower density corresponding to the messages in lower part and upper density corresponding to the upper part of the bilayer graph.

Denote  $p^{(t)}$  and  $q^{(t)}$  as the message pdf at the input of the lower and upper check nodes, respectively, at the beginning of the  $t$ -th iteration. Because each check node involves in only one kind of density (lower or upper), the update rule for check nodes of bilayer code is not different from the update rule for check nodes of standard LDPC codes. Denote  $p'^{(t)}$  and  $q'^{(t)}$  as the densities resulted from the lower and upper check update, respectively. Then:

$$p'^{(t)} = F_{L,CND}(p^{(t)}, d_c) \quad (23)$$

$$q'^{(t)} = F_{U,CND}(q^{(t)}, d_c') \quad (24)$$

where  $d_c$  and  $d_c'$  are the lower and upper check degree, respectively;  $F_{L,CND}(\cdot)$  and  $F_{U,CND}(\cdot)$  are the update functions for lower and upper check nodes, respectively.

Now we proceed with the message density at a degree  $(i, j)$  variable node at the beginning of the  $(t+1)$ -th iteration:

$$p_{i,j}^{(t+1)} = \left( \otimes^{i-1} p'^{(t)} \right) \otimes \left( \otimes^j q'^{(t)} \right) \otimes p_c, i \geq 2, j \geq 0 \quad (25)$$

$$q_{i,j}^{(t+1)} = \left( \otimes^i p'^{(t)} \right) \otimes \left( \otimes^{j-1} q'^{(t)} \right) \otimes p_c, i \geq 2, j \geq 1 \quad (26)$$

where  $p_c$  is the density of the LLR received over the channel, and  $\otimes^k f = f \otimes f \otimes \dots \otimes f$  ( $k$  terms) for  $k \geq 2$  and  $\otimes^1 f = f, \otimes^0 f = \delta$  (Dirac function).

Finally, the message densities at input of the lower and upper check nodes at the beginning of the (t+1)-th iteration is updated as

$$p^{(t+1)} = \sum_{i \geq 2, j \geq 0} \frac{i}{i+j} \lambda_{i,j} p_{i,j}^{(t+1)} \quad (27)$$

$$q^{(t+1)} = \sum_{i \geq 2, j \geq 0} \frac{i}{i+j} \lambda_{i,j} q_{i,j}^{(t+1)} \quad (28)$$

The overall message error probability at the beginning of the (t+1)-th iteration is

$$e(p^{(t+1)}, q^{(t+1)}) = \sum_{i \geq 2, j \geq 0} \lambda_{i,j} \left( \frac{i}{i+j} e_{i,j}^1(p^{(t)}, q^{(t)}) + \frac{j}{i+j} e_{i,j}^2(p^{(t)}, q^{(t)}) \right) \quad (29)$$

where  $e_{i,j}^1(p^{(t)}, q^{(t)})$  and  $e_{i,j}^2(p^{(t)}, q^{(t)})$  are the message error probability corresponding to the densities  $p_{i,j}^{(t+1)}$  and  $q_{i,j}^{(t+1)}$ , respectively, after one evolution iteration with inputs  $p^{(t)}$  and  $q^{(t)}$ , respectively.

### 3.2.3 BE-LDPC code optimization

There are many approaches to optimization the code ensemble to achieve the goal we mentioned at the beginning of this section. A simple but efficient approach is proposed in [20]. The key idea is first fix the check degrees  $d_c, d_c'$  and the lower graph, try to find the edge distributions such that the rate of the bilayer code approaches R.. Then change to another pair  $(d_c, d_c')$  and repeat the optimization process.

Step 1: Fix  $d_c, d_c'$  and the lower subgraph. The rate of the bilayer code is  $1 - (k_1 + k_2)/n$ , so to maximize it we need to minimize  $k_2$ , or equivalently, maximize  $\eta$  because  $\eta = \frac{d_c k_1}{d_c k_1 + d_c' k_2}$ . By fixing the lower graph, we also fix its edge distribution  $\lambda_i$ , which is related to  $\lambda_{i,j}$  by

$$\lambda_i = \frac{1}{\eta} \sum_{j \geq 0} \frac{i}{i+j} \lambda_{i,j} \Leftrightarrow \sum_{j \geq 0} \frac{i}{i+j} \lambda_{i,j} - \eta \lambda_i = 0 \quad (30)$$

Therefore, our problem at step 1 can be formulated as follows:

$$\max_{\{\lambda_{i,j}, \eta\}} \eta \quad (31)$$

$$\text{subject to } \sum_{j \geq 0} \frac{i}{i+j} \lambda_{i,j} - \eta \lambda_i = 0 \quad (32)$$

$$\sum_{i \geq 2, j \geq 0} \lambda_{i,j} \left( \frac{i}{i+j} e_{i,j}^1(p^{(t)}, q^{(t)}) + \frac{j}{i+j} e_{i,j}^2(p^{(t)}, q^{(t)}) \right) < \mu^h e(p^{(t)}, q^{(t)}) \quad (33)$$

$$\sum_{\substack{i \geq 2 \\ j \geq 0}} \lambda_{i,j} = 1 \quad (34)$$

where  $h$  is the optimization iteration number,  $0 < \mu^h < 1$  is a convergence factor that is increased in each iteration towards 1. This problem is a linear programming problem and can be solved iteratively [20]

**Step 2:** Establish an appropriate range for  $d_c, d'_c$  and searching over this range to find the optimal value. To do this, repeat Step 1 for each feasible pair  $(d_c, d'_c)$ .

### 3.3 Designing Bilayer-Lengthened LDPC Code

In this section, we consider the second codes proposed in [20], namely, the bilayer-lengthened LDPC codes (BL-LDPC). Its representation graph is shown in Figure 13. Both the lower graph and the overall graph have the same number of check nodes.

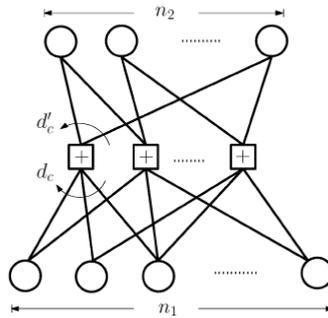


Figure 12. Bilayer-lengthened LDPC code

First, the source transmits the codewords from the  $(n_1 + n_2, k_1)$  LDPC codes (denoted as  $C_1$ ) represented by the overall graph. The relay then decodes the source codeword. It encodes the  $n_2$  bits corresponding to the upper part of the graph, using a  $(n_2, k_2)$  LDPC code (denoted as  $C_2$ ) (by adding  $k_2$  parity bits), and then sends these  $k_2$  parity bits to the destination, using a codebook of rate  $R_1 = R_+$  -  $R_-$ . Our goal is to design the source and relay codes such that the lower code can approach the capacity  $R_-$  and the bilayer code can approach the capacity  $R_+$ .

For decoding at the destination, in each block, the  $k_2$  parity bits are decoded first, then they were used to decode the  $n_2$  upper variable nodes of the source codeword transmitted in the previous block. The destination then removes the upper part of the overall graph and updates the parity check equations. Finally, the remaining part (the lower part) of the graph is decoded to get the data.

This code structure has good performance in the channel conditions which have large gap between  $R_+$  and  $R_-$ .

#### 3.3.1 Bilayer-lengthened LDPC code ensemble

The BL-LDPC code is dual to the BE-LDPC code in the sense of interchanging between the role of variable nodes and check nodes. So we have the similar definitions as the previous section.

- A lower edge is an edge connecting a check node to a lower variable node.
- An upper edge is an edge connecting a check node to an upper variable node.
- Variable node degree of a variable node is the number of edges connected to it.
- Variable degree of an edge is the degree of the variable node it is connected to.
- Lower variable degree distribution  $\lambda_{i,1}$  is the percentage of lower edges with degree  $i$  ( $i \geq 2$ ).
- Upper variable degree distribution  $\lambda_{i,2}$  is the percentage of upper edges with degree  $i$  ( $i \geq 2$ ).
- $d_c, d'_c$  are the number of edges in the lower and upper subgraph, respectively.

A Bilinear-Lengthened LDPC code ensemble is defined by  $\left\{ \left\{ \lambda_{i,1} \right\}_{i \geq 2}, \left\{ \lambda_{i,2} \right\}_{i \geq 2}, d_c, d'_c \right\}$ .

### 3.3.2 Bilinear Density Evolution

Denote  $p^{(t)}$  and  $q^{(t)}$  as the message pdf in the lower and upper parts, respectively, at the beginning of the  $t$ -th iteration. Denote  $p^{*(t)}$  and  $q^{*(t)}$  as the densities resulted from the lower and upper check update, respectively. Denote  $\oplus$  as the check density-update operation, and  $\oplus^d f = f \oplus f \oplus \dots \oplus f$  ( $d$  terms),  $\oplus^1 f = f$ ,  $\oplus^0 f = 1$ . Then:

$$p^{*(t)} = \left( \oplus^{d_c-1} p^{(t)} \right) \oplus \left( \oplus^{d'_c} q^{(t)} \right), d_c > 1 \quad (35)$$

$$q^{*(t)} = \left( \oplus^{d_c} p^{(t)} \right) \oplus \left( \oplus^{d'_c-1} q^{(t)} \right), d'_c \geq 1 \quad (36)$$

The update rules for the message densities at a variable node of degree  $i$  in the lower and upper subgraphs, respectively, are:

$$p_i^{(t+1)} = \left( \otimes^{i-1} p^{*(t)} \right) \otimes p_c, i \geq 2 \quad (37)$$

$$q_i^{(t+1)} = \left( \otimes^{i-1} q^{*(t)} \right) \otimes p_c, i \geq 2 \quad (38)$$

where  $p_c$  is the density of the LLR received over the channel.

Finally, the message densities in the lower and upper parts at the beginning of the  $(t+1)$ -th iteration is updated as

$$p^{(t+1)} = \sum_{i \geq 2} \lambda_{i,1} p_i^{(t+1)} \quad (39)$$

$$q^{(t+1)} = \sum_{i \geq 2} \lambda_{i,2} q_i^{(t+1)} \quad (40)$$

The overall message error probability at the beginning of the  $(t+1)$ -th iteration is:

$$e(p^{(t+1)}, q^{(t+1)}) = \sum_{i \geq 2} \eta \lambda_{i,1} e_{i,1}(p^{(t)}, q^{(t)}) + (1-\eta) e_{i,2}(p^{(t)}, q^{(t)}) \quad (41)$$

where  $\eta = \frac{d_c}{d_c + d'_c}$  and  $e_{i,1}(p^{(t)}, q^{(t)})$ ,  $e_{i,2}(p^{(t)}, q^{(t)})$  are the message error probability corresponding to the densities  $p_i^{(t+1)}$  and  $q_i^{(t+1)}$ , respectively, after one evolution iteration with inputs  $p^{(t)}$  and  $q^{(t)}$ , respectively.

### 3.3.3 BL-LDPC code optimization

**Step 1:** Fix  $d_c, d'_c$ , find the lower variable distributions  $\lambda_{i,1}$  such that the LDPC code represented by the lower subgraph has the rate approaching to  $R$ .

**Step 2:** Fix  $d_c, d'_c$ , and the lower variable distributions  $\lambda_{i,1}$  found from step 1. Our goal at Step 2 is to maximize the rate of the overall bilayer-lengthened code:  $1 - k/(n_1 + n_2)$  where  $k$  is the number of check nodes. To do this, we need to maximize the number of upper variable nodes  $n_2$ , which is given

by  $n_2 = d'_c k \sum_{i \geq 2} \frac{\lambda_{i,2}}{i}$ . Because  $k$  and  $d'_c$  are fixed, our problem can be formulated as follows:

$$\max_{\lambda_{i,2}} \sum_{i \geq 2} \frac{\lambda_{i,2}}{i} \tag{42}$$

$$\text{subject to: } \sum_{i \geq 2} \eta \lambda_{i,1} e_{i,1}(p^{(t)}, q^{(t)}) + (1-\eta) e_{i,2}(p^{(t)}, q^{(t)}) < \mu^h e(p^{(t)}, q^{(t)}) \tag{43}$$

$$\sum_{i \geq 2} \lambda_{i,2} = 1 \tag{44}$$

**Step 3:** Search for optimal  $d_c, d_c'$  over a reasonable range. For each feasible  $(d_c, d_c')$ , repeat Step 1 and Step 2 to find the optimal code and record its performance.

### 3.4 Simulation results

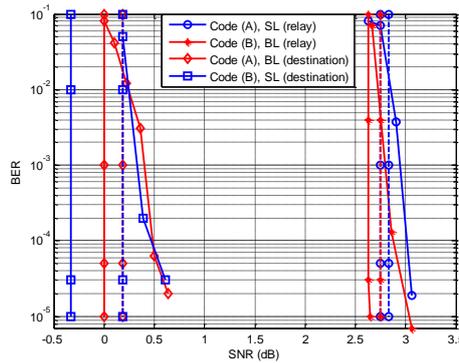


Figure 13. Comparison between expurgated code and lengthened code

In this section, the performance of the bilayer-expurgated and bilayer-lengthened LDPC codes are compared together. Code (A) (expurgated code) and code (B) (lengthened code) are compared in Figure 14. The performance of code (C) and (D) are illustrated in Figure 15 and Figure 16, respectively, to show that the expurgated code is good for the small gap between  $R_+$  and  $R_-$ , while the lengthened code is good for the large gap between  $R_+$  and  $R_-$ . All of these codes are designed from the optimization procedure mentioned above. The target rates ( $R_-$ ,  $R_+$ ) for each code are as follows: (0.3, 0.4) for code(A), (0.5, 0.7) for code (B), (0.3, 0.9) for code (C) and (0.65, 0.7) for code (D). The maximum variable degree for all codes is set to 20. The maximum number of iterations is 600, and the block length is chosen to be 100000. The solid straight lines in each figure represent the theoretical rate limits, while the dashed lines represent the convergence thresholds.

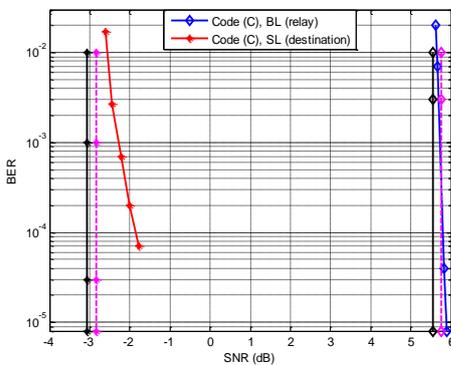


Figure 14. Bilayer-lengthened LDPC code for large SNR gap between relay and destination

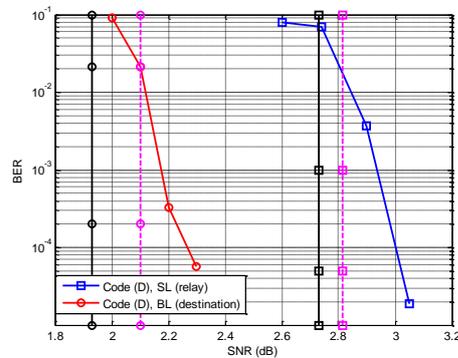


Figure 15. Bilayer-expurgated LDPC code for small SNR gap between relay and destination

The following are some important observations:

- The optimal pair  $(d_c, d'_c)$  for the codes from (A) to (D) are (15, 4), (8, 6), (5, 33) and (15, 8), respectively.
- The expurgated code does better in the condition of small gap between  $R_+$  and  $R_-$ , and the lengthened code does better in the condition of large gap between  $R_+$  and  $R_-$ .

The convergence thresholds is close to the theoretical limits (less than 0.5dB), and the SNR gap between the BER curves and the corresponding convergence thresholds are also small, that confirms the asymptotic convergence.

#### 4 Conclusion and Further Work

In this paper I do a survey on how to apply LDPC codes, which can approach the capacity-limit of the communication channels, to the cooperative communications system. Two main concepts has been introduced in this paper. The first one is using iterative decoding and factor graph representation to analyze the performance of LDPC-coded relay systems, in which, the complexity of the analysis is reduced by using factor graph decoupling method. As a result, some algorithms to design the optimized LDPC codes for relay systems have been proposed. The second concept is parity-forwarding. Based on this concept, two new kinds of LDPC codes have been presented, namely, the Bilayer-expurgated LDPC codes and the Bilayer-lengthened LDPC codes. It has been show that these two codes can simultaneously approach the capacity limits of two Gaussian channels (source-relay channel and source-destination channel) at two different SNRs. Further works can be developed from these concepts, for example, consider a multiple relay networks.

#### REFERENCES

- [1]. Aria Nosratinia, Ahmadreza Hedayat, *Cooperative communications in Wireless Networks*, IEEE Communication Magazine, 2004.
- [2]. T. M. Cover and A. A. E. Gamal, *Capacity Theorems for the Relay Channel*, Information Theory, IEEE Transaction on, 1979. 25(5): pp. 572–84.
- [3]. A. Sendonaris, E. Erkip, and B. Aazhang, *User Cooperation Diversity Part I and Part II*, Communication, IEEE Transaction on, 2003. 51( 11): pp. 1927–48.
- [4]. T. E. Hunter and A. Nosratinia, *Cooperation Diversity through Coding*, Proceedings of 2002 IEEE International. Symposium on Information Theory (ISIT'02), Lausanne, Switzerland, 2002. pp. 220.
- [5]. K.J. Ray Liu, Ahmed K. Sadek, Weifeng Su, Andres Kwasinski, *Cooperative Communications and Networking*, Cambridge University Press, 2009.
- [6]. J. N. Laneman, G. W. Wornell, and D. N. C. Tse, *An Efficient Protocol for Realizing Cooperative Diversity in Wireless Networks*, Proceedings of IEEE ISIT, Washington, DC, June 2001. p. 294.
- [7]. G. Kramer, M. Gastpar, and P. Gupta, *Cooperative strategies and capacity theorems for relay networks*, Information Theory, IEEE Transaction on, 2005. 51(9): pp. 3037–3063.

- [8]. R. G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA: MIT Press, 1963.
- [9]. T. Richardson and R. Urbanke, *The Renaissance of Gallager's Low-Density Parity-Check Codes*, IEEE Communications Magazine, 2003, 41: pp. 126–131.
- [10]. F. R. Kschischang, B. J. Frey, and H. A. Loeliger, *Factor Graphs and the Sum-Product Algorithm*, Information Theory, IEEE Transaction on, 2001. 47(2): pp. 498-519.
- [11]. M. A. Khojastepour, N. Ahmed, and B. Aazhang, *Code design for the relay channel and factor graph decoding, Proceedings of 38<sup>th</sup> Asilomar Conference on Signals and Systems, Computing. (Asilomar '04)*, Pacific Grove, CA, Nov. 2004. pp.2000-2004.
- [12]. T. J. Richardson, M. A. Shokrolahhi, and R. L. Urbanke, *Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes*, Information Theory, IEEE Transaction on, 2001. 47(2): pp. 619-637.
- [13]. T. J. Richardson, and R. L. Urbanke, *The Capacity of Low-Density Parity-Check Codes under Message-Passing Coding*, Information Theory, IEEE Transaction on, 2001. 47(2): pp. 599-618.
- [14]. E. Sharon, A. Ashikhmin, and S. Litsyn, *EXIT Functions for the Gaussian Channel*, Proceedings of 40th Annual Allerton Conference on Communication, Control, Computing (Allerton '03), Allerton, IL, Oct. 2003. 2:pp. 972-981.
- [15]. S. ten Brink, G. Kramer, and A. Ashikhmin, *Design of Low-Density Parity-Check Codes for Modulation and Detection*, Communications, IEEE Transaction on, 2004. 52(4): pp. 670-678.
- [16]. S. ten Brink, *Convergence behavior of iteratively decoded parallel concatenated codes*, Communications, IEEE Transaction on, 2001. 49(10): pp. 1727–1737.
- [17]. G. Kramer, *Communication strategies and coding for relaying*, in *Wireless Communications: IMA Volumes in Mathematics and its Applications*, P. Agrawal, D. M. Andrews, P. J. Fleming, G. Yin, and L. Zhang, Eds. New York: Springer-Verlag, 2007. 143: pp. 163–175.
- [18]. P. Razaghi and W. Yu, *Parity-forwarding for multiple-relay networks*, in *Proceedings of IEEE International Symposium on Information Theory*, Seattle, WA, Jul. 2006. pp. 1678–1682.
- [19]. C. Li, G. Yue, M. A. Khojastepour, X. Wang, and M. Madhian, *LDPC-coded Cooperative Relay Systems: Performance Analysis and Code Design*, Communications, IEEE Transaction on, 2008. 56(3): pp. 485-496.
- [20]. P. Razaghi, and W. Yu, *Bilayer Low-Density Parity-Check Codes for Decode-and-Forward in Relay Channels*, Information Theory, IEEE Transaction on, 2007. 53(10): pp. 3723-3739.