Use of the LDPC codes Over the Binary Erasure Multiple Access Channel

Sareh Majidi Ivari

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

March 2017

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Sareh Majidi Ivari

Entitled: Use of the LDPC Codes Over the Binary Erasure Multiple Access Channel
and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____ Chair

Dr. R. Raut

____ External Examiner

Dr. A. Youssef

____ Internal Examiner

Dr. Y.R. Shayan

____ Supervisor

Dr. M.R. Soleymani

Approved by: __________________________________

Dr. W.E. Lynch, Chair

Department of Electrical and Computer Engineering

______________ 20___ __________________________________

Dr. Amir Asif, Dean,

Faculty of Engineering and Computer

Science

ABSTRACT

Use of the LDPC Codes Over the Binary Erasure Multiple Access Channel

Sareh Majidi Ivari.

Concordia University, 2017

Wireless communications use different orthogonal multiple access techniques to access a radio spectrum. The need for the bandwidth efficiency and data rate enhancing increase with the tremendous growth in the number of mobile users. One promising solution to increase the data rate without increasing the bandwidth is non-orthogonal multiple access channel. For the noiseless channel like the data network, the non-orthogonal multiple access channel is named: Binary Erasure Multiple Access Channel (BEMAC). To achieve two corner points on the boundary region of the BEMAC, a half rate code is needed. One practical code which has good performance over the BEMAC is the Low Density Parity Check (LDPC) codes. The LDPC codes receive a lot of attention nowadays, due to the good performance and low decoding complexity. However, there is a tradeoff between the performance and the decoding complexity of the LDPC codes. In addition, the LDPC encoding complexity is a problem, because an LDPC code is defined with its parity check matrix which is sparse and random and lacks of structure.

This thesis consists of two main parts. In the first part, we propose a new practical method to construct an irregular half LDPC code which has low encoding complexity. The constructed code supposed to have a good performance and low encoding complexity. To have a low encoding complexity, the parity check matrix of the code must have lower triangular shape. By implementing the encoder and the decoder, the performance of the code can be also evaluated. Due to the short cycles in the code and finite length of the code the actual rate of the code is degraded. To improve the actual rate of the code, the guessing algorithm is applied if the Belief Propagation is stuck. The actual rate of the code increases from 0.418 to0.44. The decoding complexity is not considered when the code is constructed.

Next in the second part, a regular LDPC code is constructed which has low decoding complexity. The code is generated based on the Gallager method. We present a new method to improve the performance of an existing regular LDPC code. The proposed method does not add a high complexity to the decoder. The method uses a combination of three algorithms: 1- Standard Belief Propagation 2- Generalized tree-expected propagation 3- Guessing algorithm. The guessing algorithm is impractical when the number of guesses increases. Because the number of possibilities increases exponentially with increasing the number of guesses. A new guessing algorithm is proposed in this thesis. The new guessing algorithm reduces the number of possibilities by guessing on the variable nodes which are connected to a set of check nodes. The actual rate of the code increases from 0.41 to 0.43 after applying the proposed method and considering the number of possibilities equal to two in the new guessing algorithm.

DEDICATION

This thesis is dedicated to my family for their love, endless support and encouragement.

To the memory of my beloved mom.

ACKNOWLEDGEMENT

First and foremost, I owe my deepest gratitude to my supervisor, Professor M. Reza Soleymani, to give me the opportunity to work in his group and the freedom to explore on my own. I am grateful for his support and encouragement throughout my research. Without his guidance and instruction, I would have not been able to complete this work.

I am grateful to all members of my Master's defense committee: Prof. A. Youssef, Prof. William E. Lynch, Prof. Yousef R. Shayan, Prof. C. Assi and Prof. R. Raut. I appreciate their valuable comments and all the time they spent reading this thesis.

I also want to thank all current and previous members of the Wireless and Satellite Communications Laboratory for their help and support. In addition, I wish to thank NSERC CRD as well as Interdigital Canada Ltd., and PROMPT. More than anyone else, I would like to thank my parents, whom, without their love and unlimited sacrifices, I would have not been where I stand at the moment and I only hope I have made their support worthwhile. Last but not least, I would like to express my gratitude to my husband, Amir, for all of his support and encouragement, and love.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Symbols

C	Channel Capacity
l_{avg}	LDPC average variable degree
r_{avg}	LDPC average check degree
$\rho(x)$	LDPC Right Degree polynomial
$\lambda(x)$	LDPC Left Degree polynomial
$R(\rho, \lambda)$	LDPC Design Rate
(λ, ρ)	LDPC normalized degree distribution pair from an edge perspective
Pr_l^{BP}	Probability of Erasure under BP algorithm after l iterations
$\varepsilon^{BP}(\lambda, \rho)$	LDPC maximum Tolerable Loss
$\delta^{BP}(\lambda, \rho)$	LDPC Multiplicative Gap
ε	Channel Erasure Probability
g	Size of the gap in the parity check matrix
H	Parity Check Matrix
k	The number of the message bits
H_{QC}	Parity Check Matrix of Quasi-Cyclic LDPC code
H_{RA}	Parity Check Matrix of Repeat Accumulate code
m	Number of Parity Check equations
n	Codeword Length
R	Achievable Rate

r	Channel output
S	Binary Source
v	Encoded message
w_c	Column weight
w_r	Row weight
$L(x)$	Likelihood of a binary variable
$L(x\|y)$	Conditional Likelihood
$\Pr(x)$	Probability of the variable x
ln L(x)	log-likelihood of a binary variable
m_{cv}^{l}	Message from check node c to variable node v in lth iteration
m_{vc}^{l}	Message from variable node c to check node v in lth iteration
$P_{l}^{BP}(\varepsilon)$	The probability that a message has erasure at the lth iteration
$\|\bar{k}\|$	The number of erased bits
Δ_N	The lower bound on the density of the parity check matrix of LDPC code
$\Delta(H)$	The density of the parity check matrix
$\overline{x_E}(\delta, \pi)$	The encoding complexity
$\overline{x_D}(\delta, \pi)$	The decoding complexity

List of Acronyms

AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BEMAC	Binary Erasure Multiple Access Channel
BP	Belief Propagation
DDP	Degree Distribution Pair
DE	Density Evolution
FDMA	Frequency Division Multiple Access
GTEP	Generalized Tree-Structure Expected Propagation
IRA	Irregular Repeat Accumulate
LDPC	Low Density Parity Check
MAC	Multiple Access Channel
MAP	Maximum a posteriori
ML	Maximum Likelihood
QC-LDPC	Quasi-Cyclic Low Density Parity Check
RA	Repeat Accumulate
TDMA	Time Division Multiple Access
TEP	Tree-Structure Expected Propagation

1.Chapter 1 Introduction

In wireless communications and mobile networks, channel access method allows several terminals connecting to the same spectrum or transmission medium. Channel access method has different types like FDMA, TDMA, and CDMA etc. For example, in frequency division multiple access (FDMA) a frequency spectrum is divided into several bands and each band is allocated to one user. Therefore, users use separated frequency bands and there is no interference between them. In time division multiple access (TDMA) the whole frequency spectrum is allocated to a user at each time and one user transmits and uses the whole channel at any time. Therefore, there is no interference between users. In all channel access methods, users transmit over the orthogonal channels, therefore, there is no interference between them. However to increase the bandwidth efficiency, users can transmit over the non-orthogonal channels. The existing channel can be exploited between sources and the destination. This channel is multiple access channel (MAC). This channel is not any more orthogonal. The MAC increases the channel capacity and the bandwidth efficiency and finally results in increasing the transmission rate. If the channel is noiseless, therefore the MAC is named Binary Erasure Multiple Access Channel (BEMAC). The BEMAC is a channel which is simple to analyze. After the emergence of the internet, the BEMAC is promoted onto the class of “real world” channel. To model data networks, binary erasure multiple access channel is used.

Achieving the capacity of the binary erasure multiple access channel is not possible without using forward error correcting codes. Turbo codes and low-density parity-check (LDPC) codes and also rate less codes like the Raptor codes are good candidates for the BEMAC. They have good performance and achieve rates near the channel capacity C. Due to the randomness of the LDPC codes and their simple and fast decoding and they receive a lot of attention nowadays and they are more popular than the Turbo codes. Also, the complexity of the Raptor code is higher than the LDPC codes and the LDPC codes are one of the main blocks of the Raptor codes. Therefore, the LDPC codes are selected in this thesis to evaluate their performance and encoding and decoding complexity.

There are two different types of LDPC codes: regular and irregular. The performance of the irregular LDPC codes is higher than regular ones. Although, the LDPC codes are popular and have the good performance over the BEC however, there are some disadvantages of the LDPC codes. One of the disadvantages of the LDPC codes is its encoding complexity which is not time linear. Also, there is a tradeoff between the performance of LDPC codes and their encoding and decoding complexity. In this thesis we investigate the LDPC codes in terms of the encoding and decoding complexity and also the performance.

In this thesis an irregular half rate LDPC code is constructed. The generated code has low encoding complexity, because the shape of the parity check matrix of the code is lower triangular. The actual rate of the code is less than one half, due to the limited length of the code and short cycles. The actual rate of the generated irregular LDPC code can be increased by applying the guessing algorithm. If we fix the number of guesses at one, then, the actual rate can increase from 0.418 to 0.44. Next, a regular Gallager half rate LDPC code with low decoding complexity is generated. The theoretical threshold of the code is 0.429. But, the actual rate is less than 0.429. In this thesis with proposing a new decoding method the performance of the regular LDPC code can be increased. By applying this method, the performance of the regular code increases from 0.41 to 0.43. The decoding complexity does not increase highly.

In this Chapter, we present the motivation, the problem statement, the literature review, the contribution of the thesis and finally the thesis outline. In the next Chapter, we present the background of the encoding and decoding of LDPC codes over the binary erasure channels. Also, in Chapter 2, the techniques to generate the LDPC parity check matrix and the decoding algorithms will be investigated. In Chapter 3 an irregular LDPC code will be constructed. The constructed code has low encoding complexity. To improve the performance of the code the guessing algorithm will be used after the decoding. In Chapter 4, the performance of a Gallager LDPC code will be evaluated. The code has low decoding complexity. A method will be presented in this chapter to improve the performance of a regular LDPC code without increasing the decoding complexity highly.

1.1 Motivation

In 1948, Shannon with his paper "A mathematical theory of communication" opened up a very important new field for modern digital communications, called information theory [1]. In his famous channel coding theorem, he showed that information can be transmitted reliably, i.e., with an arbitrarily small probability of error, across a given channel at any rate below the channel capacity. The construction of the practical capacity-achieving codes has been the main goal of the coding theory. Shannon analyzed the channel capacity for a single user scenario. However, a channel is usually share by more than one user. Actual communication systems are consisting many networks links. In 2-user MAC when one source is using the channel, another source as an interfering source can use the same channel. At the destination, messages of both sources can be detected and decoded correctly. Figure 1 shows the MAC in wireless network. In this channel each source has independent data to transmit to the destination. The destination receives two messages from two sources simultaneously. If we consider the channel noiseless then the channel is called binary erasure multiple access channel (BEMAC). Figure 1.2 shows system model for 2-users BEMAC.

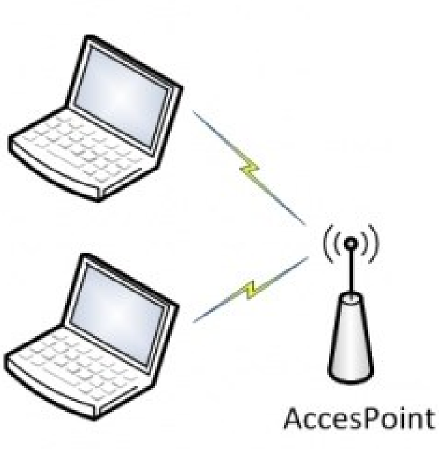

Figure 1.1. Multiple access channel

In 2 users-BEMAC, if each sources transmits equally likely binary data $\{0,1\}$, the destination received the combined stream. The combined stream is $\{0,1,2\}$ with probability of $\{0.25,0.5,0.25\}$ respectively. It means that if the source one sends 0 with probability of 0.5 and the second source also sends 0 with probability of 0.5, then the destination receives $(0+0=0)$ with probability of 0.25. When the destination receives $\{0\ or 2\}$, it knows that both sources have sent $0\ or\ 1$ respectively. But, if one source sends 0 and the other one sends 1 the destination receives 1 and it does not know which one sent 0 and 1. The destination considers this bit as erasure. On the average, half of the time the received message is erased or lost. To solve this problem, a code of half rate is required to determine erased bits in the destination. Shannon shows that the capacity of BEMAC is 1.5. Figure 1.3 shows Shannon capacity region of the BEMAC.

If the main or primary source sends at rate one and the other one as the secondary or interfering source sends at rate of half, then the corners of the capacity region is achieved. It means that the main transmitter should not change its transmission rate and also its transmission power. Therefore, it sends at rate one. The interfering source has to change its transmission rate and sends at half rate. Because, in BEMAC half of the time received steam is lost or erased. Therefore, if the secondary source encodes its data with a code of rate half, then the receiver can decode the received stream if half of it is erased. The receiver at destination has to detect two signals and decode each signal successfully. It uses successive decoding. In successive decoding, the receiver first decodes the message of the second transmitter with half rate coding, then substrates it from the received stream to determine the message of main source. Two corner of the capacity region are achieved by this scheme and the other points can be achieved with time sharing.

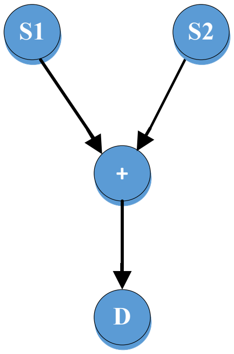

Figure 1.2. System model for MAC

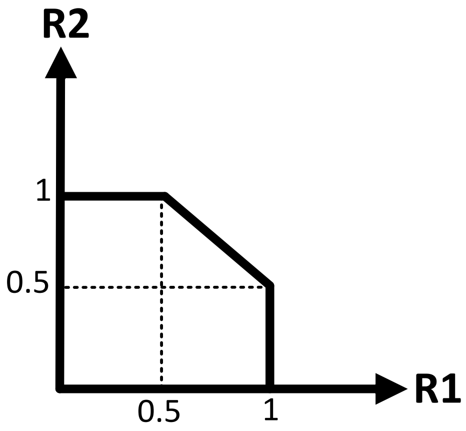

Figure 1.3. Capacity region for binary erasure multiple access channel

A lot of work has been done to achieve the Shannon capacity of MAC. Jabbari Hagh et al. in [2] showed that the capacity is achievable if one source encodes at rate one and the other one encodes at rate of one half using Rateless codes like Raptor code. The destination performs successive interference cancellation and the data for both sources can be recovered and also, two corners of the capacity region are achieved. Khoueiry in his thesis [3] proposed a new scheme for achieving the capacity. The proposed scheme uses joint decoding. In the proposed scheme two sources encode their data and joint decoding is used at the destination. Low-Density Parity-Check codes are used. Both sources can encode their data at any rate and different points of the capacity region are reached. If two sources use codes of half rate then the middle point of the capacity region is achieved. All of these works improve the capacity of the MAC to achieve near the Shannon capacity. Another way for achieving the Shannon capacity is to improve the performance of the code in the MAC. Low-Density Parity-Check codes are good candidate over the BEMAC, due to their good performance over the binary erasure channel.

1.2 Problem statement

Low-Density Parity-Check (LDPC) codes were introduced by Gallager in the early 1960’s [4]. At that time, computers could not simulate codes with large length. Hense, LDPC codes were not practical and they were forgotten for several decades. In 1990’s they were rediscovered by D. MacKay and Neal [5]. Due to the good performance and simple and fast decoding, they received a lot of attentions. LDPC codes show good properties over the Binary Erasure Channel (BEC). Therefore, they can be good candidate for BEMAC and achieve near the Shannon capacity. LDPC codes utilize iterative decoding algorithms [6]. This class of algorithms is named message passing algorithms. One of the important classes of these algorithms is the belief propagation algorithm (BP) [6]. BP is a suboptimal decoding procedure, but, approximates near the maximum likelihood decoding [6]. LDPC codes are usually easy to decode due to sparseness of their parity check matrices. However, due to the randomness of their parity check matrices their encoding is complex. Also, there is a tradeoff between complexity of decoder and performance of the LDPC codes.

A significant research has been done for designing LDPC codes with good performance. The objective of these works is to determine the pair distribution (λ, ρ) which yields the best performance. These codes are known as performance-optimized codes [7], [8]. The problem with these codes is that their decoding complexity. The decoding complexity increases because the number of iterations for the convergence of the decoder is large. For some applications when real time decoding is needed, decoder would stop after a defined number of iterations. Thus, the decoder cannot get to the maximum achievable rate. On the other hand, a part of the research has been done to design low complexity LDPC codes. These codes are denoted by complexity-optimized codes [9]-[10]. All these optimizations have been done to design a desired LDPC code that achieves the best tradeoff between complexity and performance.

Some works have been done to reduce the complexity of decoder or increase the speed of decoding for a given code by improving the iterative algorithms. In [11] Layered Belief Propagation L-BP algorithm has been proposed. In [11] standard Belief propagation has been modified. In this algorithm the check nodes are divided into subgroups called layers and each iteration is broken into multiple sub-iterations. It has been shown that the convergence for decoding LDPC codes improves by using a simple and efficient layering strategy.

Authors in [12] took a different approach. Instead of trying to find a good degree distribution, the performance of an existing code have been improved over the binary erasure channel (BEC). In [12] for the first time, the performance of an existing code was improved by guessing on unknown variable nodes for short-length LDPC codes. Authors in [12] proposed three algorithms, algorithm A is the same as the standard belief propagation. In algorithm B, if algorithm A fails, it makes some assumption on some of the erased bits, check-sum determines if guesses are correct or not. Algorithm B guesses on the variable nodes with higher degree. The drawback of this method is that the complexity of decoder grows exponentially with increasing the number of guesses and there is a limitation on the number of guessing variable nodes and also it has the probability of error greater than the maximum likelihood. For reducing the complexity and improving the probability of error, they proposed algorithm C. In algorithm C, the decoder defines a set of equations as basic equations and if and only if the set of basic equations have a unique solution then the received codeword is maximum likelihood decodable.

When iterative algorithms like Belief Propagation are used for decoding of LDPC code, density evolution is used to determine the performance of LDPC codes over BEC [14]. density evolution uses asymptotic analysis that assume that the Tanner graph of a LDPC code is cycle free and also code length is infinite [14]. The actual rate of a LDPC code is lower than the Maximum a posteriori (MAP) decoder, due to the cycle in the Tanner graph and also the LDPC code length being finite. In [13] Maxwell decoder is presented to achieve MAP capacity when BP gets stuck i.e. when there is no more check nodes of degree one and there is still erased bits in the codeword. In this situation, Maxwell decoder makes assumption on one or more remaining erased bits, until a check node of degree one is created. Then, BP runs. The process of guessing is repeated until all the erased bits are recovered successfully. The check sums determine whether the assumptions made were correct or not. If check sums are zero, then our assumptions are correct otherwise the decoder has to make another assumption. Maxwell decoder is not practical because the complexity of decoder grows exponentially with increasing number of guesses. Maxwell decoder is a powerful tool to derive the LDPC codes MAP capacity and its performance [21].

Authors in [15] proposed Tree-Structure Expected Propagation (TEP) algorithm. TEP works as Maxwell decoder. But, its complexity is the same as BP algorithm. TEP in each iteration removes one check node of degree two and one of the variable nodes connected to it. If two variable nodes connected to a check node of degree two were also connected to a check node of degree three, then a check node of degree one is released. Then, BP can continue decoding. In [16], Authors proposed Generalized TEP (GTEP) algorithm. GTEP removes one check node and one variable node in each iteration. TEP is a special case of the GTEP. In this paper [16], the authors proposed that at the beginning it is better to put some constraints on the structure of the matrix to improve performance of GTEP decoder.

The objective of a lot of research on LDPC codes is either finding a good pair distribution that achieves better performance and achieves near Shannon capacity with complexity as low as possible or try to improve the performance of existing codes without adding higher complexity. The complexity of the LDPC codes is the sum of the complexity of encoder and decoder. The complexity of LDPC decoder is related to the number of ones in the parity check matrix. The complexity of LDPC encoder is related to the gap in the parity check matrix [17]. In [17] greedy algorithms for transforming the parity check matrix to a lower triangular matrix are proposed.

Related works

1.3.1 LDPC encoder

Low-Density Parity-Check (LDPC) codes received a lot of attention due to the fast and simple decoding. LDPC codes have good performance with small probability of error. The problem of the LDPC codes is their encoding complexity. A LDPC encoder has complexity quadratic in the block length. It means for a code of length n, the encoder has a complexity of n^2. However, Turbo codes can be encoded in linear time. A lot of work has been done for reducing the complexity of the encoder.

In [18] and [19] instead of using bipartite graph, they use cascade graph. In this method each stage is cascade to the next one and each stage acts like a small code which the size of these sub codes is considerably smaller than the overall code. According to the density evolution, the performance of the code degrades if the code length decreases. The drawback of this method is reducing the performance of the overall LDPC code, but, results in the real time encoding.

For decreasing the encoding complexity, the parity check matrices in LDPC codes have to be lower triangular. Authors in [20] proposed a new method for generating parity check matrices that are lower triangular. In this method, for generating a parity check matrix, two constraints have been applied. One of them is the degree constraint and the other is the constraint for having a lower triangular matrix. Generally, this method results in performance reduction.

In [22] proposed iterative encoding. The proposed algorithm is based on an iterative matrix inversion technique. The proposed algorithm can find the value of parity check bits if and only if $(H_P \oplus I)^{\ k} = 0$. A parity check matrix which satisfies this condition can be used in this method. This method can results in loss of performance in general.

Richardson et al. [17] proposed greedy algorithms for making an existing parity check matrix to a lower triangular matrix. Richardson proved that greedy algorithms don not change the degree distributions and just transform a matrix to lower triangular. Greedy algorithms with column and row permutation change the parity check matrix to lower triangular. In [17], authors showed that the complexity of encoder for a LDPC matrix with gap of g is $\mathrm{O}(n + g^2)$. They proved that the minimum achievable gap for a regular a $(3,6)$ LDPC code is $0.017n$, n is the code length. They proved that the expected gap is of order less than $\sqrt{n}$ which results in real time encoding, because the encoding complexity is $\mathrm{O}\left(n + \sqrt{n}^2\right) = \mathrm{O}(2n)$.

1.3.2 LDPC decoder

Low density parity check (LDPC) codes constitute a class of the powerful codes. Based on traditional sum-product and max-product algorithms, various modified algorithms are used to improve the performance of LDPC codes in terms of error rate, complexity and latency. Large size of LDPC codes leads to large complexity in both encoding and decoding LDPC codes. This is why LDPC codes were ignored for a long time.

Tanner graph were introduced to describe linear block codes [21]. The graphical representation such as factor graphs promotes the trend of iterative processing in signal processing [18]. The decoder will pass messages between variable nodes and check nodes iteratively. Iterative decoder decreases the complexity and makes the implementation of LDPC codes practical. For LDPC decoding when hard decision is applied, decoder will decode the codeword iteratively until a legal codeword is found. LDPC decoder will not guarantee the result is the true codeword that was sent from the transmitter. But it will make sure it is a legal codeword.

Message passing algorithm is an iterative algorithm and is a powerful way to compute the marginal probabilities in a graph. Good LDPC codes should avoid short cycles because short cycles will lead to bad performance. When the factor graph is cycle-free, message passing algorithm is guaranteed to converge and offer an optimal result. However, when the graph contains cycles, it may converge to a local optimum or even fail to converge [19].

The two main message passing algorithms are sum-product algorithm (or belief propagation algorithm or probability propagation algorithm) and max-product algorithm (or min-sum algorithm). Sum-product decoder is an iterative process and aims at computing the sum-marginal. In the message passing algorithms, messages are often computed in the logarithmic domain. Max-product decoder (MPD) aims to compute the Max-marginal.

Another way for decoding the LDPC code is linear programming. The goal of linear programming decoder is to find the maximum likelihood codeword [20]. The complexity grows exponentially when the degree of check nodes increases. It is too high to implement for large size LDPC codes. It is optimal for small length codes.

1.3 Contribution of the thesis

In thesis, we consider all challenges in constructing an LDPC code i.e., the complexity of encoder and the decoder and also their performance. We generate regular and irregular LDPC codes. Since an irregular LDPC code has a better performance, an irregular half rate LDPC code with low encoding complexity is generated. To have low encoding complexity LDPC code, the parity check matrix of the code must be in the lower triangular shape. In this thesis we propose a method to generate a lower triangular matrix. The method keeps the density of the LDPC parity check matrix uniform. The gap in the lower triangular matrix can be any desired value. In the method three constraints in constructing the LDPC code is applied. One constraint is the degree distribution and another one is for the gap. The last constraint is for the density of the parity check matrix. The proposed method considers all these constraints and the constructed code has low encoding complexity and good performance. The low complexity encoder and the decoder are implemented. The performance of the code is evaluated and to increase the performance of the code the guessing algorithm is added. We apply the guessing algorithm on this code and investigate the performance improvement. Therefore, a half rate LDPC code with low encoding complexity and good performance is constructed. Next, we want to generate a half rate code which has low encoding complexity and improve the performance of the code. The ensemble $(3,6)$ is selected. This ensemble has the best performance and the lowest complexity among the other ensembles.

In this thesis we generate a regular LDPC according to the ensemble $(3,6)$. The parity check matrix of this code is not lower triangular. Hence, the complexity of encoder is not low. However, the complexity of the decoder is low which and the theoretical threshold of the code is 0.429. The Performance of the code is lower than irregular LDPC codes. We propose a new method to improve the performance of the existing regular LDPC code. The proposed method improves the performance of existing LDPC codes without increasing the decoding complexity dramatically. It has been done by applying three decoding algorithm efficiently which results in the performance improvement while keep the decoding complexity low. Applying GTEP and guessing algorithm can improve the performance of standard BP. The complexity of GTEP is the same as iterative decoding. However, the complexity of the guessing algorithm is not as low as BP and GTEP. The complexity of the guessing algorithm increases with increasing the number of guesses. Running the GTEP before the guessing algorithm decreases the complexity of the guessing algorithm. In this thesis to reduce the complexity of guessing algorithm, instead of guessing on any unknown random variable node, the decoder guesses on variable nodes connected to a check node. The number of possibilities reduces by half.

1.4 Thesis outline

In Chapter 2 we review the required background material. In this chapter, the different representation of the LDPC codes and various method of constructing these codes are studied. The methods of decoding from the iterative decoding algorithms to the maximum likelihood decoding are discussed. Next, the performance of the iterative decoding algorithms is evaluated.

In Chapter 3, after investigating parameters which affect the encoder complexity, we propose a method to generate an irregular half rate LDPC parity check matrix that is lower triangular. In the proposed methods in addition to the degree distribution constraint, two other constraints are applied to have a lower triangular shape matrix and keep the density of the matrix uniform. The performance of the generated matrix is evaluated in this chapter. Also, to improve the performance of the code the guessing algorithm is applied. Simulation results are presented in this chapter.

In Chapter 4, a regular half rate LDPC code is generated. The LDPC code has low decoding complexity. We present a new decoding algorithm that increases the actual rate of an existing

LDPC code. The main advantage of the proposed scheme is that it improves the performance of an LDPC code without changing the degree distribution and increasing decoding complexity considerably. The proposed algorithm uses a combination of three algorithms: 1- standard belief propagation 2- Generalized tree-expected propagation 3- guessing algorithm. If the decoder cannot recover the erasure in the received codeword at the first step of the algorithm, then the next step is run, until, all the erased bits are solved. The guessing algorithm at the third step increases the decoding complexity. Therefore, in this chapter some ideas to improve the actual rate and reduce the complexity of the guessing algorithm is investigated.

Finally, in Chapter 5, we conclude our work and offer suggestions for future research.

2.Chapter 2 Background

2.1 Introduction

In this chapter we will discuss about the background material required in the rest of thesis. First we will overview different presentation of Low-Density Parity-check (LDPC) codes. Then we will take a look at the structure of LDPC codes which is needed in constructing an LDPC code and also we talk about ways to construct an LDPC code. Then, we will talk about the LDPC decoding algorithms. Since LDPC codes use message-passing algorithms, the analysis of their performance is different from the linear block codes. Therefore, the performance and the analysis of the LDPC codes will be presented and at the end we will conclude the chapter.

2.2 The Representation of LDPC codes

In this section, first we talk about matrix, graphical and polynomial representation methods of LDPC codes. The advantages of the LDPC codes are also presented. These representation concepts help in the designing LDPC codes and analyzing the performance of the code.

2.2.1 The Matrix Representation

Linear channel error correction codes are expressed by both the generator matrix G and the parity check matrix H, since there is:

$$G.H^T = 0 \tag{2.1}$$

There are some linear block codes which are defined just by parity-check matrix H. One important code of this class is Low-Density Parity-check (LDPC) codes. LDPC codes are specified by the parity check matrix H. The H matrix should be very sparse, i.e., the number of ones or nonzero elements in the parity check matrix H should be much smaller than the total elements in the H matrix. Because of this, this class of linear block codes are named Low-Density Parity-check codes. The dimension of the parity-check matrix H is $m \times n$. Where, n is the length of the codeword and m is the number of parity bits. The H matrix has n columns and m rows. The design rate R of the code, which also called design rate, is:

$$R = \frac{n-m}{n} \tag{2.2}$$

In this thesis, the field is considered Galois field. Therefore, the elements in the H matrix are either 0's or 1's. A codeword in linear codes is the null space of the parity check matrix H:

$$\vec{v}H^T = 0 \tag{2.3}$$

Where $\vec{v} = [v_1, v_2, \dots, v_n]$ is a $n - tuple$ codeword and $v_i \epsilon \{0,1\}$. In every Gallager LDPC code, the parity check matrix H has the following structural properties:

1- Each row consists of ρ ones.

2- Each column consists of λ ones. Properties 1 and 2 determine degree distribution of LDPC codes.

3- The number of ones in common between any two columns is no longer than 1. This property guarantees that the matrix is cycle free.

4- The code is random and has no structure.

5- The length of LDPC codes is much larger than ρ and λ. This property ensures the sparseness of H.

According to the definition of the parity check matrix, there is a cycle in the parity check matrix when the number of ones in common between any two columns is greater than one. Cycles in the LDPC codes are destroying and cause the degradation in the code performance. We will explain the cycle in LDPC codes by using the graph representation in the next section.

2.2.2 The Bipartite Representation

Tanner for the first time represented an LDPC code by using bipartite graph in 1981 [21]. After that the representation of the LDPC codes using bipartite graph is called the Tanner graph. A Tanner graph is used to demonstrate the iterative decoding process of an LDPC code.

A Tanner graph is composed of a set of nodes or vertices and a set of edges. The nodes are grouped into two subgroups: variable nodes and check nodes. Edges are used to connect nodes of these two subgroups together. An edge can only connect two nodes of two different subgroups in the Tanner graph. When two nodes are connected by an edge in the Tanner graph, we say that this edge is incident with these two nodes. The degree of a node is the number of edges that are connected to it. The Tanner graph can be derived from the parity check matrix H with m rows and n columns easily. The graph can be induced by using the following rules:

1- The m rows corresponding to the set of parity check constraints form the m check nodes (or check sum vertices), denoted by $c_1, \dots, c_m$ while the n columns corresponding to the codeword bits form n variable nodes (or code bit vertices), denoted by $v_1, \dots, v_n$.

2- There is an edge between a check node and a variable node if and only if the element in H is equal to one.

According to the above rules, we can obtain two conclusions. The first conclusion is that the degree of a check node (or variable node) is equal to its corresponding row (or column) weight. The second one is that there is at most one edge between any two nodes. A cycle in a Tanner graph is referred to as a closed loop. In the Tanner graph, the length of a loop is the number of the edges in the loop. The length of the shortest cycle in the graph is called the graph' girth. In the LDPC codes the cycle of length four is avoided strongly.

Based on these rules, the Tanner Graph of the following matrix H can be induced, which is shown in Figure 2.1. In the Tanner graph, the variable nodes are shown by circles and the check nodes by squares.

This example shows a Gallager LDPC code. The number of ones in each row and column is four and two, respectively. Therefore, the rate of the code is half. In Figure 2.1, the four green edges indicate a cycle. In fact, this cycle is four which is the shortest cycle. Therefore, the girth of the graph is four. The girth plays an important role in an LDPC code. The girth affects the performance in the iterative decoding algorithms. In constructing LDPC codes, large girths are always desired. The role that the girths play in the LDPC codes will be discussed in detail when we describe the Belief Propagation.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (2.3)$$

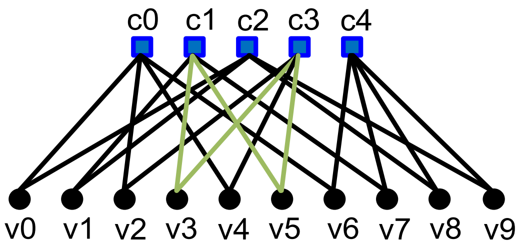

Figure 2.1. The Tanner graph for the parity check matrix of (2.6)

2.2.3 The Degree Distribution Polynomial

Another representation of the LDPC codes is the degree distribution polynomial. The degree distribution polynomials were introduced by Richardson to represent an ensemble of LDPC codes combined with the Tanner graph [31]. The degree distribution polynomial is used to specify the degree distributions of the variable nodes and check nodes in Tanner graph or the H matrix by using the following format [14], [31-32]:

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i \,.\, x^{i-1} \text{ for variable nodes} \tag{2.4}$$

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i \,.\, x^{i-1} \text{ for check nodes} \tag{2.5}$$

Where $d_v$and d_c are the maximum degrees of the variable nodes and check nodes respectively; λ_i and ρ_i denote the fraction of all edges incident to variable nodes with degree i and check nodes with degree j. Based on a pair of degree distribution polynomials and a given code length, we can calculate some parameters of this given LDPC code. We can see that the degree distribution polynomials describe an ensemble of LDPC codes, but not a specific LDPC. However, this definition is very helpful in expressing a code's structure and in generating an LDPC code, which will be demonstrated in the next section. However, the Tanner graph and the H matrix describe a specific LDPC code.

2.3 Construction of the LDPC codes

The selection of a particular Tanner graph or a parity check matrix H from the ensemble is an issue in constructing a good LDPC code. At a particular block lengths and degree distribution pair, certain Tanner graphs (or certain parity check matrices H) have the best performance among all the other graphs. Due to the distribution of edges in the bipartite graph or ones in the parity check matrix which results in the larger girth. Thus, the problem of the code construction in the LDPC codes is choosing a Tanner graph (or a matrix) among all the possible Tanner graphs (or matrices). The selected graph must satisfy all the constraints of the code and also provides a good performance under iterative decoding like belief propagation algorithm.

Several approaches to constructing a good LDPC code have been proposed. It is worth to mention that the related graph to a good LDPC code should have large girth and fewer cycles and fewer stopping set. We will introduce some methods for constructing LDPC codes in this section. In the next chapter we will talk about the proposed constructing method.

2.3.1 Pseudorandom codes

2.3.1.1 Gallager codes

Gallager in his thesis proposed LDPC codes in the 1960s [4]. Gallager in his thesis proposed a general method to construct pseudo-random regular codes. Also, he investigated the performance of LDPC codes. In his thesis he just talked about regular LDPC codes in which each row has ρ ones and each column has λ ones. For constructing a regular LDPC code, he proposed to construct sub-matrices $H_1, H_2, \dots, H_\lambda$.

$$\begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_\lambda \end{bmatrix} \tag{2.6}$$

The sub-matrices have the following structure properties: 1- every row of each sub-matrix has ρ ones but every column of each sub matrix has a single one. 2- The number of ones in each sub-matrix is: $\rho \times \frac{m}{\lambda}$. 3- The other submatrices are the column permutations of the first sub-matrix H_1. 4- In the first sub-matrix H_1, for $1 < i < \frac{m}{\lambda}$, the ith row of H_1 contains ρ ones in columns $(i-1)\rho + 1$ to $i\rho$. For example, the ensemble $(12{,}3{,}6)$ is given; the H matrix is given as follow:

$$H = \left[\begin{array}{cccccccccccc} \hline 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}\right] \tag{2.7}$$

The parity-check matrix generated by the above rules is called Gallager parity-check matrix or Gallager code. The Gallager construction method does not purposely avoid forming the cycles of length four. Therefore, the Gallager matrix may contain these kinds of cycles which will severely degrade the performance of iterative decoding. In order to improve the performance of a Gallager code, we should try to eliminate the cycles with length four, i.e., to avoid more than one 1s in any two rows or two columns in the parity-check matrix H when constructing H.

2.3.1.2 Mackay codes

Another class of pseudo randomly LDPC codes are Mackay codes, which are presented by Mackay in 1997 [5]. In the Mackay method for generating LDPC codes the parity check matrix is constructed column by column. New columns with appropriate weight randomly generated and added to the matrix until appropriate matrix with predefined row distribution is constructed. If the desire matrix is not generated, then the whole part of the matrix or partially is reset and the process restarted. In addition to the row distribution, another constraint has to be checked. The constraint is the cycle of length 4. Also, at each column placement the short cycles has to be checked. The complexity of generating codes increases as longer cycles are considered.

2.3.2 Random codes

Mackay and Gallager method for generating LDPC codes are not fully random. Also, construction of the code is based on the parity check matrix. The random code construction approach has been presented in [14] and [23]. In the random method is based on the bipartite graph. In the random process, an appropriate number of sockets for each variable node and check node are set. A random interlivear determines the connection between two types of sockets (variable node and check node). Finally, the graph has to be checked. Checking the graph is performed to ensure that randomly graph satisfies the basic constraints for designing LDPC codes. One of constraints is that there should be at most on connection between any variable node and check node pair. Another constraint is to check the cycles of short length. If any of the constraints is not satisfied the graph is reset and reconstructed. The first constraint grantees that the graph satisfying the design rules and parameters. The second constraint improves the structural properties and the performance of the code.

2.3.3 Structured LDPC codes

2.3.3.1 Quasi-cyclic LDPC Codes

Quasi-cyclic LDPC codes have the main characteristics of both cyclic codes and low-density parity-check (LDPC) codes. The main characteristic of the LDPC code is their performance and for the Quasi-cyclic codes is their structured which results in low encoding complexity. Therefore, Quasi-cyclic LDPC codes achieve good performance while exploiting the structure of Quasi-cyclic codes for reducing the encoding and decoding complexity.

Quasi-cyclic LDPC codes are a particular class of Quasi-cyclic codes characterized by parity-check matrix H while the H matrix is sparse. Every row in the parity check matrix of Quasi-cyclic (QC-) LDPC codes is t circular shift of the previous row [24]. The implementation of the encoder is based on the shift register and results in the low-complexity encoding [33] [34]. Therefore, the encoding complexity is related linearly to the block length of the code. In [35]-[37] demonstrate the algebraic method to construct QC-LDPC codes.

The structure of the QC-LDPC parity-check matrix H_{QC} is as the following:

$$H_{QC} = \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,n'} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,n'} \\ \vdots & \vdots & \ddots & \vdots \\ H_{m',1} & H_{m',2} & \cdots & H_{m',n'} \end{bmatrix} \qquad (2.8)$$

According to the equation (2.7), the parity check matrix H_{QC} is consist of submatrices H_{ij}, $i\,\epsilon\{1,2,\dots,m'\}$ and $j\epsilon\{1,2,\dots,n'\}$. Each submatrix is either a circulant permutation matrix or a null matrix.

2.3.3.2 Repeat Accumulate Codes

Repeat Accumulate (RA) codes were discovered by Divsalar *et al.* in 1998 [30]. It is a class of low-density parity-check codes. The construction of the RA code is based on the parity-check matrix H. To have a good insight about RA codes, it is better to start from the encoder of the RA code. The encoder of the RA codes works as follow:

1- The encoder take k source bits.

$$s_1 s_2 s_3 \dots s_k,$$

2- Repeat each bit k' times, where $N = k'k$ bits.

3- Permute

4- Accumulate and then transmit.

The form of the parity check matrix H is like:

$$H_{RA} = [H_1 \; H_2], \tag{2.9}$$

Where H_2 is the dual-diagonal matrix with one column of weight one:

$$H_2 = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & 1 & \ddots & & \\ & & \ddots & 1 & \\ & & & 1 & 1 \end{bmatrix}, \tag{2.10}$$

The reason for the structure of H_2 is for the accumulator in the encoder. There are two types of RA codes: 1- regular RA codes, 2- Irregular RA codes. The H_1 matrix determines that the code is regular or irregular. In the regular RA code, the H_1 matrix is low density and all columns have the same weight and the weight is equal to the repetition of the sequential encoder. Also, the weight of all rows is one. For the irregular version of the regular RA which named the Irregular Repeat Accumulate (IRA) class of codes, the column weights of H_1 vary and correspond to a variable repetition code and row weights correspond to the combined inputs to the accumulator represented by H_2. In both case, the position of the entries in H_1 define the interleaver in the sequential view of the code. Figure 2.2 shows the block diagram of systematic encoder of RA codes. According to figure 2.2, an encoder consists of a repetition R, interleaver and combiner C and generaor polynomial $\frac{1}{1+\mathrm{D}}$. The repetition R defines the column and row weights in the H matrix. The pair distribution $\lambda(\mathrm{x}), \rho(\mathrm{x})$ defines the repetition R and combiner C. According to the row weight of the matrix H_1, the bits emerging from the interleaver are combined with the combiner C. The current input and the previous output of that block are simplified with the generator polynomial $\frac{1}{1+\mathrm{D}}$.

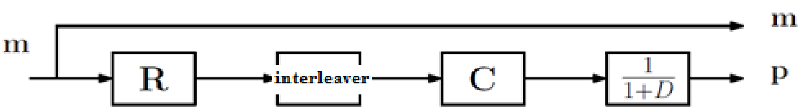

Figure 2.2. The schematic of the encoder of the RA Codes

2.4 LDPC codes structures

In this section, we talk about the structure of Low-Density Parity-Check (LDPC) codes. There are two types of LDPC codes: regular and irregular LDPC codes. Gallager's LDPC codes are referred to as regular LDPC codes because of their regular structures in the parity-check H matrices. In the regular LDPC code, the degree of each check node or row is ρ and the degree of each variable node or column is λ. The total number of ones in the H matrix or the number of edges in the Tanner graph is E, and there is:

$$E = n.\lambda = m.\rho \rightarrow m = \frac{n.\lambda}{\rho} \qquad (2.11)$$

Since, the design rate of a linear code is:

$$R = 1 - \frac{m}{n} \qquad (2.12)$$

Therefore, by substituting (2.8) in (2.9) the code rate R can be computed as:

$$R = 1 - \frac{\lambda}{\rho} \qquad (2.13)$$

R is referred as design rate. The rows of the H matrix are considered linear independent. Usually, independencies among rows of the H matrix are not possible. Therefore, the actual rate is lower than the design rate.

The ensemble of a regular LDPC code is described as (n, λ, ρ). Where n is the length of the code and λ and ρ are the column and row weight, respectively. For example a $(n, 2, 4)$ LDPC code refers to a code with variable nodes of degree 4 and check nodes of degree 2. The design rate of this code from (2.10) is $\frac{1}{2}$. In asymptotic analysis of the LDPC codes, if n is large enough, the average behavior of almost all instances of this ensemble concentrates around the expected behavior [12]. Although regular LDPC codes show good performance over the binary erasure channels (BEC) but still they show a larger gap to capacity than Turbo codes. The main advantage of regular LDPC codes over turbo codes is their better "error floor" and their simple and fast decoding.

Another type of LDPC codes is irregular LDPC codes. If the degree of check nodes and variable nodes are not fixed any more, the structure of the LDPC code is called irregular LDPC code.

Luby et. al showed that the capacity of the irregular LDPC codes reaches more close to the Shannon capacity than the regular one [32]. By designing the irregular LDPC codes carefully, LDPC codes perform very close to the capacity. An ensemble of irregular LDPC codes is defined by the degree distribution of its variable nodes $\{\lambda_1, \lambda_2, \dots, \lambda_{d_c}\}$ and check nodes $\{\rho_1, \rho_2, \dots, \rho_{d_v}\}$. Where λ_i denotes the fraction of edges incident on variable nodes of degree i and ρ_j denotes the fraction of edges incident on check nodes of degree j. Another way of describing the ensemble of the irregular LDPC code is by the degree distribution polynomial using the equations (2.4) and (2.5). The Tanner graph of the irregular LDPC code is presented in the terms of the fraction of edges of each degree. In this thesis to define an irregular LDPC code, a variable (check) distribution means a variable (check) edge degree distribution.

Similar to regular codes, it is shown in [42] that the average behavior of almost all instances of an ensemble of irregular codes is concentrated around its expected behavior, when the code is large enough. Also, the expected behavior of the ensembles converges to the cycle-free case. The number of edges in the Tanner graph or the number of ones in a parity-check matrix H of an irregular LDPC code is E and there is:

$$n = E \sum_i \frac{\lambda_i}{i} = E \int_0^1 \lambda(x) dx \tag{2.14}$$

$$m = E \sum_i \frac{\rho_i}{i} = E \int_0^1 \rho(x) dx \tag{2.15}$$

Therefore, the design rate of an irregular LDPC code is achieved by substituting (2.14) and (2.15) in (2.13):

$$R = 1 - \frac{m}{n} = 1 - \frac{\int_0^1 \rho(x) dx}{\int_0^1 \lambda(x) dx} \tag{2.16}$$

As far as the performance of the irregular LDPC codes is better than the regular codes, a lot of researches have been done to find irregular LDPC codes which have the best performance. Finding a good asymptotically long family of irregular codes is equivalent to finding a good degree distribution. A lot of researches have been done for finding the best degree distribution of the LDPC codes over the binary erasure channel [25] and [38].

2.5 Decoding of LDPC codes

Decoding over the binary erasure channel (BEC) is a process in which a decoder makes a decision on the erased bits to find a codeword that minimizes the probability of error. It means that the decoder chooses a codeword that maximizes a posteriori probability (MAP) which is called MAP decoding. A MAP decoder tries to find a codeword based on the received codeword r such that [29]:

$$\max_{\mathrm{v_j \epsilon V}} \mathrm{Pr}\{\mathrm{v_j}|\mathrm{r}\} \tag{2.17}$$

In the random codes like Low-Density Parity-Check (LDPC) codes, the code length is large. The size of the code set $|V|$ grows exponentially with the size of code length. Therefore, searching for a codeword is practically impossible in the LDPC codes with the large code length. Thus, another way to find the most likelihood codeword is needed. First we will discuss the maximum likelihood decoding of the LDPC codes over the BEC.

2.5.1 Maximum Likelihood decoding of LDPC codes over the binary erasure channel

Considering that the codeword v is sent and r is received then the maximum likelihood decoder chooses a codeword from a set of codewords which maximizes the following probability:

$$\Pr(v|r) \tag{2.32}$$

The maximum likelihood decoding tries to find the closest codeword to the received message. The maximum likelihood decoding achieves the MAP solution [39]. The set of codewords for LDPC codes is large, due to the large length of the codeword. Therefore, searching for finding a codeword takes time and the decoding is not time linear.

If the transmitted codeword is $v = (v_1, \ldots, v_N)$ and the received message is $r = (r_1, \ldots, r_N)$ where $v_i \in \{0,1\}$ and $r_i \in \{0,1,e\}$. e denoted erasure. Then there is [39]:

$$H_k . {v_k}^T = H_{\bar{k}} . v_{\bar{k}}^T = H_k . {r_k}^T = z^T \tag{2.33}$$

Where k is the set of known bits in r, $k = \{i: r_i \neq e\}$. Similarly, $\bar{k}$ is the set of erasures which $\bar{k} = \{i: r_i = e\}$. H_k and $H_{\bar{k}}$ corresponding to the columns of H which are known and unknown respectively [39]. z is the length of known bits. Maximum Likelihood (ML) over the BEC sums up to solve the above linear system. If the probability of channel erasure is ε, then according to the weak law of large number:

$$|\,\bar{k}| = N(\varepsilon + o(1)) \tag{2.34}$$

Therefore, $|\bar{k}|$ is the number of erased bits. If and only if the columns of $H_{\bar{k}}$ are linearly independent then, ML will have a unique answer for the equation (2.34). The equation is a linear system with $|\bar{k}|$ variables. In the maximum likelihood decoding, the decoder to solve the equation (2.34) uses Gaussian elimination. Totally, the complexity of solving the equation (2.34) is equal to:

$$((1 - R)\beta + \gamma\delta)\, \varepsilon^2 N^3 \tag{2.35}$$

The value of β and γ are chosen, according to the algorithm is used to solve the equation. The first method that reduced the number of operation to perform Gaussian elimination was proposed by Stassen. According to the Stassen method, the number of required operation is $O(N^{2.81})$ operations [40]. Another method which is the fastest method and impractical is presented in [41] and requires $O(N^{2.376})$ operation to perform the Gaussian elimination. Therefore, ML is impractical for LDPC codes, due to the large length of the codes.

In [39] proposed an algorithm for reducing the complexity of ML decoding for LDPC codes over the BEC. In the proposed algorithm the complexity of ML decoding remains $\mathrm{O}(\mathrm{N}^3)$. The constants are while significantly reduced and the proposed method is a practical method. In [39] a simple practical probabilistic algorithm is presented for efficient ML decoding of LDPC codes over the BEC. Generally, these algorithms to perform Gaussian elimination can be views as the standard iterative decoding algorithm.

The iterative decoding algorithms like BP can be reinterpreted as a Gaussian elimination procedure. In the iterative algorithms, in each iteration one column of the parity check matrix is left with a nonzero entry, like the Gaussian elimination procedure. BP performs the Gaussian elimination in which we only process columns that have at least one connected row of degree one, i.e., a row with a single nonzero entry. TEP and GTEP also perform Gaussian elimination [16]. The TEP accounts for rows of degree two and the GTEP is able to process any column, no matter the degree of the connected rows [16].

Iterative algorithms are the best candidate for the decoding of LDPC codes. Gallager in his thesis [4] proposed several iterative decoding algorithms for LDPC codes over the binary erasure channel (BEC). The proposed algorithms are message-passing algorithms. In the message passing algorithms messages pass iteratively between nodes through the edges in the bipartite graph. The message can be the probability of being a symbol. For example in the Galois binary field $GF(2)$, symbols are 0 or 1, then the messages through the edges are the probability of being 0 or 1. MacKay and Neal [5] rediscovered LDPC codes over the Additive White Gaussian Noise (AWGN) [5]. They proposed Belief Propagation algorithm which is sum-product algorithm for the decoding. They showed that BP reaches the same result as the MAP decoder when a code has no short cycles in the bipartite graph and received symbols are independent of each other. Since, the parity check matrix H is sparse the iterative decoding algorithms reduce the decoding complexity. We will talk about the BP algorithm. Iterative algorithms are the best choice for the LDPC codes decoding.

2.5.2 Message passing algorithm

One class of iterative decoding is Message Passing algorithm. Message passing algorithm uses the structure of the Tanner graph. In the message passing algorithm the messages pass from variable nodes to check nodes and from check nodes back to variable nodes. Variable nodes calculate the message based on the values they observed and the message passed from their adjacent check nodes. In the algorithm, the message that is sent from the check node c with degree i to the variable node v with degree j through the edge e at lth iteration calculates as follow: the message is the summation of the messages came from the adjacent variable nodes to the check node c through the edges other than e in the previous iteration. Then the variable nodes sent their message to the check nodes. Iterations continue until the variable nodes reach the fixed point or after a defined number of iterations.

2.5.3 Belief Propagation algorithm

One of the important subclass of the message-passing algorithm is Belief Propagation (BP) algorithm. BP supposes that the Tanner graph is tree. It means that there is no cycle in the graph or rows are linearly independent. When the graph is tree, BP calculates the exact marginal probability. If the Tanner graph is not cycle free, BP cannot calculate the exact marginal probability and it approximates maximum likelihood decoding. In each iteration the message sent from a check node to its adjacent variable node and comes back from the variable node to the check node. In the other word, the message passed from the variable node v to the check node c is the probability. This probability is computed based on the observed value of the variable node v and the messages come from check nodes to the variable node v in the previous iteration. More precisely, the message passed from a message node v to a check node c is the probability that v has a certain value given the observed value of that message node, and all the values communicated to v in the prior round from check nodes incident to v other than c. Though, the message passed from c to v is the probability that v has a certain value given all the messages passed to c in the previous round from message nodes other than v.

In the BP algorithm, if the messages for each variable node converge to a fixed point or after a defined number of iterations the beliefs for each of the variable node are obtained. In the BP likelihoods or even log-likelihoods are used instead of probabilities or beliefs. Likelihood of a binary random variable is:

$$L(x) = \frac{\Pr(x=0)}{\Pr(x=1)} \tag{2.18}$$

Given another random variable y, the conditional likelihood of x denoted $L(x|y)$ is defined as [6]:

$$L(x|y) = \frac{\Pr(x = 0|y)}{\Pr(x = 1|y)} \tag{2.19}$$

The relation between the conditional likelihood of x ($L(x|y)$) and the conditional likelihood of y ($L(y|x)$) is:

$$L(x|y) = \frac{\frac{\Pr(y|x = 0)\Pr(x = 0)}{\Pr(y)}}{\frac{\Pr(y|x = 1)\Pr(x = 1)}{\Pr(y)}} = \frac{\Pr(y|x = 0)}{\Pr(y|x = 1)}.\frac{\Pr(x = 0)}{\Pr(x = 1)},$$

$$L(x|y) = L(y|x).\frac{\Pr(x=0)}{\Pr(x=1)} \tag{2.20}$$

If the probability of $\Pr(x = 0) = \Pr(x = 1)$ then:

$$L(x|y) = L(y|x) \tag{2.21}$$

Similarly, the log-likelihood of x is $\ln \mathrm{L(x)}$ and the conditional log-likelihood of x given y is $\ln \mathrm{L(x|y)}$. If $y_1, y_2, \dots, y_n$ are independent random variables, because we assumed independence assumption, then [6]:

$$lnL(x|y_1,y_2,\dots,y_n) = ln\frac{\Pr(x=0|y_1,y_2,\dots,y_n)}{\Pr(x=1|y_1,y_2,\dots,y_n)} = \ln\left(\frac{\Pr(x=0|y_1)}{\Pr(x=1|y_1)}\cdots\frac{\Pr(x=0|\,y_n)}{\Pr(x=1|\,y_n)}\right)$$

$$= ln\sum\nolimits_{i=1}\tfrac{\Pr(x=0|y_i)}{\Pr(x=1|y_i)} \tag{2.22}$$

We would like to calculate $lnL(x_1+x_2+\cdots+x_n|y_1,y_2,\dots,y_n)$. Where $(x_1,x_2,\dots,x_n)$ are binary random variables and $(y_1,y_2,\dots,y_n)$ are random variables.

In [6] consider $p = 2\Pr(x_1=0|y_1)-1$ and $q = 2\Pr(x_2=0|y_2)-1$, then $\Pr(x_1+x_2=0|y_1,y_2) = \frac{1+p+q+pq}{4} + \frac{1-p-q+pq}{4} = \frac{2+2pq}{4} \rightarrow 2\Pr(x_1+x_2=0|y_1,y_2)-1 = pq$.

Then:

$$2\Pr(x_1+x_2+\cdots+x_n=0|y_1,y_2,\dots,y_n)-1 = \prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1) \tag{2.23}$$

Therefore, $\mathrm{L}(x_1+x_2+\cdots+x_n|y_1,y_2,\dots,y_n)$ is [6]:

$$\ln\mathrm{L}(x_1+x_2+\cdots+x_n|y_1,y_2,\dots,y_n) = \ln\frac{\Pr(x_1+x_2+\cdots+x_n=0|y_1,y_2,\dots,y_n)}{\Pr(x_1+x_2+\cdots+x_n=1|y_1,y_2,\dots,y_n)} \tag{2.24}$$

Then according to the (2.6), we can simplify (2.7) as follows:

$$\ln\frac{\Pr(x_1+x_2+\cdots+x_n=0|y_1,y_2,\dots,y_n)}{\Pr(x_1+x_2+\cdots+x_n=1|y_1,y_2,\dots,y_n)} = \ln\frac{1/2(1+\prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1))}{1-1/2(1+\prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1))}$$

$$= \ln\frac{1/2(1+\prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1))}{1/2(1-\prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1))} = \ln\frac{1+\prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1)}{1-\prod_{i=1}^{n}(2\Pr(x_i=0|y_i)-1)} \tag{2.25}$$

$$L = \frac{\Pr(x=0)}{\Pr(x=1)} = \frac{\Pr(x=0)}{1-\Pr(x=0)} \rightarrow L - L\Pr(x=0) = \Pr(x=0) \rightarrow \Pr(x=0) = \frac{L}{L+1}$$

$$\rightarrow 2\Pr(x_1 = 0|y_1) - 1 = \frac{l-1}{l+1} = \tanh(\frac{l}{2}) \tag{2.26}$$

Then the log-likelihood of $\ln \mathrm{L}(x_1 + x_2 + \cdots + x_n|y_1, y_2, \ldots, y_n)$ is:

$$\ln \mathrm{L}(x_1 + x_2 + \cdots + x_n|y_1, y_2, \ldots, y_n) = \ln \frac{1+\prod_{i=1}^{n} \tanh(l_i)}{1+\prod_{i=1}^{n} \tanh(l_i)} \tag{2.27}$$

Where $l_i = \ln L(x_i|y_i)$. According to these formulas, we can calculate the message at variable nodes. The messages from check node c to variable node v and from variable node v to check node c in lth iteration are defined as m_{cv}^{l} and m_{vc}^{l}, respectively. Belief propagation algorithm continues the iterations until the m_{vc}^{l} reaches the fixed point or after a defined number of iterations.

$$m_{vc}^{l} = \begin{cases} m_v & l=0 \\ m_v + \sum_{c'\epsilon C_v \backslash \{c\}} m_{c'v}{}^{l-1} & l \geq 1 \end{cases} \tag{2.28}$$

$$m_{c'v}^{l} = \ln \frac{1+\prod_{v'\epsilon V_C \backslash \{V\}} \tanh(\frac{m_{v'c}{}^{l}}{2})}{1-\prod_{v'\epsilon V_C \backslash \{V\}} \tanh(\frac{m_{v'c}{}^{l}}{2})} \tag{2.29}$$

The belief propagation algorithm for LDPC codes can be derived from these two observations. In round 0, for example the variable node v observes the received message and sends the log-likelihood of the observed message m_v along all its outgoing edges. Then, the check node c calculates m_{cv} and sends it to the variable node v. In the calculation of m_{cv}, the message that is sent from the variable node v from the previous iteration is excluded.

2.6 Asymptotic analysis of LDPC Codes

In the asymptotic analysis of LDPC codes, we consider that an ensemble represent the behavior of the all LDPC codes. The behavior of an LDPC code is close to its ensemble if the code length is large and the code does not have the short cycle. In this section, we evaluate the performance of the LDPC codes over the binary erasure channel based on the asymptotic analysis.

2.6.1 Density evolution for LDPC codes

In the iterative decoding algorithms like Belief Propagation (BP) over the Binary Erasure Channel (BEC), the probability of erasures reduces after each iteration. In the BP, consider that the bipartite graph is tree. In the other words, there is no cycle in the graph.

The probability that a message has erasure at the lth iteration is denoted by $P_l^{BP}(\varepsilon)$. Therefore, the probability of erasure at the first iteration $P_{l_0}^{BP}(\varepsilon)$ is equal to the channel erasure ε. If $P_l^{BP}(\varepsilon) = x_l$, then $P_{l+1}^{BP}(\varepsilon) < x_l$.

A check node of degree i along a particular edge is erasure in the $(l+1)th$ iteration if any of the $(i-1)$ messages coming from the variable nodes to this check node in the lth iteration is erasure. The probability that all $(i-1)$ messages coming from the variable nodes are not erasure is $(1-x_l)^{i-1}$. Therefore, the probability that any of them is erasure is $1-(1-x_l)^{i-1}$. The probability that a check node has a degree i is equal to ρ_i. Thus, the expected erasure probability of a check node to variable node message in the $(l+1)th$ iteration is equal to $\sum_{i=2}^{d_c} 1-(1-x_l)^{i-1}$ which can be written as $1-\rho(1-x_l)$. Next, we can consider the erasure probability of the variable node to the check node in the $(l+1)th$ iteration. If the message along a particular edge of a variable node of degree j is erasure, if the received value of the associated variable node is an erasure and all incoming message to the $(j-1)$ edges are erasure. It can be written as $\varepsilon(1-\rho(1-x_l))^{j-1}$. Since the edge has probability λ_j to be connected to a variable node of degree j then the erasure probability of a variable node to check node in the $(l+1)th$ iteration is equal to $\sum_{j=2}^{d_v} \varepsilon(1-\rho(1-x_l))^{j-1} = \varepsilon\lambda\big(1-\rho(1-x_l)\big)$. Since the probability of erasure at $(l+1)th$ iteration is x_{l+1}, then $\varepsilon\lambda\big(1-\rho(1-x_l)\big) = x_{l+1}$. The probability of erasure after iteration reduces therefore $\varepsilon\lambda\big(1-\rho(1-x_l)\big) > x_{l+1}$.

Density evolution equation gives a precise characterization of the asymptotic performance of Low-Density Parity-Check codes. The threshold ε in the density evolution gives on the average of the codes with ensemble (n, λ, ρ). In the asymptotic analysis of the Low-Density Parity-Check codes, the length of the codes consider infinite. Therefore, for the codes with finite length the performance would be less than the expected performance ε.

2.6.2 Threshold

According to the density evolution, if probability of channel erasure is zero, it means that the probability of the erasure after lth iteration is zero. It is worth to mention that this condition is satisfied if the number of iteration goes to infinity. Also, if the probability of channel erasure is one then the probability of the erasure after lth iteration is one [26].

$$Pr_l^{BP}(\varepsilon = 0) = 0 \qquad \text{And} \qquad Pr_l^{BP}(\varepsilon = 0) = 0 \qquad \text{for} \qquad l \to \infty \tag{2.30}$$

There is a well-defined supremum of ε for which $Pr_l^{BP}(\varepsilon) \xrightarrow{l\to\infty} 0$. This supremum is called the Threshold. For a given pair degree distribution, the threshold is defined as [26]:

$$\varepsilon^{BP}(\lambda,\rho) \triangleq Sup\{\varepsilon \in [0,1]: Pr_l^{BP}(\varepsilon) \xrightarrow{l\to\infty} 0\}. \tag{2.31}$$

Over a binary erasure channel with erasure of ε, messages can be transmitted reliably over the channel using Low-Density Parity-Check codes with large length and pair degree distribution (λ, ρ) with $\varepsilon < \varepsilon^{BP}(\lambda, \rho)$. It means that for the channel with probability of erasure ε and $\varepsilon < \varepsilon^{BP}(\lambda, \rho)$ after l iteration the probability of erasure in the codeword goes to zero $Pr_l^{BP}(\epsilon) \xrightarrow{l\to\infty} 0$. Reliable transmission is not guaranteed over the channel with $\varepsilon > \varepsilon^{BP}(\lambda, \rho)$. Also, the threshold is defined as the minimum of $T(x) = \frac{x}{\lambda(1-\rho(1-x))}$.

For the codes with small length or for the codes with small cycles, the actual threshold is smaller than this theoretical threshold. The threshold determines the actual rate of the LDPC codes. The actual rate of the LDPC codes is less than the design rate. Thus, the threshold shows the performance of the LDPC codes.

2.7 Conclusion

In this chapter, we presented some background about Low-Density Parity-Check (LDPC) codes and decoding techniques which will be used for these codes. First, we discussed about different presentations of LDPC codes which are helpful in understanding of the decoding and the performance of these codes. Then, we presented the iterative decoding and the performance of this class of decoding. We also explained about the way that evaluates the performance of the iterative decoding. Finally, we presented the concept of maximum likelihood decoding for LDPC codes over the binary erasure channel.

3. Chapter 3 Fast encoding of the LDPC codes over the binary erasure multiple access channel (BEMAC)

3.1 Introduction

Two robust and practical channel codes are: Turbo codes and Low-Density Parity-Check (LDPC) codes which nearly reach Shannon capacity. There are some advantages of LDPC codes over Turbo codes [29]. The advantages are: 1- they do not need a long interleaver to improve the performance, 2- Thrills methods are not used in the decoding, 3- they decrease block error and the Bit Error Rate (BER) of error floors. Due to these advantages, LDPC codes are more popular than Turbo codes. LDPC codes are one of the hottest topics in error correcting codes. They have good performance under message-passing decoding, and achieved significant fraction of the channel capacity at low decoding complexity. Therefore, a lot of research and development has been done on LDPC codes, and they are used in digital communication standards like DVBS2.

The LDPC code is specified with its parity check matrix H. A codeword in the LDPC code is the null space of the parity check matrix H which is random, sparse and large. LDPC codes use iterative algorithms for the decoding. Due to the sparseness of the parity check matrix H and using the iterative algorithms, the decoding of the LDPC code is fast and simple. On the other hand, the encoding complexity of an LDPC code is an issue. Since the H matrix lacks structure and is random, the encoding complexity is high. Encoding complexity grows quadratically with increasing code length. The encoding complexity of the LDPC code is not time linear. However, the encoding complexity of Turbo codes is time linear, which is an advantage over LDPC codes. A lot of work has been done to reduce the encoding complexity [17]-[20].

The research shows that one way of reducing the encoding complexity is the cascade code [18], [19]. Authors in [20] proposed a method to construct the LDPC parity check matrix in a lower triangular shape. In this method the ensemble of the code is restricted by both degree distribution and parity check matrix has lower triangular shape. Authors in [22] showed that if H_p is an identity matrix, then encoding complexity is time linear. The problem of all these method is the performance loss.

In this chapter we construct a lower triangular LDPC code, where the size of the gap can be flexible. In this method, a lower triangular LDPC code is constructed without a loss of performance. This is achieved with permutations in the elements of the matrix. In this chapter we talk first about the binary erasure multiple access channel. In section 3.2, we show how a half rate code can be helpful in recovery of erased bits in the binary erasure channel, therefore the goal is designing a half-rate code. The low-density parity check code is a good candidate for use over the BEC. The problem of the LDPC codes is their encoding complexity.

In section 3.3, we talk about the best ensemble of the half rate LDPC codes and show the simulation results. In section 3.4, we will evaluate the encoding complexity of the LDPC codes. Next, in this section we show that the encoding complexity of the LDPC code reduces if the parity check matrix H is lower triangular in shape. In section 3.5, we present the generating parity check matrix for lower triangular and non-triangular shape scheme. In section 3.5, we will construct an irregular LDPC code and present the simulation results, and Section 3.6 concludes the chapter.

3.2 Binary Erasure Multiple Access Channel

For increasing bandwidth efficiency, multiple access channel (MAC) is used. In multiple access channel more than one user uses the channel simultaneously. It is a non-orthogonal multiple access channel. Consider two binary users use the noiseless channel simultaneously. The channel is called 2-users binary erasure multiple access channel (2-users BEMAC). The sources are denoted by S1 and S2. Each source generates the binary bits equally $\{0,1\}$. Since, the channel is noiseless the received signal at the destination is superposition of both source messages and is given by $y = u + v$, where u and v are messages of sources S1 and S2, respectively.

If both sources sent the same bits, i.e. both sent 0 or 1, at the destination the received signal is 0 or 2. Then the receiver can decide that both sources sent 0 or 1. If both sources sent 0 with probability of 0.5 then the received signal is 0 with probability of 0.25. At the destination the receiver can decode both messages successfully. Also, if both sources sent 1 with probability of 0.5, the received signal is 2 with probability of 0.25. At the destination both messages can be decoded successfully. However, if sources sent different bits which means one source sent 0 and the other one sent 1 then the received signal is 1. The decoder cannot decide which source sent 1 and which one sent 0. The decoder just knows that sources sent opposite bits. Therefore, those bits at the destination are erased. If one source sent 1 with probability of 0.5 and the other one sent 0 with probability of 0.5, then, the received signal is 1 which is erased with probability of 0.25. We can conclude that the probability of erasure at the destination is also 0.5. The probability of known bits in the received signal at the destination is 0.5. For example if source 1 sent the message (1 0 0 1) and the second source sent (1 1 0 0), the received signal is (2 1 0 1). The decoder at the destination can decide that both sources sent (1 e 0 e). Figure 3.1 shows the binary erasure channel with erasure probability of ε. In 2- users binary erasure multiple access channel the erasure probability of ε is 0.5.

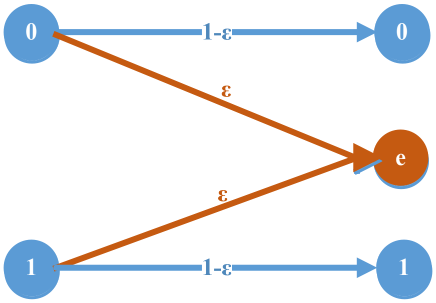

Figure 3.1. Binary erasure channel with erasure probability ε

Since half of the received signal at the destination is erased; a code of half rate can recover erased bits. Assume source 1 transmits at full rate $R1 = 1$ bit per channel use (uncoded stream) and source 2 can transmit at rate $R2 \leq 0.5$ bit per channel use (coded stream). This channel can be modeled as a binary erasure channel (BEC) as shown in figure 3.1 with probability of erasure ε equal to 0.5. Therefore, on average half of the received message at the destination is erased. The receiver first decodes the message of the second source. The second source has encoded its data with a half rate code. It means that if half of its messages at the destination is erased, it can be recovered by decoding the message. Therefore, the receiver can first decode the message of the second source and then the message of the source one can be determined. The capacity of this channel is $1 - \epsilon = 0.5$. This means that the maximum sum rate which is $R1 + R2 \leq 1.5$ can be achieved on this channel. Thus, for a binary erasure multiple access channel the secondary source needs to encodes its data with a code of half rate. It is worth mentionly, in this chapter we generate half rate codes.

3.3 Half rate LDPC codes

In 2-users binary erasure multiple access channel, for recovering the erased bits at the destination, a code of half rate is needed. Since, the design rate of the LDPC code with the ensemble (n, λ, ρ) is:

$$r = \frac{k}{n} = 1 - \frac{n-k}{n} = 1 - \frac{\int \rho(x)dx}{\int \lambda(x)dx} \tag{3.1}$$

To have a code of half rate, n should be equal to $2k$ or $2\int \rho(x)\, dx = \int \lambda(x)dx$. The length of the codeword is twice the message length. The ensemble of the half rate regular LDPC code can be written as:

$$(n, m, 2m), \qquad m\epsilon\{2,3,\dots\} \tag{3.2}$$

According to the density evolution, the actual rate or the threshold of the half rate codes is:

$$x = \varepsilon\lambda\big(1 - \rho(1-x)\big) \quad for\; x \in (0,1) \rightarrow$$

$$x = \varepsilon(1 - (1-x)^{2m-1})^{m-1} \tag{3.3}$$

Where m is the number of ones in each column which is integer and $\lambda(.)$ and $\rho(.)$ are the pair degree distribution. The pair degree distribution of the regular half rate code is as follows:

$$\lambda(x) = x^{m-1} \;, \quad \rho(x) = x^{2m-1} \tag{3.4}$$

According to the threshold, we would like to find the best value of m which results in the maximum threshold α. To find the threshold we have to take the derivative of $\alpha(x)$ with respect to x as the following:

$$\varepsilon = \frac{x}{(1-(1-x)^{2m-1})^{m-1}} \rightarrow$$

$$\frac{\partial\varepsilon}{\partial x} = \frac{\partial}{\partial x}\left(\frac{x}{(1-(1-x)^{2m-1})^{m-1}}\right) \rightarrow$$

$$\frac{\partial\varepsilon}{\partial x} = \frac{(1-(1-x)^{2m-1})^{m-1} - x[(m-1)(2m-1)(\left(1-(1-x)^{2m-1}\right)^{m-2})(1-x)^{2m-2}]}{((1-(1-x)^{2m-1})^{m-1})^2} \qquad (3.5)$$

Then, we put $\frac{\partial\varepsilon}{\partial x} = 0$ to find the value of x which is a function of m:

$$1-(1-x)^{2m-1} - x[(m-1)(2m-1)(1-x)^{2m-2}] = 0 \rightarrow$$

$$1-(1-x)^{2m-2} = \mathrm{x(m(2m-3))}(1-x)^{2m-2} \rightarrow \qquad (3.6)$$

After finding roots of the equation (3.6), we choose one of the roots which is between 0 and 1, then put in the equation (3.3). Table 3.1 shows $T(x)$ versus x according to the different values of m. However, the threshold is the minimum of $T(x)$ and $T(x) = \frac{x}{\lambda\left(1-\rho(1-x)\right)}$.

According to the table 3.1, the equation (3.6) does not have a root between 0 and 1 for $m = 2$. If m is equal to 3 then the threshold peaks at 0.428. The threshold of the LDPC decreases after $m = 3$. It means that with increasing the density necessarily the threshold does not increase. Thus, the best ensemble is $(n, 3,6)$ which results in the best threshold in regular LDPC codes. Figure 3.2 shows the performance or the threshold of these ensembles.

Also, authors in [43] showed that the ensemble $(\mathrm{n}, 3,6)$ is the best ensemble and achieves the best performance among all the other ensembles over the BEC.

Table 3-1. The threshold according to the different values of m

	$\boldsymbol{x}$	$\boldsymbol{\varepsilon}$
m=2	$2x^3-3x^2=0 \rightarrow x=0,\frac{3}{2}$	$x\nexists(0,1)$
m=3	$(x-1)^5-10(1-x)^4x+1=0 \rightarrow$ $x=0.26057\,,1.5115$	$\varepsilon=\frac{x}{(1-(1-x)^{2m-1})^{m-1}} \rightarrow$ $x=0.26057 \rightarrow \varepsilon=0.42944$
m=4	$\frac{1}{(1-(1-x)^7)^3}-\frac{21(1-x)^6x}{(1-(1-x)^7)^4}=0$ $\rightarrow x=0.263641, x=1.56061$	$x=0.263641 \rightarrow$ $\varepsilon=0.383447$
m=5	$(x-1)^9-36(1-x)^8x+1=0 \rightarrow$ $x=0.246559\,, x=1.60313$	$x=0.246559 \rightarrow$ $\varepsilon=0.34155$
m=6	$\frac{1}{(1-(1-x)^{11})^5}-\frac{55(1-x)^{10}x}{(1-(1-x)^{11})^6}=0$ $\rightarrow x=0.263641, x=1.56061$	$x=0.2228108 \rightarrow$ $\varepsilon=0.307646$

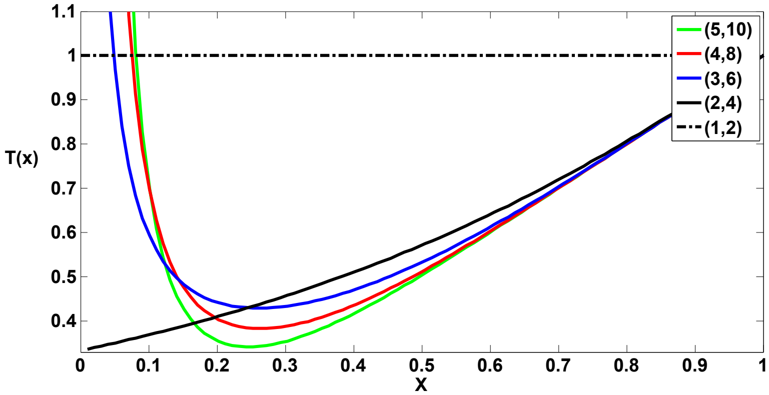

Figure 3.2. T(x) versus x for different values of m

3.4 Encoding complexity of LDPC codes

The drawback of LDPC codes is their encoding complexity which is not time linear. The order of the encoding complexity is n^2. In this section we show the encoding complexity of the LDPC code, also how this complexity reduces if the parity check matrix H is lower triangular.

For a linear code which has a generator matrix of G, the generated codeword is x:

$$S.G = x \tag{3.7}$$

The encoding complexity is:

$$\mathrm{O}\ (n(n-m)) \tag{3.8}$$

Since LDPC codes are specified by parity check matrix H, they do not have a generator matrix. For a (n, k) LDPC code, a codeword is the null space of the parity check matrix and there is:

$$H.x^T = 0^T \tag{3.9}$$

Which shows a codeword is the null space of parity check codes. If the encoding is considered systematic, therefore, the codeword is (x_s, x_p) which x_s and $x_p$are the message and parity, respectively. Therefore, the encoder has to find x_p:

$$H.(x_s, x_p)^T = 0^T \tag{3.10}$$

$$(H_s, H_P).(x_s, x_p)^T = 0^T \tag{3.11}$$

$$H_s.{x_s}^T = H_P.{x_P}^T \tag{3.12}$$

$$H_P{}^{-1}(H_S.x_S{}^T) = x_P{}^T \tag{3.13}$$

To encode a message and determine the parity bits, the encoder has to calculate the inverse of H_P. The Dimensions of H_S and H_P are $m \times (n-m)$ and $m \times m$, respectively.

The complexity (the number of operation) of calculating $H_S.x_S{}^T$ is equal to:

$$\mathrm{O}\,((n-m)) \tag{3.14}$$

The complexity of calculating the inverse of H_P is:

$$\mathrm{O}\,(m^2) \tag{3.15}$$

The complexity (the number of operation) of calculating $H_P{}^{-1}(H_S.x_S{}^T)$ is equal to:

$$\mathrm{O}\left((n-m)+m+m^2\right) \tag{3.16}$$

Then the total number of calculation for calculating the parity bits is:

$$\mathrm{O}\,(n+m^2) \tag{3.17}$$

If the matrix is lower triangular and the complexity of inversion is:

$$\mathrm{O}\,(g^2) \tag{3.18}$$

Then the encoding complexity of an LDPC code is:

$$\mathrm{O}\,(n+g^2) \tag{3.19}$$

Where g is the gap in the H matrix. Figure 3.3 shows the lower triangular H matrix with a gap of g and parity check matrix H of LDPC codes of DVBS2. With reducing the gap in the H

matrix, the encoding complexity reduces quadraticly. Richardson in [17] proposed greedy algorithms for transforming a parity-check matrix H to a lower triangular shape. Using the greedy algorithms does not change the pair distribution of the matrix H.

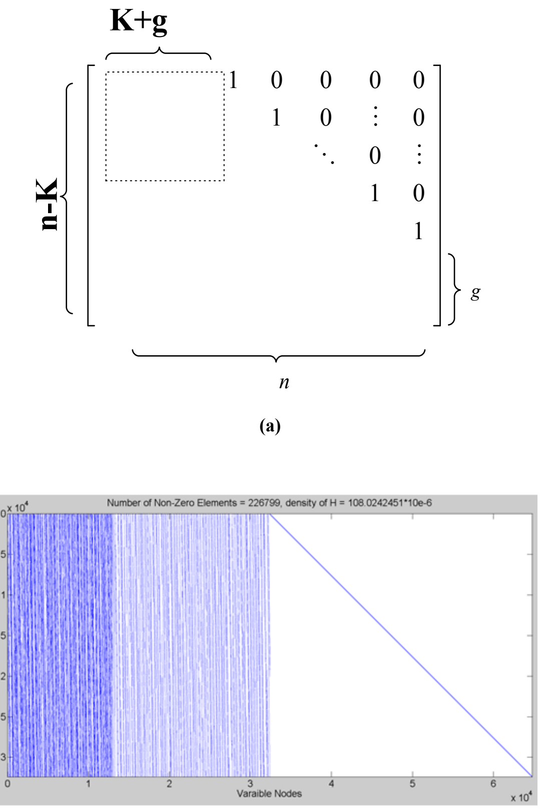

Figure 3.3. a) lower triangular parity check matrix H with gap of g, b) - Shape of the DVBS2 Matrix

However, using greedy algorithms does not always guarantee to reach the considered gap. Here we propose a method for generating a parity-check matrix H which has lower triangular shape with the considered gap. In this situation, the matrix is restricted not only by the degree constraints but also by the constraint that the parity-check matrix has a lower triangular shape. After generating LDPC matrix, the encoding procedure is the same as Richardson proposed in [17].

3.5 Constructing LDPC codes

In a 2 users BEMAC, to recover messages, a code of half rate is needed. An LDPC code is used in the 2-users BEMAC. The problem with the LDPC code is its encoding complexity. To reduce the encoding complexity, a lower triangular parity-check matrix H is needed. First, we need to generate a lower triangular LDPC code. In chapter two we talked about the different way for constructing an LDPC code. One of the conventional method is the Gallager method. We will use this method in the next chapter for generating a regular LDPC code. In this section, we propose a method for constructing LDPC code and lower triangular LDPC codes. Next, we calculate the probability of decoding failure and the outage capacity.

3.5.1 The method for constructing regular LDPC code

In order to generate a random regular low-density parity-check matrix H, we must first specify the ensemble (n, λ, ρ). According to the ensemble, ρ represents the number of ones each row, whereas λ represents the number of ones in each column. The number of columns is n and the number of rows is $\frac{n\lambda}{\rho}$. According to the LDPC ensemble, if we put ρ ones randomly in a row of n elements then, the probability that each elements being one is $\frac{\rho}{n}$. The probability that a column does not contain 1 is:

$$\lambda_0 = (1 - \tfrac{\rho}{n})^{n-k} \tag{3.20}$$

The probability that a column has just one non-zero elements is:

$$\lambda_1 = \binom{n-k}{1} \tfrac{\rho}{n} (1 - \tfrac{\rho}{n})^{n-k-1} \tag{3.21}$$

As we can see from the equation (3.20) the probability that a column has no 1 is not zero. Therefore, the dimension of the matrix decreases from $(n-k) \times k$ to $(1-\lambda_0)n \times (n-k)$. Hence, reducing the design rate of the code to: $1 - \frac{n-k}{(1-\lambda_0)n} = \frac{k-n\lambda_0}{(1-\lambda_0)n}$ which is less than $1 - \frac{n-k}{n}$.

A way to maintain the design rate as$1 - \frac{n-k}{n}$, is to permute the elements of the matrix. This problem can be solved without changing the design rate or the matrix dimension. For example, figure 3.4.a shows a matrix in which the column c_x does not have 1 in its elements and c_y has two or more ones. One of the elements in the column c_y which is 1 chosen randomly. The selected one is permuted with the elements in its same row of the column c_x. It results that the row distribution is remained unchanged and all columns have at least one 1. With this permutation the weight distribution of the rows does not change but the weight distribution of columns is changed as needed. According to the figure 3.4.b, the weight of the second column is zero. On the other hand the weight of the last column is 8. If we choose one of the 1s in the last column randomly and swap it with the corresponding element in the second column, the distribution of row remains unchanged and the columns of weight zero are removed.

$$\begin{array}{c} \begin{array}{ccc} c_x & \ldots & c_y \end{array} \\ \begin{bmatrix} 0 & \ldots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{bmatrix} \end{array}$$

(a)

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

(b)

Figure 3.4: a) a matrix with columns of weight zero and non-zero, b) the permutation in a matrix with a column of weight zero

We can construct regular LDPC codes in a way in which there is no need for permutation. In this way, at each row we choose ρ elements randomly and put 1s at those places. This means that ρ places are chosen randomly out of the n places. The elements corresponding to the chosen places would be 1 and the reset elements are zero. After choosing a place or elements, that element or place have to be removed from the available places or elements. Then in the next row, ρ places have to be chosen randomly out of $n-\rho$ places. Continue until there no available places. Until now, this method guaranties there is no column of having no 1. If there is no more available places, and there are rows not corresponding to ρ, then the available places reset to n and continue until the weight of all rows is ρ. This method guarantees that all rows have the weight ρ and also, the weight of columns is λ. Therefore, there is no need for permutation. If a regular matrix is required, then we continue this method until all columns have λ ones.

The problem of this method is that the degree of freedom reduces at each row, because the number of available places reduces after each row. This method is explained for a regular LDPC code with the ensemble $(12,36)$.

At the first row the number of available places is 12 and they are: $\{0,1,2,3,4,5,6,7,8,9,10,11\}$. Six places are chosen randomly. For example, at the first row $\{2,3,6,7,8,10\}$ are chosen. The available places for the second row are: {0,1,4,5,9,11}. Hence, we have to choose six places out of these six places. There is no available place anymore. Therefore, for the next row, the available places reset to {0,1,2,3,4,5,6,7,8,9,10,11}. Six places are chosen out of twelve places in the third row. These places are: {0,2,4,6,8,10}. This process continues until the weight of all rows is 6. The equation (3.22) shows the constructed matrix.

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{3.22}$$

3.5.2 Proposed method for generating lower triangular LDPC matrix

To generate a lower triangular matrix, the parity-check matrix of the code is restricted not only by the degree constraints, but also by the shape of the matrix. To construct a lower triangular parity check matrix, first we must determine the desire gap. The size of the gap g can even be zero.

In this work, first we consider that the density of a lower triangular LDPC matrix H should be the same as a standard LDPC matrix H and remain unchanged. We can describe this position as follows:

In the parity-check matrix H, ρ ones are spread among n positions in each row. However, in the lower triangular parity-check matrix, ρ ones are spread among $(n-m+g)$ positions. The gap is equal to g. Therefore the rest $(m-g)$ positions have to be zero. Therefore, the density is changed in the lower triangular matrix. For keeping the density unchanged, the number of ones at each row should not be ρ. In this case, if ρ ones are spread in n places, then $\frac{\rho(n-m+g)}{n}$ ones are spread in $(n-m+g)$ places.

For a lower triangular LDPC code with the ensemble (n, λ, ρ), the number ones at each row can be achieved as follows:

$$\rho'(i) = \frac{\rho}{n} \times \left[n - m + i + g(\frac{m-i}{m})\right] \quad i \, \epsilon \, \{1,2, \dots, m\} \tag{3.22}$$

Where ρ is the number of ones in each row and n and m are the number of columns and rows. g is the gap in the parity check matrix, i is the number of each row. For example for a regular (3,6) LDPC code, there are six ones in each row. If the gap is 0, then the number of ones in the first row $(i = 0)$ is $\frac{\rho}{n} * [n - m] = \rho * \left[1 - \frac{m}{n}\right] = \rho * r = 6 * 0.5 = 3$.

First, we evaluate the performance of the generated matrix by considering this constraint for the density. The total number of ones in the H matrix is the summation of ones in each row. Then, according to (3.3) the total number of ones in the matrix is:

$$\sum_{i=1}^{i=m} \rho'(i) = \frac{\rho}{n} \times \left[n - m + i + g\left(\frac{m-i}{m}\right)\right] = \frac{\rho}{n} \times \left[\sum_{i=1}^{m} n - m + g + \left(1 - \frac{g}{m}\right) \sum_{i=1}^{m} i\right]$$

$$= \frac{\rho}{n} \times \left[(n - m + g)(m) + (1 - \frac{g}{m})(\frac{m(m+1)}{2})\right] \tag{3.23}$$

Where l_{avg} and r_{avg} are the average variable and check degrees [26]. Then, the average check degree is:

$$r_{avg} = \frac{\sum_{i=1}^{m} \rho'(i)}{m} = \frac{1}{m}\left(\frac{\rho}{n} \times \left[(n - m + g)(m) + \left(1 - \frac{g}{m}\right)\left(\frac{m(m+1)}{2}\right)\right]\right)$$

$$= \frac{\rho}{n} \times \left[(n - m + g) + \left(1 - \frac{g}{m}\right)\left(\frac{(m+1)}{2}\right)\right] \tag{3.24}$$

And the average variable degree is:

$$l_{avg} = \frac{\sum_{i=1}^{m} \rho'(i)}{n} = \frac{1}{n}\left(\frac{\rho}{n} \times \left[(n-m+g)(m) + \left(1-\frac{g}{m}\right)\left(\frac{m(m+1)}{2}\right)\right]\right) \quad (3.25)$$

Since $\lambda = \frac{m\rho}{n}$, then we can rewrite the equation (3.25) as the following:

$$l_{avg} = \frac{\lambda}{n} \times \left[(n-m+g) + \left(1-\frac{g}{m}\right)\left(\frac{(m+1)}{2}\right)\right] \quad (3.26)$$

We can show that $r_{avg} \leq \rho$ and $l_{avg} \leq \lambda$.

$$\frac{\rho}{n} \times \left[(n-m+g) + \left(1-\frac{g}{m}\right)\left(\frac{(m+1)}{2}\right)\right] \leq \rho \rightarrow$$

$$\frac{1}{n} \times \left[(n-m+g) + \left(1-\frac{g}{m}\right)\left(\frac{(m+1)}{2}\right)\right] \leq 1 \quad (3.27)$$

And

$$\frac{\lambda}{n} \times \left[(n-m+g) + \left(1-\frac{g}{m}\right)\left(\frac{(m+1)}{2}\right)\right] \leq \lambda \rightarrow$$

$$\frac{1}{n} \times \left[(n-m+g) + \left(1-\frac{g}{m}\right)\left(\frac{(m+1)}{2}\right)\right] \leq 1 \quad (3.28)$$

According to the equations (3.27) and (3.28), if the ensemble $(n, 3,6)$ is chosen, the ensemble of the generated matrix is $(n, \lambda \leq 3, \rho \leq 6)$.

If the gap is equal to m, then the ensemble of the resultant matrix H is $(n, 3,6)$. If the gap is zero then the resultant ensemble is:

$$\left(n, \frac{3}{n} \times \left[(n-m+0) + \left(1-\frac{0}{m}\right)\left(\frac{(m+1)}{2}\right)\right], \frac{6}{n} \times \left[(n-m+0) + \left(1-\frac{0}{m}\right)\left(\frac{(m+1)}{2}\right)\right]\right)$$

$$= (n, \frac{3}{n} \times \left[\left(n-\frac{m}{2}+\frac{1}{2}\right), \frac{6}{n} \times \left(n-\frac{m}{2}+\frac{1}{2}\right)\right] = (n, 3\left(1-\frac{r}{2}\right), 6\left(1-\frac{r}{2}\right)) \quad (3.29)$$

Therefore, this constraint results in decreasing the performance. We can first apply this constraint on the number of ones in the row distributions; then according to the column distribution, construct the column by adding more ones in the columns. The constructed matrix has the desire degree distribution and gap while keeping the density as minimum.

3.6 Simulation Results and Discussion

We generate a lower triangular LDPC code based on the method which is presented in the previous section. The pair distribution and the actual rate of the generated matrix must be checked. Richardson in [17] proposed the encoding method of the lower triangular LDPC matrix. We will use the method for the encoding. The complexity of the encoding is low. After encoding, the decoding must be designed based on the Belief Propagation (BP) over the Binary Erasure Channel (BEC). Then, for increasing the performance and compensating the drop in the performance, the guessing algorithm will be used. The performance of the generated code is improved after assuming the values of the erased bits. This section we will demonstrate the simulation results of the code.

3.6.1 Construction of half rate LDPC code

In constructing the parity check matrix H, three aspects must be considered: 1) the encoding complexity 2) the decoding complexity 3) the performance. The gap g in the H matrix determines the encoding complexity. A smaller gap results in a lower encoding complexity. The decoding complexity is proportional to the number of ones in the H matrix or the number of edges in the Tanner graph. The decoding complexity increases with the density of the matrix. The pair degree distribution is the key to having good performance; for example, irregular pair distribution has a better performance than a regular one. There is a tradeoff between the decoding complexity and the performance in LDPC codes. The code with better performance has higher complexity. The degree distribution is designed to maximize the performance, also increases the density, hence, increasing the complexity.

The starting point in designing H matrix is choosing an appropriate gap. The next step is choosing the degree distribution pair (λ, ρ) that achieves the best threshold for a given finite codeword length and rate. Here, the best degree distribution is the primary concern, making the decoding complexity the second concern. A lot of work has been done to find the best degree distribution [25] and [38]. We use the degree pair distribution which was obtained and used in [3]. Authors in [3] used an optimization tool for finding the best right regular degree distribution [26]. The pair distribution is as the following [3]:

$$\lambda(x) = 0.4021x + 0.2137x2 + 0.0768x3 + 0.3902x7 \tag{3.29}$$

$$\rho(x) = x5, \tag{3.30}$$

This pair degree distribution yields a threshold of $\varepsilon^{BP} = 0.472$. Figure 3.5 shows $T(x)$ versus x and the minimum of $T(x)$ is the threshold. Therefore, the pair distribution is chosen and the theoretical threshold of this pair distribution is 0.472.

The next step is choosing an appropriate gap. For choosing the gap of the matrix, we can choose any gap as we want. Here, according to the Richardson greedy algorithm [17], the minimum achievable gap is 0.017n. Therefore, we consider the gap equal to $0.02n$.

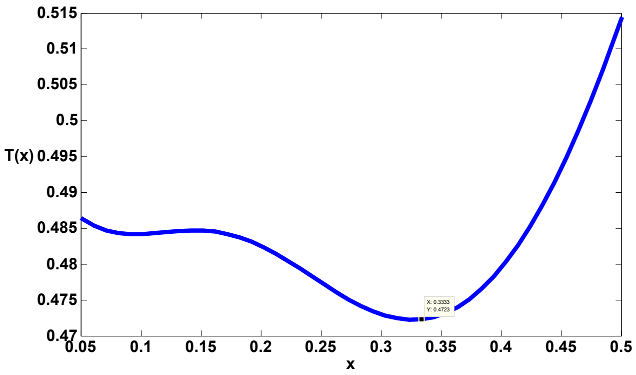

Figure 3.5: T(x) versus x and the threshold

The design rate of the code is 0.5. The code length is chosen to be10^3. The dimension of the H matrix is equal to 500×1000. We consider the gap equal to 20. The simple way of constructing H starts from setting all elements of the matrix H equal to zero except the diagonal of the lower triangular matrix. Figure 3.6 shows the matrix.

The number of ones that is inserted uniformly in each row is then drawn from the variable degree distribution ρ'. According to the degree distribution and lower triangular constraints, we put ρ' ones in each row.

$$\rho'(i) = \frac{6}{1000} \times \left[500 + i + 20(\frac{500-i}{500})\right] \;\; i \,\epsilon\, \{1,2,\ldots,500\} \tag{3.31}$$

$$\rho'(i) = \frac{6}{1000} \times \left[520 + \frac{24}{25} i\right] \;\; i \,\epsilon\, \{1,2,\ldots,500\} \tag{3.32}$$

The second step is to first check the resulting matrix from the columns perspective to avoid columns with weight $w_c \; < 2$. To have columns $w_c \; < 2$, some permutation in the elements of the matrix is needed. It is worth to mention that the permutation should not change the triangular shape. The next step is to have the column degree distribution like the equation (3.29).

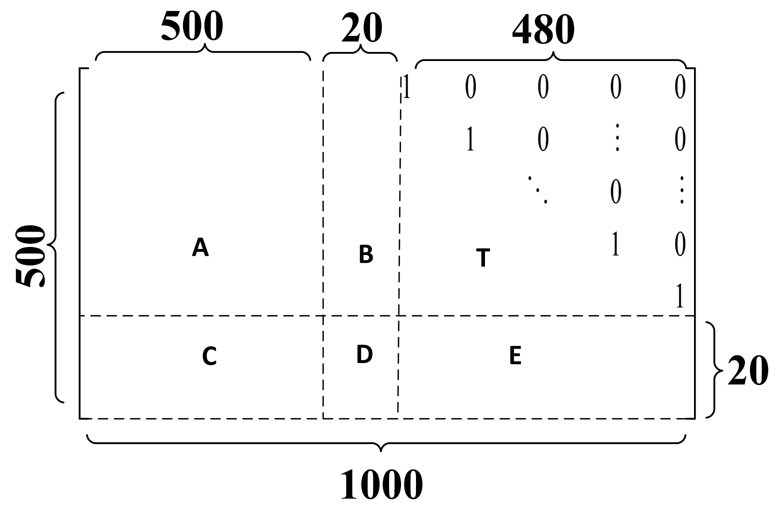

Figure 3.6. The Lower triangular matrix with the gap of 20

To have the column degree distribution similar to equation (3. 29), we need to increase the weight of some of the columns by adding some ones. This results in changes of the row degree distribution.

In each realization of H, we check the following:

1- The matrix should be lower triangular with the gap of 20.

2- According to the figure 3.6 the sub-matrix D should be invertible.

3- According to the constructed matrix, the pair degree distribution and the theoretical threshold should be checked.

Then we select the matrix which achieves the highest threshold. Also, the related D matrix should be invertible. At this stage, H is produced. Note that the construction of H is done once throughout the simulation algorithm for a specific dimension $m \times n$.

According to the constructed H matrix, the pair distribution of the generated matrix is as follows:

$$\rho(x) = 0.0159x^2 + 0.073x^3 + 0.153x^4 + 0.225x^5 + 0.173x^6 + 0.168x^7 + 0.062x^8 + 0.07x^9 + 0.03x^{10} + 0.011x^{11} + 0.004x^{12} + 0.008x^{13} \qquad (3.33)$$

$$\lambda(x) = 0.346x + 0.22x^2 + 0.102x^3 + 0.33x^7, \qquad (3.34)$$

Or we can rewrite them as:

$$\rho(x) = \frac{3*17}{3190}x^2 + \frac{4*59}{3190}x^3 + \frac{5*98}{3190}x^4 + \frac{6*120}{3190}x^5 + \frac{7*79}{3190}x^6 + \frac{8*67}{3190}x^7 + \frac{9*22}{3190}x^8 + \frac{10*23}{3190}x^9 + \frac{11*9}{3190}x^{10} + \frac{12*3}{3190}x^{11} + \frac{13*1}{3190}x^{12} + \frac{14*2}{3190}x^{13} \qquad (3.35)$$

$$\lambda(x) = \frac{2*552}{3190}x + \frac{3*234}{3190}x^2 + \frac{4*82}{3190}x^3 + \frac{8*132}{3190}x^7, \qquad (3.36)$$

The threshold of the generated code is equal to the 0.468. Figure 3 shows the threshold versus the different value of x.

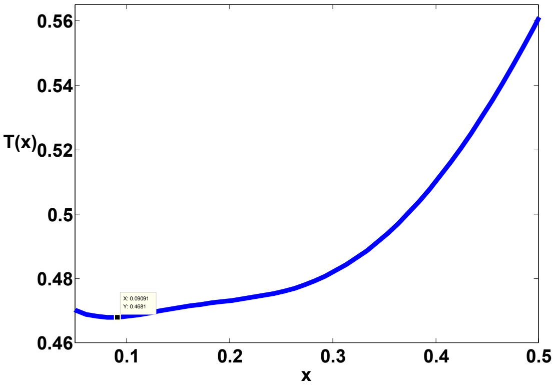

Figure 3.7: $T(x)$of the constructed matrix versus x

3.6.2 Encoding of the lower triangular parity-check LDPC matrix

Encoding of the lower triangular parity-check LDPC matrix is described in [17]. The encoding complexity is reduced in the lower triangular H matrix. The table 2 shows the size of the constructed submatrices in the equation (3.37).

$$H = \begin{pmatrix} A & B & T \\ C & D & E \end{pmatrix} \tag{3.37}$$

Table 3-2: The size of the submatrices in the equation (3.37)

The matrix	Size of the matrix
H	$m \times n = 500 \times 1000$
A	$(m-g) \times (n-m) = 480 \times 500$
B	$(m-g) \times g = 480 \times 20$
C	$g \times (n-m) = 20 \times 500$
D	$g \times g = 20 \times 20$
T	$(m-g) \times (m-g) = 480 \times 480$
E	$g \times (m-g) = 20 \times 480$
$\emptyset := -\boldsymbol{E}\boldsymbol{T}^{-1}\boldsymbol{B} + \boldsymbol{D}$	$g \times g = 20 \times 20$

In the process of systematic encoding, the encoder gets the message s and produces parity bits P_1 and P_2. The encoder sends $[s, P_1, P_2]$ to the channel. The table (3.3) shows the process of the encoding. According to the table 3.3 the encoder can encode the messages with low complexity.

Table 3-3: The procedure of the encoding

The procedure of the encoding	Size of the output matrix
$\mathrm{A}\boldsymbol{s}^{\boldsymbol{T}}$	$(m-g) \times 1$
$\boldsymbol{T}^{-1}[\mathbf{A}\boldsymbol{s}^{\boldsymbol{T}}]$	$(m-g) \times 1$
$-\boldsymbol{E}[\boldsymbol{T}^{-1}[\mathbf{A}\boldsymbol{s}^{\boldsymbol{T}}]$	$g \times 1$

$\boldsymbol{Cs^T}$	$g \times 1$
$-\boldsymbol{E}[\boldsymbol{T^{-1}}[\mathbf{A}\boldsymbol{s^T}] + \boldsymbol{Cs^T}$	$g \times 1$
$\boldsymbol{P_1} = -\emptyset^{-1}[-\boldsymbol{E}[\boldsymbol{T^{-1}}[\mathbf{A}\boldsymbol{s^T}] + \boldsymbol{Cs^T}]$	$g \times 1$
$\boldsymbol{BP_1^T}$	$(m-g) \times 1$
$\boldsymbol{BP_1^T} + \mathbf{A}\boldsymbol{s^T}$	$(m-g) \times 1$
$\boldsymbol{P_2} = -\boldsymbol{T^{-1}}[\boldsymbol{BP_1^T} + \mathbf{A}\boldsymbol{s^T}]$	$(m-g) \times 1$

3.6.3 Decoding of the lower triangular parity-check LDPC matrix

Belief propagation is an iterative decoding algorithm and one of the powerful tools for decoding of the LDPC codes. Due to the features of the binary erasure channel, positions of the erased bits are known and the value of bits is 0 or 1. Therefore, BP algorithm is very simple and fast over the BEC and can be described as following [6]:

1- Put the value of each check nodes equal to zero.

2- If each variable node is received, then, calculate the value of all adjacent check nodes. Remove all known variable nodes and their associated edges from the graph.

3- Look for a check node of degree one. If there is a check node of degree one, substitute its value into its adjacent variable node. Then remove all known variable nodes and their edges from graph and again look for a check node of edge one. Continue until either there is no more check node of degree one or all unknown variable nodes are de-erased.

We implement the BP algorithm. According to the BP, the performance of the generated code is evaluated. Figure 3.8 shows the capability of correcting code versus the different number of iterations. Figure 3.8 shows how the probability of erasure versus the number of iteration. According to the Figure 3.8, the probability of erasure drops with increasing the number of iterations. If the threshold of the code is closer to the channel erasure, the code converges faster and the number of decoding iteration decreases. However, if the channel erasure is greater than the threshold of the code $\epsilon > \varepsilon$, then the probability of erasure does not go to zero even if the number of iterations goes to infinity. The actual rate of this code on the average is $r = 1 - 0.418 = 0.482$.

3.6.4 Guessing algorithm

Due to the short cycles in the code and the finite length of the code, the performance of the code is less than the theoretical threshold. The short cycles can be removed when the code is constructed. Another way for preventing the short cycles in the low-density parity check codes is increasing the code length. The probability of short cycles decreases with increasing the code length. If the cycles do not remove from the code, the performance of the code can be increased in another way. To improve the performance of the code, the guessing algorithm can be added at the decoder.

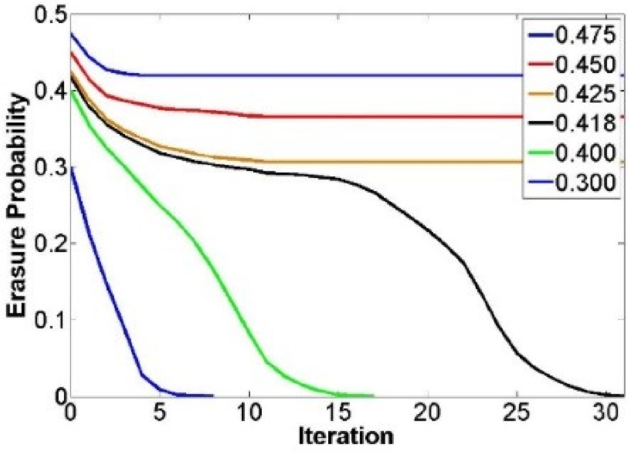

Figure 3.8: The simulation result of the constructed code; the probability of erasure versus the number of iteration

The guessing algorithm is proposed in [12]. Authors in [12] took a different approach. Instead of trying to find a good degree distribution, the performance of an existing code have been improved over the binary erasure channel (BEC). In [12] for the first time, the performance of an existing code was improved by guessing on unknown variable nodes for short-length LDPC codes. Authors in [12] proposed three algorithms, algorithm A is the same as the standard belief propagation. In algorithm B, if algorithm A fails, it makes some assumption on some of the erased bits, then check-sum determines if guesses are correct or not. Algorithm B guesses on the variable nodes with higher degree. The drawback of this method is that the complexity of the decoder grows exponentially with increasing the number of guesses and there is a limitation on the number of guessing variable nodes and also it has probability of error greater than the maximum likelihood. To reduce the complexity and improve the performance, they proposed algorithm C. In algorithm C, the decoder defines a set of equations as basic equations and if and only if the set of basic equations have a unique solution then the received codeword is maximum likelihood decodable.

The simulation results show that the code can correct up to 418 continuous erased bits out of 1000 bits. Figure 3.8 depicted the simulation results. It shows that if the number of erasures is equal or bellow of 418, the decoder can decode successfully. Then, we apply the guessing algorithm B on this code. If we make just one guess then the performance of the code can be improved. It can correct up to 440 erased continuous bits out of 1000 bits. Figure 3.9 shows the simulation results. If we make more than one guess on the V set, the performance would improve more, however on the other side, the complexity of the decoder increases more. We can see from the simulation results that just with one guess, the performance of the code increases by 2%.

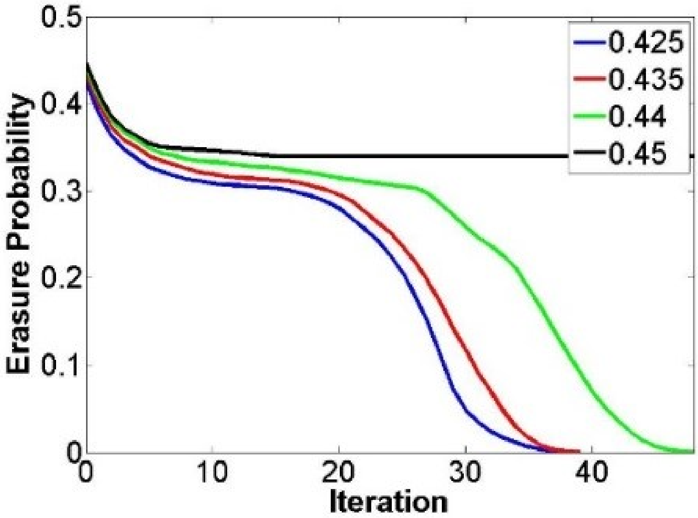

Figure 3.9: The probability of erasure versus the number of iteration for the constructed code after adding the guessing algorithm

3.7 Comparison with the half rate LDPC code in DVBS2 Standard

In this chapter we generated an LDPC code which has a low encoding complexity. In this section we would like to compare the decoding and the encoding complexity and also the performance of the constructed code in this chapter with a half rate LDPC code of the standard DVBS2.

The pair distribution of the constructed code is given in the equations (3.35) and (3.36). Therefore the theoretical threshold for these pair distribution degrees is equal to 0.468. As far as the gap of the constructed code is 20, therefore the encoding complexity of the code is $o(g^2) = o(20^2) = o(400)$. While the gap for the DVBS2 LDPC codes is equal to zero then the encoding complexity of this code is equal to zero.

The pair distribution of the half rate LDPC code of the standard DVBS2 is as the following [44]:

$$\rho(x) = \frac{6}{226799}\, x^5 + \frac{7*32399}{226799} x^6 \tag{3.38}$$

$$\lambda(x) = \frac{1}{226799} + \frac{2*32399}{226799} x + \frac{3*19400}{226799} x^2 + \frac{8*12960}{226799} x^7 \tag{3.39}$$

According to the density evolution and the pair distribution in the equations (3.38) and (3.39) is equal to 0.465. Therefore, both codes have the same theoretical performance.

The decoding complexity of an LDPC code is in correspondence with number of edges in Tanner graph of the code or the number ones in the its parity check matrix$H$. Therefore, the decoding complexity of the LDPC codes is defined in term of the density. The density of parity-check matrix in the LDPC codes is defined as the following [26]:

$$\Delta(H) = \frac{1}{nr}\left|\{(i,j): H_{i,j} \neq 0\}\right|. \tag{3.40}$$

Since, the number of one in the parity check matrix of the constructed code is equal to 3190 and according to the equation (3.40), the density of the constructed code in this chapter is equal to:

$$\Delta(H) = \frac{1}{nr}\left|\{(i,j): H_{i,j} \neq 0\}\right| = \frac{1}{1000\times0.5} \times 3190 = \frac{3190}{500} = 6.38 \tag{3.41}$$

The number ones in the half rate LDPC code is equal to 226799 then, the density of the DVBS2 LDPC code is equal to:

$$\Delta(H) = \frac{1}{nr}\left|\{(i,j): H_{i,j} \neq 0\}\right| = \frac{1}{64800\times0.5} \times 226799 = \frac{226799}{32400} = 6.99 \tag{3.42}$$

After comparing the equations (3.41) and (3.42), we can conclude that the density of the DVBS2 is greater than the constructed code in this chapter. Therefore, the decoding complexity of the DVBS2 LDPC codes is greater than the generated code in this chapter.

3.8 Conclusion

In this Chapter, we studied 2-users binary erasure multiple access channel (2-users BEMAC). From 2-users BEMAC we conclude that a code rate of half is needed. Low-Density Parity-Check codes (LDPC) are one of the good candidates for 2-user BEMAC. Due to the good performance of the LDPC codes over BEC. The problem of the LDPC codes is the encoding complexity which is not time linear. In this chapter we talked about the encoding complexity and how lower triangular LDPC matrix reduces the encoding complexity.

For the half rate LDPC code, using the LDPC codes which their matrices are lower triangular decreases the encoding complexity. Also, in this chapter we proved that which ensembles results in the best threshold. We proposed a method for generating the lower triangular LDPC matrix which its density remained unchanged. The performance of the constructed code is evaluated. Also, for increasing the performance of the code guessing algorithm is used. Applying the guessing algorithm at the decoder increased the performance of the code by 0.02.

4. Chapter 4 Fast decoding of LDPC codes over binary erasure multiple access channel (BEMAC)

4.1 Introduction

In the previous chapter an irregular Low-Density Parity-Check (LDPC) code with low encoding complexity has been generated and also the performance of the generated code increased. In this chapter, we have developed a regular LDPC code which has low decoding complexity. Also, a new method has been proposed that increases the actual rate.

Since the parity check matrix of the LDPC codes is sparse and they utilize iterative algorithms for the decoding process, their decoding complexity is low. Iterative algorithms are named message passing algorithms [6]. One important class of these algorithms is the belief propagation algorithm (BP) [6]. BP is a suboptimal decoder, but, approximates the maximum likelihood decoding [6].

There is a tradeoff between complexity of decoder and performance in LDPC codes. Performance-optimized LDPC codes, usually needs large number of iteration to convergence, therefore, they are not complexity-optimized codes. Irregular LDPC codes have a better performance than regular codes; however, the regular LDPC codes are lower complex. In the half rate LDPC code, regular (3,6) LDCP codes have best performance and lowest complexity among all the regular ensembles of rate half.

In this chapter, a technique is presented for improving the performance of the existing codes, without increasing the complexity greatly. Instead of using an optimized-performance code, the performance of an existing code is improved. We do this by applying iterative decoding algorithms, standard BP, generalized tree-expected propagation (GTEP) and guessing algorithm. Complexity of the iterative decoding algorithms like BP and GTEP is low. However, the complexity of the guessing algorithm increases exponentially with the number of guesses. A new guessing algorithm is proposed, which reduces the number of possibilities, hence the decoding complexity. In the new guessing algorithm, instead of making assumption on a set of variable nodes, the decoder makes assumption on the variable nodes which are connected to a set of check nodes. Regarding the binary field, the number of possibilities is reduced by half. The proposed method is applied to a regular LDPC (3, 6) codes with lengths of 1000 and 2000. The threshold of a regular LDPC code is 0.428 according to the density evolution. Due to the cycles and finite code length, the performance of the code decreases to less than 0.42. Applying this new method and considering a maximum 3 set of guesses, the actual rate increases to 0.43. With increasing the number of guesses the actual rate increases more.

In the next section the relationship between performance and complexity of the LDPC code is evaluated.

4.2 Performance and complexity of LDPC codes

Iterative decoding algorithms are suboptimal. It means that the performance of LDPC codes would be degraded if there are cycles in the Tanner graph. In this case the actual rate of the code is less than the design rate. Performance of the iterative decoder is evaluated by density evolution technique [6]. For evaluating performance of iterative decoders, assumed that the Tanner graph is a tree or in other words, rows are linearly independent. When rows are linearly independent it means that the H matrix is cycle free. An LDPC code is cycle free [6], if the number of ones in common between any two columns in H matrix is no greater than one. According to the density evolution, the performance of LDPC codes is characterized by the threshold, denoted by ε. The threshold can be calculated according to the pair distribution [6] and as the following:

$$\varepsilon.\lambda\big(1-\rho(1-x)\big) < x\ for\ x \epsilon\ (0,\varepsilon) \tag{4.1}$$

$\lambda(.)$ and $\rho(.)$ are degree distribution of columns and rows of the LDPC matrix. To have an optimized-performance LDPC codes, pair distribution have to be selected carefully [27]. The performance of an LDPC code is related to the weight distribution of the columns and rows in the H matrix and usually the performance of the irregular LDPC code is better than the regular one.

On the other hand, the complexity of the decoder is related to the number of edges in the Tanner graph, or, the number of ones in the H matrix. According to Richardson [26], the density of parity check matrix of LDPC code H denoted by $\Delta(H)$ is defined as the following [26]:

$$\Delta(H) = \frac{1}{nr}\big|\{(i,j): H_{i,j} \neq 0\}\big|. \tag{4.2}$$

This shows that if the number of ones in the matrix increases, the density of the matrix increases, and therefore, the complexity of decoder increases. The lower bound on the density of LDPC matrices in terms of the threshold and design rate has been defined as [26]:

$$\lim_{N\to\infty} \inf \Delta_N > \frac{K_1 + K_2 ln\frac{1}{\delta}}{1-\delta} \tag{4.3}$$

$$K_1 = \frac{\varepsilon ln\frac{\varepsilon}{1-\varepsilon}}{(1-\varepsilon) ln\frac{1}{1-\varepsilon}} \tag{4.4}$$

$$K_2 = \frac{\varepsilon}{(1-\varepsilon) ln\frac{1}{1-\varepsilon}} \tag{4.5}$$

Where δ is multiplicative gap and defined as follows [26]:

$$r = (1-\varepsilon)(1-\delta) \tag{4.6}$$

$$\delta = \frac{1-\varepsilon-r}{1-\varepsilon} \tag{4.7}$$

If the actual rate is equal to the design rate, then, the gap would be zero. The value of δ increases with decreasing performance and we can conclude that the lower bound on the complexity decreases. If we simplify the equation (4.3) then we have:

$$\lim_{N\to\infty} inf\ \Delta_N > \frac{\varepsilon}{r} \frac{ln\frac{\varepsilon}{1-\varepsilon-r}}{ln\frac{1}{1-\varepsilon}} \rightarrow \tag{4.8}$$

$$\frac{No.of\ ones}{nr} > \frac{\varepsilon}{r} \frac{ln\frac{\varepsilon}{1-\varepsilon-r}}{ln\frac{1}{1-\varepsilon}} \rightarrow \tag{4.9}$$

$$No.of\ ones > n\varepsilon \frac{ln\frac{\varepsilon}{1-\varepsilon-r}}{ln\frac{1}{1-\varepsilon}} \rightarrow \tag{4.10}$$

$$No.of\ ones > n \frac{(ln\frac{\varepsilon}{1-\varepsilon-r})^{\varepsilon}}{ln\frac{1}{1-\varepsilon}} \tag{4.11}$$

The equation (4.11) shows the minimum number of ones for a certain performance. This shows that an increase in the threshold ε of the LDPC code results in a logarithmic increase in the lower bound on the density.

A. Khandekar and et al. in the paper “One the complexity of Reliable Communication on the Erasure Channel,” show the relationship between the decoding complexity under iterative message-passing algorithm and the asymptotic achievable rate [28]. The authors show how the decoding complexity increases if the gap between the design rate r and the asymptotic achievable rate ε tends to zero $\delta \to 0$ [28].

The encoding and decoding complexity in [28] are dented as $\overline{x_E}(\delta, \pi)$ and $\overline{x_D}(\delta, \pi)$, respectively. Where, π is a decoded error probability and δ is the multiplicative gap. The multiplicative gap is defined in the equation (4.7).

Authors in [28] presented that for the ensemble of the LDPC code of rate r, the encoding complexity and the decoding complexity with maximum-likelihood decoding over the binary erasure channel can be obtained as the following [28]:

$$\lim_{\delta \to 0} \overline{x_E}(\delta, \pi) = O(\tfrac{1}{\delta^2}) \tag{4.12}$$

$$\lim_{\delta \to 0} \overline{x_D}(\delta, \pi) = O(\tfrac{1}{\delta^4}) \tag{4.13}$$

Also, the authors in [28] demonstrate that for the irregular ensemble of LDPC codes under the message passing algorithms over the binary erasure channel, the complexity of the decoder per each iteration is equal to [28]:

$$\lim_{\pi \to 0} \overline{x_D}(\delta, \pi) = O(\log 1/\varepsilon) \tag{4.14}$$

The equation (4.14) shows that if the gap between the achievable rate and the design rate decreases, the complexity of the decoder increases logarithmically in each iteration. Therefore, decreasing the gap between the design rate and the actual rate increases the complexity of the decoder and also the encoder. The pair degree distribution determines the actual rate and the gap between the design rate and the actual rate. Therefore, if the performance is the primary concern in constructing an LDPC code, then an appropriate pair distribution has to be chosen which have a good performance. Irregular pair distribution has better performance. Though, the decoding complexity increases in the regular pair distribution.

The actual performance of a constructed LDPC code is less than the theoretical threshold due to the cycles in the graph. To improve an existing code and overcome the cycles, a new method will be presented in the next section.

4.4 Proposed method

Standard belief propagation (BP) algorithm is very fast and simple over the binary erasure channel. In the BP algorithm, the decoder after removing known variable nodes and their associated edges in the Tanner graph, looks for check nodes of degree one. The decoder transfers the value of the degree one check node to its adjacent variable node [6]. BP stops and declares failure when there is no more check nodes of degree one.

When belief propagation gets stuck the erased bits are either the member of a stopping set or not, therefore, there are two scenarios. The first scenario happens when the number of unknown variable nodes is equal to the number of independent equations. In other words, the matrix is full rank and there is a unique answer for the equations, however, BP cannot find. The erased bits are the member of the stopping set. Figure 4.1 shows the Tanner graph after removing known variable nodes and associated edges when the first situation happens.

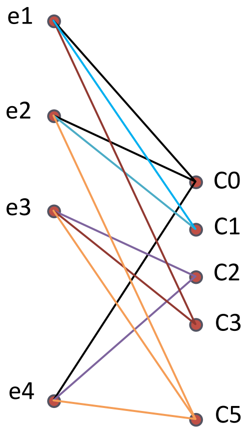

Figure 4.1. The Tanner graph of the first scenario

Equations at the check nodes for the Tanner graph in Figure 4.1 are equal to:

$$\begin{cases} e_1 + e_2 + e_4 = C_0 \\ e_1 + e_2 = C_1 \\ e_3 + e_4 = C_2 \\ e_1 + e_3 = C_3 \\ e_2 + e_3 + e_4 = C_5 \end{cases} \tag{4.15}$$

The number of erased bits in this Tanner graph is four. The number of independent equation and rank of the matrix is also four. Therefore, there is a unique answer for the set of equations in the (4.15), however BP cannot solve them. Due to the cycles in the Tanner graph, the erased bits are in the stopping set. However, the unknown variables in the equation (4.15) can be determined by Gaussian elimination; the answer for e_4 is equal to $e_4 = C_0 + C_1$, after finding e_4 the next erased bit e_3 can be determined. Finally e_1 and e_2 would be found.

The second scenario happens when the number of unknown bits is more than the number of independent equations. The matrix of coefficients is not full rank. Therefore, there is no unique answer for the unknown bits. Figure 4.2 shows the Tanner graph for the second scenario.

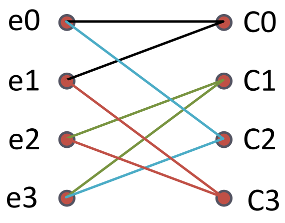

Figure 4.2. The Tanner graph for the second scenario

Equations at the check nodes in figure 4.2 are:

$$\begin{cases} e_0 + e_1 = C_0 \\ e_2 + e_3 = C_1 \\ e_0 + e_3 = C_2 \\ e_1 + e_2 = C_3 \end{cases} \tag{4.15}$$

The rank of the matrix for the set of equations in the equation (4.15) is three and there are four unknown variable nodes. The matrix is not full rank. Therefore, there is no unique solution for these equations and BP cannot solve these equations.

This is the weakness of the standard BP which cannot find the erased bits in the first and second scenario due to the cycle in the Tanner graph. To solve these undesirable situations, we propose a new algorithm in which the performance of an LDPC code can be improved without increasing the decoding complexity considerably. A couple of researches have been done to improve the performance of the Belief Propagation algorithm, however, these methods are not always efficient. It means that sometimes they add high complexity, however, the performance can be improved without increasing the complexity too highly. We proposed a method which improves the performance of the code according to the scenario that happened. Therefore, the proposed method is more efficient. The proposed algorithm is a combination of three decoding algorithms: standard BP, Generalized Tree-expected propagation (GTEP), and guessing algorithm. Figure 4.3 shows the proposed algorithm.

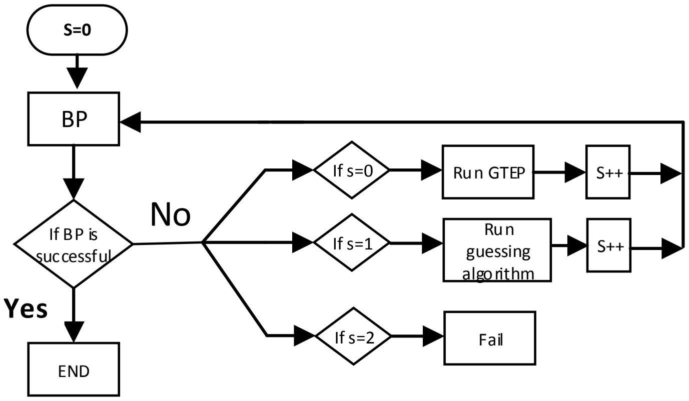

Figure 4.3. The block diagram of the proposed method

The complexity of GTEP is the same as BP and this algorithm can solve some of the erased bits in the first scenario. Therefore, it does not add more complexity. Though sometimes, GTEP can solve all the erased bits and there is no need to run guessing algorithm and adds more complexity. The guessing algorithm adds a higher complexity. The complexity of the guessing algorithm increases exponentially with increasing the number of guesses. Therefore, the guessing algorithm is run after GTEP. Running GTEP decreases the number of unknown variable nodes and results in decreasing the number of guesses in the guessing algorithm, which results in reducing the decoding complexity. In addition, the complexity of the guessing algorithm can be reduced, if the decoder chooses a check node and makes an assumption on the variable nodes are connected to it, instead of choosing a set of variable nodes and making an assumption on them. At the next section we will talk about the GTEP and the guessing algorithm.

4.3.1 Generalized tree-expected propagation

Generalized tree-expected propagation (GTEP) is like belief propagation and at each iteration one check node and one variable node are removed from the graph. If the condition for the successful decoding of GTEP is satisfied, then it can solve the erased bits. After solving one erased bit, a couple of erased bits can be defined with standard BP. Generalized TEP (GTEP) can find the value of some of the erased bits if the first scenario happens. GTEP works as a Maxwell decoder but with the same complexity as BP [16]. GTEP algorithm is as the following [16]:

1- In each iteration, it selects one check node.

2- If the degree of the check node is one it runs BP.

3- If the degree of the check node is greater than one then, it removes the check node and one of the variable nodes and its associated edges. The check nodes that are connected to the removed variable node are reconnected to all of the variable nodes connected to the removed check node. If the removed variable node is parity one, then, flip the check nodes.

4- Continue and go to step 1 until there is no check node of degree one or all the check nodes are removed.

Tree-expected propagation (TEP) is a special case of GTEP. In TEP, the decoder looks for check nodes of degree two and removes it from the graph with one of its associated variable node. The condition to decode successfully is that two variable nodes of a check node degree two also share a check node of degree three. If this condition is satisfied, then check nodes of degree one are appeared and BP can start the decoding again. GTEP decodes successfully, if y variable nodes are connected to a check node of degree y and also are connected to another check node of degree $y + 1$. For example if three eased bits share a check node of degree three and also share a check node of degree four, then the forth bit can be solved by GTEP. Figure 4.4 shows the process of TEP. It shows that after one iteration, two check node of degree one is released.

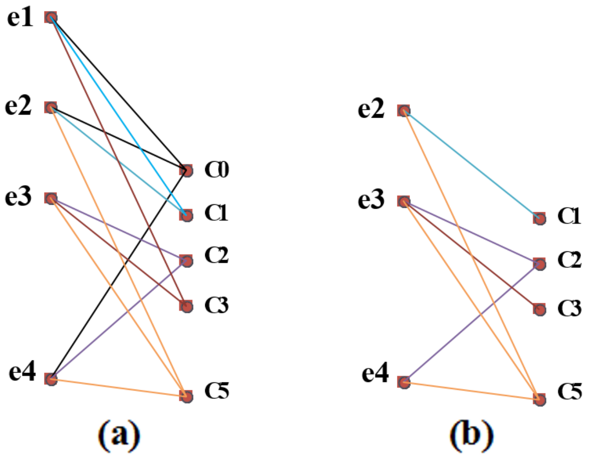

Figure 4.4. a) The Tanner graph before running GTEP b) The Tanner graph after running GTEP

GTEP have the complexity as low as BP. Since, at each iteration it remove one check node and one variable node its complexity is the same as BP. GTEP works successfully in the first scenario, if the condition for successful GTEP decoding is satisfied. Then, it can solve the erased bits. Otherwise, if the conditions for successful decoding are not satisfied, GTEP fails. The remaining erased bits can be solved by the guessing algorithm.

4.3.2 The new guessing algorithm

If BP and GTEP could not find the value of some of the erased variable nodes, the guessing algorithm will be used. In the first scenario, when the equations at the check nodes have a unique answer but standard BP cannot find it, the guessing algorithm by guessing on the erased bits can find the answer. The guessing algorithm in the first scenario finds the unique answer. One of the problems of the guessing algorithm is its complexity. The complexity of the guessing algorithm grows exponentially if the number of guesses increases linearly. Hence, there is a limitation on the number of guesses [12]. Another problem of the guessing algorithm is its probability of error. When the second scenario happens and there is no unique answer for the equations, it is possible that the guessing algorithm declares a wrong codeword as the output of the decoding.

In the guessing algorithm, the algorithm makes assumption on a set of erased variable nodes. We consider that the size of the set is x. In the binary field, for x guesses on the erased variable nodes there are 2^x possibilities. The guessing algorithm for finding the correct guess it has to check all 2^x possibilities. The way for finding the correct assumption is checking the check nodes. The correct assumption gives zero check-sum. If there is one unique answer, then there is only one possibility gives the zero check-sum among all the 2^x possibilities.

In a binary field, we can reduce the number of possibilities using the new guessing algorithm. In the guessing algorithm, the decoder chooses a set of variable nodes and poses assumptions on them. However, in the new guessing algorithm the decoder chooses a set of check nodes and makes assumption on the variable nodes connected to them. In this algorithm if the decoder chooses a set of k check nodes $\{c_0, c_1, \dots, c_k\}$ which are:

$$\begin{cases} b_{11}.a_1 + & \cdots & +b_{1N}.a_N + c_1 & = 0 \\ \vdots & \ddots & \vdots \quad \vdots & \vdots \\ b_{k1}.a_1 + & \dots & +b_{kN}.a_N + c_k & = 0 \end{cases}, \qquad \text{b's and c's } \epsilon\{0,1\} \tag{4.16}$$

Where $(+)$ is the addition in a binary field (Galois field). The number of unknown variables in these equations is N':

$$N' = \sum_{j=1}^{N} max(\{b_{ij} \middle| i = 1, \dots, k\}) \tag{4.17}$$

Then the number of possibilities is:

$$2^{N'-k} \qquad (4.18)$$

Proof: All possible combinations of the binary N-tuple $(a_1, \dots, a_N)$ is 2^N. In linear algebra, given M binary independent set of linear equations, the number of the possible solutions is 2^{N-M} [29].

For example if one check node is selected and the degree of the chosen check node is two like Figure 4.5 (a), then the number of possibilities and the independent equation are two and one, respectively. Table 4.1 shows the possibilities for e_1 and e_2 according to the value of c.

Table 4-1: The possible values for the variable nodes in the figure 4.5.a

e_1	e_2	C	e_1	e_2	C
1	0	1	0	0	0
0	1	1	1	1	0

If the decoder chooses two check nodes of degree two and they have one in common variable node similar to Figure 4.5.b then the number of unknown variable nodes and independent equations are three and two, respectively. Therefore, the number of possibilities is two. The decoder poses assumptions on three variable nodes $e_1e_2e_3$ in the new guessing algorithm while the number of possibilities is two which is equal to one guess in the guessing algorithm. Table 4.2 shows the possibilities if $c = 1$ and $c_x = 0$. In this paper we consider $e_1e_2e_3$ as one set of guess.

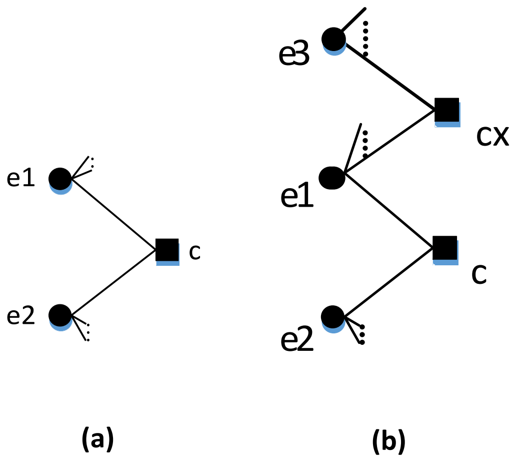

Figure 4.5. a) A check node of degree two and its associated variable nodes b) two check nodes of degree two and their associated variable nodes

Table 4-2: the possible values of the variable nodes in the figure 4.5.b

e_1	e_2	c	e_1	e_3	c_x	e_1	e_2	e_3	c	c_x
1	0	1	0	0	0	1	0	1	1	0
0	1	1	1	1	0	0	1	0	1	0

3.9 Simulation Results

In this section, the proposed method will be first applied on an existing LDPC code, then the performance of the code would be evaluated. The regular LDPC codes are chosen for the test of the method. In this thesis two regular (3,6) half-rate LDPC codes are constructed based on the Gallager method. The lengths of the constructed codes are 10^3 and 2×10^3. These two codes have the same design rate and theoretical threshold. However, they have different performance, due to different code length. The pair distribution of these two codes is:

$$\lambda(x) = x^2 \tag{4.19}$$

$$\rho(x) = x^5 \tag{4.20}$$

According to the equation (2. 29) the theoretical threshold for the pair distribution of (4.19) and (4.20) is equal to 0.428. Figure (3.2) shows the threshold of the ensemble $(3,6)$ versus x. However, the performance of the codes is less than 0.429, due to the cycles in the graph and length of the code. The performance of these codes is evaluated according to the different channel erasure rates. Figure 4.6 shows the probability that a packet for the code of length 10^3 received correctly over different channel erasure rates. This figure depicts that for the small channel erasure rate, less than 0.41, the probability that a packet received correctly is almost one. It means that all the erased bits solved by the decoder. With increasing the channel erasure rate after 0.41, the probability that the decoder can correct all the erasure drops sharply.

Figure 4.6 also shows that how applying the proposed method (BP+ GTEP+ Guessing algorithm) improves the performance from 0.41 to 0.42. The new guessing algorithm is applied. In this code, one check node according to figure 4.5.a or two check nodes similar to the figure 4.5.b are chosen. Therefore, the number of possibilities is two which is equal to the guess on one variable node. As we can see from the results, if the channel threshold is 0.43, the probability that a packet has erasure after running BP is 0.7 and this probability reduces to 0.55 after running GTEP algorithm. At the last step, after running the new guessing algorithm, this probability is reduced to 0.23.

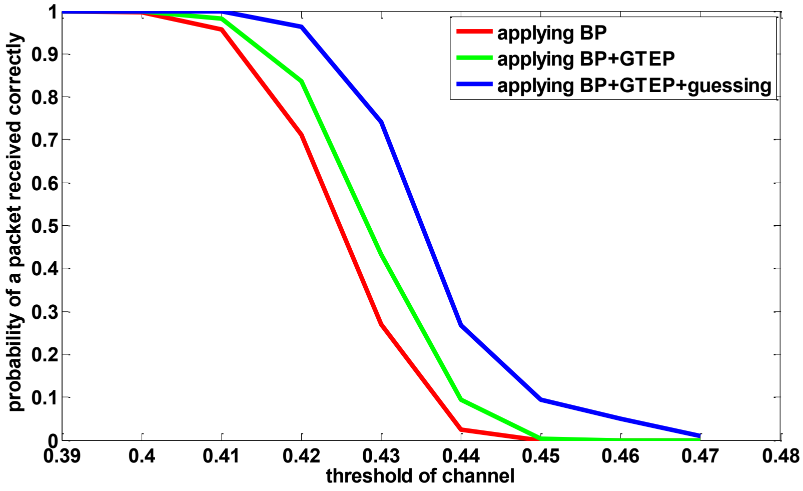

Figure 4.6. The simulation results for the code of length 10^3

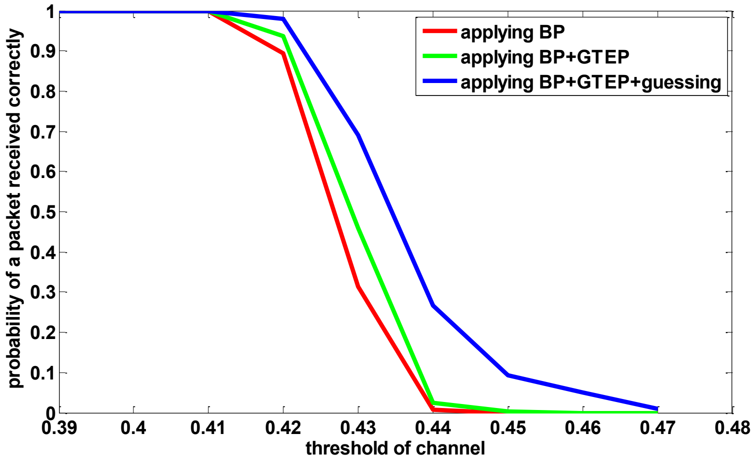

Figure 4.7. The simulation results for the code of length 2×10^3

Figure 4.7 shows the simulation results for the code of length 2×10^3. The performance of this code is more close to the theoretical threshold than the code with length of 10^3. Simulation results in figure 4.7 show that the performance of this code increased after applying the proposed method. At the channel erasure rate of 0.43, the probability of erasure before and after the applying the method are 0.1 and 0, respectively. The number of guesses used for both codes is identical. Therefore, both simulations in figure 4.7 and 4.8 use the same maximum number of guesses.

Another simulation is done for the number of guesses for both codes. Figure 4.8 shows the probability of guesses number that the decoder needs to make on the erased variable nodes for correcting a codeword. In this simulation, the channel erasure rate is considered to be 0.43. According to the simulation results, on average the LDPC code of length 2×10^3 needs larger number of guesses than the code of length 10^3. It is obvious that codes with larger length need more number of guesses.

Another comparison is done between the guessing algorithm and the new guessing algorithm. Figure 4.9 presents the number of guesses in the guessing algorithm and the new guessing algorithm. In this simulation, the channel erasure rate and the code length are 0.43 and 2×10^3, respectively. The simulation result shows that the new guessing algorithm needs smaller number of guesses than the guessing algorithm. Therefore, the complexity of the new guessing algorithm is less than the guessing algorithm.

4.4 Conclusion

In this chapter, we proposed a new method for improving the performance of an existing regular LDPC code without increasing the decoding complexity dramatically. In this work we showed that the combination of the three decoding algorithms: the standard BP, GTEP, the guessing algorithm increases the performance of the LDPC code. The problem of the guessing algorithm is its complexity. The complexity of the guessing algorithm reduces by reducing the number of possibilities. The number of possibilities can be degraded, if the decoder chooses a set of check nodes and poses assumptions on their adjacent variable nodes. The complexity of the new guessing algorithm does not increase exponentially with increasing the number of guesses. We applied the proposed algorithm on a regular (3,6) LDPC code, the simulation results in this paper show that the performance of a regular LDPC code can be increased from 0.42 to 0.43 with considering the maximum number of guesses equal to three.

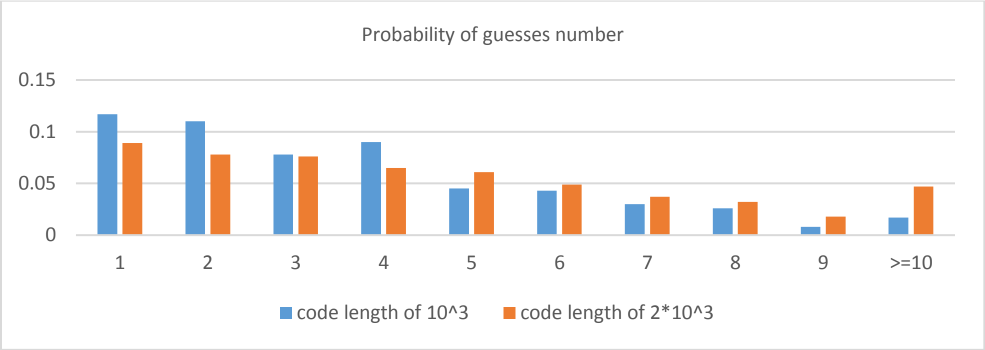

Figure 4.8. Probability of guesses number for channel erasure rate of 0.43

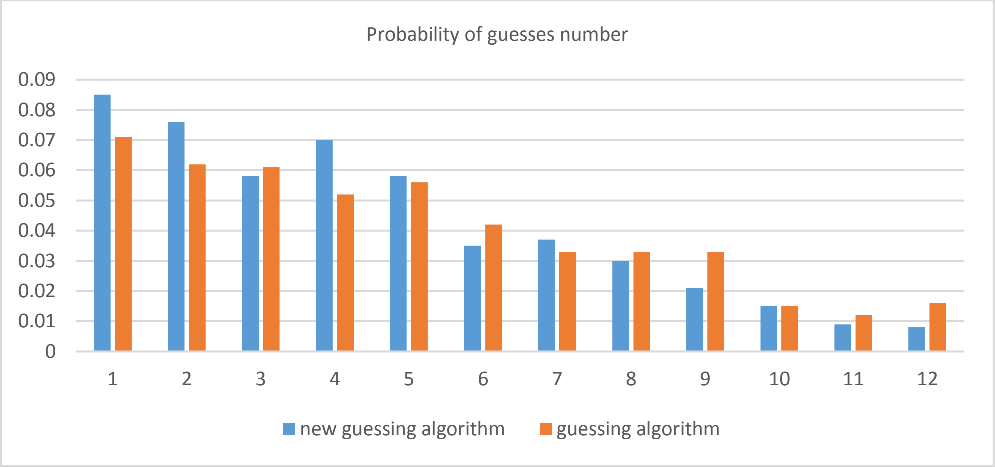

Figure 4.9. Probability of guesses number for channel erasure rate of 0.43 and code of length 2×10^3

5. Chapter 5 Conclusion and Future Works

5.1 Introduction

In this thesis we evaluated Low-Density Parity-Check (LDPC) codes in different terms. From constructing the LDPC code to the encoding and the decoding procedure of the LDPC code was evaluated. The performance and the encoding and the decoding complexity and the tradeoff between then are investigate in this thesis. Some methods to construct an LDPC code and to improve the performance are proposed in this thesis.

To increase the bandwidth efficiency, multiple access channel is needed. In chapter 3, first we investigated the 2 users-binary erasure multiple access channel (BEMAC). Since, the transmissions over the multiple access channel are not orthogonal, 2 users send information simultaneously. On the average half of the received message is erased and because of that, the channel is named BEMAC. In order to recover the messages for both sources a half rate code is needed. If one source send at full rate and the other one encode its message with a half rate code, at the receiver both messages can be recovered. According to the Shannon capacity, the capacity of the 2 users-BEMAC is 1.5. To achieve near the Shannon capacity a code of half rate which has a good performance over the Binary Erasure Channel (BEC) is needed. LDPC codes are good candidate and have good performance over the BEC. Therefore, an LDPC code of half rate is needed.

In this chapter, according to the density evolution, the best ensemble for the regular half rate code was theoretically found. The best ensemble is $(n, 3,6)$ which has the highest threshold among all the other ensembles. This ensemble can be considered as an optimum ensemble which results in the best performance with lowest decoding complexity. Therefore, the ensemble $(n, 3,6)$ is an optimum ensemble. There is two types of the LDPC codes; regular and irregular. The performance of the irregular LDPC code is better than the regular one. Since, the best ensemble is $(n, 3,6)$ and irregular LDPC codes have better performance, then an irregular LDPC code with a degree distribution was achieved from this ensemble is our interest.

There are advantages of the LDPC codes over the Turbo codes. However, the problem or the disadvantage of the LDPC code is its encoding complexity. The encoding complexity of the turbo codes is time linear, since the encoding complexity of the LDPC code of the length n is equal to the n^2. Therefore, the encoding complexity of the LDPC code is not time linear. In this chapter, we proved that to reduce the encoding complexity, the parity check matrix of the code must be approximately in the lower triangular shape. In this chapter we want constructed a half rate LDPC code which has good performance and low encoding complexity.

We proposed a method to construct LDPC codes. In the proposed method, 1's are spread in each row randomly. At the end the columns weight must be checked. We showed that in this method the probability that all columns have the weight greater than zero is zero. To remove the columns or rows of weight zero, some permutations are needed. Another method is that we can put a constraint that 1's are put in the selected places in each row. The selected place is removed from the available places list. This method guaranteed the weight of all columns and rows are greater than one. Next, we proposed a method to construct a lower triangular parity check matrix. To construct a lower triangular parity check matrix there are more constraint. The first constraint is the degree distribution pair. The second one is the constraint to keep the shape of matrix the lower triangular. In this thesis, we also consider the density of the parity check matrix in all rows of the matrix equally. For example, according to the ensemble $(n, 3,6)$ the number of ones in each row is equal to 6. This results in the different density in the rows of the lower triangular matrix. However, in the proposed method we kept the density in each row equally. It was done by changing the number of ones according to the number of elements that can be non-zero in each row.

In the rest of this chapter, an irregular half rate LDPC code is generated. The selected pair degree distribution polynomial has the same right degree distribution as the ensemble $(n, 3,6)$. The theoretical performance or the threshold of the constructed code according to the density evolution is 0.468. The generated parity check matrix of the code has the density as low as possible and also the density of the matrix in all rows is remained approximately equal. The encoder and the decoder for the generated code is implemented. The performance of the code is evaluated and the actual performance of the code is less than 0.468, due to the short cycles and finite length of the code. To overcome these problems and improve the performance of the code the guessing algorithm is applied. After the Belief Propagation (BP) algorithm gets stuck, the guessing algorithm is run. The guessing algorithm makes an assumption on a set of erased bits. After the assumption, the BP again starts the decoding. If all the erased bits are solved, then there is no need for another assumption otherwise another assumption is needed. The correct assumption results in the zero check-sum. If the check-sum is not zero, then the decoder change the value of assumed bits and continue the decoding until the check-sum zero is achieved.

In the chapter 4, the encoding complexity is not the primary concern. In this chapter, the performance and the decoding complexity is the primary concern. The decoding complexity of an LDPC code is first evaluated in this chapter. The lower bound of the code density to achieve the actual rate demonstrated that to have a better actual rate the lower bound on the density increases. Therefore, there is a tradeoff between the performance of the code and the complexity. In this thesis, we decided to choose a code with low decoding complexity and then improve the performance of the code.

To evaluate the performance of an LDPC code we started with the iterative decoding algorithms. One important class of these algorithms is BP. We investigate two scenarios when BP gets stuck and cannot solve the erased bits, therefore, the performance of the code degraded. The first scenario happened when there is a unique answer for the remained erased bits. BP cannot solve them due to the cycles in the graph and however, using the Gaussian elimination can find the answer. The second scenario happened when there is no unique answer for the erased bits. Bits in the first scenario can be found by Generalized Tree-Expected Propagation (GTEP) Algorithm if the successful decoding is satisfied for this algorithm. The complexity of the GTEP is as low as BP; therefore it does not add higher complexity.

The rest of the erased bits can be solved with the guessing algorithm. The complexity of the guessing algorithm is higher than the BP and GTEP. The complexity of the guessing algorithm increases exponentially with increasing the number of guesses. In this thesis to improve the performance of the existing code, we proposed a method. The proposed method is the combination of these three algorithm; BP, GTEP, guessing algorithm. The proposed method improves the performance of the code without increasing the complexity highly. The proposed method improve the performance is an efficient way.

In the proposed method first BP is run. If BP gets stuck then GTEP is run. Final, if there are still erased bits guessing algorithm is run. If after the BP guessing algorithm is run, the complexity of the decoder increases highly. When GTEP is run before the guessing algorithm the number of erased bits reduces and the set of guesses reduces. As a result, the decoding complexity reduces if the set of guesses decreases. Also, to decrease the decoding complexity more, a new guessing algorithm is proposed. In the new guessing algorithm the algorithm chooses a set of check nodes and makes an assumption on the variable nodes connected to them. However, the guessing algorithm chooses a set of variable nodes and makes an assumption on them. In the new guessing algorithm the number of possibilities reduces by the half. Therefore, the complexity of this algorithm reduces more. The proposed algorithm was applied on a regular half rate code. The optimum ensemble $(n, 3,6)$ was chose. The code was constructed based on the Gallager method. The proposed method was applied on the code and the simulation results showed an improvement in the code performance.

5.2 Future work

In this thesis, we just considered the 2 users-BEMAC. The performance and the complexity of the LDPC are improved over the binary erasure channel which is noiseless. In the future we will consider the Binary Symmetric Channel (BSC). The performance and the complexity of the LDPC code will be improved over the BSC.

Also, the proposed method to improve the performance of the regular LDPC code will be applied on the LDPC code of the Raptor code. In the future after applying the method, we will investigate the Raptor code in terms of the performance and complexity.

Bibliography

[1] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol.27, pp.379-423, 1948.

[2] M. Hagh and M. R. Soleymani, "Raptor Coding for Non-Orthogonal Multiple Access Channels", IEEE International Conference on Communications, vol., no., pp.1–6, June 2011.

[3] Khoueiry, B. (2016). Capacity Approaching Coding Strategies for Machine-to-Machine Communication in IOT Networks. (Unpublished doctoral dissertation). Concordia university, Montreal, Canada.

[4] R. G. Gallager, LowDensity Parity-Check Codes. MIT Press, Cambridge, MA, 1963.

[5] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” Electron. Lett., vol. 32, no. 18, pp. 1645–1646, March 1996, reprinted Electron. Lett, vol. 33(6), pp. 457–458, March 1997.

[6] A. Shokrollahi, “LDPC codes: An introduction,” Digital Fountain, Inc., Fremont, CA, Tech. Rep., Apr. 2, 2003.

[7] H. Tavakoli, M. Attari, and M. Peyghami, “Optimal rate for irregular LDPC codes in binary erasure channel,” in Proc. IEEE ITW, Paraty, Brazil, Oct. 2011, pp. 125–129.

[8] H. Tavakoli, M. Ahmadian, and M. Peyghami, “Optimal rate irregular low-density parity-check codes in binary erasure channel,” IET Commun., vol. 6, no. 13, pp. 2000–2006, Sep. 2012.

[9] W. Yu, M. Ardakani, B. Smith, and F. R. Kschischang, “Complexity-optimized LDPC codes for Gallager decoding algorithm B,” in Proc, IEEE Int. Symp. Information Theory (ISIT), Adelaide, Australia, Sep. 2005, pp. 1488–1492.

[10] M. Ardakani, W. Yu, B. Smith, and F. R. Kschischang, “Complexity-Optimized Low-Density Parity-Check Codes,” in Proc, IEEE Int. Symp. Information Theory (ISIT), Adelaide, Australia, Sep. 2005, pp. 1488–1492.

[11] N. Laouini, L. Ben Hadj Slama, A. Bouallegue, “Fast decoding of low density parity check codes,” Proc. 9th International Conf. HONET, pp. 52-56, 2012.

[12] H. Pishro-Nik and F. Fekri, “On decoding of low-density parity-check codes over the binary erasure channel,” IEEE Trans. Information Theory, vol. 50, no. 3, pp. 439–454, Mar. 2004.

[13] C. Measson, A. Montanari, and R. Urbanke, “Maxwell Construction: The Hidden Bridge Between Iterative and Maximum a Posteriori Decoding,” IEEE Trans. on Information Theory, vol. 54, no. 12, pp. 5277–5307, 2008.

[14] T. Richardson and R. Urbanke, “The capacity of low-density parity check codes under message-passing decoding,” IEEE Trans. Inform.Theory, vol. 47, pp. 599-618,2001.

[15] P. M. Olmos, J. J. Murillo-Fuentes, and F. P´erez-Cruz, “Tree-structure expectation propagation for decoding LDPC codes over binary erasure channels,” in Proc. 2010 IEEE International Symp. Inf. Theory, pp. 799–803.

[16] L. Salamanca, P. M. Olmos, J. J. Murillo-Fuentes, and F. P´erez-Cruz, “Tree Expectation Propagation for ML Decoding of LDPC Codes over the BEC,” IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 61, NO. 2, FEBRUARY 2013.

[17] T. Richardson and R. Urbanke, “Efficient encoding of low-density parity-check codes”, IEEE Trans. Inform. Theory, vol. 47, pp, 638-656, Feb 2001.

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

[19] M. Sipser and D. Spielman, “Expander codes,” IEEE Trans. Inform. Theory, vol. 42, pp. 1710–1722, Nov. 1996.

[20] D. J. C. MacKay, S. T. Wilson, and M. C. Davey, “Comparison of constructions of irregular Gallager codes,” in Proc. 36th Allerton Conf. Communication, Control, and Computing, Sept. 1998.

[21] M. R. Tanner, “A recursive approach to low complexity codes,” IEEE Trans. Inform. Theory, vol. 27, pp. 533-547, 1981.

[22] D. Haley, A. Grant, and J. Buetefuer, “ Iterative encoding of low density parity-check codes,” in Proc. GLOBECOM 2002, Taipei, Taiwan, 2002, pp. 1289–1293.

[23] MacKay, D. J C, “Good error-correcting codes based on very sparse matrices”, IEEE Transactions on Information Theory, Vol.45, No.2, pp.399-431, Mar. 1999.

[24] Fossorier, M.P.C., “Quasicyclic low-density parity-check codes from circulant permutation matrices”, IEEE Transactions on Information Theory, Vol.50, Vo.8, pp.1788-1793, Aug. 2004.

[25] H. Tavakoli, M. Attari, and M. Peyghami, “Optimal rate for irregular LDPC codes in binary erasure channel,” in Proc. IEEE ITW, Paraty, Brazil, Oct. 2011, pp. 125–129.

[26] T. Richardson and R. Urbanke, "Modern Coding Theory", Cambridge University Press, 2008.

[27] P. Oswald and A. Shokrollahi, “Capacity-achieving sequences for the erasure channel,” IEEE Trans. Inf. Theory, vol. 48, no. 12, pp. 3017-3028, Dec. 2002.

[28] A. Khandekar and R. McEliece, “On the complexity of reliable communication on the erasure channel,” in Proc. IEEE ISIT, Washington, DC, USA, 2001, p. 1.

[29] S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications. 2nd Edition, Prentice-Hall, 2005.

[30] D. Divsalar, H. Jin, and R. J. McEliece. "Coding theorems for ‘turbo-like’ codes." Proc. 36th Allerton Conf. on Communication, Control and Computing, Allerton, Illinois, Sept. 1998, pp. 201–210.

[31] T. Richardson, A. Shokrollahi, and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. Information Theory, vol. 47, pp. 619-637, Feb. 2001.

[32] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Analysis of Low density codes and improved designs using irregular graphs," IEEE Trans. Inf. Theory, vol. 47, pp. 585-598, 2001.

[33] Cai, Z., Hao, J., Tan, P.H., Sun, S., Chin, P. S., “Efficient encoding of IEEE 802.11n LDPC codes”, Electronics Letters , Vol.42, No.25, pp.1471-1472, Dec. 2006.

[34] Li, Z., Chen, L., Zeng, L., Lin, S., Fong, W.H., “Efficient encoding of quasi-cyclic low-density parity-check codes”, IEEE Transactions on Communications, Vol.54, No.1, pp.71-81, Jan. 2006.

[35] Lan, L., Zeng, L., Tai, Y.Y., Chen, L., Lin, S., Abdel-Ghaffar, K., “Construction of Quasi-Cyclic LDPC Codes for AWGN and Binary Erasure Channels: A Finite Field Approach”, IEEE Transactions on Information Theory, Vol.53, No.7, pp.2429-2458, July 2007.

[36] Zhang, L., Huang, Q., Lin, S., Abdel-Ghaffar, K., Blake, I.F., “Quasi-Cyclic LDPC Codes: An Algebraic Construction, Rank Analysis, and Codes on Latin Squares”,IEEE Transactions on Communications, Vol.58, No.11, pp.3126-3139, Nov. 2010.

[37] Zhang, G., Sun, R., Wang, X., “New quasi-cyclic LDPC codes with girth at least eight based on Sidon sequences”, International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Aug. 2012.

[38] H. Tavakoli, M. Ahmadian, and M. Peyghami, “Optimal rate irregular low-density parity-check codes in binary erasure channel,” IET Commun., vol. 6, no. 13, pp. 2000–2006, Sep. 2012.

[39] D. Burshtein and G. Miller, “An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel,” IEEE Trans. Inform. Theory, vol. 50, no. 11, pp. 2837–2844, nov 2004.

[40] V. Strassen, “Gaussian elimination is not optimal,” Numerische Math., vol. 13, pp. 354–356, 1969.

[41] D. Coppersmith and S. Winograd, “Matrix multiplication via arithmetic progressions,” in Proc. 19’th ACM Symp. Theory Comp., pp. 1-6, 1987.

[42] M. G. Luby, M. Mitzenmacher, M. A. Shokrohalli, and D. A. Spielman, “Improved low-density parity-check codes using irregular graphs,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[43] T.J.Richardson and R.Urbanke, “The capacity of low-density parity-check codes under message passing decoding”, *IEEE Trans.Inform*. Theory, vol. 47, pp. 599-618, Feb, 2001.

[44] U. Reimers and A. Morello, “DVB-S2, the second generation standard for satellite broadcasting and unicasting,” Int. J. Satell. Commun. Network., vol. 22, no. 3, May–Jun. 2004.