DESIGN OF HIGH PERFORMANCE LOW LATENCY RATELESS CODES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
NADHIR IBRAHIM ABDULKHALEQ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF PHILOSOPHY OF DOCTORATE IN
THE DEPARTMENT OF
ELECTRONIC AND COMMUNICATION ENGINEERING

SEPTEMBER 2017

Title of the Thesis: **Design of High Performance Low Latency Rateless Codes**

Submitted by **NADHIR IBRAHIM ABDULKHALEQ**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Philosophy of Doctorate.

Prof. Dr. Kemal İDER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Philosophy of Doctorate.

Assoc. Prof. Dr. Orhan GAZİ
Supervisor

Examination Date: 13.09.2017

Examining Committee Members

Prof. Dr. Ali Özgür YILMAZ	(ODTÜ)
Assoc. Prof. Dr. Orhan GAZİ	(Cankaya Univ.)
Assist. Prof. Dr. Barbaros PREVEZE	(Cankaya Univ.)
Assist. Prof. Dr. Javad RAHEBI	(THK Univ.)
Assist. Prof. Dr. Göker ŞENER	(Cankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Nadhir Ibrahim BDULKHALEQ
Signature :
Date : 13.09.2017

ABSTRACT

DESIGN OF HIGH PERFORMANCE LOW LATENCY RATELESS CODES

ABDULKHALEQ, Nadhir Ibrahim
Ph.D., Electronic and Communication Engineering Department
Supervisor: Assoc. Prof. Dr. Orhan GAZI

September 2017, 109 pages

Luby Transform (LT) codes are one of the best rateless codes mainly designed for binary erasure channel. The characteristics of such codes perfectly performing when used with bulk data files, however a performance degradation has been observed when using them with short length messages. In this thesis, we present a new design for rateless codes, particularly an efficient LT codes using robust soliton distribution (RSD) as a degree generation method and tested in both binary erasure channel (BEC) and noisy channels like the additive white Gaussian noise (AWGN) channel. First, a new proposed decoding technique is defined as belief propagation-pattern recognition (BP-PR) is implemented to enhance the decoding ability of the conventional (BP) algorithm to overcome the problem of losing degree-one coded symbols which caused early decoding termination. The simulation results approve the improvement of the BP-PR when used with LT-RSD and outperforms the bit error rate (BER) records for the state of art techniques like memory-based robust Soliton distribution using conventional BP (LT-MBRSD-BP) or the Gaussian elimination assisted belief propagation (LT-RSD-BP-GE) and improve the records for the BER when used with MBRSD, ISD and optimal degree distribution (ODD), to form the new code called (LT-MBRSD-BP-PR),(LT-ISD-BP-PR) and (LT-ODD-BP-PR) respectively.

Second, a new efficient deterministic encoding technique using deterministic degree generator with random data selection (LT-DE) is applied for extremely short data lengths. The degree generation method is based on creating the degrees in a repeated frame with a limited upper value called repetition period (R_p) and the data symbols are chosen sequentially from a truncated data file. The data file is truncated to segments of length (R_p) and each segment is chosen based on a random sequence. Testing this (LT-DE) against (LT-RSD-BP-PR), (LT-MBRSD-BP-PR) and (LT-ODD-BP-PR) in a BEC environment had approved the superiority of such code over all the other mentioned techniques. It has the lower error floor and higher successful decoding rate with minimum overhead and computational cost. The formation of this (LT-DE) associates a mutual relation between the successive coded symbols which motivate us to present a new sequential decoding technique mainly used over (AWGN) channel. With such new encoding-decoding technique LT codes can approach the decoding complexity cost of Raptor codes with smaller overhead and less encoding complexity as well.

Keywords: Luby transform codes, short length, Pattern recognition, deterministic encoding, sequential coding.

ÖZ

YÜKSEK PERFORMANSLI DÜŞÜK GECİKMELİ ORANSIZ KOD TASARIMI

ABDULKHALEQ, Nadhir Ibrahim
Doktora., Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Tez Yöneticisi: Doç. Dr. Orhan Gazi

Eylül 2017, 109 sayfa

İkili silme kanalları için tanımlanan Luby dönüşüm kodları bilim camiasında var olan ilk oransız kodlardır. Bu kodların kodlayıcı kısımları veri paketlerinin rasgele bir şekilde seçilerek birleştirilmesi mantığına göre hareket eder. Veri paketleri ise rasgele olarak belli bir dağılıma göre üretilen sayılara göre yapılırlar. Oransız kodların alıcı tarafındaki çözümlenme başarıları bu kodların gönderici tarafında çözümlenebilir şekilde kodlanmalarına bağlı olmaktadır. Eğer kodlama tasarımı etkili bir şekilde yapılmazsa alıcı tarafında başarılı çözümleme yapılamaz ve veri paketleri birbirlerinden ayırt edilemez. Alıcı tarafında derecesi bir olan kod sözcüğü çözümleme işleminin herhangi bir anında bulunamazsa çözümleme işlemi devam etmez, ve çözümleme işlemine son verilir. Bu da o andan itibaren sonraki veri paketlerinin elde edilememesi demektir. Bu tez çalışmasında çözümleme işlemi esnasında derecesi bir olan kod sözcüğü bulunmaması durumunda çözümleme işlemine devam edebilmek için bir yöntem öneriyoruz. Önerilen yöntemle çözümleme esnasında meydana gelen tıkanmalar giderilerek çözümleme işlemine devam edilmesi sağlanmaktadır. Bilgisayar benzetimleri ile yapılan çalışmalarda önerilen yöntemin LT kodlarının performanslarını arttırdığı görülmüştür. Ve bu performans yükselmesi LT kodlarının kullandığı derece üretimi için kullanılan dağılımlara bağlı olmadığı görülmektedir. Önerilen yöntem LT-RSD ve LT-MBRSD-BP kodlarına bütünleştirilmiş ve bu kodlarda elde edilen performans artışı bilgisayar benzetimleri ile gözlemlenmiştir. Önerilen performans artırım yöntemi karar yayılımı örüntü tanıma ile isimlendirilmiştir. Önerilen yöntem Gaussian elimine karar yayılımı yöntemiyle benzeşimler gösterse de, önerilen yöntemin işlem karmaşıklığı daha azdır. Bu nedenle önerilen yöntemin zaman gecikmesi literatürde var olan Gauss elimine karar yayılımı yöntemine göre daha azdır.

LT türü kodlar büyük uzunluklara sahip olan veri dosyaları için iyi performans göstermektedir. Diğer yandan kısa uzunluktaki veri dosyaları için bu tür kodların performansları iyi olmamaktadır. Bunun ana sebebi kısa uzunluktaki dosyalar için çözücü tarafında bir dereceli kod sözcüklerinin daha yüksek olasılıkla bulunmamalarıdır. Bu tez çalışmasında kod derecelerinin üretimi için deterministik bir yöntem öneriyoruz. Önerilen yöntem küçük uzunluktaki veri dosyaları için oldukça iyi performans göstermektedir. Önerilen deterministik derece üretimi derecelerin periyodik bir şekilde üretilmesi kuralına dayanmaktadır. Dereceler R_p periyodu ile periyodik bir şekilde üretimektedirler. Veri dosyası R_p adet pakete bölünüp daha sonra ise üretilen dereceler göz önüne alınarak birleştirilmektedirler. Bilgisayar benzetimleri sonucunda elde edilen sonuçlara göre önerilen yöntem rasgele derece üretim yöntemlerine göre daha düşük BER oranlarına, daha küçük başlık bilgisine ve daha az işlem miktarına sahip olmaktadır.

Anahtar Kelimeler: Luby dönüşüm kodları, oransız kodlar, karar yayılımı, kısa uzunluk, örüntü tanıma, deterministik kodlama, sıralı kodlama.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assoc. Prof. Dr. Orhan GAZI for his supervision, special guidance, suggestions, and encouragement through the development of this thesis.

Many thanks for the thesis committee represented by Prof. Dr. Ali Özgür YILMAZ and Assist. Prof. Dr. Barbaros PREVEZE, for their valuable advices during preparing this thesis.

I would like to express my truthful respect for the government of Iraq represented by the Ministry of Higher Education and Scientific Research for their financial support to complete my study.

TABLE OF CONTENTS

CHAPTERS:

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

LT	Luby Transform
BEC	Binary Erasure Channel
AWGN	Additive White Gaussian Noise
SDC	Stochastic Discrete Channel
DMC	Discrete Memoryless Channel
FEC	Forward Error Correction
BC	Block Codes
BW	Bandwidth
SNR	Signal-to-Noise Ratio
BCH	Bose-Chaudhuri-Hocquenghem
TPC	Turbo Product Codes
CC	Convolution Codes
TCC	Turbo Convolutional Code
LDPC	Low density parity check code
RSD	Robust Soliton Distribution
MBRSD	memory-based robust Soliton distribution
BP	Belief Propagation
BP-PR	Belief Propagation with Pattern Recognition
BER	Bit Error Rate
DE	Deterministic Encoding
MLP	Maximum Likelihood Probability
SEA	Sequential Encoding Approach
SDHD	Soft-Demodulation-Hard-Decoding
SD	Sequential Decoding

BP-GE	Belief Propagation-Gaussian Elimination
XOR	Exclusive OR
SPA	Sum-Product Algorithm
BPSK	Binary Phase Shift Keying
LLR	Log-Likelihood Ratio
RSR	Ripple Size Ratio
UEP	Unequal Error Protection
SLT	Shifted Luby Transform
DSLT	Diverse Shifted Luby Transform
MRSD	Modified robust soliton Distribution
DLT	Distributed Luby Transform
EEP	Equal Error Protection
GF	Galois Field
PRSD	Poisson Robust Soliton Distribution
AFSA	Artificial Fish Swarm Algorithm
RBD	Ripple-Based Distribution
BSC	Binary Symmetric Channel
SLT	Systematic Luby Transform
GE	Gaussian Elimination
OFG	On the Fly Gaussian elimination
LT-W	Luby Transform-Wiedemann
LTAM	LT code with Added Memory
BAWGN	Binary Additive White Gaussian Noise
LT-URT	Luby Transform-Unequal Recovery Time
SC-LT	Spatially Coupled-Luby Transform
PSNR	Peak Signal-to-Noise Ratio

EEP	Equal Error Protection
WBAN	Wireless Body Area Network
RS	Reed-Solomon
FSO	Free Space Optical
DNC	Distributed Network Coding
NGB-W	Next Generation Broadcast Network Wireless
TUs	Terminal Users
LTDC	LT codes based Distributed Coding
WSN	Wireless Sensor Networks
UAC	Underwater Acoustic Communications
AD	Average degree
SO	Symbol Operations
ENS	Extra Needed Symbols
URS	Unrecovered Symbols
CRF	Complete Recovery Frames
OVHD	Overhead

CHAPTER 1

INTRODUCTION

1.1 Basics of channel coding

The forefather of the recent information and coding theory is Claude E. Shannon [1,2], who establishes the bases for the digital communication when publishing his paper in 1948. His ideas to form reliable and efficient communication system has still the constitution for all the designers whom intended to develop the performance of such systems. An optimistic proof stated in his paper, Shannon proved that reliable communication of data can be achieved over noisy channels which means that even though part of the transmitted message has been corrupted or lost, the receiver is able to predict the content of that message. This is the essential meaning of what called channel coding theorem [3,4]. This communication reliability is feasible on condition; which state that:

$$R \leq C \tag{1.1}$$

where R is defined as the number of transmitted bits per channel use, and C represents the channel capacity which indicates the amount of data that can be send during each channel use. In such a case, Shannon has set the big challenge for the code designers to construct a code with a rate R close to the channel capacity C. The code rate is defined as (the ratio between the information bits, usually denoted by k to that of the code bits, usually denoted by N). and mathematically will be:

$$R = k/N \tag{1.2}$$

Sending these information bits over a noisy channel which corrupts or destroys some of them forces the transmitter to guard the data with a redundant bit, in other words the length of the coded bits usually is larger than that of the information bits (i.e. $N > k$). İn order to emphasize the function of the channel coding, let us figure out the main elements of any digital communication system. As shown in Figure 1, the source output which could be either analog or digital signal is first converted to a set of binary symbols with efficient representation and compressed shape. This part of the system is mentioned as the source encoding or data compression. The new shape of the source signal which we shall call it the information stream will be treated to make it immune from the effect of the channel noise. This action in the digital system describes the channel coding part in which the information stream will be represented by a coded stream with extra redundant symbols. These coded symbols will be used to help the receiver in extracting the information stream by using the previous knowledge of the encoding technique. One of the main advantages of the channel coding is to increase the transmission reliability and enhance the fidelity of the received signal [5] .

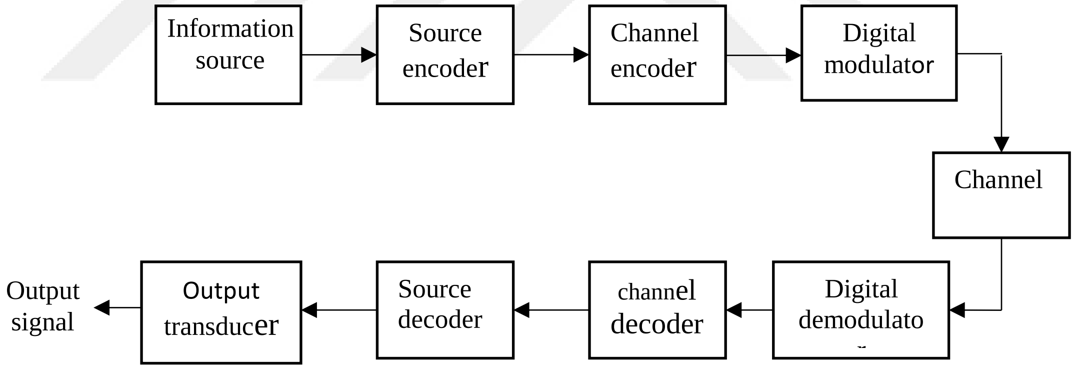


Figure 1 Basic elements of a digital communication system

The digital modulator acts as an interference block to adapt the coded stream to be suitable for transmission through the communication channel. This encoded stream will be mapped to waveform patterns, each waveform could represent one or more information symbols, i.e. b coded symbols can be transmitted at a time by using $M = 2^b$ different waveforms $s_i(t), i = 1,2, \ldots\ldots, M - 1$ each waveform represents one state of the encoded stream. it is clear that when $b = 1$, then we deal with binary modulation and when ($M > 2$), it will be called M-ary modulation. The communication channel represents the connecting medium between the transmitter and the receiver. This channel could be the free space for wireless communication while it may be in the form of wire lines or optical fiber cables for telephone channels. More details with mathematical representation for the communication channels will be presented in the next section. A process replica will be held on the receiver side, first the digital demodulator reshapes the received corrupted waveforms into estimated binary symbols. The channel decoder will be equipped with coding technique that will be used to extract the information symbols from these estimated coded symbols out from the digital demodulator. The combination of the modulator-decoder plays a significant rule in reducing the probability of error occurrence while extracting the source information from the received noisy waveforms. For some modulation methods which have orthogonal signaling waveforms, they let the number of waveforms $M \rightarrow \infty$ which leads to make the probability of error arbitrarily small. This means that we can approach the capacity of the AWGN channel when the bandwidth expansion factor $B_e = W/R \rightarrow \infty$, but this will be an expensive price and inefficient use of the bandwidth especially when large information blocks k are sent [1-6]. The designers have found a solution for such problems through the insertion of the coding techniques combined with the modulation process. By definition, coding is the mapping of an information symbols to a set of coded symbols. Shannon [1,2] comprises three main kinds of coding named as source coding, error-control coding or channel coding and secrecy coding (also called cryptography). İn this introduction, we focus on the channel coding in which a brief definition will be presented for the conventional block codes then discuss the needs for invention of the rateless codes and their applications. In the following section a brief description for the well-known communication channels will be presented.

1.2.Channel characteristics

The communication channel can be defined as the physical link between the transmitter and the receiver. According to the type of the transmitted information the physical shape of this link will be adjusted. A pair of wires is the channel when the transmitted data is an electrical signal while an optical fiber is used as a communication channel when the modulated light beam is used to carry the information message. Also, the free space is another shape of the channel when the antenna is used as an interface tool to adapt the electrical signals to the electromagnetic radiation. The sound or an acoustic wave is the suitable shape for the signals that can be transmitted in the water communication channels. Another class of communication channels is data storage media, such as magnetic tape, magnetic disks, and optical disks [5]. To describe a model of the communication channel, we need to define the channel as a stochastic discrete channel (SDC). Information theory has established the concepts of entropy and mutual information which will be used to calculate the channel capacity which is an essential property of the channel. In this thesis, two standard classes of discrete communication channels will be discussed. These channels are the binary erasure channel (BEC) and the binary additive white Gaussian (AWGN) channel. Any discrete memoryless channel (DMC) can be characterized by the three elements ($A_{in}, A_{out}, P_{y/x}$), where A_{in} and A_{out} represent the input and output alphabet respectively and $P_{y/x}$ is the transition probability of the channel. If the input alphabet represents a random variable X and has a priori distribution (x) , then the probability of receiving y when sending x through the channel is calculated using Bay's rule as [9]:

$$P(y) = \sum_{x \in A_{in}} P(y/x) \,.\, P(x) \tag{1.3}$$

Definition 1.1. The extent of information of any random variable defined on a field of alphabet A_{in} is measured by the entropy, which mathematically written as [9]:

$$H(x) = -\sum_{x \in A_{in}} P(x) log_2 P(x) = \sum_{i=1} P_i \log_2 P_i \tag{1.4}$$

where P_i represents the probability of $i-th$ element of the input random variable$X$. For the binary case ($q = 2$), the entropy is bounded by:

$$0 \leq H(x) \leq log_q(x) \tag{1.5}$$

Definition 1.2. The mutual information between two random variables ($X\ and\ Y$) or the input and output alphabets of the DMC, is the indication for how much information one variable discovers about the other. This can be interpreted as [9]:

$$I(X:Y) = \sum_{x\in A_{in}, y\in A_{out}} P(x,y) log_2 \frac{P(x,y)}{P(x)P(y)} \tag{1.6}$$

Using Bay's theorem, we can re-write the mutual information as a function of the predefined entropy, which produces:

$$I(X;Y) = H(Y) - H(Y/X) \tag{1.7}$$

$$I(X;Y) = H(X) - H(X/Y) \tag{1.8}$$

Definition 1.3. C denotes the channel capacity of a DMC and will be defined as the maximum of mutual information over all priori distributions [9].

$$C = \underbrace{\max}_{P(x)} I(X;Y) \tag{1.9}$$

The capacity is measured in bits per channel use. This capacity will be changed according to the type of the channel. In the digital communication systems, there are many models for such channels, in the following sections we will present two of these channels which we use to examine our modification on the Luby transform (LT) codes.

1.2.1. Binary erasure channel (BEC)

This channel represents one of the basic channel models applied in coding and information theory. Figure 2 illustrate the description for such channel, dealing with a binary transmission, the transmitter sends either 1 or 0 as a data or encoded bits. Differ from other channel models, there will be no decision on the corrupted received bits (which are denoted by the question mark "?") and the receiver declares that these are lost bits and a retransmission request will be sent by the receiver in order to correctly receive them.

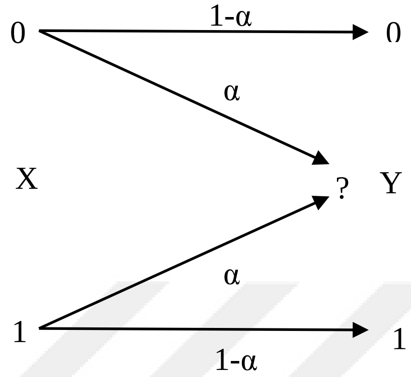


Figure 2 BEC model

To calculate the channel capacity, we need to recall equation (1.8):

$$\because I(X;Y) = H(X) - H(X/Y)$$

Where $H(X/Y)$ represent the conditional entropy, which defines the amount of information about the random variable X when all the cases of the random variable Y are given, and as the case for the above channel has three states for Y ($y = 1. y = 0\ and\ y = ?$) then the mutual information formula will be given as [9,10]:

$$I(X;Y) = H(X) - \sum_{\varphi \in Y} P(y = \varphi) H(X/y = \varphi) \qquad (1.10)$$

Substituting the three states of y in the above equation, to get:

$$I(X;Y) = H(X) - p(y = 0)H(X/y = 0) - p(y = 1)H(X/y = 1) - p(y = ?)H(X/y = ?) \qquad (1.11)$$

Keeping in mind that the case of (y =?) represents the erasure case. The conditional entropy $H(X/Y)$ means that we can conclude the value of the x when have a full knowledge of y. For the state y =? There will be no information can be used from the conditional entropy and hence $H(X/y = 0) = H(X/y = 1) = 0$, and $H(X/y =?) = H(X)$. We denote the probability of erasure as $p(y =?) = \alpha$, substituting all these conclusions in equation (1.11) to get:

$$I(X;Y) = H(X) - \alpha H(X) = (1-\alpha)H(X) \qquad (1.12)$$

Then in order to find the channel capacity, we need to get the maximum of mutual information given by equation (1.12) over all the possible events of the random variable X.

$$C = max\ \underbrace{I(X;Y)}_{p(x)} = (1-\alpha) maxH(X) = (1-\alpha) \qquad (1.13)$$

1.2.2. Additive white Gaussian noise channel (AWGN)

For some cases in the digital communication system we need to simulate certain signals in to a continuous form. For such cases continuous variables instead of the discrete variables are modeled. As example of such signals, the background noise in the digital communication systems which is typically Gaussian (i.e. Normal distributed). The continuous representation of these signals can be represented as:

$$Y = X + Z \qquad (1.14)$$

Where X represent the information part and the Z is the Gaussian noise added signal. To proceed with the derivation of the AWGN channel capacity, first the power of the transmitted information is defined as:

$$P = E[X^2] \tag{1.15}$$

where E is the expectation of a random variable. Then we can define the Gaussian channel as a time -discrete channel defined by equation (1.14) and the Z notation in this equation is a normal distributed Gaussian noise usually denoted by $Z \in \mathcal{N}(0, N)$.

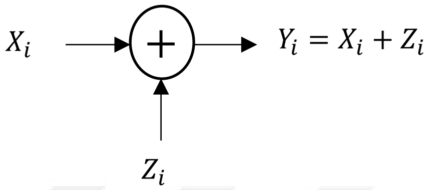


Figure 3 AWGN channel model

The channel capacity of the AWGN channel will be derived under a power constrain at the transmitter side, i.e. $E[X^2] \leq P$. Now following the same procedure while calculating the channel capacity for the BEC, first the mutual information specified in equation (1.7) will be derived for this channel case:

$$I(X; Y) = H(Y) - H(Y/X)$$

The second part of the above equation will be modified according to information gained from equation (1.14) [9]:

$$H(Y/X) = H((X + Z)/X) = H(Z/X) = H(Z) \tag{1.16}$$

The above equation gives another explanation for the mutual information in this channel which states that the mutual information is the deference in entropy between the output signal and the noise,

$$I(X; Y) = H(Y) - H(Z) \tag{1.17}$$

the noise has represented as a normal distribution with zero mean and variance N ,then this will be used as [10]:

$$H(Z) = \frac{1}{2}\log(2\pi e\ N) \qquad (1.18)$$

Now maximizing the output entropy over x will be bounded by the Gaussian entropy,

$$\underbrace{max}_{f(x),P} H(Y) \leq \frac{1}{2}\log(2\pi e\ \sigma^2) \qquad (1.19)$$

The equal sign in equation (1.19) leads to $Y \in \mathcal{N}(0, \sigma^2)$,but the result of addition of two Gaussian distribution variables leads to a Gaussian one, then by letting $X \in \mathcal{N}(0, \sqrt{P})$ and $\sigma^2 = P - N$, the expression for the channel capacity will be [10]:

$$\begin{aligned} C &= \underbrace{max}_{f(x),P} I(X;Y) \\ &= \underbrace{max}_{f(x),P} H(Y) - H(z) \\ &= \frac{1}{2}\log(2\pi e\ (P+N)) - \frac{1}{2}\log(2\pi e\ N) \\ &= \frac{1}{2}\log(\frac{2\pi e(P+N)}{2\pi eN}) \\ C &= \frac{1}{2}\log(1+\frac{P}{N}) \qquad (1.20) \end{aligned}$$

Equation (1.20) gives the formula for the capacity of a Gaussian channel with a power constrain P and noise variance N. This equation will be modified when dealing with band limited signals where $f_{max} \leq W$ and a band limited noise with power spectra $R_z(f) = \frac{N_o}{2}, |f| \leq W$,then the capacity for a channel $y(t) = x(t) + z(t)$ to be:

$$C = W\log(1+\frac{P}{WN_o}) \qquad (1.21)$$

Shannon [1, 2] had approved that the channel capacity is attainable no matter how the channel is noisy. In order to achieve this goal, the designers try to adjust the transmitted information to be more immune to the channel effects. One of the most efficient solution for such challenges was applying the channel coding which succeeds to transmit an information amount almost equal to that of the channel capacity. In the following section a short definition for the main types of the channel codes or error-control-codes with a condense introduction to the essential features of the rateless codes.

1.3. Features of the rateless codes

The circumstances and the channel conditions for each application in the digital communication systems are the main factors that will affect the choice for which type of error control codes can be used. Some of these factors like, the available physical properties of the transmission channel (the available bandwidth (BW), signal-to-noise-ratio(SNR), distortion level…... etc.), the requirement for the error probability after decoding, the available transmitted power, the allowable delay caused by decoding process or the amount extra processing operation for the used codes. The rate of the code will be the factor that can be used to classify the codes as:

1.3.1. Fixed rate codes:

All block codes, convolution codes as well as trellis coded modulation codes may be considered to have fixed rate when using as FEC codes. In the FEC technique the receiver utilizes the coding mechanism in order to correct the detected errors in the received messages. Differ from that of the FEC codes, another pre-determined rate codes usually called error detection codes which are used in an automatic repeat request (ARQ) mechanism. This ARQ techniques sends little redundant extra bits with the code which will be used in receiver to detect the occurrence of the errors, and if there is, the receiver will be asked for either retransmission or send more redundant bits.   The rate of the code is denoted by $R = k/N$ with code length N greater than that of the data length k. An extra coded symbol called parity symbols are added to that of the data symbols to form the code. The decision for the code length is taken to be competent for the worst case of the channel conditions. The receiver utilizes the added redundant symbols within the code to correct the errors occurs in the received symbols. Linear block codes are one of such codes which has simple encoding and decoding structure based on algebraic operations to form the codewords. The main criterion for such codes is the sum of two codewords leads to another codeword. The linear block code C_i is $q - ary$ usually represented as $(N, k, d_{min})_q$ where the input information x_i and the codewards $\in \mathbb{F}_q$. This $\mathbb{F}_q$ denotes the Galois field representation and d_{min} is the symbol for the minimum hamming distance which is defined by:

$$d_{min} = min\{d_H(a,b)|a,b \in C, a \neq b\} = min\,\{w_H(a)|a \in C, a \neq 0\} \qquad (1.22)$$

There are many types of such codes whom have fixed rate and designed to be used in FEC without the need for a feedback request. In this brief introduction, we can mention four groups of these well-known codes, first: error detection codes such as parity check codes, arithmetic redundancy check codes and cyclic redundancy check codes. Second: block error correction codes like, Hamming codes, BCH (Bose-Chaudhuri-Hocquenghem) cyclic block codes, Reed-Solomon cyclic block codes and Turbo Product Codes (TPC). Third: Convolution codes like, Tradition, Viterbi Decoding, Turbo Convolutional Code (TCC) and Low density parity check code (LDPC).

1.3.2. Rateless codes:

This type of coding provides an excellent solution for the demands which need to distribute large, intense files on the internet to enormous receivers. The achievement of such transmission should be done with high reliability, low network overhead and minimum infrastructure complexity. The first fountain or rateless code, form theoretical point of view, was the Reed Solomon codes [11]. This code was an efficient fountain code cause receiving k coded symbols will able to recover the k source message symbols. However, two important limitations for this code makes it more suitable for short length messages. Number one is the limitation of the constructed coded symbols, which is limited by the field size $F = 2^8$ or 2^{16}. The second limitation is the quadratic decoding complexity for such codes. An efficient solution for the pre-mentioned limitations especially for bulk data files is presented by John W. Byers, Michael Luby, and Michael Mitzenmacher in their work which they call it a digital fountain coding [12,13]. Dealing with bulk data files in a broadcasting scenario has several serious problems such as the case when some of the re-transmitted packet are not lost for some destinations which makes retransmission overcast the channel overcast the channel. In the case that the transmitter can respond to the requests, the retransmitted packets are often useful for a small portion of the receivers which means a waste of the bandwidth usage. The early attempts for constructing such types of codes has the shape of FEC erasure codes. The reason for shaping the code as erasure code is that each file will be sliced into k data packets and the code will be constructing by adding several redundant packets and the transmitter continues sending these coded packets. When the receiver collects sufficient number of coded packets N, it can be processed to recover the data packets even if some of them have been lost. It is clear that N is not fixed or even determined by the transmitter but it will be decided on the fly by the receiver, as a result such codes have variable code rate and will be named as rateless codes. The interesting feature of such fountain codes is that we can distribute a large file to number of receivers and the receiver can fill their decoder with any sufficient coded packets and they will be able to recover the original source file. In such scenario, all the receivers just send one acknowledgment to the sender after recovering the k data packets and this will be the best solution for the congestion that occurs when using the tradition fixed rate codes when a retransmission is done to recover the lost packets. These types of codes have several developments stages.

A notable advance in the development of such codes was the invention of Tornado codes [14,15], which are a type of LDPC codes considered for erasures [16]. Briefly, these codes can be represented by a graph has many layers of nodes. The first layer connects each data packet to one coded packet while the subsequent layers produce their packets from combining random number of the previous layer packets. A simplification for Tornado codes has been presented in [14,15] which uses only two encoding layers while maintains the property of having linear encoding and decoding times which is proportional to $Nln(1/\varepsilon)$ required for recovering k data packets after collecting $N = k + \varepsilon$ packets. Later, Luby [18] has designed the first typical definition of a rateless code with his Luby transform (LT) code. More simplification for the design of the code has been applied by Luby, the graph representation seems implicit, instead of explicit as it was for the Tornado codes. A pre-designed degree distribution is used to generate an integer d which determined the number of data packets must be combined to form the coded packets. Both sides of the communication system share the same seed to generate this random degree distribution. In chapter two we will present more details for the design of LT code and literature review for several types of degree distributions which represent the key for improving the performance of such codes.

A further step in the design of rateless codes has been achieved by Shokrollahi [18] with his patent in designing the new Raptor code. This code has overcome the drawbacks of LT codes mainly in reducing the average degree for producing each encoded packet from $O(\ln(k/\delta)$ for LT codes [18], where δ represent the probability of decoder failure, to $O(\ln(1/\varepsilon))$ for Raptor codes [19]. In addition, Raptor codes succeed with high probability to recover the k source packets from any $k(1+\varepsilon)$ output packets using only $O(kln(1/\varepsilon))$ symbol operations compared to that of LT codes which needs to collect any $k + O(\sqrt{k}ln^2(k/\delta))$ to recover the k source packets by using $O(k(\ln(k/\delta))$. This improvement has been done for the raptor codes by cascading the LT code with a pre-code has fixed rate such as LDPC code.

The linearity of encoding-decoding complexity as well as the compatibility of the rateless codes to the applications of multicasting transmission have nominate these codes to be one of the standards codes used in internet and mobile communications. More details for an intense vision of the applications of rateless codes will be listed in detail in the next chapter [20-23].

1.4. Overview of thesis contributions

Luby transform codes approved their suitability as an excellent erasure codes that could be used to distribute large data files to multiusers communication systems.

During the last decade, many studies have been presented to enhance the performance of such codes, most of them deal with bulk data files.

In this thesis, more investigation has been done on the analysis of the degree distribution which is the core for any improvements in LT code design. As it was emphasized in our introduction, the LT codes [18] is mainly designed for bulk data sizes and we found that the degree distribution of this code which is known as robust Soliton distribution (RSD) has some serious problems when dealing with short length messages. ***First***, we had presented a detailed survey for the contributions of several researches that dealing with LT codes. The survey classifies the previous works in three major fields which are degree distribution, decoding approaches, and the code applications and among these fields also presented some attempts dealing with short length Lt codes.

Secondly, in an erasure channel environment, a treatment for improving the performance of LT codes with short length messages has been presented. Shortly, we overcome the problem of early termination of the decoding process when the decoder hasn't find any degree-one coded symbol to start or continue the decoding process. The treatment has present a new belief propagation (BP) decoding algorithm called belief propagation with pattern recognition (BP-PR). In which we look inside the remaining coded symbols decoding matrix, if there is no degree-one coded symbol, then we look for certain patterns that if combined they produced new degree one coded symbol and resume decoding again. The utilization of this new decoding algorithm has been witnessed a performance improvement for LT codes which used (RSD) or with one of its modification which called memory-based robust Soliton distribution (MBRSD). A lower bit error rate (BER) and better successful decoding ratio have been recorded for LT codes with lengths of $k = 32$ and $k = 256$ for both pre-mentioned degree distribution even when the well-known Gaussian elimination approach BP-GE has assisted the decoding BP.

Motivating by the enhancement that has been achieved when using this new BP-PR, new deterministic encoding (DE) has been designed for an LT codes in erasure channel. This new (DE) uses the property of the certain patterns to regenerate degree-one coded symbol when the decoding matrix hasn't any one of them. This DE generates the degrees from 1 up to a maximum predetermined degree called repetition period (R_p). The data file is truncated to number of segments of length equal to (R_p), and to generate the coded symbol a set of data symbols equal to its degree are chosen serially from one of the data segments. The selection of the data segment follows a common random sequence shared between the two communication sides. The test of such deterministic encoding LT codes has been achieved in both BEC and AWGN channel. For the case of BEC, LT-DE performance has been tested against LT-RSD-BP-PR, MBRSD-BP-PR and optimal degree distribution based on the state of art work presented in [42] and we denoted as ODD-BP-PR. The performance of this LT-DE outperforms that of the prementioned codes short lengths of $k = 32$ and 16 data symbols in BER, successful decoding rate and the required overhead.

The same encoding scheme of LT-DE is used in an AWGN channel environment. The sequential selection for the data symbols to be combined while encoding of the (DE) gives this encoding technique a new name as sequential encoding approach (SEA) . We examine an LT code generated using four deferent types of degree generation which are RSD, MBRSD and ODD with our DE over an AWGN channel. The decoding process which is called soft-demodulation-hard-decoding (SDHD) consists of two steps, first the corrupted coded symbols are softly demodulated using maximum likelihood probability (MLP) probability which used to decide on the code values. Then, the coded symbols will be applied to a hard decoding using three decoding algorithms. We decode both RSD, MBRSD and ODD using BP-PR, while the SEA has been decoded using new sequential decoding (SD) algorithm. The simulation results shown in our paper [A2-2] approved the superiority of the sequential encoding -decoding algorithm when compared to that of other degree distributions using supported by the proposed decoding modification BP-PR.

1.5. Layout of the thesis

In this thesis, we focus on the first practical rateless codes which is LT code [18]. The thesis has arranged in to six main chapters. Chapter 1 is dedicated for a brief introduction for the essentials of any communication system with explanation for the channel characteristics and we focus on the definition of the two used channels in our code testing which are the BER and the AWGN channel. Then Chapter 1 presents a classification for the codes based on their rate, in which we classify the codes in to fixed rate codes and rateless codes. The basic need for such rateless codes and a brief history for their development has been presented. In Chapter 2, the main features of the LT codes are listed. A detail analysis has been presented for the encoding technique utilized by such codes as well as the effect of the degree distribution on the performance of the code. Simplified examples with the aid of matrix representation and Tanner Graph figures are used to explain the main steps od decoding approach used by such codes which is the BP approach. The implementation of this BP in both BEC and AWGN channel has been discussed. Chapter 3 is used for presenting a detailed literature review for the previous works concerning the development of LT codes. The review is divided in to three parts, the first is dedicated for the related works in degree distribution, while the second part listed the related work in decoding approaches. The final section for this review is dedicated for the related work in LT codes applications. Our proposed BP-PR approach has been discussed in Chapter 4. First a description for the problem of losing degree-one coded symbol and its effect on the decoding performance is presented. Then the solution produced by BP-PR with matrix representation and Tanner graphs are presented. Finally, in this chapter we present the testing of this approach in BEC to approve its improvement for the regular BP approach for both RSD and MBRSD. Chapter 5 is dedicated to present our new LT-DE with its encoding analysis as well as its SD approach. The results for testing this LT-DE in both BEC and AWGN channel are listed to approve its superiority against the state of art distributions like MBRSD and ODD. Finally, Chapter 6 is used to list our conclusion and future works.

CHAPTER 2

LT CODES: DESIGN FEATURES

2.1. Preliminaries

In this chapter, a clear introduction for the basic design features of LT codes will be presented. Essentially, the need for such codes arises concurrently with the steady development of distributing huge data file to multi destinations. The best solution for such scenarios was inventing the fountain codes. These codes must fulfill a special requirement such as to be entirely reliable and have small network overhead as well as serve enormous numbers of receivers having deferent design features, and, they should be installed with a minimum transmission cost [12]. To send such large data files, the server tries to slice it into packets which may have deferent sizes according to the application and the transmission control. Due to channel effects on the sent data packets, part of them are lost and a request for retransmission must be sent. For the case of unicast communication system this problem can be handled easily while for the multicast scheme this will flood the back channel with deferent retransmission requests. Even if the server can deal with such demands, this will be a kind of wasting the channel bandwidth since deferent receivers may ask for deferent packets and the server can’t send them simultaneously. Several alternative solutions for such limitations have been proposed in [24-28]. It is obvious that fountain codes offer the best solution for the above limitations as well as the case of satellite networks and deep space communication system where are the feedback channel is hard to implement or it takes unacceptable time delay.

LT codes [18], translate the actual meaning of the rateless codes or digital fountain codes [12,13] in which the transmitter will be able to flood the channel with endless coded symbols. The length of this symbol is variable in length from single bit to number of bits. Another interesting feature of such codes that the coded symbols are universal in composition i.e. when the receiver can collect any number N of them it can start decoding. When the receiver starts its mission in recovering the k data symbols from the received N coded one, it could succeed or need an extra coded symbol and then the length of the code will obviously change which explain the name of rateless codes. The following sections will give a clear illustration for the main features of LT codes. We shall start by explaining the encoding process in which the encoding algorithm will be illustrated. Then a graphical representation for the code design will be sketched to emphasize the superiority and flexibility of such codes than other fountain block codes like the tornado codes. Also, the characteristics and importance of the design of the degree distribution will be explained in the encoding algorithm section. Then the decoding algorithm in both types of channels (BEC and AWGN channel) will be presented in the following section.

2.2. Encoding algorithm

Let's consider that we have a certain file and we intend to distribute it to multi receivers, the first step is to slice this file into several symbols (in all thesis sections each symbol will be represented by a single bit). Then the encoding process will be achieved by following this algorithm [18]:

Algorithm 1: LT encoding

```
Repeat
    Randomly choose a number ($d$) called degree of coded symbols.
    Uniformly at random select ($d$) data symbols to be connected to the coded symbol.
    Using bit-wise addition of the selected data symbols to get the value of the coded symbol.
Until (Acknowledgement arrives to inform of full data recovery)
```

It is worth to mention that the decoder needs to know the degree and the set of the connected data symbols for each received coded symbol, or in other word a full description of the code generator matrix. This information may be delivered to the decoder using deferent ways, like [18]:

1. This information may be sent to the receiver explicitly, i.e. the header of the coded packet contains the degree and the set of neighbor's data symbols.
2. The degree and the set of neighbors may be computed by the receiver. For example, this can be done by calculating the time of arrival or the position of the coded symbol relative to the arrival of the others.
3. Both the encoder and the decoder may use a certain key to represent the degree and the set of neighbors, these keys could be generated randomly by the encoder and both encoder and decoder using a certain function to get the degree and set of connected neighbors from that key.
4. This key may be used as a seed generated using the pseudo random generator that is common used by both the encoder and decoder to inform the decoder of the required information for each received coded symbol.

These four ways are not the only means to deliver such information to the decoder and there are a variety of other ways depending on the application.

This code is mainly designed to be an efficient erasure code for distributing bulk data transmission and it could be used in deferent applications as will be explained in the next chapter. To get closer to the encoding algorithm mentioned above, the encoding process can be described as a bipartite graph that joining the input symbols through a connecting edge to that of the coded symbols. For instance, let us consider this prototype example, suppose that the data file consists of only five data symbols ($u_1\ u_2\ u_3\ u_4\ u_5$) such that each symbol represented by a single binary digit, then according to algorithm 1 the sender can generate endless length code. The graph representation for the encoding process is illustrated in Fig. 4. It is clear from Fig. 4 that the coded symbols theoretically have infinite length, however let’s focus on the first drown coded symbols. Fig. 4 tell us that the LT encoder will use the random generator to produce the degree and the set of data symbols in order to get the value of the coded symbols. For Fig. 4 the random generator produce these random numbers for the first four coded symbols as (2 1 3,3 1 2 3,2 3 4,4 4……...). using these numbers, we can write the following equations using the bitwise addition or the (XOR) operations using the formula, $C = u \cdot G$:

$$\left.\begin{aligned} C_1 &= u_1 \oplus u_3 = 1 \oplus 1 = 0 \\ C_2 &= u_1 \oplus u_2 \oplus u_3 = 1 \oplus 1 \oplus 0 = 0 \\ C_3 &= u_3 \oplus u_4 = 1 \oplus 0 = 1 \\ C_4 &= u_4 = 0 \end{aligned}\right\} \tag{2.1}$$

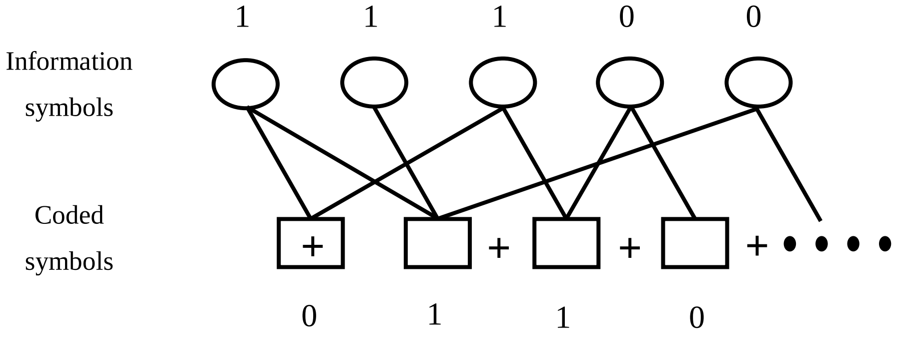


Figure 4 Bipartite graph for LT encoding algorithm

We can describe Fig. 4 as two-sided graph, the upper side represents the data symbols side in which the set of the selected data symbols will be chosen from it, while the lower side represents the coded symbols side. The generator matrix G will be constructed to be a $(k\times N)$ matrix. Each column has number of ones located randomly utilizing the random degree generator by using one of the degree distributions that will be explained in the next section while the allocation of these ones represents the selected data symbols which is also taken to be uniformly at random. The random distribution of the data selection is likely to be Poisson distribution for large data lengths. The generator matrix which describes the Fig. 4 is written as:

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots\cdots\cdots\cdots \\ 0 & 1 & 0 & 0 & \cdots\cdots\cdots\cdots \\ 1 & 0 & 1 & 0 & \cdots\cdots\cdots\cdots \\ 0 & 0 & 1 & 1 & \cdots\cdots\cdots\cdots \\ 0 & 1 & 0 & 0 & \cdots\cdots\cdots\cdots \end{bmatrix} \tag{2.2}$$

The degree generation is the most important part of the LT code design. In the following section a detailed discussion for the design of the degree distribution which have been used in LT codes.

2.2.1. Degree Distribution

The fountain of coded symbols produce by the LT encoder is designed to ensure a high probability of successful decoding ratio which is the ratio of the successful recovery of the data symbols to that of the total received or collected one. However, the design of the LT code must ensure this high decoding ratio with small amount of overhead or extra needed symbols, this compromising requirements can be approached by the well-design of the degree distribution. To get involved in deep of the design characteristics of the degree distribution, let’s have these important definitions:

Definition 2.1: degree distribution $\Omega(d)$ is a probabilistic function which generates the weight of each degree (d) according to the function parameters.

Before proceeding into the main degree distribution types which utilized in the design of LT codes, let’s point out the design objectives required for such design. The main design objectives of degree distribution can be listed as [18]:

1. The success of the LT process must be achieved with minimum coded symbols on average. Keep in mind that the success of the LT process mainly depends on the coded symbols that will led to full data recovery.
2. The encoding-decoding complexity is as low as possible. This complexity is measured by the number of symbol operations during the LT process. The symbol operations can be calculated in average as the multiplication of the average degree of the coded symbols by the number of the coded symbols collected and processed by the decoder.

These two features of the degree distribution design must compromise in a fitting process as possible to insure a well-design for such distribution. The degree distributions that present by the first paper dealing with the LT code design [18] will be listed below [18, 29,30].

Definition 2.2 (All -at-once distribution), let (d) represnt an integer called degree then this distribution can be defined as:

$$\Omega(d) = \begin{cases} 1\,, & for\ d = 1 \\ 0, & for\ d > 1 \end{cases} \tag{2.3}$$

Equation (2.2) illustrates that this distribution generates only degree one at each encoding step while the data symbol which connects to the coded symbol will be chosen randomly and following the rule of the ball and bins. This rule describes the coded symbols as a group of balls that thrown randomly towards a set of bins which are considered as the data symbols. To illustrate the inefficiency of such distribution, let's imagine that we aim to throw number of balls into k number of bins such that this k is chosen to be large enough ($k > 10^3$). it is important to ensure that every bin has its own ball, however after throwing an N balls, the probability that we have an empty bin is [31],

$$Pr(empty\ bin) = (1 - \frac{1}{k})^N \simeq e^{-N/k} \tag{2.4}$$

Now it is clear that, when N equal to k which means that no extra coded symbols will be need, the total number of the empty bins will be around $(1/e)$ and this number will be decreased as the number of balls will increase, i.e. by using this distribution, even if the sender generates a code length of triple times the data length there is still about (5%) of the total data symbols not connected, which totally inefficient. In general, after throwing an N balls the expected number of the empty bins will be [31]:

$$E(empty\ bins) = ke^{-N/k} \tag{2.5}$$

İf we denote the expectation in equation (2.5) as (δ), then to find the required number of balls to ensure that all the bins are occupied, (i.e. to find N with probability of $(1 - \delta)$) **,**this number has to be[31]:

$$N > kln(k/\delta) \qquad (2,6)$$

Equation (2.5) is taken to be a general formula for the minimum sum of the degrees of the coded symbols to ensure covering all the data symbols [18]. İt seems that the all-at-once distribution may has the minimal symbol operations but it needs unpractical number of coded symbols during the encoding process to successfully covered all the data symbols. Luby [18], succeeded in developing new distributions which can cover all the k data symbols with coded symbols have the sum of degrees in the order of $O(k.\ln(k/\delta))$. These two Soliton distributions represent the key of the LT codes design. Before getting in the essential formulas for such distributions, Luby presents some probabilistic definitions which are necessary for understanding the main requirements for a good distribution [18,31].

Definition 2.3 (*coded symbol release*)**:** it represents the ability of the decoder to release a coded symbol when L data symbols still unrecovered and this coded symbol is used to recover the $(k-L)$ th data symbol.

Definition 2.4 (*degree release probability*)**:** it is denoted by $q(i,L)$,which is defined the probability of releasing a coded symbol of degree i where at the same time still an L unrecovered data symbols.

Definition 2.5 (*overall release probability*)**:** it represents the total probability of the ith degree release probability for any coded symbol, i.e. if $r(i,L) = P_i q(i,L)$,then the total release probability for any coded symbol when it is released and still an L data symbols unrecovered, is: $r(L) = \sum_i r(i,L)$.

According to the pre-mentioned definitions and depends on the value of the degree of the coded symbol we can conclude two main types for these probabilities:

1. The LT process depends mainly on the coded symbols of degree one which represent the key for starting the decoding process, this led to:

$$q(1,k) = 1 \tag{2.7}$$

2. For the coded symbols that have degrees greater than one and for the rest unrecovered data symbols, i.e. for $i = 2,3,\dots\dots\dots,k$ and $L = k-i+1,\dots\dots\dots..,1$, the probability of the degree release will be:

$$q(i,L) = \frac{i(i-1)\cdot L \cdot \prod_{j=0}^{i-3} k-(L+1)-i}{\prod_{j=0}^{i-1} k-j} \tag{2.8}$$

and for the rest values of the degrees $q(i,L)=0$.

These probabilities and definitions are the bases for proceeding to understand the two well-known degree distributions proposed by Luby. These proposed distribution focus on an important design property which states that: the rate of adding new data symbol to the decoding box (ripple) should be the same as that for processing it. Luby [18] was inspired by the behavior of the soliton wave [32]. The ripple can be defined as the set of all connected data symbols that have not been processed yet. This ripple size is a compromise issue, so the designer should ensure that:

1. The ripple size preferred to be small and not include redundant coded symbols that used to recover the same data symbol.
2. The ripple size in other hand should be kept large enough in order not to be vanished before recovering all the data symbols or which is called premature decoding process.

Definition 2.6 (*ideal soliton distribution*)**:** LT code can be generated using the ideal Soliton distribution which is denoted by $\rho(d)$ such that d is the degree for the coded symbols and its generated according to probabilistic function defined by [18]:

$$\rho(d) = \begin{cases} \dfrac{1}{k}, & for\ d = 1 \\ \dfrac{1}{d(d-1)}, & for\ d = 2,3,\dots k \end{cases} \tag{2.9}$$

And as probabilistic function, $\sum_d \rho(d) = 1$. Several important properties for this degree distribution, such as:

1. It has a uniform overall release probability, i.e. $r(L) = 1/k$ for all $L = k, \ldots\ldots, 1$.
2. It produces an ideal encoding-decoding behavior but with an ideal channel conditions, it implies that the decoder needs to collect only k coded symbols to recover all of the k data symbols. This means that at each decoding step only one coded symbol is released to recover its connected data symbol.
3. The average degree generated by such distribution can be calculated as:

$$H(k) = \sum_{d=1}^{k} d/d(d-1) \tag{2.10}$$

It can be approximated that $H(k) \backsimeq \ln(k)$.

4. The sum of degrees of k coded symbol will be on average $k \cdot \ln(k)$ which the same for that of the all-at-once distribution. Which conclude that it will be the essential average sum of the degrees required for any distribution. Even the average for the sum of the degrees of the coded symbols is identical for both distribution but the ideal soliton distribution needs less coded symbols to cover all the data symbols than that of the all-at-once distribution.
5. Despite having an ideal expected number for the needed coded symbols to recover the data symbols, this distribution shows a weak performance for the practical channel conditions. That is because its fragile characteristics of the ripple size which is filled by only one coded symbol at each decoding step which quite critical where for any channel fluctuation this symbol could be lost and the decoding process will be halt.

In Fig. 5, Fig. 6 and Fig. 7 the ideal soliton distribution is sketched for deferent data length values ($k = 10$ and $k = 50$ and $k = 200$). It is clear that degree (2) has the largest probability among all degrees and always it is (0.5), while degree (1) probability is inversely proportional with the data length and the rest degrees has less probabilities which are inversely proportional to the degree value.

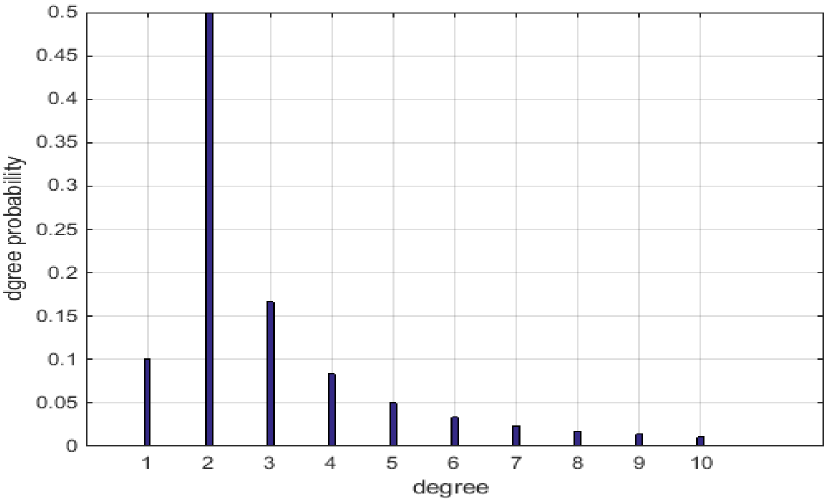


Figure 5 Ideal Soliton Distribution for k=10

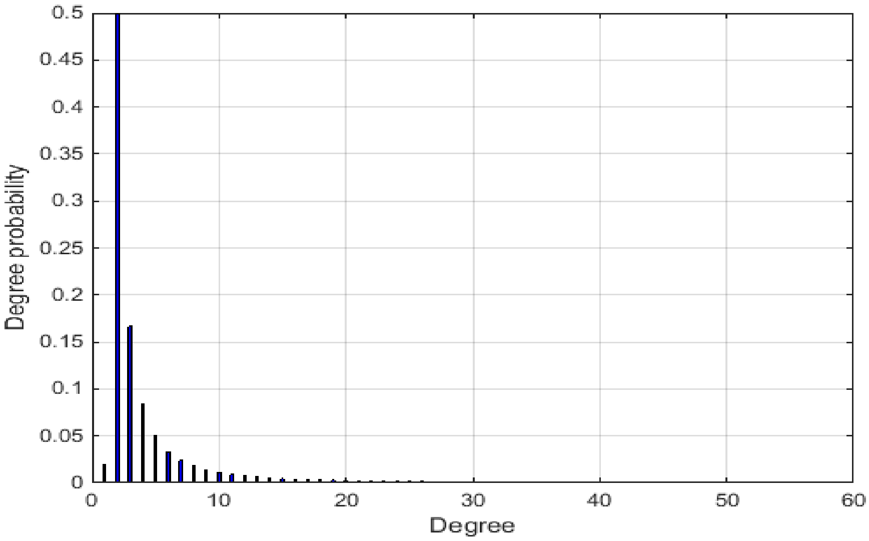


Figure 6 Ideal Soliton Distribution for k=50

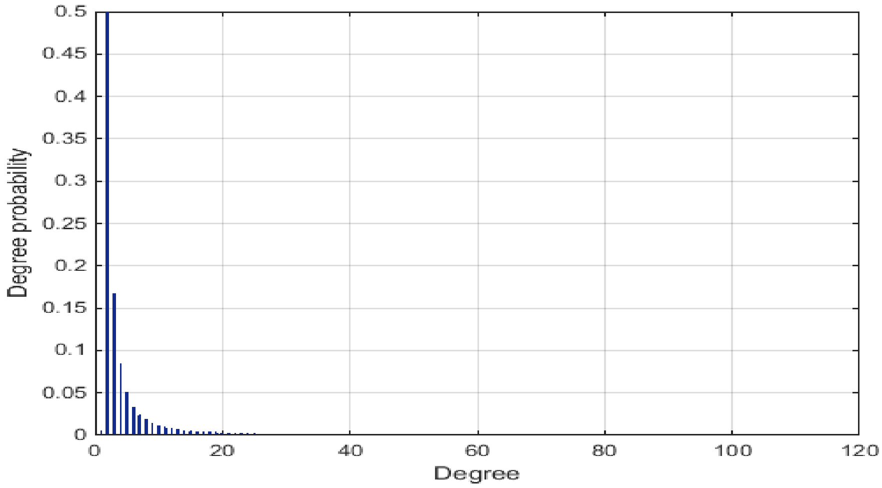


Figure 7 Ideal Soliton Distribution for k=200

Until now, it is obvious that that this ideal distribution constructs some essential facts for good degree distribution design. As a result, Luby [18] introduces his modified version of this ideal soliton distribution which he calls the robust soliton distribution.

Definition 2.7 (*robust soliton distribution*)**:** for some constant value $c \in (0,1)$ and an $S = c \cdot \ln(k/\delta)\sqrt{k}$ to represent the expected number of degree one coded symbols where δ is the decoder failure, used to define new probabilistic function $\tau(d)$ as [18]:

$$\tau(d) = \begin{cases} \frac{S}{k}\frac{1}{d} & for\ d = 1,2,\cdots,\frac{k}{S} - 1 \\ \frac{S}{k}\ln\left(\frac{S}{\delta}\right) & for\ d = \frac{k}{S} \\ 0 & for\ d > \frac{k}{S} \end{cases} \tag{2.11}$$

Add this function $\tau(d)$ to that of the ideal soliton distribution mentioned above in (2.9) to get the normalizing factor as:

$$\beta = \sum\nolimits_{d} \rho(d) + \tau(d) \tag{2.12}$$

By using all of the above definitions, the robust soliton distribution will be formulated as [18]:

$$\Omega(d) = \frac{\rho(d) + \tau(d)}{\beta}\ , \qquad for\ all\ d = 1,2, \dots \dots .., k \tag{2.13}$$

Now, like what we had done to specify the main features of the ideal soliton distribution, here are the main properties and treatments that had been added by the robust soliton distribution:

By inserting the function in (2.11), this distribution and with high probability insures that the expected ripple size will kept high enough and during the decoding process and will not vanished so as not to have a premature decoding process.

The group of the coded symbols had been set to $N = k\beta$, i.e. for any degree $(d)$the total number of the coded symbols will be $k \cdot (\rho(d) + \tau(d))$. Substituting the functions of $\rho(d)$ and $\tau(d)$, this total number will be calculated as [18]:

$$N = k + O(\sqrt{k} \cdot ln^2(k/\delta)) \tag{2.14}$$

The average degree of the coded symbols is [18]:

$$D = O(ln(k/\delta) \tag{2.15}$$

The added function $\tau(d)$ is also useful in ensuring complete covering with high probability of all the k data symbols. This has been done by inserting a spike probability at certain high value degree which is $d = k / S$ .It is worth to mention that this distribution is mainly designed for bulk data length ($k > 10^3$) and for small message length this distribution will faced serious problems mainly characterized by the absence of degree one coded symbols.

From the above-mentioned properties, RSD offers an ultimate distribution for the LT codes especially commensurate with the large data length. In order to get more deeply sight inside the RSD, we draw this distribution for two deferent data lengths, particularly for ($k = 100$) and ($k = 10000$). In Fig. 8 both distribution functions that construct the RSD, which are ISD ($\rho(d)$) and the function ($\tau(d)$) for the parameters ($c = 0.2, \delta = 0.05$) are join. There are two spikes in this graph, the first is for degree (2) which takes this large probability from the ISD while the other spike is inserted by the ($\tau(d)$) function at degree ($d = 41$) and as we explain before that this high degree with this high probability is useful to insure covering all the data symbols during the encoding process. It is also noted that the probability portion of degree one is enhanced by adding the new ($\tau(d)$) function. The resultant RSD for the same parameters of ($c = 0.2, \delta = 0.05$) is illustrated in Fig. 9. The effect of increasing the decoder failure probability (δ) from (0.05) to (0.4) has been shown in Fig. 10 which clarifies that this increase will be noticed in decreasing the level both the high degree probability spike and the degree one probability in addition to shift the high degree spike to towards higher degree value. In Fig. 11, it is obvious that when decreasing the constant (c) from (0.2) to (0.1) while keeping low decoding probability failure of ($\delta = 0.05$), this will approximately vanish the high degree probability and the RSD will close to that of the ISD. All the above-mentioned results will be noticed again but more emphasized because the effect of the data length with these parameters of ($c = 0.2, \delta = 0.05$) increases the high degree spike in Fig. 12 and Fig. 13 to a noticeable value which also can be controlled by varying c and δ parameters.

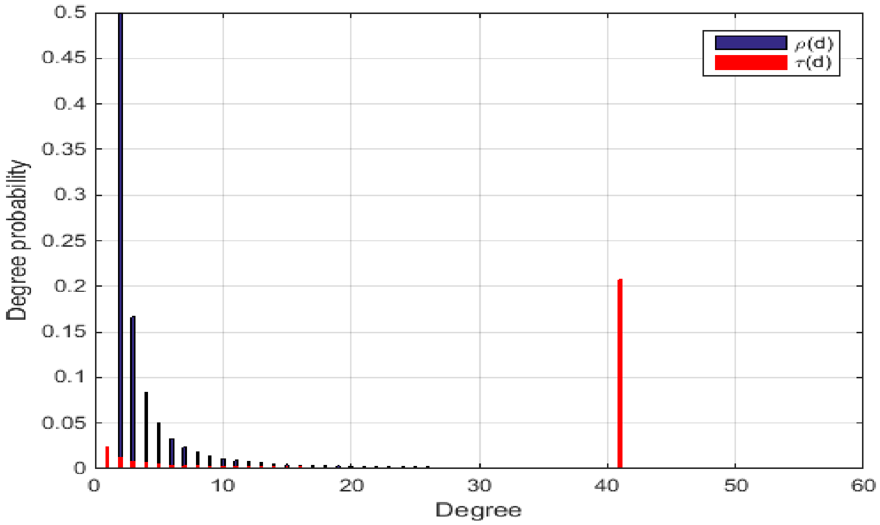


Figure 8 ISD $\rho(d)$ and $\tau(d)$ function for k=10000 and $c = 0.2, \delta = 0.05$

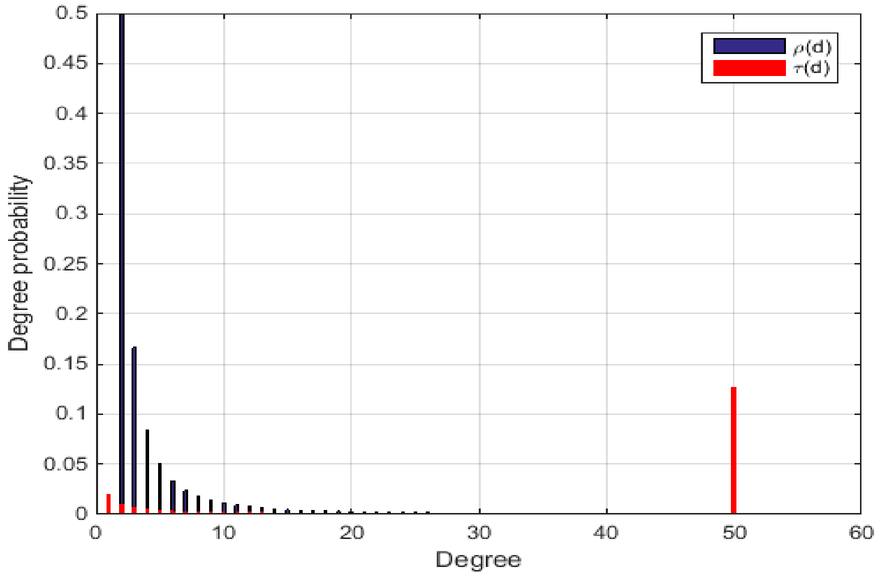


Figure 9 RSD   for k=10000 and $c = 0.2, \delta = 0.05$

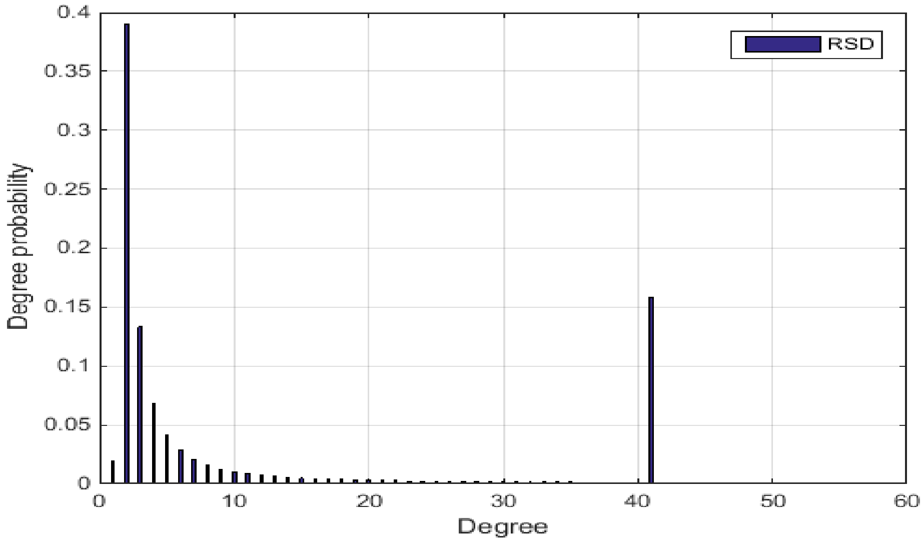


Figure 10 ISD $\rho(d)$ and $\tau(d)$ function for k=10000 and $c = 0.02, \delta = 0.4$

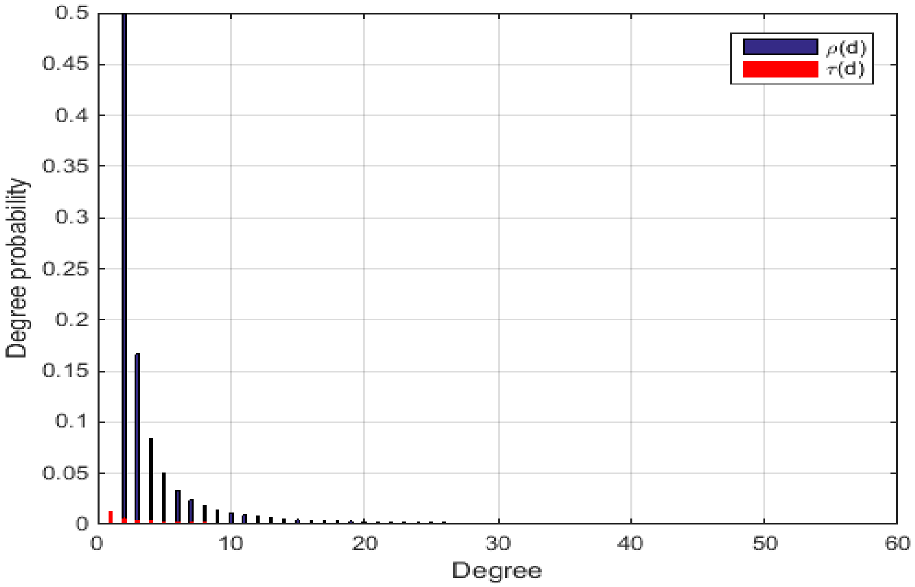


Figure 11 ISD $\rho(d)$ and $\tau(d)$ function for k=10000 and $c = 0.1, \delta = 0.05$

For the figures which is used to illustrate the RSD, it is obvious that the addition of the two probability functions $\rho(d)$ and $\tau(d)$ with the normalization formula, this will adapt each probability on the graph for the RSD.

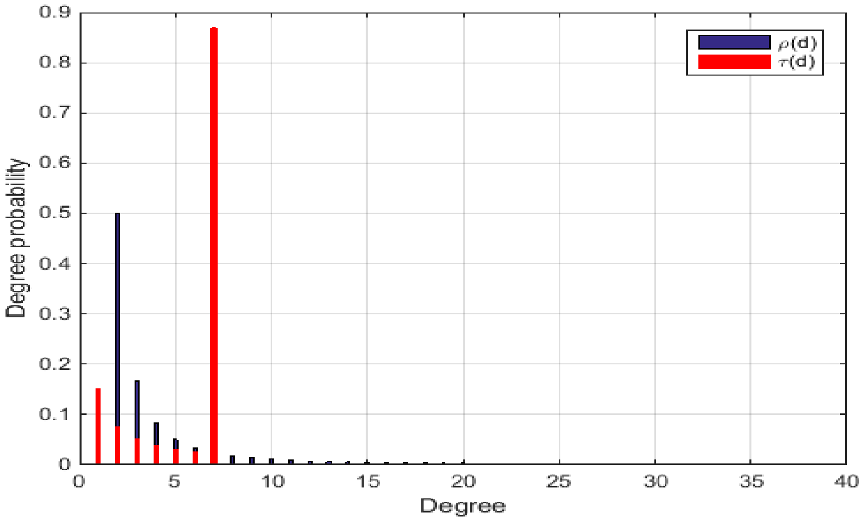


Figure 12 ISD $\rho(d)$ and $\tau(d)$ function for k=100 and $c = 0.2, \delta = 0.05$

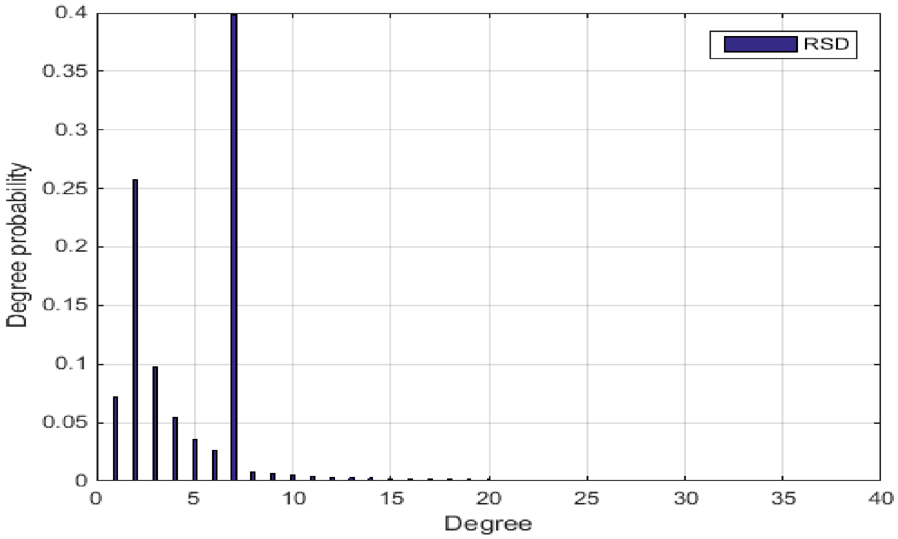


Figure 13 RSD for k=100 and $c = 0.2, \delta = 0.05$

The expected number of degree one is an important factor in the design of degree distribution. Below and referring to the summery presented in [29] we plot the expected number of degree one (S) for deferent value of (δ) and see the effect of changing the constant (c) on this (S). Fig. 14 illustrates this effect for large data length ($k = 10000$) while Fig.15 is drawn for small data length, particularly ($k = 100$). The effect is identical for both figures, this average of degree one has its maximum value with maximum (c) and minimum (δ**)** and this is logical since increasing the the degree one coded symbols plays a significant rule in ensuring a successful decoding process. It is quite obvious that this average of degree one is critical issue for the short data length as recorded on the Fig. 15.

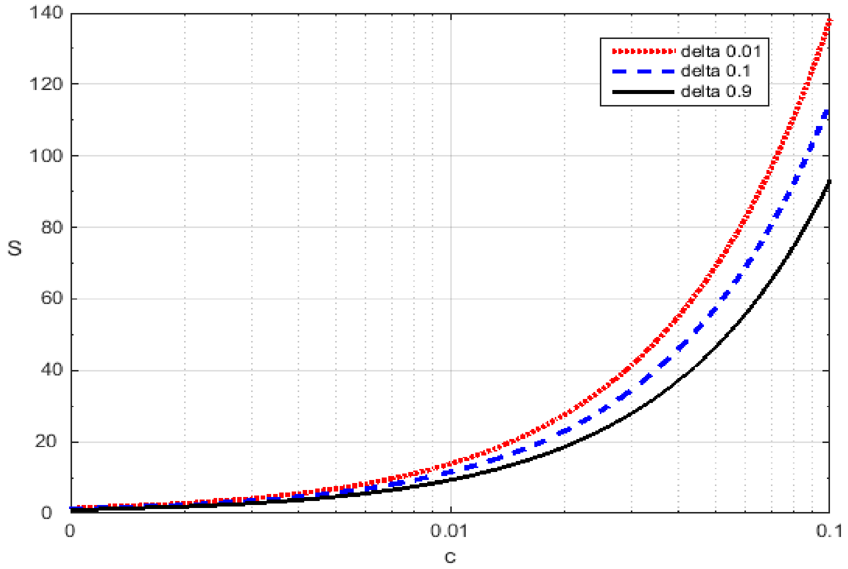


Figure 14 expected number of degree one versus c , for $k = 10000$

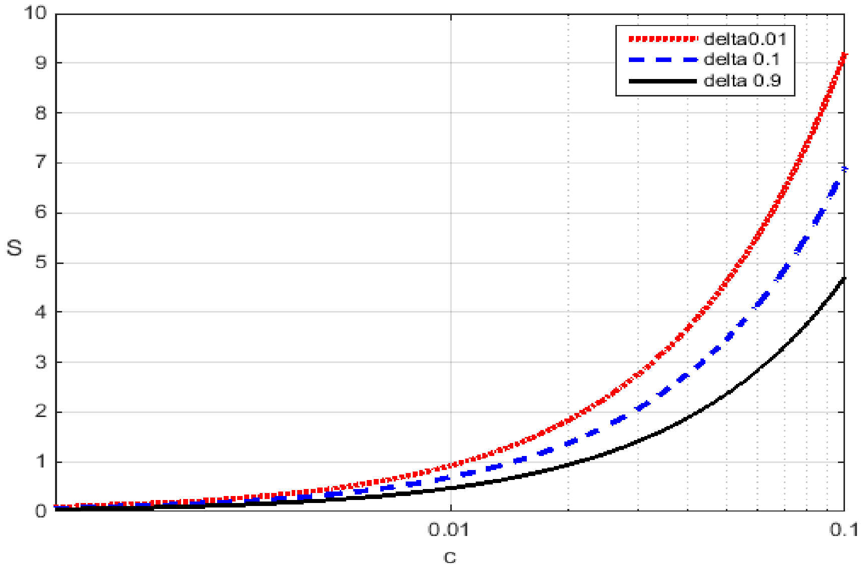


Figure 15 expected number of degree one versus c , for $k = 100$

From all what have been presented in the last sections, it has been noticed that to construct a good LT code, the designer has to spent a sufficient time for the design of the degree distribution. In the following section, we shall present a detail explanations for the decoding algorithm and its application in both BEC and AWGN channel.

2.3. Decoding algorithm

As a coding scheme, the relation between the encoder and decoder is so strong such that the decoder translates the coded symbols to their original required information symbols. Again, the degree distribution has its noticeable effect on the success of the decoding process and to get more in deep, let us present the main decoding algorithm utilized by Luby [18], suppose that the receiver could collect an N coded symbols such that $N = k + \varepsilon$,wher ε represent the overhead extra coded symbols. If the coded symbols stream is denoted by C and the data vector is denoted by u,then the decoding algorithm will be [18]:

Algorithm 2: LT decoding

```
Repeat
Search for a coded symbol $C_n$ which has a single connection (degree one).
Copy its value to its connected data symbol $u_k$.
Update the coded symbols values of the neighbors to the recovered data symbol
    using bitwise addition.
Release all the recovered data symbol connections.
Until (recovering all the data symbols, or no degree one coded symbol found)
```

So, the decoding algorithm depends mainly in its starting and survival on the presence of degree one coded symbol. If for any reason in the decoding step, the decoder fail to find this degree one coded symbol, the decoder will halt and the rest of the unrecovered data symbols will be declared to be an error. For simplicity, let us take this toy example, presume that the data vector represented by ($u_1\, u_2\ \ u_3\, u_4\, u_5$) and the decoder starts its process after collecting these coded symbols vector

(C_1 C_2 C_3 C_4 C_5 C_6), then the above-mentioned algorithm 2 can be illustrated in the form of bipartite graph as in set of figures below. First, the encoding was done as described in Fig. 16, the red edge represents the degree one connection which is assigned for $C_1$and C_5.

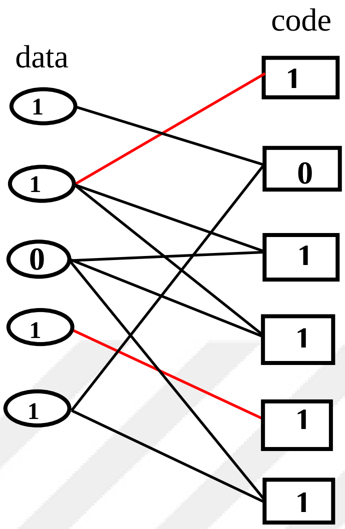


Figure 16 LT encoding for k =5 and N = 6

In Fig. 17, the decoder receives these codes and has prior information about their connections represented by the degree of each coded symbol and the its connected data symbols. As shown in Fig. 18, the decoder should find the coded symbol of degree one (the coded symbol with the red edge) and copy its value to its connected data symbol.

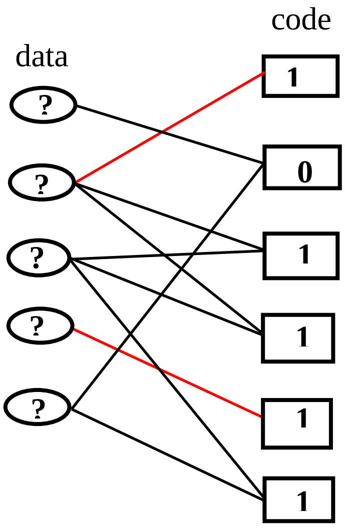


Figure 17 first step of LT decoding for k =5 and N = 6

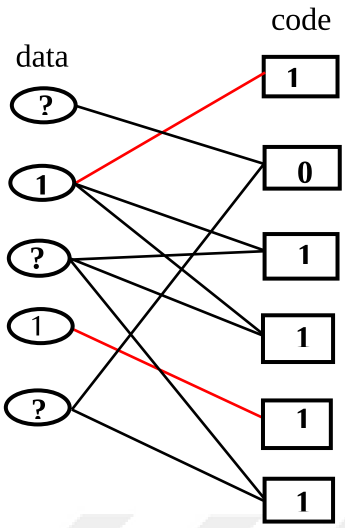


Figure 18 Second step of LT decoding for k=5 and N=6

The recovered data symbol value is using to update all the neighbors coded symbols which are connected this data symbol, as shown in Fig. 19. Then the last decoding step in the per-mentioned algorithm 2, in which the decoder will release all the connecting edges of the recovered data symbol and this illustrated in Fig. 20.

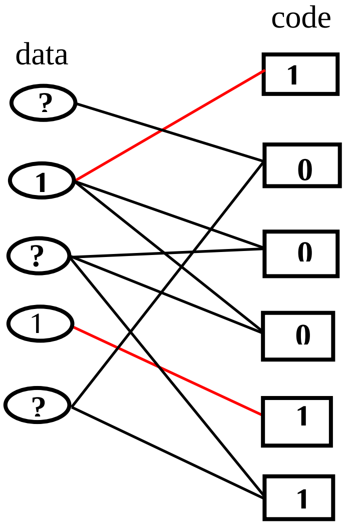


Figure 19 third step of LT decoding for k=5 and N=6

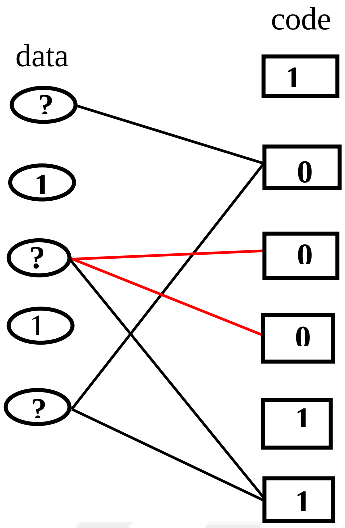


Figure 20 fourth step of LT decoding for k=5 and N=6

It is clear from Fig. 20 that after releasing all the connected edge of the recovered data symbol, a new degree coded symbol is created which is necessary for starting a new decoding step and continue recovering the remaining data symbols. It also seen in Fig. 20 that there are two degree one coded symbols are connected to the same data symbol which unwanted case since it adds a symbol operation that will cause a redundant recovery. After this simplified example with its illustrated figures, now it is obvious that the well-design of the degree distribution has its great effect on a successful decoding mission. Also, we can analyze why RSD and ISD give the highest probability for degree two which is the responsible for providing a new degree one to the ripple budget after releasing one of its edges in the previous decoding step. This decoding algorithm has been called a belief propagation (BP) algorithm or message passing algorithm since the messages between both sides of the bipartite graph or between the data side and the coded symbols side will be transferred in a ping-pong style. In the following section, we shall give a brief description for the BP algorithm in both the BEC and the AWGN channel.

2.3.1. Belief propagation algorithm in BEC

Referring to the channel model that has been described in section (1.2.1.), the effect of the channel will be in the shape of losing some of the transmitted coded symbols. The decoder must recover all the data symbols after collecting an N coded symbols usually greater than that of the data symbols length. Each coded symbol is defined for the decoder by its equation which is came from its degree and the locations of the data symbols that was combined using bitwise addition to generate that coded symbol. Let us recall the generator matrix used in equation (2.2) to represent the encoding matrix for the LT code shown in Fig. 16 which is used to encode data message of (1 1 0 1 1) and let the received code frame is (1 0 1 1 1 1). As shown in equation (2.16) that $G(i,j) = 1$ represent an edge connect the coded symbol C_j with the data symbol u_i . For the decoding case, let us define the decoding matrix G_d which is the transpose of the matrix G, as shown in (2.17), the column outside the matrix represent the coded symbols value produced using the bitwise addition of the data symbols connected to this coded symbol and assigned by the 1 value in the matrix raw.

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & u_2 & u_3 & u_{41} & u_5 \end{array} & \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \end{array} \tag{2.17}$$

For the decoding in BEC, after collecting an N coded symbols (in this example is 6), the following steps in an application for the decoding algorithm 2:

1. Fetch the decoding matrix and determine the coded symbols of degree one (marked in red), as shown in (2.18).
2. For each coded symbol of degree one, copy the value of the determined coded symbol to its connected data symbol, (2.19).
3. Update the value of all coded symbols (marked in blue) which is connected to the recovered data symbol column, (2.20).
4. erase all the values (1s) of the recovered data symbol column, (2.21)

repeat the above four steps again, if there are new degree one coded symbols and there are still unrecovered data symbols (2.22) to (2.30).

$$G_d = \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & u_3 & u_{41} & u_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \tag{2.18}$$

$$G_d = \begin{array}{c} \begin{array}{ccccc} u_1 & 1 & u_3 & u_{41} & u_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \tag{2.19}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & 1 & u_3 & u_{41} & u_5 \end{array} & \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \end{array} \tag{2.20}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & 1 & u_3 & u_{41} & u_5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \end{array} \tag{2.21}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & 1 & u_3 & 1 & u_5 \end{array} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \end{array} \tag{2.22}$$

$$G_d = \begin{array}{c} \begin{array}{ccccc} u_1 & 1 & u_3 & 1 & u_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \tag{2.23}$$

$$G_d = \begin{array}{c} \begin{array}{ccccc} u_1 & 1 & 0 & 1 & u_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \tag{2.24}$$

$$G_d = \begin{array}{c} \begin{array}{ccccc} u_1 & 1 & 0 & 1 & u_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \tag{2.25}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & 1 & 0 & 1 & u_5 \end{array} & \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \end{array} \tag{2.26}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & 1 & 0 & 1 & 1 \end{array} & \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \end{array} \tag{2.27}$$

$$G_d = \begin{array}{cc} \begin{array}{ccccc} u_1 & 1 & 0 & 1 & 1 \end{array} & \\ \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] & \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \end{array} \tag{2.28}$$

$$G_d = \begin{array}{cc} \begin{matrix} u_1 & 1 & 0 & 1 & 1 \end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{matrix} \end{array} \qquad (2.29)$$

$$G_d = \begin{array}{cc} \begin{matrix} 1 & 1 & 0 & 1 & 1 \end{matrix} & \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{matrix} \end{array} \qquad (2.30)$$

It is worth to mention that, the effect of the channel will be increased as the erasure probability (α) increased, which imposed more erased coded symbols that is why the degree distribution must compromise between offering enough degree one coded symbols and in the other hand ensuring higher degree coded symbols to cover all the data symbols during the encoding process. It is also noticed that, if there is degree one coded symbol, the decoding process continue but for the case of no degree one coded symbol found, the decoder will halt and the rest unrecovered data symbols will be added to the error vector.

2.3.2. Belief propagation algorithm in AWGN channel

Basically, the rateless codes are designed to be a FEC codes for a wired internet environments in which the packet lose is modeled by the erasure probability (α) as described in section (1.2.1). However, applying such codes in the wireless scenarios will be faced by some serious communication problems such as noise, fading and signal variations [33]. Since the first invention of the rateless codes and especially the LT code, several attempts were presented to examine or enhance the behavior of these codes in the noisy channels (details for such researches will be presented in the next section). It was found that LT codes were suffering from high error floor when using them to face the effect of the noisy channels such as AWGN channel. The decoding procedure for the LT code in the AWGN channel will be composed of two main parts. The first step will be either hard or soft demodulation for the received noisy coded symbols to estimate their values. Then these estimated coded symbols will be fed to a hard BP decoder to recover the information data symbols. Due to the nature of message passing technique utilized in the belief propagation algorithm, if there is any error occurred while deciding on the estimated value of the noisy received coded symbol, this error will be copied to the connected data symbol and propagated through the successive steps to next decoding symbols. This weakness in the LT code performance over AWGN channel motivated the researchers to adapt the decoding matrix of the LT code (shown above in equation (2.18)) to be close to that of the LDPC codes [34,35]. In other words, the matrix should be reshaped to have a check nodes connected to variable nodes, by imposing such construction for the decoding matrix, it will be suitable to use the well-known Sum-Product-Algorithm (SPA) with its iterative soft decoding [36]. This adaption for the generator matrix and therefore to the decoding matrix play a significant rule in enhancing the rateless codes performance over the noisy channels like AWGN channel. The first attempt presents by Shokrollahi [19], where he concatenates the LT code as an inner code has degree distribution given by (2.31) with a high rate LDPC code as an outer code.

$$\begin{aligned}\Omega(x) = \; & 0.007969x + 0.493570x^2 + 0.166220x^3 \\ & +0.072646x^4 + 0.082558x^{25} + 0.056058x^8 \\ & + 0.037229x^9 + 0.055590x^{19} + 0.025023x^{65} \\ & + 0.0003135x^{66} \end{aligned} \qquad (2.31)$$

Return to our interest in LT code, referring to Fig. 21, suppose an LT code is generated using equation (2.31) and concatenated with high rate LDPC code and form a raptor code, this new generator matrix will be multiplied by the data symbols vector u to create the coded symbols stream C. The coded symbols stream C is modulated using binary-phase-shift -keying (BPSK) to generate the vector X. The modulated vector will be corrupted by an AWGN which has Gaussian distribution of zero mean and noise power represented by (σ^2) . The decoder receives this noisy vector Y and starts decoding after collecting a sufficient N noisy coded symbols. The decoding method will be based on the soft representation of the output from the channel. The main function of the decoder will be focused on the estimation of the coded symbols based on their maximum a posteriori (MAP). The (SPA) will be utilized in an iterative manner on the bipartite graph which is consists of two sides represented by k variable nodes and N check nodes. Referring to the channel output will be fed to the decoder by the vector Y.

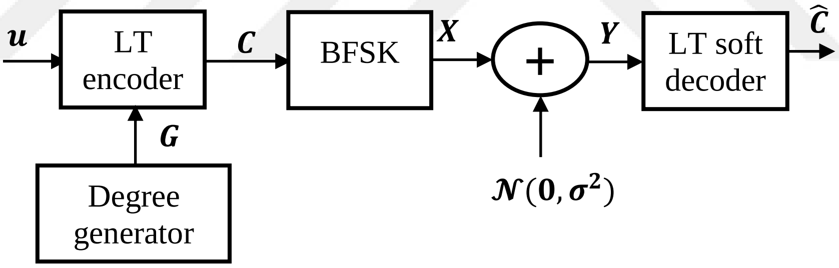


Figure 21 system model for LT coding over AWGN channel

the soft decoding procedure will be run as follows [37]:

5. computing the channel log-likelihood ratio (LLR) to be represented in a vector and calculated as:

$$L_{ch}(j) = ln\frac{p(y_i/x_j = +1)}{p(y_i/x_j = -1)}\ , \qquad j = 1,2, \dots \dots .., N \qquad (2.32)$$

Where $p(y_i/x_j)$ represent the channel probability density function.

6. Several updating iterations take place through a message passing between the check node and the variable node, each message from the check node to the variable node can be calculated to be [38]:

$$L_{c_j \to v_i} = 2tanh^{-1}\ \left(\tan\left(\frac{L_{ch(j)}}{2}\right) \prod_{i \in \mathcal{N}(c_j)\backslash(v_i)} \tanh\left(\frac{L_{v_i \to c_j}}{2}\right)\right) \tag{2.33}$$

Where $\mathcal{N}(c_j)\backslash(v_i)$ represent the group of the variable nodes that are connected to the check node j , such that the variable node i will be excluded.

7. The message back from the variable node to the check node in the same iteration step, will be [38]:

$$L_{v_i \to c_j} = \sum_{j \in \mathcal{N}(v_i)\backslash(c_j)} L_{c_j \to v_i} \tag{2.34}$$

Similarly, $\mathcal{N}(v_i)\backslash(c_j)$ represent the group of the check nodes that are connected to the variable node i , such that the check node j will be excluded.

8. Finally, an estimation for the recovered information symbol will be calculated as:

$$\widehat{u_\imath} = \sum_{j \in \mathcal{N}(v_i)} L_{c_j \to v_i} \tag{2.35}$$

It is worthily mention that the efficiency of the soft decoding approach is much higher than that of the hard-decoding approach in the case of the noisy channel but on the expense of complexity cause this decoding will achieved in an iterative manner to update the probabilities before deciding on the estimated information symbol. Also, we should remember that the shape of the decoding matrix has been changed in order to get a parity check matrix close to that of the LDPC code which would make it suitable for the iterative soft decoding and that what had be done by Shokrollahi in his raptor codes [19].

CHAPTER 3

LT CODES: LITREATUE REVIEW

3.1. LT codes invention

This chapter summarizes the valuable attempts in improving the main features of the LT codes. Since the first appearance of this code in [18], it witnessed a great interest and cited in several research works in the past decade. LT code represents the first identical paradigm of rateless or fountain codes. The generator matrix for such codes can be classified as low density generator matrix. For an erasure channel like the internet, and to send a certain file, this file has to be sliced into group of packets, these packets could have deferent sizes according to the application. During our study for LT codes [18], we found that these codes are mainly based in their design on the characteristics of the degree distribution and whether this distribution can fulfill the essential requirements in encoding and decoding efficiency. Most of the previous works were focused on the analysis of the degree distributions and determined the weak points that can be overcome by either suggesting new distribution or by adapting the main LT distribution which is RSD. As we mentioned above, the first applicable system for the LT codes was the wired internet communication which can be modeled as an erasure channel with a packet or symbol loss represented by the erasure probability (α), however after the invention of raptor codes [19], many new ideas succeed in adapting the conventional LT codes to be suitable for wireless transmission after testing these codes in noisy channels environments like AWGN channel. These new ideas widened the range of the LT codes applications. Even the name is rateless codes for the case of LT codes, but the amount of the extra needed packets or the overhead is a competent factor for good design of such codes. So many researches were presented to test new decoding approaches to decrease the overhead amount and also increase the successful decoding ratio. In the following sections, we try to classify the review of the previous works into their interests which could be in the fields of degree distribution, new decoding approach, application, or special design for short length LT codes.

3.2. Related works in degree distribution

It is well known that LT codes are degree distribution dependent, good degree distribution is the key for better performance. Luby [18], presents his LT codes and states the analysis for both ISD and RSD. In [18], it has been proved that LT code represent the practical paradigm for rateless codes in which the coded stream look likes to be a stream of drops from a water fountain such that we can fill up our cup from any number of drops, scientifically many receivers with deferent channel conditions and deferent accessing time can accumulate enough output coded symbols and decode them to extract the required information messages. Through a comprehensive analysis for the design of the ISD, Luby [18] shows that this distribution offer an ideal behavior with respect to the expected number of the coded symbols required to recover the data symbols however the degree one coded symbols was too small and could be disappear at and decoding step due to channel conditions and as result cause a decoding failure. He also modifies this distribution by inserting some new parameters like the constant (c) and the decoding failure probability (δ) to form the new function ($\tau(d)$) which is added to that of the ISD ($\rho(d)$) and form the new RSD, as described in section (2.2). This new RSD can generate each encoded symbol independently using $O(\ln(k/\delta))$ symbol operations and the k information data symbols can be extracted from any $(k + O(\sqrt{k}ln^2(k/\delta))$ coded symbols with a probability of success $(1-\delta)$ by computational price of $O(k \cdot ln(k/\delta))$. It has been mentioned several times in this chapter that this RSD represent the ultimate distribution for large data lengths, however many efforts during the last decade have been presented to impose more improvements on it, either to decrease the overhead amount or to enhance the error floor or even to compress the complexity price. In [29], a neat discussion for the two main rateless codes LT and raptor codes has been presented. It was shown through histogram graphs the effect of changing the parameter (c) of the RSD on the required number for successful decoding and with large data block of ($k \cong 10000$). He concludes that with such sizes of data blocks the overhead will be in the range of (5%) to (10%) and can be controlled by adjusting the main parameters of the RSD. Esa Hyytiä et al. [39], proposed a performance analysis for short length LT codes to optimize the degree distribution. Two optimizing criteria have been used [40,41], the first optimizing rule was the minimum average overhead required for successfully decode the information message and the second optimization was tested to get maximum successful decoding probability with certain N received coded symbols. The research utilizes a Markov chain approach to reduce the size of the state space as much as possible which was valid for very small values of N. An alternative optimizing approach was combinatorial approach which leads to recursive equations for the success probability. The conclusion result from this research was the degree distribution can be optimized by tuning just few parameters and the freedom is allowed for the other conditions to reach an optimum degree distribution. Another effort was presented by the same authors in [42], they used an iterative optimization algorithm for a parametrized degree distribution. The algorithm basically used an objective function which is built by means of importance ratios, this idea found in importance sampling theory. Like their previous work [39], they concluded that optimum degree distribution needs to fulfill only few conditions and they agree with other previous work [18,29,43] that RSD is the best distribution for bulk data length while the optimum distribution for small messages is still an open issue for further research. An equivalent matrix representation for the LT encoder has been investigated in [44], their investigation focusses on some weakness properties that could appear in the combination of the coded symbols. The proposed equivalent matrix succeeds in overcoming the limitation in decoding performance which came from a repetitive identical row and all zero columns and has been tested for short length messages of ($k = 30\ and\ 60$). Multi-objective evolutionary algorithm based on decomposition is an optimization approach for the degree distribution of LT codes has been proposed in [45]. The research aims to optimize the degree distribution according to application, either by decreasing the overhead required for complete recovery or by minimizing the complexity price which must paid in the form of more symbol operations. The result indicates that the degree distribution can be modified to be suited for different requirements with the aid of these evolutionary algorithms. In [46], a simulation analysis for the RSD and ISD with deferent values for the main parameters (c, δ) to compare the efficiency of decoding success as well as the time required for full recovery, the study approves the superiority of the RSD over that of the ISD and for specific values of (c, δ). An interesting study for the Optimization of degree distributions in LT code with covariance matrix adaptation evolution strategy has been presented in [47]. The evolution for the degree distribution has been done by numerical calculations for each degree probability to get an optimized degree distribution with respect to minimum overhead and decoding failure rate. The usage of this evolution strategy helps to get the optimized distribution and successes in decreasing the required overhead to be less than (10%) and reduces the decoding failure rate for deferent purposes. Xiao and Jiang [48], also interested in small size LT codes and their application in multimedia broadcasting and multicast servers. They had design a new type of rateless codes with special generator matrix, the matrix is a special $0-1$ random matrix in which all the elements are generated independently and has equal probability. In [49], the effect of changing the RSD parameters $(\delta\ )$ and the factor $(x = k/S)$on the average degree and the decoding efficiency of LT code has been studied and analyzed through simulations. The BP early termination has been treated via new encoding technique in [50], the idea aims to reduce the loss probability by enlarging the mean chain size. The input symbols were divided into two groups and each degree two coded symbol should assign one data symbol from each group. By such way, the authors insure that at least two coded symbols will share the same neighbors. By such technique, the results successes in decreasing the unrecovered data symbols for early termination to be in the bound $(0 < n < k/2)$. The same authors in [51] , try to improve the BP decoding algorithm by changing the focus from degree two coded symbol to degree one coded symbol. They force the degree one coded symbol to connect to data symbol from deferent group in order not to get recitative degree one connection. In both research [50,51], even the enhancement has been achieved in the decoding performance but the encoding nature of the LT codes has been broken, since the selection of the data symbols is not uniformly at random any more cause the algorithm will force either degree one or degree two coded symbol to choose his data symbol. Tsai et al [52], utilize a covariance matrix evolution algorithm to optimize a sparse matrix for LT codes generation. The optimization algorithm is used to decide on the best non-zero entries in the sparse matrix. A novel degree distribution based on both RSD and ISD joined to gather in one modified degree distribution was the work presented in [53]. The research uses the fact of certain degrees importance in the RSD which are degree one (necessary for decoding starting) and degree two (important for providing new degree one to the ripple size) and high degree like ($d = 100$) which is useful in keeping the ripple size survival for the last decoding steps. So, the novel degree distribution assign certain probabilities for (d=1,2 and100) while the rest of the degrees have been generated using ISD. By such combining technique, the authors try to gain the strong properties of the both distributions and joined them in one modified probability function. The ripple size has been reshaped by adjusting the RSD focusing on the probability of degree 1 and 2 and the maximum degree in the distribution to set the ripple size close to a fraction of the data length k [54]. This fraction was called the ripple size ratio (RSR) and during the coding procedure the ripple size is reshaped to minimize the mean square error between the expected ripple size and the RSR. The Adjusted RSD performs much better conventional RSD with such ripple size monitoring. With the same degree attention but slightly deferent approach, Yen et al [55] try to increase the proportion of degree 1 and decreasing the other low degree probabilities in the RSD. By this approach less coded symbols that collected by the receiver were able to recover the data information symbols. Also, to prevent degree one from assigning to the same data symbol a non-repetitive integrating encoding scheme was used to improve the LT code ability for unequal error protection (UEP) in the video streaming applications. In [56], different class of LT codes called shifted LT (SLT) codes has been discussed. A feedback is used in this type of codes which inform the encoder about the recovered symbols which help it to shift the generation of the RSD. Usually the shifting is limited to small set of degrees, however in this research a diverse SLT or (DSLT) has been proposed to fully benefit from the feedback acknowledgement, the best results have been achieved with the hybrid SLT/DSLT used in wireless environment in the shape of communication complexity and memory usage. The ripple size issue is reconsidered with a modified RSD (MRSD) in [57]. The same idea presented in [55] was proposed here, to decrease the complexity even with increasing the code length. The authors present a MRSD with only two variables and try to adjust the ripple size by maximizing the expected mean value of it while trying to minimize its variance. An optimizing distributed LT (DLT) has been proposed in [58]. They proposed a generalized design for a rate adaptive degree distribution of LT codes to be transmitted on noisy channels like AWGN channel. A density evolution mean is used as an optimization tool for the DLT codes and the authors present a derivation for the constraints that had been used for this optimization. The paper concludes that the good design LT codes should compromise between the performance and complexity, also, the constraints that they derived during the optimization has not impose further complexity on the rate adaptive degree distribution design. In an erasure channel, Iqbal H. el al [59] study the performance of a (DLT) in a communication system has multi-transmitters connected to multi-relays and serve a single destination. The research proposed a hybrid encoding scheme represented by an encoding process at the transmitters and a combining process at the relays. The authors enhance the equal error protection (EEP) in the erasure channel and overcome the Performance limitation influenced by the information length at the senders when using the conventional DLT codes. For the case (UEP) erasure scenarios, the research enable the new DLT codes to work efficiently with arbitrary number of senders in the network. A statistical measurements comparison has been presented in [60], the conventional RSD, binomial, pareto and, parabolic distributions have been compared to an optimized distribution based on power law. The simulations approve the quality of the proposed degree distribution to be used in the internet communication environment because this distribution got the best records for BER, overhead amount, time for successful recovery and number of iteration when compared to the other distributions. It is also concluded that the pareto distribution came in the second level of this performance evaluation. In [61], based on some important works [62,63], another attempt to use an LT codes with novel degree distribution that can be used in finite fields (or Galois Field, $GF(q)$) representation, has been proposed. The research tries to improve the decoding success rate for such codes but with the same coding cost represented by the overhead packets. The degree distribution uses Raptor degree distribution [19] as the basic frame, and it concatenates with limited random degree values to establish a new degree distribution. With such approach, low degrees of 1 and 2 have high probability about 50% while the rest of the 50% will be generated randomly. The research concludes that as the finite field size increase, a better performance has been noted. By focusing on improving the intermediate performance of rateless codes, Jun et al [64] have proposed a new class of rateless codes utilizing patch zigzag coding approach [65] and testing the code in an erasure channel conditions. The encoder maintains a shifting technique for the input packets before XOR them to make the zigzag decoding applicable. A Poisson Robust Soliton Distribution(PRSD) has been proposed in [66], three main parameters are effected the LT code performance using PRSD which are ($\lambda, c\ and\ \delta$), the mathematical characteristics of the Poisson distribution [67] as well as the average degree of the coded symbols will determine the suitable value for the parameter (λ). The appropriate values for the RSD parameters (c, δ) have been chosen according to the expected ripple size property by applying the artificial fish swarm algorithm (AFSA) [68]. An adaptive degree distribution attempt presented by Yen et al in [69], this has been done through a backward relation between the expected ripple size and the degree distribution. The main approach is to optimize the degree distribution by inserting some constrains to make the error between a predetermined function and the ripple size as minimum as possible. The authors called the result of such mapping a ripple-based distribution (RBD).

3.3. Related works in LT decoding approaches

It has been clearly noted that the degree distribution design for an LT code is the most important criteria for good code performance. However, many attempts in the literature are presented as decoding enhancing approaches where they are used in both erasure and AWGN channels. In chapter two, we present in detail the decoding approach of LT codes which is mainly based on the BP algorithm. One of the first attempts to test LT codes performance in the noisy channels like an AWGN channel has been presented in [70]. The research focused on testing the performance of LT and Raptor codes on two noisy channels, particularly are the binary symmetric channel (BSC) and AWGN channel. The simulation results had approved the superiority of the Raptor codes on that of LT codes in these noisy channels. A simplification for the analysis of error probabilities encounter during BP decoding algorithm, is presented in

[71]. The researchers apply the results gained from the analysis of Darling and Norris [72] to limit the BER of the LT codes as the code length enlarged. The paper also introduced an analysis for the finite length LT codes based on the work of Karp et al [73], by using Poisson model, the authors succeeded in decreasing the iterative programming complexity from $O(N^3log^2N)$ to $O(k\,^2logk)$. A novel systematic LT (SLT) code that used over noisy channels, is the work presented by Nguyen et al in [74]. A reshaping for the generator matrix of an LT code has been done to be composed of two matrices A and B. The matrix A is an identity matrix of $(k\times k)$ and B matrix is an ordinary non-singular LT code generator matrix of $(k\times M)$ size. The resultant $G(k\times N)$ matrix can produce a parity-check matrix H that is suitable to use the powerful soft-decoding iterative approach for the classic LDPC codes. By such matrix formation using this new SLT, the performance over an AWGN channel records very low error floor at low SNR values in the order of $(3-4)$ dB for a code length $(1000{,}3000)$ symbols.

A modified Gaussian elimination (GE) approach for decoding an LT codes over BEC has been called on the fly GE (OFG) [75]. This approach differs from the conventional GE used in [76] that the decoder doesn’t wait for the first k output symbols to start decoding and instead of that it uses the GE procedure while receiving the output coded symbols and arrange them in the matrix one by one following the GE way. OFG approach successes in reducing the complexity price for both triangularization and back-substitution steps used in the conventional GE. Hussein et al [77], try to enhance the performance of the LT codes over AWGN using soft - decoding approach. It is well-known that, LT codes efficiency lowers in an AWGN due to the error propagation when the receiver decides to apply the hard-decoding procedure and even if the decision has been made to switch to the soft-decoding, it also faced by high check nodes complexity cost. The authors in [77] proposed an algorithm that simplified the check node equation by combining both the hard-decoding and the soft-decoding BP approach. They first decide on the code values by using the LLR describes equation (2.32) then use these values as an input to the hard-decoding BP approach. The simulation results approve the quality of such merging approach when using with an LT code of length $(k=1024)$ by applying a number of iterations bounded by 50.

An important factor in the design of the LT code is the ripple size, which represent the number of decodable coded symbols in each decoding step. In [78], an emphasized study on the ripple size has been presented. The authors intend to decrease the ripple size while decoding progress rather than keeping fixed one as proposed by Luby [18]. The research proposes an adaptive function for ripple size decreasing that can be used for any degree distribution. Through a details analysis and simulation for LT codes of lengths (256,512 ,1024,2048), the research concludes that the contribution of the high degree coded symbols in adding a redundant entry to the ripple size is much higher than that coded symbols with low degrees. With such decreasing ripple size code design, the authors succeeded in achieving lower error rate with less overhead compared to the state of art works. The Wiedemann solver [79] has been used as a tool to enhance the decoding performance of LT codes in a new named code called LT-W code [80]. The research presents this LT-W approach as solution for the premature decoding process occur with short length LT codes. This decoding approach decoding ensures effective decoding when a full-rank coefficient matrix is received. The Wiedemann's method is called whenever no degree one coded symbol found in the ripple box, using this method new degree one is generated and resuming the decoding process to minimize the overhead size which is a critical factor for short length LT codes. Wang et al [81] introduce new scheme for fountain codes in which they establish a relation between the coded packet by using an added memory. The research presents a new class for LT codes called LTAM (LT code with added memory), the memory is used as a history that can be used before encoding each new packet, i.e. the new coded packet depends on the past one. This technique successes in decreasing the overhead required for complete data recovery as well as it provides an efficient data transmission among wireless sensor networks by fast data recovering by the receiving nodes which as a result turn them off earlier and saving energy. In an erasure channel environment, an idea for recovering part of the source packets on the fly by using Gaussian elimination decoder was presented in [82]. This technique also introduced an outline for the calculation of gradual packet extraction. The authors also compared several fountain codes and founded that the systematic codes scheme have the best performance in terms of gradual data extraction where these codes can recover about half of the data packets and progress to get the remaining by receiving a minimum number of transmitted packets compared to the other.

A different usage of a memory while encoding is used in [83]. The research using a memory to track the connections of the source packets and decide for the highest connection one. The new encoding technique using RSD is called memory-based RSD (MBRSD). The encoding procedure isolates degree one in one group and use it after encoding all the other degrees, then connect degree one coded symbol to the highest connection data symbol. This LT-MBRSD provides high edge release speed and then successes in achieving minimum error floor for small and large data lengths LT codes compared to the conventional LT-RSD. In [84], an improvement for the BP decoding approach has been achieved by using the well-known GE. The authors present the problem of losing degree one coded symbol for short length LT codes over a binary additive Gaussian noise (BAWGN) channel. An edition has been applied for the generator matrix of the received coded symbols $G(\mathrm{N}\times k)$ by inserting an identity matrix I to get a parity check matrix $H\ = [G, I]$ that is suitable to utilize the efficient iterative BP that used by the LDPC codes. The GE approach is used to remove any stuck in the decoding process where it helps to re-produce degree one coded symbol by using some raw operations in decoding matrix. The same authors in [85] present an analysis for the behavior of the BP decoding based on LLR values calculation used for LT codes over BAWGN channel. They conclude via iterative BP that data nodes can be classified into different classes according to the time at which they start to receive non-zero LLR values. They prove that the data nodes that update their LLR values earlier have great chance to be decoded correctly. Because of such analysis, the research concludes that by increasing the class of fast updating data nodes a lower BER values can be achieved.

Hayajneh and Yousefi in [86] propose a new class of LT codes called an overlapped LT codes. In their code generation, they make the adjacent coded packets overlapped by assigning an overlapped data packets to them. The overlapped data packets between any adjacent coded packets is limited by $0 < \mu < os$ where os represent the overlapped segment of the data packets, if $\mu = 0$ then there is no overlapping between coded symbol representation. The data are discriminated according to priority scheme like that used by [87] also the data symbols are chosen for encoding with different probabilities as the case with unequal error protection (UEP) LT codes utilized in [88,89]. The new rateless code has been defined with four parameters (k, os, μ, l) where l represents the data segment length and the authors approved its compatibility to be used with any degree distribution rather that of RSD and guarantee its performance enhancement. Another application for the concept of LT-UEP is used in [90], the research forced the decoding algorithm to recover the high priority data symbols before the other and produce a new class of rateless codes called LT code with unequal recovery time LT-URT. The encoding process is executed in different levels using different data segments and the coding scheme utilize an intermediate feedback to inform the transmitter to switch to use another data segment cause the last one has been recovered. A new class of rateless codes called Spatially Coupled -LT (SC-LT), is presented by [91,92] into two different channels. In SC-LT the data message also divided into several groups and a synchronized randomness between transmitter and receiver to allocate the space of encoded packets and recover its on data segments. Cheong et al [93] applying the efficiency of the OFG approach to assist the regular BP to reduce the number of overhead packets required for successful recovery when using with short length LT codes. The authors also present a simulation results to approve their complexity saving by using the OFG approach compared to the regular GE as an assisted decoding tool.

3.4. Related works in LT code applications

The invention of rateless codes was an effectual solution for distribution a data file from a certain server to multi-receivers having different channel conditions and allow them for free time accessing. Based on this main idea and since that time of producing LT codes [18], many applications in a diverse communication system applications have been produced. An adaptive video streaming using rateless codes scheme has been presented into two researches by the same authors in [94,95]. The rateless codes are used as FEC to compensate the packet loss effect in the internet communication system. A practical Internet communication system using the H.264 video coder to decide on the average utilizing bandwidth and the average peak SNR (PSNR). This practical data transmission has been applied to compare its result to the simulated work presented in [96]. The comparison has been done with the case of fixed rate FEC codes in which the channel conditions must be available before deciding on the code features and with an updated version FEC code that can be modified in real time situations. Utilizing this real LT code for video streaming outperforms both the pre-mentioned codes with respect to data recovery period and gain in PSNR. The superiority of the rateless codes such as LT [18] and Raptor [19] present them as an efficient solution for scalable data distribution over erasing channel networks [97-100]. New approach for using UEP expanding window fountain (EWF) codes [101] has been proposed to be an enhancing tool for real-time scalable video multicast [102]. It is found that using EWF codes based on UEP is more suitable for this multimedia streaming than the conventional equal error protection (EEP) fountain codes. The rateless codes represented by LT or Raptor codes are used as a mean for data repairing when switching from unicast to multicast video streaming [103]. Similar work is proposed in [104] by Li et al, the LT codes have been used to deliver streaming files from single server to many users even with different decoding approaches. The research try to reach an optimized degree distribution for LT codes that used for such scenario, they do this antically and via simulation results. The influence of different factors like channel characteristics, random requests, and coding approach on delay in communication, effective channel usage and throughput have also been approved via the optimization analysis and results. The optimized LT codes success in reduce the total required bandwidth when has been compared to the case of transmitting individual data streams for multi-users or even for one of the users. One of the first implementation of an optimized degree distribution LT codes over a WiMAX networks has been presented in [105]. The optimization of the degree distribution has been done using a convex function to reach the best optimized results.

Wireless body area network (WBAN) is another interesting application that can use the efficiency of rateless codes to mitigate the packet errors while transmission. In [106] the realization of LT code as a low power consumption correcting code (due to its simple construction) is used for increasing the fidelity of the communication link in a WBAN system. The experiment has been achieved by using a WBAN to transmit a body medical data using the IEEE 802.15.6 communication specifications to a receiving monitoring device and compare the error mitigation capabilities between

Reed-Solomon (RS) code as a fixed-rate code with that of LT code as a rateless code. The implementation results approve the superiority of the LT codes to be a flexible, less power consumption more efficient error correcting code especially for bad channel conditions when the error values exceed the error correction capabilities of the fixed rate codes. The UEP [107] approach which gives a priority for some source symbols to recovered first has been an attractive coding technique for some application which requires such priority. Arsalan et al [108] proposed a modified LT code based on the UEP scheme and suited for progressive decoding application when not all the data parts have the same importance, such as image or video sources compressed in a progressive or scalable style. The coded symbols are modulated using BPSK scheme ad transmitted over free space optical (FSO)Channel. The authors [109] demonstrate the enhancing effect of using LT codes in recording lower BER values for the signal over FSO systems in which the performance degrades due to fluctuations of atmospheric turbulence of the FSO channel. The performance of LT codes in a wireless sensors network is the work presented in [110]. In [111] an investigation for the degree distribution parameters of LT codes proposed to work for distributed network coding (DNC) scheme. The authors proposed a degree distribution based on the degree value and the odd degrees and the simulation results approve the performance enhancement for the DNC using LT code new design. An effective employing of the rateless codes properties proposed in a cooperative coding scenarios. Two scenarios for the next generation broadcast network wireless (NGB-W) system are simulated. The first scenario tests the ability of the rateless codes to improve the communication link. between the terminal users (TUs) and the base station when the broadcasting tower and the base station used different degree distributions and the (TUs) try to decode the received signal by merging both degree distribution. While in the second scenario, the TUs cooperate between them to receive the signal from the base station when there is no direct link between them. in both scenarios, the rateless coding support the TUs to get more reliable signals. The breakable nature of the communication link in deep space as well as the high importance of the downlink makes the deep space communication a serious challenge for the conventional fixed rate channel coding. A research presented in [112], proposed cascading LT code with high rate LDPC code to form a rateless code that fit the requirements for reliable communication link in deep space communication. The application of such rateless codes decreases the amount of errors and increases the fidelity of the received signals. Another special case of communication system is the wireless sensor networks (WSN) in which applying the LT code features is highly support the performance of such networks. In [113,114] an efficient application of LT codes based distributed coding (LTDC) scheme used in wireless sensor network in order to increase the strength of data storage and efficiency of data recovery. Al-Awami et al [115], study the effect of applying LT codes as an effectual coding to reduce the complexity in a limited resources system like the (WSN). The proposed (LTDC) , is built in a way that be compatible for the WSN, The (RSD) is used to allocate the choice probabilities to storage nodes , and randomly distributes its data over the storage network. The simulation results show a noticeable energy consumption while approaching the storage demands.

The concatenating of LT codes with LDPC codes to form the well-known Raptor codes [19], motivates the researchers to present another copy of such cascading procedure, like the research presented in [116]. An LT code is used as an outer code with a polar code [117] as an inner code are concatenated and utilized in an underwater acoustic communications (UAC) where impulsive noise is considered as unavoidable and the receiver records severe BER. It is shown via simulation results and derivation analysis that BER values of the suggested polar-LT code can be enhanced in the impulsive noises and it gives better result than that of the concatenating LDPC-LT code and can be effectively utilized in other impulsive noise channel applications like radar and UAV communication systems. The same cascading scheme of polar-LT codes has been proposed in [118] as an effective secure coding scheme in wiretap model on BEC. The proposed code can perform in such communication system without the need of previous information about the channel characteristics. The analysis and the simulation results approve the rateless code features superiority on that of the finite rate codes to provide more secure transmission over such channels. one of the most recent application of LT codes is the one proposed in [119] where the codes are used in cloud computing as a virtual storage for the data of internet users.
The research depends on the flexibility of the LT codes structure to retrieve the source data stored in the distributed cloud storage. The authors conclude that with good

coding structure of LT codes the users can easily retrieve their resources even when they demand the same file type and in the same time.

In the previous sections of this chapter, we summarize some of the enormous applications of the rateless codes and particularly LT codes in order to clarify the great importance of the investigation for improving the performance of such codes for different new applications, which is the work that will be presented in the following chapters.

CHAPTER 4

NEW PATTERN-RECOGNITION DECODING APPROACH

4.1. Contribution summery and related background

Luby Transform (LT) codes and Raptor codes are the first practical rateless erasure codes over erasure channels. The degree distribution represents the core in the design of such codes and it strongly effects the performances of these codes. As described in chapter 2, the BP algorithm is utilized in iterative structure to recover the information symbols from that of the received coded symbols. The existence of degree-one coded symbols is essential for the starting and continuation of the decoding process. The absence of degree-one coded symbol at any instant of the iterative decoding operation results in the decoding failure. To overcome this problem, we proposed a novel approach that can be used in the absence of degree-one coded symbol to break the decoding stuck and continue with the decoding operation.

In order to present a complete idea for the solution presented in our contribution, a recall for several basic ideas of LT encoding-decoding structure as well as some related works to our proposed method. A noticeable attention has been dedicated for the LT codes which are considered as capacity approaching erasure codes. The name rateless codes came from the ability of such codes to produce an unlimited number of coded symbols from any k data symbols. An extra needed coded symbol (usually referred to the overhead and denoted by (ϵ)), should be collected by the receiver to make the total received coded symbols:

$$N = k(1 + \epsilon) \tag{4.1}$$

The rate of the code is changeable and depends on the successful recovery acknowledgment, when a certain N coded symbols have been used by the decoder and complete recovery of data symbols has been achieved. However, the case is different for fixed rate codes such as block codes the rate is decided to meet the worst channel conditions. This sometimes brings unnecessary overhead when channel is in good state. On the other hand, rateless codes bring a comfortable flexibility for the code rate, i.e., for the transmission overhead [78].

Luby transform codes have small encoding and decoding complexities. The k data symbols can be recovered using BP algorithm with an average decoding complexity [18]:

$$dec.complexity_{avg} = k \cdot \ln(k/\delta) \tag{4.2}$$

which is an estimation for the symbol operations for a probability of successful decoding $1 - \delta$.

Let us remember the formation of LT code, in the beginning the source file is truncated into k data symbols $(u_1, u_2, \cdots, u_k)$. The length or the size of each packet can be a single bit or a group of bits (in the whole thesis, we consider each packet as a single bit). Then, a degree distribution $\Omega(d)$ is used to generate a random digit d called a degree. For each generated degree, a set of source symbols are uniformly chosen and exclusively ORed to form the coded symbol C_i. The generated symbols are transmitted over a binary erasure channel. The effect of the channel appears in the shape of erasing some of these output symbols and the number of lost coded symbols varies according to the erasure probability of the channel, i.e., α. The decoder can start decoding if the degree information and the set of neighbors of each encoding symbol are available at the receiver [18].

When N group of coded symbols are received, and have at least one coded symbol of degree one, BP algorithm can be utilized for decoding operation. BP algorithm works in an iterative manner employing the coded symbols to recover the k data symbols. The decoder either successes in decoding all the data symbols or declares failure and this highly depends on the degree distribution and the shape of the generator matrix while forming the coded symbols. The decoder declares failure if at certain decoding step, a coded symbol of degree one cannot be found and the decoding process halts. Degree distribution, which affects the encoding and decoding efficiency, plays a critical role for the design of LT codes [120].

The rapid surge of the data flows over the internet encouraged the researchers to focus on fountain codes with low encoding and decoding complexities when compared to the previously known classical codes such as such as Reed-Solomon block erasure codes [46]. Numerous progresses have been made on the design of LT codes. Some of these studies are done to enhance the degree distribution to improve the overall performance of the code. A good degree distribution should provide enough degree-one coded symbols. Moreover, the probabilities of low degree values should be high enough such that decoding can continue until the last information symbol is recovered. The ripple size is the number of degree-one coded symbols available in each decoding step. In addition, the degree distribution should provide a large probability value for a high value degree. This is necessary to guarantee that all the information symbols are used during the encoding operation.

Robust soliton distribution (RSD), one of the distributions satisfying all the above mentioned constrains, is used to generate the degree values for coded symbols. Studies for improving RSD are revisited in a detailed review in the previous chapter. Based on using the RSD as a degree distribution for the encoding part and the conventional BP algorithm for the decoding part, the LT decoders can suffer from the absence of degree-one code symbol at any stage and declare failure. We focus on this issue and look for some ways to continue the decoding operation even in the absence of degree-one coded symbols. This is the main motivation of contribution where we present new pattern based decoding operations in which we intend to prove that using some of the patters in the Tanner graph connections decoding operation can still be continued even in the absence of degree-one coded symbol which is the reason for the declaration of decoding failure for conventional LT decoders.

4.2. LT codes construction

In this section, we provide brief revision for the main procedure of the encoding and decoding processes of the LT codes that has been presented in chapter 2.

4.2.1. LT encoding

The first step of the LT encoding is to divide the data file into packets. Then a degree d is generated according to a distribution, and the encoded packet is generated by XORing d source packets which are chosen uniformly at random. A packet can contain a single bit or a group of bits. Mathematically, the encoding operation can be formulated as:

$$C = uG \tag{4.3}$$

where u represent the information packets and G is the generator matrix whose column size is a constant number but its row size is a variable number whose value depends on the time instant at which acknowledgement signal is sent from the receiver side. G is a binary matrix. The 1's in the columns of G indicate the chosen information packets during the generation of corresponding coded packet. The number of 1's in a column of G represents the total number of information packets chosen to generate the coded packet, i.e., represent the degree of a coded packet. More explanations are illustrated in the following example.

Example 1: let the information file is divided into 4 packets (bits) which are (1 0 0 1) and by using a degree generator, the columns of G can be formed as:

$$G = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \cdots \\ 1 & 0 & 0 & 1 & 1 & 0 \cdots \\ 0 & 1 & 1 & 1 & 0 & 0 \cdots \\ 0 & 0 & 1 & 0 & 1 & 1 \cdots \end{bmatrix} \tag{4.3}$$

Then using (4.3) the coded packets are found as:

$$C = [0 \quad 1 \quad 0 \quad 0 \ 1 \ 1 \cdots] \tag{4.4}$$

Tanner graphs are used to graphically illustrate the encoding and decoding operations of LT codes. It shows the connections between the data packets usually called the left-hand side and the code packets usually called right hand side. In (4.4) if only 6 code packets are generated, its Tanner graph illustration can be given as in Figure 22.

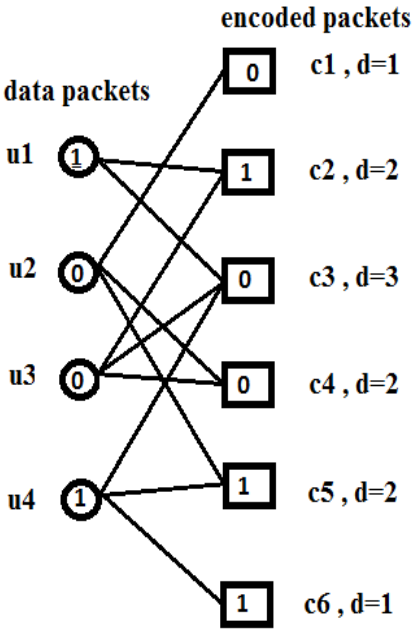


Figure 22 Tanner Graph representation of LT code for $k = 4$ and $N = 6$

After the reception of enough encoded packets, the decoder starts decoding the received packets through the use of message passing algorithm which includes message exchange between the left and right hand sides of the Tanner graph. For a data file consisting of k source packets, the decoder will achieve an average of $O\left(k \cdot \ln\left(\frac{k}{\delta}\right)\right)$ packet operations. The k source symbols can be recovered from any $k + O\left(\sqrt{k}\ln^2\left(\frac{k}{\delta}\right)\right)$ encoded symbols with probability $1 - \delta$ where δ represent the decoder failure probability [18]. The probability distribution $\Omega(d)$ of the right-hand side degrees is a precious part of the design for the LT codes, below we should remember the essential degree distribution that has been proposed with the first appearance of LT codes in [18] which is the RSD. As it is prementioned in chapter 2, the RSD is a modified version of the ISD, which is formulated as [18]:

$$\rho(\mathrm{d}) = \begin{cases} \frac{1}{\mathrm{k}} & \text{for } \mathrm{d} = 1 \\ \frac{1}{\mathrm{d}(\mathrm{d}-1)} & \text{for } \mathrm{d} = 2, 3, \cdots \mathrm{k} \end{cases} \tag{4.5}$$

After adding some modification on this ISD, RSD is given by [18]:

$$\Omega(d) = (\rho(d) + \tau(d))/\beta \tag{4.6}$$

Where $\beta = \sum_d \rho(d) + \ \tau(d)$ and $\tau(d)$ is a probabilistic function with the formula:

$$\tau(d) = \begin{cases} \frac{S}{k}\frac{1}{d} & for\ d = 1,2, \cdots, \frac{k}{S} - 1 \\ \frac{S}{k}\ln\left(\frac{S}{\delta}\right) & for\ d = \frac{k}{S} \\ 0 & for\ d > \frac{k}{S} \end{cases} \tag{4.7}$$

in (4.7), the parameter S represent the expected number of degree one coded symbol which given by:

$$S = c\,.\ln(\frac{k}{\delta})\sqrt{k} \tag{4.8}$$

with c is a constant between 0 and 1 and δ represent the decoder failure probability.

4.2.2. LT decoding

Again, we have revisit the decoding algorithm based on iterative BP which is used for decoding LT codes, as mentioned in section (2.3). The BP algorithm depends mainly on finding the degree one-coded symbol to start the decoding or to keep continue recovering the remaining data symbols.

Algorithm 2: LT decoding

```
***Repeat***
***Search*** for a coded symbol $C_n$ which has a single connection (degree one).
***Copy*** its value to its connected data symbol $u_k$.
***Update*** the coded symbols values of the neighbors to the recovered data symbol
    using bitwise addition.
***Release*** all the recovered data symbol connections.
***Until*** (recovering all the data symbols, or no degree one coded symbol found)
```

4.3. Memory-based robust soliton distribution (MBRSD) LT code

We choose MBRSD [83] as a new coding scheme for LT codes which uses the RSD and modifies the encoding technique by splitting the degrees into two groups. One group for only degree one and the second group is dedicated for the other higher degrees. Assigning degree one coded symbol is delayed until completing encoding the higher degrees ($d > 2$) coded symbols. Then degree one coded symbols are assigned to the higher degree data symbols, i.e. the data symbols having the higher connection edges. The memory is needed to track the data symbol connections to decide on the data symbols with higher degrees. The above description can be summarized by the following algorithm as introduced in [83].

Algorithm 3: LT-MBRSD encoding approach

```
***Repeat***
***Generate*** a degree $d$ from a right-hand side distribution like RSD
    RSD.
 ***If*** $d = 1$, then
    ***choose*** the data symbol with highest instantaneous degree without
    replacement.
***If*** $d \neq 1$, then
    ***choose*** $d$ data symbols uniformly distributed
***Perform*** XOR of the chosen $d$ data symbols to generate the code symbol $C_n$and
transmit it.
***Until*** acknowledgement signal is received.
```

The effect of such encoding scheme on the decoding part appears in increasing the number of released edges or connections in each decoding step, and this causes to make the decoding process perform better since the chance to add new degree one coded symbol to the ripple size increases as a result. To illustrate the MBRSD concept, let's consider the following example:

Example 2: suppose a data message consists of 8 data symbols and the encoder of an LT code generates the coded symbols using RSD, if the decoder receives 10 coded symbols with a degree vector $D = [2 \quad 3 \quad 2 \quad 3 \quad 1 \quad 3 \quad 3 \quad 1 \quad 2 \quad 3]$, let us define the decoding generator matrix as $G_d = G^T$ which has fixed columns and variable rows, each row represents a coded symbol equation in which an indication for its degree and data symbols connection is illustrated. As shown in (4.9), G_d is (10×8) decoding matrix for the 10 received coded symbols. For instance, it's clear that we have two degree one coded symbols C_5 and C_8, connected to u_9 and u_8 respectively. When applying MBRSD the formation of the generator matrix G is changed causing to change the decoding matrix. The memory tracking for the data symbol connections nominates u_4 and u_8 to be connected to the coded symbols C_5 and C_8 respectively as indicated in (4.10).

$$G_d = \begin{array}{c} \begin{array}{cccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right] \end{array} \qquad (4.9)$$

$$G_d = \begin{array}{c} \begin{array}{cccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right] \end{array} \qquad (4.10)$$

4.4. LT with Pattern Recognition (LT-PR)

As mentioned previously the absence of degree-one code symbol at any instant of the iteration results in decoding failure. In LT encoding, coded symbols are formed by taking XOR of randomly selected data symbols. When a degree-one symbol is found, the data symbol connected to degree-one symbol is decoded directly. And the resolved data symbol is added to the other coded symbols which contain the data symbol in its XOR formation. In this way, the degrees of the coded symbols are reduced.

This can be mathematically explained as: if

$C_i = u_l$ and

$$C_m = u_l \oplus u_n \oplus \cdots \tag{4.11}$$

then C_m is reduced as:

$$C_m = C_m \oplus C_i \tag{4.12}$$

This operation can be generalized for the coded symbols having degrees more than one.

Let C_i be a coded symbol such that $deg(C_i) > 1$ where $deg(\cdot)$ is the degree function whose output is the degree value of coded symbol C_i. Let's define the run-set as:

Definition 4.1: The run-set of the coded symbol C_i represents the set of data symbols used while generating the coded symbol C_i . and the symbolic representation can be formulated as:

$$rs(C_i) = \{u_l, u_s, \cdots\} \tag{4.13}$$

For the coded symbols C_i and C_j **,** if

$$rs(C_i) \subset rs(C_j) \tag{4.14}$$

then C_j can be reduced as

$$C_j = C_j \oplus C_i \tag{4.15}$$

after which the degree of C_j reduces to:

$$deg(C_j) = deg(C_j) - deg(C_i) \tag{4.16}$$

This is the motivation of this approach. In other words, if we cannot find a degree-one code symbol then we can look for degree-two, degree-three, etc. code symbols and try to reduce higher degree symbols using lower degree symbols using (4.14) -(4.16). That is, we look for some code patterns in other code patterns paying attention to the degrees and run-set of the code symbols.

Pattern searching can be done either using the connections between data and coded symbols of the Tanner graph or inspecting the columns of the generator matrix of the LT encoder. The following example gives the required information to be close for understanding the proposed approach.

Example 2: Assume that the decoding generator matrix of an LT code is as given in (4.17):

$$G_d = \begin{array}{c} \begin{array}{cccccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \tag{4.17}$$

It is clear that, there is no degree one in all the received coded symbols, investigating the matrix in (4.17) shows the following degrees, indicating in Table 1.

Table 1: degrees of the coded symbols for the decoding generator matrix

$\boldsymbol{C_i}$	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
$\mathbf{deg}(\boldsymbol{C_i})$	2	2	2	3	2	2	2	3	3	2

where it is seen that there is no degree-one code symbol. A conventional LT decoder using belief propagation declares failure at the beginning of the decoding procedure. However, when the run-set of the code symbols are inspected we see that:

$$rs(C_3) = \{u_7, u_9\}$$

$$rs(C_4) = \{u_7, u_8, u_9\}$$

$$rs(C_3) \subset rs(C_4) \tag{4.18}$$

The connections for the code symbols C_3 and C_4 in the Tanner graph of the LT code represents the required pattern among all patterns constructed in the Tanner graph, these special patterns are the core of the proposed LT-PR. The shape of the pattern that constructed by the C_3 and C_4 coded symbols is shown in Figure 23.

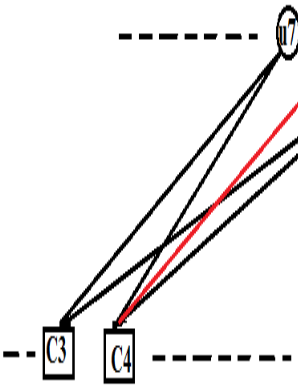


Figure 23 Tanner graph connections for the required pattern

Using (4.18) C_4 can be simplified by using bitwise addition as:

$$C_4 = C_4 \oplus C_3$$

reducing the degree of C_4 to:

$$deg(C_4) = deg(C_4) - deg(C_3)$$

$$\deg(C_4) = 3 - 2 \to 1 \tag{4.20}$$

which is nothing but degree-one. Since degree-one coded symbol is again available, we can continue decoding with conventional approach. In addition, a similar relation is also available for the code symbols C_7 and C_8, i.e., $rs(C_7) \subset rs(C_8)$ and $C_8 = C_8 \oplus C_7$. Thus, it can be concluded that even in the absence of degree-one code symbols it may be possible to continue decoding by searching some coded patterns in others. The simplest approach can be searching degree-two coded patterns inside degree-three coded patterns.

The Tanner graph gives us the image of these patterns that can help us on continuing the decoding process. Let us define some terms in Tanner graph as follows. Code nodes and data nodes are the points in the Tanner graph where edges are connected to each other. Cycle is a set of nodes and edges such that a node can be reached from any one of the other nodes tracing the connecting edges. For instance, in Fig. 2 the nodes labelled by C_3, C_4, u_7, u_9 and the black edges connecting all these nodes form a cycle. If two code nodes are in the same cycle then the coded symbol with smaller degree representing one of these nodes can be used to reduce the degree of the coded symbol representing other node.

The decoding approach using Tanner graphs can also be interpreted using the generator matrix of the LT code. Let r_i and r_j be the two rows of G_d, $d(r_i)$ and $d(r_j)$ are the number of ones in the rows r_i, r_j respectively. The $\deg(C_i) = d(r_i)$ and $\deg(C_j) = d(r_j)$. If $d(r_i \oplus r_j) < d(r_i)$ or $d(r_i \oplus r_j) < d(r_j)$ then $C_i$or C_j can be simplified using either $C_i = C_i \oplus C_j$ or $C_j = C_j \oplus C_i$. While choosing C_i or C_j for simplification, we can use the criteria:

$$C_k = \underbrace{\text{argmin}}_{k} \left| d(r_i \oplus r_j) - d(r_k) \right|, k \epsilon \{i, j\} \tag{4.20}$$

If $d(r_i \oplus r_j) > d(r_i)$ and $d(r_i \oplus r_j) > d(r_j)$ then $C_i$or C_j cannot be used to simplify each other. For simplicity of the illustration, we can consider the algorithm-3. In this algorithm, we search for the coded symbols that only differ by a single packet in their run-sets.

Algorithm 4: Pattern-Recognition decoding approach (LT-PR)

```
1: Let $G_d$ represent an $Nxk$ binary decoding generator matrix, for $N$
   received coded symbols
2: ***Investigate*** $G_d$, if $\deg(C_i) = 1$ ,then
3: ***follow*** the same steps for conventional BP decoding in algorithm 1
4: ***else***
5: ***calculate***   $d(r_i \oplus r_j), i = 1, j = 2, \cdots, N$
6: ***If***  $d(r_i \oplus r_j) = 1$,then
7: ***proceed*** as in step-3.
8: ***else***
9:  ***increment*** the $i$ value, i.e., $i = i + 1$ and repeat step-5
```

In order to prove the ability of the proposed LT-PR decoding approach for the mitigation of the decoder stuck, we present a simulation results to compare our approach with the state of art decoding approach represented by, the MBRSD [83] as an encoding-decoding approach and the BP assisted by GE approach as an efficient decoding approach for the case of short length LT codes. To present a complete vision for the performance of each decoding approach, let us present a summarized revision for the BP-GE approach as well since we had illustrated both of our approach and the MBRSD one in previous sections.

4.5. LT with BP assisted by Gaussian-Jordan elimination (BP-GE)

It is clear now that the BP algorithm needs the degree-one coded symbol in order to start decoding. Especially with short length LT codes and due to the small fraction of degree one while the generation of the degrees using RSD, these degree-one coded symbols may be lost and cause the decoding failure. So, to solve such problems the designers proposed some backing tools that helps the decoder to continue decoding. One of the efficient tool is the well-known Gaussian-Jordan elimination method and we shortly denoted by GE. We had mentioned that the generator matrix of the LT code represents a set of linear equations and the BP algorithm as proposed in [18] is used to solve these equations by using the message-passing technique. In the BP-GE, we deliver the received decoding matrix to the GE procedure first, which is summarized by the following steps:

1. Triangularization step:

   As described in [76], The main object of the triangularization is to use row operations and row and column reorganization to adjust the decoding matrix in the shape of an upper triangular $(k \times k)$ matrix having a diagonal of ones and zeros below it by deleting the lower rows if exists. If the GE procedure successes in this step then it proceeds to the second step, if not according to the BP-GE decoding approach, it returns to the conventional BP algorithm described in algorithm 1.

2. Back-substitution step:

   It represents the final decoding step which is achieved when the triangularization has been succeeded. In this step, the decoder proceeds to adapt the triangular matrix into the identity matrix, which means a successful decoding mission.

4.6. Simulation results and statistical calculations

The main purpose of our proposed approach is to improve the performance of the LT codes of short length messages. The improvement in the decoding performance is illustrated into two parts. The first comparison is done by comparing our improvement in the decoding performance, when comparing our LT-RSD-BP-PR with conventional LT-RSD-BP, LT-MBRSD-BP and LT-RSD-BP-GE for extremely short length message of ($k = 32$ and $k = 16$). where in the second part, we approve the improvement in the decoding performance for LT-RSD-BP and LT-MBRSD-BP for larger data length, particularly of ($k = 256$). The LT code is generated using RSD with parameters $c = 0.02, \delta = 0.1$ and is transmitted over a BEC characterized by an erasure probability of ($\alpha = 0.02$). Simulations are implemented in a MATLAB environment and the results are collected after (100) erroneous frames are received. This means that the number of transmitted packets changes for every rate. For our proposed approach, we tried the technique described in algorithm-3.

In Figure 24, the BER performances of LT code using different decoding approaches that assisting the Belief Propagation (BP) are presented. The comparisons are done between our proposed Pattern Recognition assisted BP, i.e., BP-PR, with that of the regular BP, BP-MBRSD [83] and the well-known Gaussian Elimination assisted BP-GE [76]. It is clear from Figure 24 that the LT code with RSD employing the proposed method outperforms the conventional BP-RSD and BP-MBRSD methods for all the rated under concern. For BER performance, our proposed method has similar score to that of the BP-GE [76].

For the sake of the complexity issue, it is known that for the regular BP in order to recover the k source symbols from any N coded symbols with probability $1 - \delta$ an average of $O\left(kln\left(\frac{k}{\delta}\right)\right)$ symbol operations were needed [18]. On the other hand, the required number of additive operations for BP-GE is about $O(N^2)$ [76]. For our proposed method, the additive complexity is about $O(l * m)$, where $1 \leq l \leq N$ and $1 \leq m \leq N - 1$. When $N = k + \epsilon$ and $l = m = N$, the worst case additive complexity of our proposed method is like that of BP-GE approach. However, it is obvious that the probabilistic average complexity of our proposed approach is less than that of the BP-GE method. From this point of view, it is obvious that's the proposed method is more efficient in terms of computational complexity.

To emphasize the need for the PR to assist the conventional BP and particularly for the short length messages, let us discuss the statistical calculations presented in Table 2 and Table 3. In both tables, we intend to illustrate the contribution of the PR for the decoding enhancement of the regular BP. In Table 2, we generate the code using RSD with the parameters of ($c = 0.02$ and $\delta = 0.1$) and by transmitting (1000) frames of length ($k = 32$). It is shown that BP-PR gives better results for all the code lengths ($N = 32$,48 and 64) and the number of useful patterns that satisfy the condition for algorithm 3 are used to remove the decoder stuck when the decoding generator matrix has no degree-one coded symbol. The number of these patterns change randomly due to the random degree generation and the random allocation of the data symbols for each coded symbol. These numbers tell us that the decoder could use the removal stuck patterns more than one time in certain frames since the number of PR patterns is greater than the number of the decoding frames, which is (1000).

In Table 3, the same test has been done for the same data and code lengths but with different RSD parameters, we use the parameters of ($c = 0.1$ and $\delta = 0.05$). As it is clear from the records, better decoding success for all the code lengths have been achieved for both BP and BP-PR. By comparing the best results from both tables 2 and 3, we found that the best record is the one with the code length of ($N = 64$) in which the LT-RSD with BP-PR succeeded of decoding (978) frames out of (1000) frames without errors for the case of RSD with parameters of ($c = 0.1$ and $\delta = 0.05$). however, we should mention that the higher successful decoding records for the case of RSD with parameters of ($c = 0.1$ and $\delta = 0.05$) is achieved on the expense of higher decoding complexity since the average degree is about (5) compared to (3) for that of RSD with parameters of ($c = 0.02$ and $\delta = 0.1$).

In Table 4, the data length ($k = 16$) is used. We see that BP-PR still outperforms the conventional BP decoding approach. The number of useful PR patterns is also increased due to decrease the expected number of degree one coded symbols when the data length is decreased.

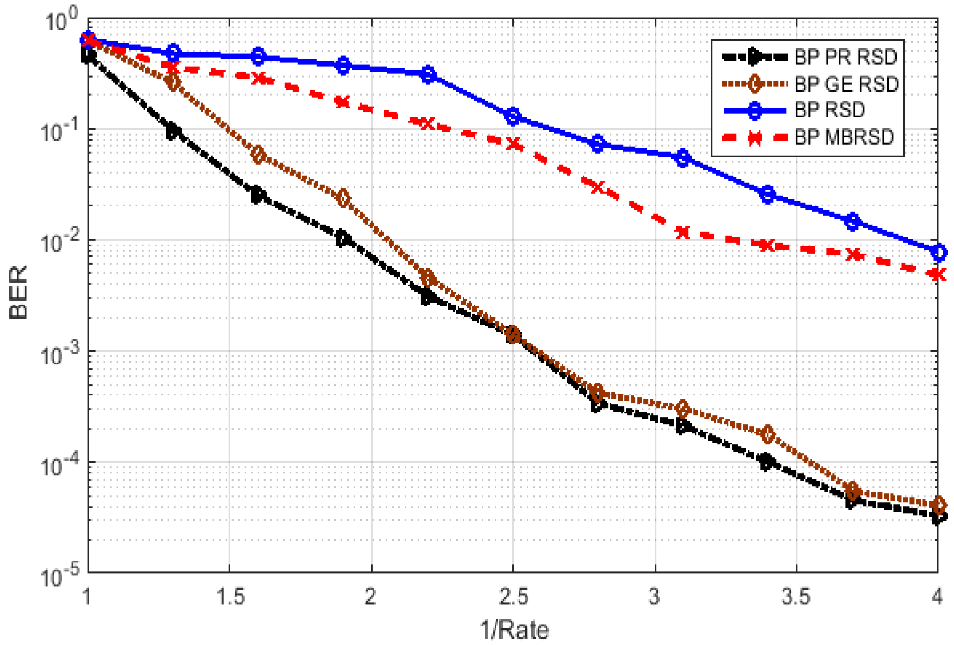


Figure 24. BER comparison between RSD-BP, MBRSD-BP, RSD-BP-GE and RSD-BP-PR for k=32 over BEC of α=0.02

Table 2. Number of useful patterns used by BP-PR for LT-RSD of $k = 32$, $c = 0.02, \delta = 0.1$ with an erasure probability, of $\alpha = 0.1$

K	N	Success (BP)	Success (BP-PR)	PR Patterns
32	32	0	2	1532
	48	394	505	1242
	64	750	854	1098

Table 3. Number of useful patterns used by BP-PR for LT-RSD of $k = 32$, $c = 0.1, \delta = 0.05$ with an erasure probability, of $\alpha = 0.1$

K	N	Success (BP)	Success (BP-PR)	PR Patterns
32	32	0	3	1287
	48	613	751	1347
	64	949	978	1016

Table 4. Number of useful patterns used by BP-PR for LT-RSD of $k = 16$, $c = 0.1, \delta = 0.05$ with an erasure probability, of $\alpha = 0.1$

K	N	Success (BP)	Success (BP-PR)	PR Patterns
16	16	1	7	1294
	24	152	689	1558
	32	530	832	1589

For larger data length $k = 256$ the BER performances of LT code using BP-RSD, BP-MBRSD, and BP-PR-RSD, BP-PR -MBRSD employing algorithm-3 are depicted in Figure 25.

It is clear that, the proposed approach enhances the performances of LT-like codes in all rates. This improvement is due to the removal of decoding block due to the absence of degree-one coded symbols.

Another criterion for the performance of rateless codes is the decoding success rate or its counterpart failure rate. The decoding success rate is a measure of decoder performance and it is defined as the ratio of total number of correctly decoded packets to the total number of transmitted packets. The simulation is performed for the two data lengths $k = 32$, $k = 256$. The performance graph is depicted in Figure 26. For the data length $k = 256$, the RSD-BP-PR achieves 100% of decoding success at a rate of 1.5 while the MBRSD-BP needs an additional rate of 0.25 achieve the same performance. For the data length $k = 32$, LT-RSD-BP-PR achieves 100% of decoding success at a rate of 2.75 while MBRSD achieves the same performance at a rate of 3.25.

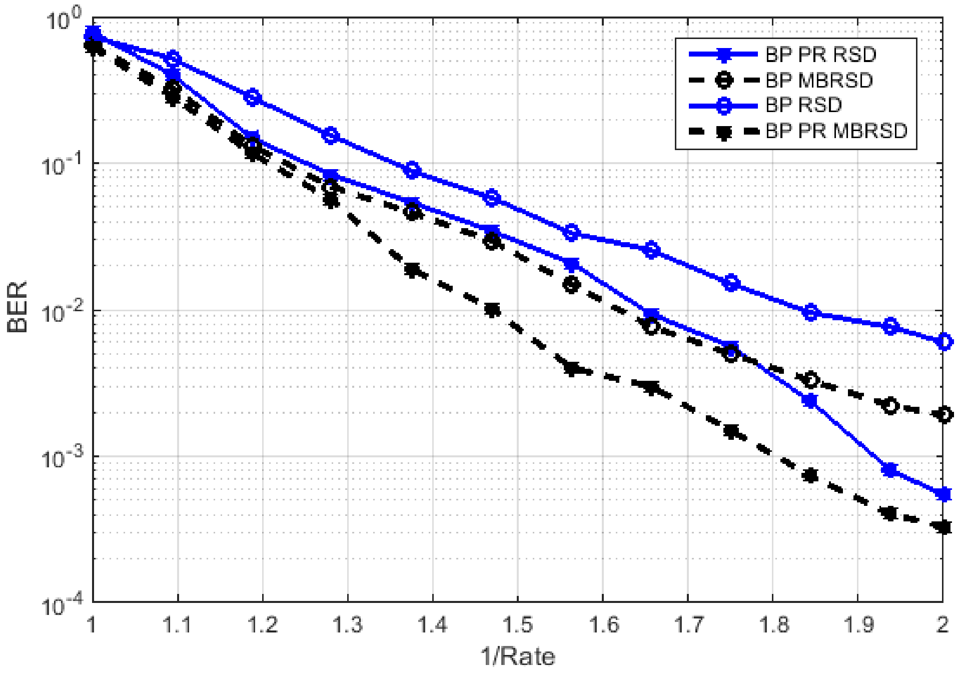


Figure 25. BER comparison between RSD-BP, MBRSD-BP, RSD-BP-PR and MBRSD-BP-PR for k=256 over BEC of α=0.02

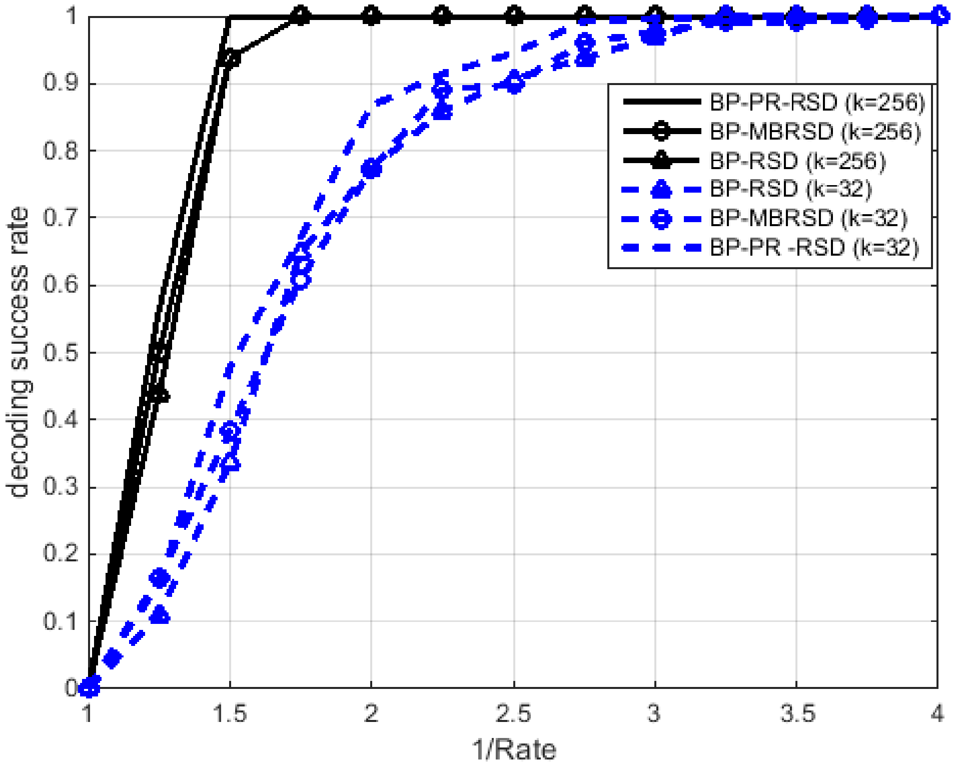


Figure 26. RSD-BP, MBRSD-BP and RSD-BP-PR for k=32 and 256 using RSD with parameters c=0.02, δ=0.1with an erasure probability of α=0.02

Finally, to see the effect of the proposed PR decoding approach as a performance improvement tool, let us compare its performance with the other three decoding approaches that have been pre-mentioned (regular BP, MBRSD-BP, and BP-GE). We shall transmit (1000) frames of length ($k = 32$) and decoding them using RSD degree generator procedure. The channel effect imposed as an erasure factor with erasure probability of $\alpha = 0.1$. The decoder collects ($N = 2k$) in its buffer, then tries to decode each frame starting with length ($N = k + 1$), if it is unable to recover all the data symbols then add another coded symbol from the buffer, and so on until data fame recovery is achieved or use all the buffered coded symbols without success. The results of this simulations recorded as a statistical calculation for three main performance factors named as (number of the successful recovery frames, the average overhead percentage, and the average unrecovered symbols). In Table 5, the results approve the performance enhancement of the decoding approaches that assist the regular BP but with different efficiencies. The MBRSD represent encoding-decoding approach at the same time and it adds a little improvement in all the prementioned performance factors. On the other hand, both BP-GE and BP-PR add a significant improvement to the decoder performance. BP-PR successes in recovering higher number of data frames compared to that achieved by BP-GE but it needs a little higher overhead coded symbol.

Table 5. Statistical calculations for the decoder performance using different decoding approaches for LT-RSD of $k = 32$, $c = 0.02, \delta = 0.1$ with an erasure probability, of $\alpha = 0.1$

Decoding approach	K	N	Recovered frames	Overhead percentage	Unrecovered data symbols	Average degree
Regular BP	32	64	777	65%	3	3
MBRSD-BP			779	62%	3	
RSD-GE			859	53%	1	
RSD-PR			865	56%	1	

From the above simulation results and the statistical calculations, we conclude that the problem of absence of degree-one code symbols during the decoding of LT codes which results in the decoding failure can be solved. By using our proposed BP-PR, we succeeded in overcoming this problem. The BP-PR which has been tested for different short and moderate data lengths approve its ability to enhance the performance of the LT-like codes over the BEC. The proposed BP-PR has recorded the lower error floor when compared with that of conventional BP and BP-GE. Even when the recorded BER are so close to that of the BP-GE but the proposed algorithm requires less computational complexity which means shorter time for recovering the data file.

CHAPTER 5

NEW DETERMINISTIC CODING APPROACH FOR SHORT LENGTH LT CODES

5.1. Contribution summery and related background

Digital fountain codes initiated by Byers et al, launch an interesting solution for dispensing bulk data file to multi receivers having different channel situations and different accessing time. Luby transform (LT) code represents the typical model for such codes which characterized by a linear encoding-decoding complexity. The most important design factor of LT codes is the structure of the degree distribution. In this chapter, a new deterministic approach for degree generators has been proposed. The degrees are generated from 1 up to a pre-determined value called repetition period R_p and the data symbols are selected sequentially from the information message divided into frames of length R_p. MATLAB simulations are done in a binary erasure channel environment utilizing the proposed encoding method to generate short length LT code and comparing the performance to an LT code using the conventional robust Soliton distribution (RSD) [18] and two state of art distributions named memory based robust soliton distribution (MBRSD) [83] and optimal degree distribution (ODD) [42]. The simulation results approve the superiority of the proposed degree generators over that of the pre-mentioned distributions with respect to error floor, successful decoding rate, overhead and the complexity price for encoding-decoding operations.

During the last decade authors paid a noticeable attention for the design of degree distribution of LT codes. The first distributions used were the ideal soliton distribution (ISD) and robust soliton distribution (RSD) introduced by Luby [18]. It has been approved that RSD has the optimal performance for the case of broadcasting bulk data files [18, 19 and 29]. Many attempts have been focused on enhancing the LT-RSD codes to achieve a better performance with smaller data files. Esa Hyytiä et al [39] proposed a performance analysis for short length LT codes to optimize the degree distribution. Two optimizing criteria have been used to minimize the average overhead required for successful decoding maximize the probability of successful decoding with certain N received coded symbols. They conclude that the degree distribution can be optimized by tuning just few parameters and freedom is allowed for the others to reach an optimum degree distribution. Another effort was presented by the same authors in [42], they used an iterative optimization algorithm for a parametrized degree distribution. The algorithm basically used an objective function which is built by means of importance ratios. They reached to the results that optimum degree distribution needs to fulfil only few conditions and they agree with other previous work [18,29] that RSD is the best distribution for bulk data length while the optimum distribution for small messages is still an open issue for further research. The BP early termination has been treated as a new encoding technique in [50] in which the idea aims to reduce the loss probability by enlarging the mean chain size. The input symbols were divided into two groups and each degree two coded symbol must assign one data symbol from each group. By this way, the authors insure that at least two coded symbols will share the same neighbors. This technique is successful in decreasing the unrecovered data symbols for early termination to be in the bound $0 < n < k/2$. Tsai et al [52] utilize a covariance matrix evolution algorithm to optimize a sparse matrix for LT codes generation. The optimization algorithm is used to decide on the best non-zero entries in the sparse matrix. A novel degree distribution based on both RSD and ISD joined to gather in one modified degree distribution was the work presented in [53]. The research focus on certain degrees in the RSD like degree one (necessary to start the decoding) and degree two (important for providing new degree one to the ripple size) and high degree like ($d = 100$) which is useful in keeping the ripple size survival for the last decoding steps. The novel degree distribution assign certain probabilities for ($d = 1,2$ and 100) while the rest of the degrees have been generated using ISD. By such combining technique, the authors try to gain the strong properties of both distributions and joined them in one modified probability function. A modified RSD (MRSD) has been presented in [57] which aims to decrease the complexity even with increasing the code length. The authors present a MRSD with only two variables and try to adjust the ripple size by maximizing the expected mean value of it while trying to minimize its variance. A statistical measurements comparison has been introduced in [60], the conventional RSD, binomial, pareto and parabolic distributions have been compared to an optimized distribution based on power law. The simulation results over binary erasure channel approve the quality of the proposed degree distribution with respect to many performance factors. A Poisson Robust Soliton Distribution(PRSD) has been proposed in [66], three main parameters (λ, c and δ) characterized the PRSD. The mathematical characteristics of the Poisson distribution and the average degree of the coded symbols are used to decide on the suitable value for the parameter λ. The appropriate values for the RSD parameters (c, δ) have been chosen according to the expected ripple size property by applying the artificial fish swarm algorithm (AFSA).   A state of art work is presented in [83] where a memory-based robust Soliton distribution (MBRSD) is proposed. In this proposal, each degree-one code symbol is connected to the data symbol which has the highest degree without replacement., in this way a non-uniform   left-hand distribution has been used. The main aim of the research is increasing the edge release and accelerating the decoding process which result in a better decoding performance.

All the degree distributions that have been reviewed focus on the optimization of the RSD. As a new attempt in this paper, we propose a deterministic approach for the generation of degrees rather than the random technique used in RSD. In addition, the selection of the data symbols is in a serial manner.

5.2. The proposed deterministic encoding algorithm

In this section, let us first list some of the degree distributions that are used to generate short length LT code and compete our proposed LT with deterministic encoding (LT-DE). We use three degree distributions beside our proposed one, they are the RSD which is explained in detail in section (4.2.1.) and the MBRSD which is also described in section (4.3) and the last one is the optimal degree distribution (ODD) which is presented in [42]. Let us describe the (ODD) first before introducing our proposed one.

5.2.1. Optimal degree distribution

A detailed analysis has been made in [42] to optimize the degree distribution via an algorithm using iterative optimization based on methods utilized in importance sampling theory to build an objective function which may be used to optimize the parametric degree distribution. The resultant optimize degree distribution has the best performance with respect to successful decoding and minimum overhead but not with complexity issue. In this paper, we truncate the optimized results for the case of $k = 100$ to be suitable with our tested length $k = 32$ and this truncation is necessary to give reasonable complexity price represented by the average degree. Referring to [42] the optimal degree distribution is given by:

$$\Psi(d) = \begin{cases} 0.18 & d = 1 \\ 0.34 & d = 2 \\ 0.27 & d = 4 \\ \frac{1}{d(d-1)} & else \end{cases} \tag{5.1}$$

The probability values listed above are normalized by the factor

$$\beta = \sum_{d=1}^{k} \Psi(d) \tag{5.2}$$

and finally, the ODD has the form of:

$$\Omega(d) = \Psi(d)/\beta \tag{5.3}$$

5.2.2. The proposed deterministic encoding (LT-DE)

It was explained in the previous sections that RSD can get the optimal performance with some modifications. However, it is mainly designed for bulk data frames while those of length less than 10^3 still have performance limitations [121]. The main reason for such limitations is the overhead required, i.e., a decoder on average requires

$(1 + \epsilon) \cdot$ k packets to successfully decode the source **k** packets. The percentage of these extra needed packets for LT codes is approximately **5**% for $k = 10000$ and it is approximately 30% for $k = 100$ [122]. For this reason, to decrease the overhead required for LT codes having short data length we propose a deterministic encoding approach. In our proposed approach, for an information sequence consisting of **k** data symbols, first we generate the degrees in a deterministic manner as follows:

$$deg(C_n) = \begin{cases} n, & n = 1, 2, \cdots, R_p \\ n - m \cdot R_p, & n = R_p + 1, \cdots, k \end{cases} \tag{5.4}$$

where $deg(C_n)$ represents the degree of the n^{th} coded symbol and R_p is the repetition period value which also represents the maximum degree. This R_p is adjusted as required and it can be considered as the main parameter in the DE approach, it may take the value bounded by $3 \leq R_p \leq k/2$. Large R_p means large average degree and more encoding-decoding complexity. The first R_p coded symbols have degrees generated serially from 1 up to R_p, then the same degrees are used again for the next R_p coded symbols and so on till reach the required code length. To fulfil this repeated frame of degrees an integer represents the frame number denoted by m and is belong to $m \in (0,1,2, \cdots, k/R_p - 1)$ for the first k coded symbols.

Let us consider this example, for an information sequence consisting of k symbols represented by $(u_1\, u_2\, u_3 \cdots u_{k/2}\ u_{k/2+1} \cdots u_k)$ with $R_p = k/2$ and $m \in \{0, 1\}$ the coded symbols $(c_1\ c_2\ c_3 \cdots c_k)$ have their degrees using (6), the degrees are tabulated into two frames to be $(1,2,3, \ldots k/2; 1,2,3, k/2)$ and the data symbols are selected after divided them into segments of length R_p. Thus, our data vector u is divided into two frames

$(u_1\, u_2\, u_3 \cdots u_{k/2})$ and $(u_{k/2+1}\, u_{k/2+2}, \cdots u_k)$. Using this scheme, the output coded symbols are generated using the following equations:

$$\begin{aligned}
c_1 &= u_1,\\
c_2 &= u_1 \oplus u_2,\\
c_3 &= u_1 \oplus u_2 \oplus u_3,\\
c_{k/2} &= u_1 \oplus u_2 \oplus u_3 \cdots \oplus u_{k/2},\\
c_{k/2+1} &= u_{k/2+1},\\
&\vdots\\
c_k &= u_{k/2+1} \oplus u_{k/2+2} \oplus u_{k/2+3} \cdots \oplus u_k
\end{aligned} \tag{5.5}$$

It is clear from (5.5) that the size of the generator matrix is $k \times k$and for further generation of coded symbols the same degree frames are generated sequentially but the data symbols are selected from random segment generator. It is clear that for the first k coded symbols, it is guaranteed that no data symbol is left out of the encoding operation even with very small code length while this may occur frequently for RSD due to the random data selection method in the encoding process. For each coded symbol two parameters are required to identify it to the decoder, these are (d, m). In our example above $c(d, m)$ with the values $c(1,0)$ is the first coded symbol connected to the first data symbol, while $c(1,1)$ is the first coded symbol in the next frame which is $c_{k/2+1}$ and is connected to the first data symbol in the next segment of the data vector which is $u_{k/2+1}$.

By sharing the value of R_p between the two sides of the communication system, the encoding generator must assign the values of d, m to complete the encoding process without the need to select the data symbols indices because they are taken serially. Figure 27 illustrate an encoding scheme for a coded symbol of degree 3 and produced by combining three data symbols from the first data segment. It is also noticed that the serial selection for the data symbols prepare an appropriate pattern to use our previous decoding enhancement approach, the so-called belief propagation-pattern recognition (BP-PR) [A2-1]. On the other hand, at each decoding step, the degree one-coded symbol is connected to the highest degree data symbol which the idea of the MBRSD [83], but without the need for extra memory.

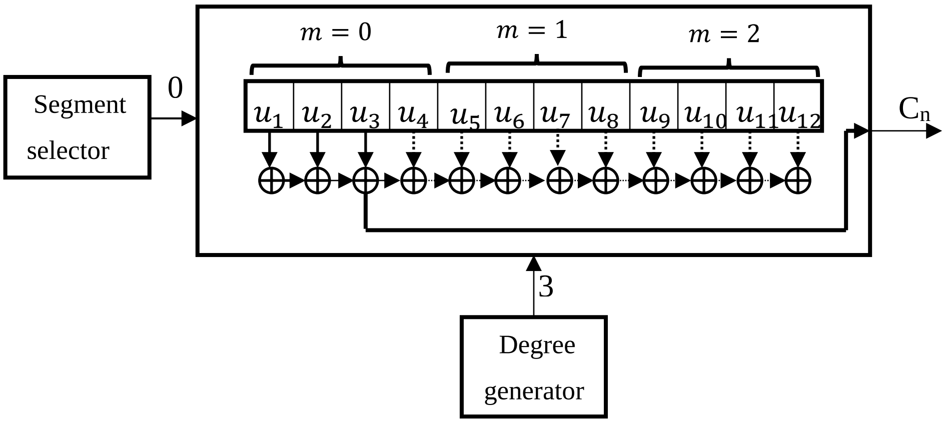
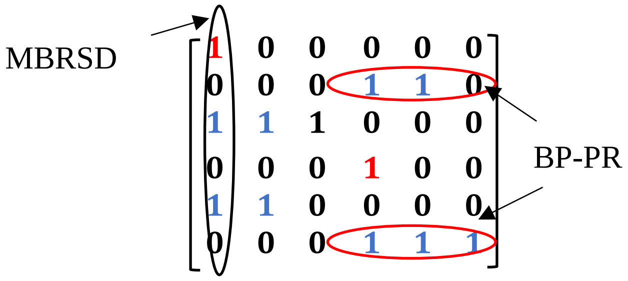

$m = 0$
$m = 1$
$m = 2$
Segment selector
0
u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} u_{11} u_{12}
C_n
3
Degree generator

Figure 27 DE scheme for $c(0,3)$

In Fig. 3 an illustration for these facts is presented. The generator matrix below is used for decoding purpose, each row represents a coded symbol equation and each column represent the data symbol connections. Let $k = 6$ and the coded symbols have been generated according DE approach such as: $[c(0,1), c(1,2), c(0,3), c(1,1), c(0,2), c(1,3)]$. This matrix fulfils the conditions for BP-PR and it is used the idea of MBRSD.



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

Figure 28 DE decoding matrix for $k = 6, R_p = 3$ and $m = (0,1)$

In order to implement the ideas that has been presented in DE approach, algorithm 5 is used for this purpose.

Algorithm 5: Deterministic Encoding (LT-DE)

```
Set the value of $R_p$
Calculate the upper value of $m \leftarrow m_{up} = \frac{k}{R_p} - 1$
Randomly generate the segment selector value from
    $m = (0,1,\dots,m_{up})$ without replacement.
***Repeat***
    generate a repeated degree frame from 1 up to $R_p$
    **if** $d = 1$, **then**
        $c = u_{mR_{p+1}}$
    **else**
        $c = u_{mR_{p+1}} \oplus u_{mR_{p+2}} \cdots \oplus u_{mR_{p+d}}$
**end if**
***Until*** (Acknowledgement is obtained from receiver)
```

5.3. Performance improvement of LT-DE over BEC

Computer simulations have been used to generate short length LT codes for $k = 32,24,16$. Four encoding techniques that have been mentioned in Section 3 are used to generate the output coded symbols. For the case of RSD and MBRSD the parameters are chosen as $c = 0.02$, $\delta = 0.1$ while for the case of DE, $\mathrm{R_p} = 4$ is chosen to be the repetition period. The code is transmitted over a binary erasure channel characterized by an erasure probability of $\alpha = 0.02$ and the decoder uses the BP-PR as a decoding approach. Two strategies are employed to implement the test, in the first one the results are shown in Figure 29 up to Figure 31. In this testing strategy, we try to transmit the code with different rates and the measures are recorded after collecting (100) error frames for each point on the graph. The decoder can collect the dedicated code length for each rate then start decoding. In the second strategy, the decoder collects a code length of N and starts decoding with code length $N = k$ then take an extra coded symbol from the buffer one by one to recover all the data symbols.

The BER comparison for an LT code of data frame length $k = 32$ is shown in Figure 29. An interesting achievement has been recorded for the DE approach at the point of zero overhead. It records a BER of 0.0345 compared to 0.4557 as a best error rate achieved by the ODD. The DE has the best error floor and records the minimum BER for all the overhead points on the graph. ODD performance came in the second level and has better BER than that recorded by RSD and MBRSD. The best error floor has been recorded at the point of $R^{-1} = 3$ which is listed as 0.000015 for DE, $0.0002$for ODD, 0.0003 for MBRSD and RSD.

In Figure 30, the frame error rate FER records comparison is shown. At the starting point of the graph (zero overhead) approximately all the frames are in error for the distributions of RSD, MBRSD and ODD while about half of them are recorded as an error frames for the case of DE. Again, it is obvious that the DE has the best FER for all the points on the graph. Figure 31 represents the negative image of Figure 30, however in this Figure the number of frames is fixed to be (1000) transmitted frames and the successful decoding rates have been drawn. For the case of DE, more than half of the 1000 frames were recovered correctly without any extra overheard and it outperforms all the other distributions for the rest of overhead points. ODD has better successful decoding rate than RSD and MBRSD but still below the rate recorded by our DE.

In our simulations, we first measured the performance of the DE approach with respect to error floor and successful decoding rate. To have a complete vision for the quality of our proposed coding approach, we use some statistical measurements to compute several important factors that could be used to clarify the ability of our approach to have minimum overhead and minimum complexity cost compared to the other distributions. In these simulations, an LT code with length $k = 32$ is encoded using DE, RSD, MBRSD and ODD, then 1000 frames have been sent over a BEC of erasure probability $\alpha = 0.02$. When the received number of symbols equals to $N = k$ the decoder starts decoding, and some extra coded symbols are stored into the buffer If successful recovery is not possible the decoder uses some more symbols from the buffered coded symbols. In Table 4, performances of RSD, MBRSD, ODD and DE are depicted with respect to the average value of:

1) Average degree(AD): simply computed as:

$$AD = round\left(\sum_{run=1}^{1000}\sum_{i=1}^{N}\frac{\deg(c_i)}{1000}\right) \tag{5.6}$$

2) Symbol operations (SO): This factor reflects the effect of the AD on the decoding complexity represented by the symbol operations that executed during the decoding process. These operations for each decoding step include: (fetch $G_{N\times K}$ for $\deg(c_i) = 1$, detecting u_j that connected to this c_i, $\widehat{u_j} = c_i$, update the value of all the coded symbols in the j column, $G(i,:) = 0\ , G(:,j) = 0$. The SO is computed as:

$$Avg(SO) = ceil\left(\sum_{run=1}^{1000}\frac{SO_{run}}{1000}\right) \tag{5.7}$$

3) Extra needed symbols(ENS): represents the number of coded symbols that have been used by the decoder while trying to recover the data frame.

$$Avg(ENS) = ceil\left(\sum_{run=1}^{1000}\frac{ENS_{run}}{1000}\right) \tag{5.8}$$

4) Unrecovered symbols(URS): it measures the error during each decoding run represented by the number of unrecovered data symbols.

$$Avg(URS) = ceil\left(\sum_{run=1}^{1000} \frac{URS_{run}}{1000}\right) \tag{5.9}$$

5) Complete recovery frames(CRF): it represents the number of successful decoding frames from that of the 1000 received frames.
6) The overhead ratio(OVHD): This ratio represents the amount of overhead needed by the decoder and it has been recorded by computing the average ENS to total available overhead in the buffer, we calculated as:

$$OVHD = \left(\frac{ENS}{N-k}\right) \tag{1}$$

In Table 6, we compare all the degree generation approaches using data frame length $k = 32$ and test all the above-mentioned factors for two code lengths. For the case of $k = 32, N = 36$, the DE shows the best results for all the factors. For instance, for the error records represented by the factor URS, it has only 2 unrecovered symbols compared to an error of $10,11,8$ for the MRSD, RSD and ODD respectively. With this short length, the DE also achieves the highest number of CRF with a total of 554 frames from the 1000 frames under test compared to the best records achieved by the other for the ODD which recovers only 96 frames. This high frame recovery achievement has been recorded with an overhead 0.75 and average symbol operation of 667 which represents the best records from all the other degree generation approaches under test.

The effect of the complexity issue is clarified by the records of the SO because the average degree seems to be fixed due to the round function. For the code length $N = 64$, it is seen that the DE outperforms the others and has the best results of 990 CRF recorded with minimum OVHD of 0.2812 and minimum SO of 787.

In Table 7, we had test the DE approach for shorter data length. for the $k = 24$ the best results recorded at $N = 48$ which has a CRF of 992 by using an OVHD of 0.25 achieved with SO of 459. When the data length was $k = 16$, it is obvious that the best results are CRF of 995 which needs only an OVHD of 0.187 and achieved using SO of 227.

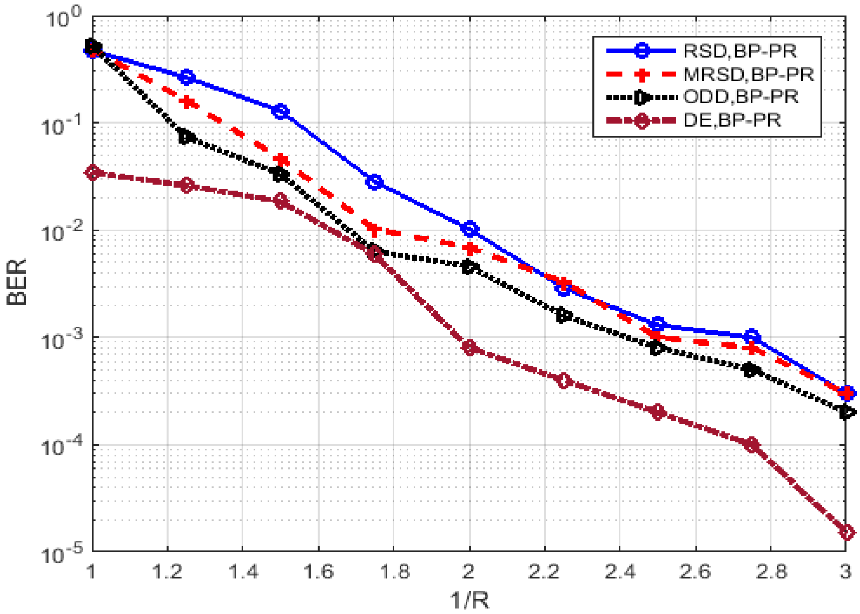


Figure 29 BER comparison for LT code of length $k = 32$ using different degree generation RSD, MBRSD, ODD and DE over BEC with erasure probability $\alpha = 0.02$.

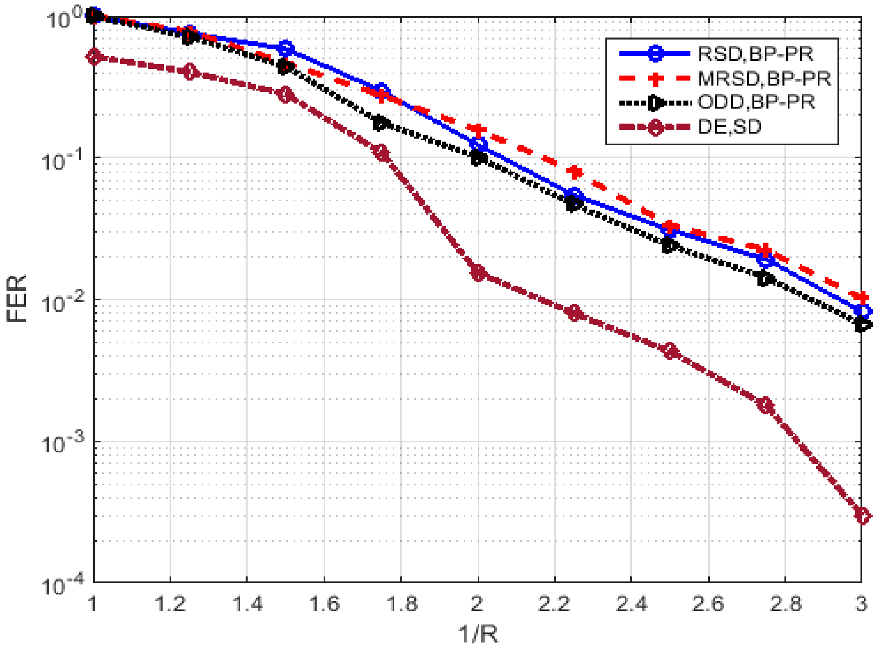


Figure 30 FER comparison for LT code of length $k = 32$ using different degree generation RSD, MBRSD, ODD and DE over BEC with erasure probability $\alpha = 0.02$.

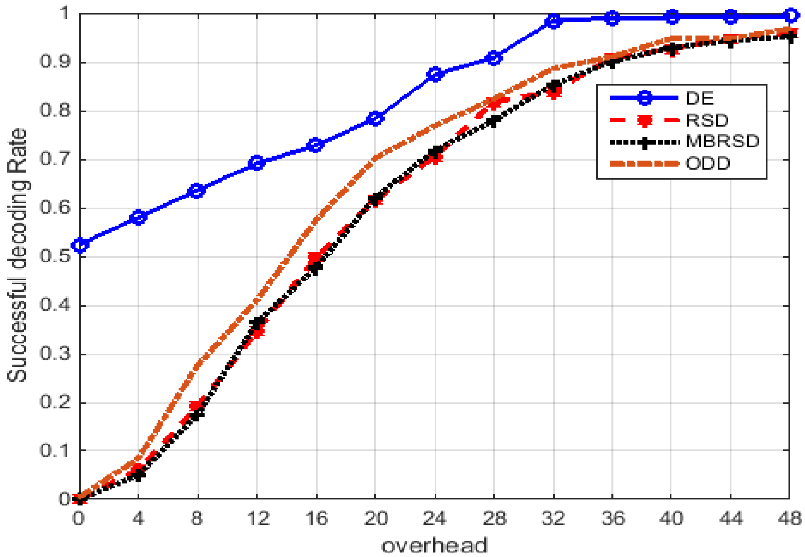


Figure 31 Successful decoding rate comparison for LT code of length $k = 32$ using different degree generation RSD, MBRSD, ODD and DE over BEC with erasure probability $\alpha = 0.02$.

Table 6 Statistical calculation comparison for LT code of length $k = 32$ using RSD, MBRSD, ODD and DE degree generations over BEC with erasure probability $\alpha = 0.02$.

Degree generation	k	N	AD	SO	ENS	URS	CRF	OVHD
DE $\mathrm{R_p} = 4$	32	36	2	667	3	2	554	0.75
		64	2	787	9	1	990	0.2812
MBRSD	32	36	3	1075	4	10	68	1.0
		64	3	1212	19	1	854	0..593
RSD	32	36	3	1215	4	11	72	1.0
		64	3	1276	18	1	875	0.562
ODD	32	36	3	949	4	8	96	1.0
		64	3	1080	16	1	879	0.50

Table 7 Statistical calculation comparison for LT code of length $k = 24{,}16$ using DE degree generation over BEC with erasure probability $\alpha = 0.02$.

Degree generation	k	N	AD	SO	ENS	URS	CRF	OVHD
DE $\mathrm{R_p} = 4$	24	28	2	408	3	1	684	0.75
		48	2	459	6	1	992	0.25
	16	20	2	211	2	1	784	0.50
		32	2	227	3	1	995	0..187

5.4. Performance improvement over AWGN channel

. In this section, short length LT codes are generated using efficient DE with random data selection and performance tests have been done over additive white Gaussian (AWGN) channel. The encoding technique illustrated in section (5.2.2) and implemented in algorithm 5, led to mutual relation between coded symbols that will be used in the decoding part. The decoding complexity of the proposed structure is decreasing to the order of $O(k)$. The simulation results show that the proposed approach has better performance in terms of error floor and number of completed recovery frames when compared to LT codes using robust soliton distribution RSD, MBRSD and ODD even when supporting these distributions by the decoding enhancement approach BP-PR.

LT [18] and Raptor [19] codes are mainly proposed as an efficient rateless codes over erasure channels, however many attempts have been presented to test the performance of such codes over noisy channels such as binary symmetric channel (BSC) and AWGN are evaluated [70]. The first attempts conclude that even with large data frames such as $k = 1000$ and $k = 10000$ these codes suffer from high error floor when applying the regular BP. Inspecting the decoding matrix of LT codes inspired many of the early efforts whom are testing these codes over AWGN to adjust this matrix to be closed to the well-known low density parity check code (LDPC) matrix [34,35]. This adjustment allows the decoder to use the iterative soft decoding of such codes [123,14]. Another idea for the generator matrix adjustment has been presented as systematic LT code [74]. The generator matrix of the received code symbols for such systematic LT codes become more suitable to use the standard belief-propagation soft decoding algorithm. In [58] a rate-compatible degree distribution of LT code for AWGN channels was proposed. In this study, several criteria, such as maximal rate or low complexity or even the constrained-iteration performance are taken in consideration for the optimization of the degree distribution. Like Shokrollahi method [19], Chen et al [124] proposed a new type of rateless code suitable for noisy channels and they had obtained their code by concatenating low density generator matrix (LDGM) rateless codes with simple post-codes, and this code structure becomes similar to that of Raptor codes. A Gaussian Jordon elimination (GJE) method has been used to overcome the problem of missing degree-one coded symbol for short length LT codes over AWGN channel has been presented in [84]. The received generator matrix $G(N\ x\ k\ )$ of the coded symbols has been used to construct a parity check matrix $H = [G\ ,I\ ]$ where $I = (N\ x\ N\ )$ identity matrix, this I matrix has been added to make check matrix suitable to use the BP decoding algorithm used in the decoding of LDPC codes [125]. Another approach has been presented in [126], where a group weight distribution of an LT code has been determined and a refined union bound used to evaluate the act of LT codes under the maximum likelihood decoding before the application of soft decoding algorithm.

In all the pre-mentioned researches there is an attempt to insert additional check nodes in the generating matrix of the received LT codes to approach the compatibility of the soft decoding algorithms. This treatment imposes significant decoding complexity and increases the time required to extract the required information symbols especially when increasing the iteration number for the soft decoding to enhance the performance. In this section, a less complex approach will be applied. we intend to use the exact generator matrix of the received coded symbols without any alteration. The decoding will be achieved into two steps: first, a soft demodulation will be done to estimate the coded symbols values. Second a hard-decoding using BP-PR approach [A2-1] is utilized for the LT codes with degree distributions of RSD, MBRSD and ODD while the DE is used the new sequential decoding (SD) approach.

In Chapter 4, a detail description for the BP-PR has been introduced, the next section is dedicated for introducing the simplicity procedure of the SD approach.

5.4.1. Sequential decoding approach (SD)

Referring to (5.5), let us suppose that the decoder can collect k coded symbols of DE approach and first it rearranges them as descried in (5.5), then we can illustrate the encoding diagram for the first($R_p = 4$),for example, as:

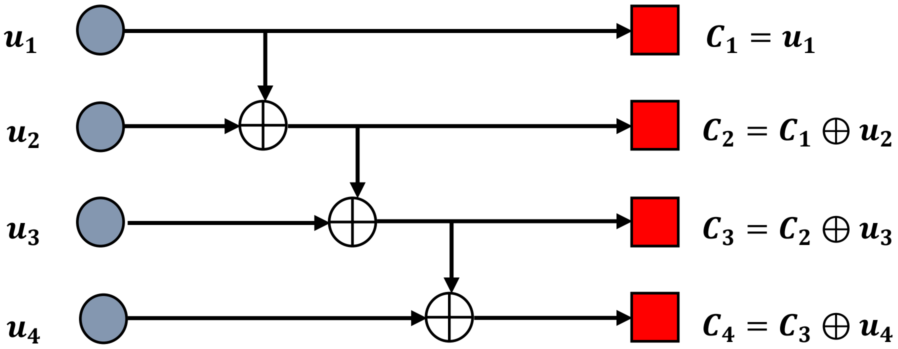


Figure 32 Sequential encoding approach for the first repetition period with $R_p = 4$.

The relations between successive coded symbols can be used by the decoder to recover the data symbols. Assume that the generator matrix in (5.5) is used for the encoding operation. Then for the set of estimated symbols $\hat{C} = (\hat{c}_1, \hat{c}_2, \hat{c}_3, \cdots \hat{c}_N)$, the coded symbol which connected to a single data symbol can be used recover its connected data symbol as:

$$u_i = \widehat{c_i} \tag{5.6}$$

while for any adjacent coded symbols having degree greater than one the decoding can be achieved using:

$$u_j = \hat{c}_j \oplus \hat{c}_{j-1} \quad for \quad j = 2,3,4 \tag{5.7}$$

So, it requires only $O(k)$ symbol operations to recover k data symbols from any k coded symbols rather than an average of $O(klog(k))$ symbols for the regular BP with $N = k + \varepsilon$ received coded symbols [18].

5.4.2. System model

Figure 33 shows the scenario used in our simulations. A file of length k will be represented by a set of random binary symbols, $u = (u_1 \; u_2 \; u_3 \cdots, u_k)$. An LT code of length N is produced by multiplying this data stream by the generator matrix G which can be generated using one of the degree distributions (RSD, MBRSD, ODD or DE). The coded symbols $C = (C_1\ C_2\ C_3 \cdots C_N\ )$ are modulated using Binary phase shift keying (BPSK) and producing the modulated signal V which is transmitted over an additive white Gaussian noise (AWGN) channel with noise power σ^2. The received signal equals to $Y = V + n$. The decoder has two sections:

1. **Soft demodulation**

Let the received noisy codeword represented by $Y = (y_1, y_2, y_2, \cdots y_n)$,then the probabilities $Pr(C_i = 1\ /y_i)$ and $Pr(C_i = 0\ /y_i)$ can be calculated as:

$$Pr(C_i = 1\ /y_i) = (1 + e^{-\frac{y_i}{\sigma^2}})^{-1} \tag{5.8}$$

and,

$$Pr(C_i = 0\ /y_i) = 1 - Pr(C_i = 1\ /y_i) \tag{5.9}$$

Comparing (5.8) and (5.9), the value of the coded symbol C_i is estimated based on the maximum likelihood probability (MLP).

2. **Hard decoding**

In this part, the estimated coded symbols are fed to the BP-PR hard decoder for the case of LT code using (RSD, MBRSD and ODD) while the case of DE is decoded with SD approach.

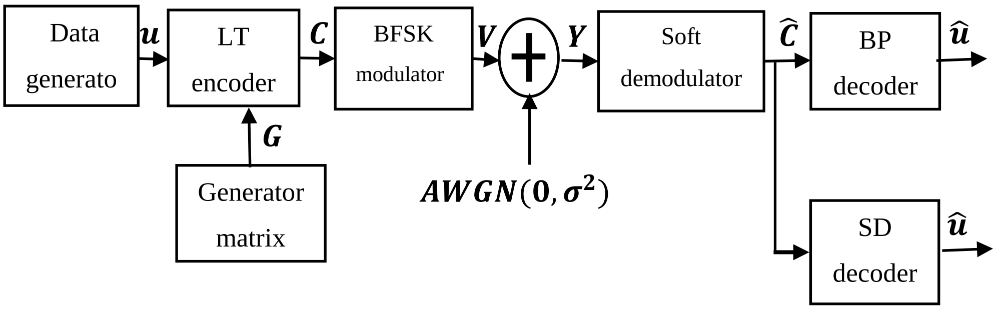


Figure 33 System model for soft-demodulation-hard-decoding approach

5.4.3. Simulation results

In this section, an implementation for the system model shown in Figure 33 has been executed using a MATLAB simulation. A data length of ($k = 32$) has been used and the LT codes have been generated using four degree generators. For the case of (RSD, MBRSD and ODD) the parameters are chosen as ($c = 0.02$ and $\delta = 0.1$) and the DE has been generated using ($R_p = 4$). A rateless LT codes are transmitted over a binary AWGN channel characterized by zero mean and noise power represented by σ^2. In the receiver side the decoder collects a specified number of coded symbols and starts its mission to recover the data symbols. In Figure 34, the BER comparison for the four types LT codes has been illustrated. The first step in the decoding process is dedicated to estimate the coded symbol values by deciding on the MLP using (5.8) and (5.9). It is obvious that the DE with SD approach using an overhead of (OVHD=0.2) outperforms all the other degree generators using an overhead of (OVHD=1.0) for a wide range of SNRs. Figure 35, shows the successful decoding rates achieved by the four types LT codes with the same overhead numbers prementioned above. The transmitter sends (1000) frames over an AWGN channel. DE with SD approach successes in recovering all the data frames earlier by (2dB) than that of the other LT codes with RSD, MBRSD and ODD. It is worthily to mention that this improvement on the BER floor and the successful decoding rate has been done with less complexity since the SD approach can recover the k data symbols from a complete set of k coded symbols. As mentioned above, the first step in the decoding process is calculating the likelihood probabilities given in (5.8) and (5.9), these probabilities can be updated by averaging these probabilities for the identical coded symbols. For the case of the SD approach, after averaging the redundant coded symbols probabilities, the redundant coded symbol is deleted. The remaining coded symbols are rearranged from (C_1 up to C_k) as indicated in (5.5). The data symbols are recovered using (5.6) and (5.7) in a sequential manner. According to this procedure SD approach needs less time and less symbol operations and as it is clear from Figure 34 and Figure 35, it requires less overhead to achieve lower BER and higher successful decoding rates.

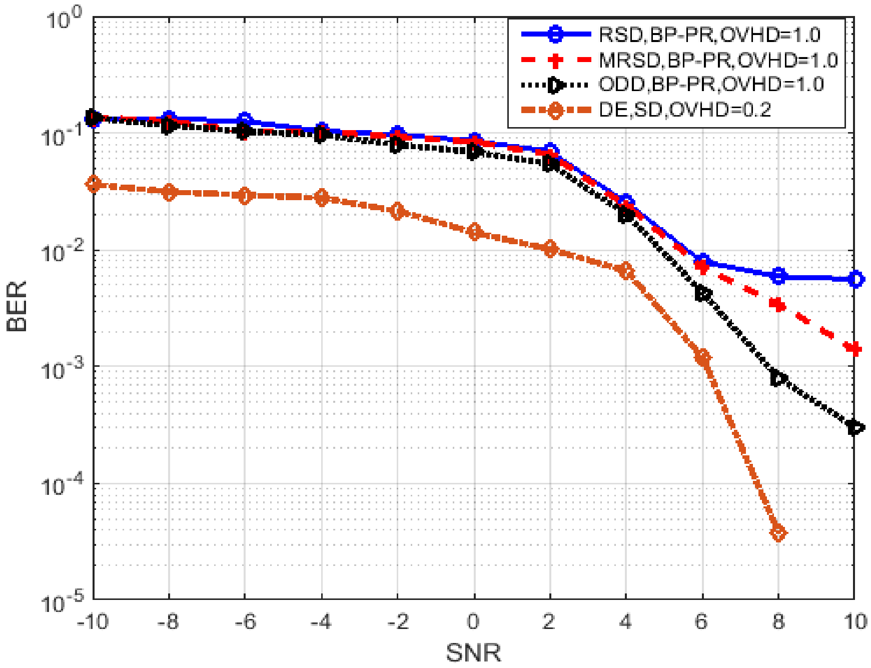


Figure 34 BER comparison for LT code of length $k = 32$ using different degree generation RSD, MBRSD, ODD and DE over AWGN channel

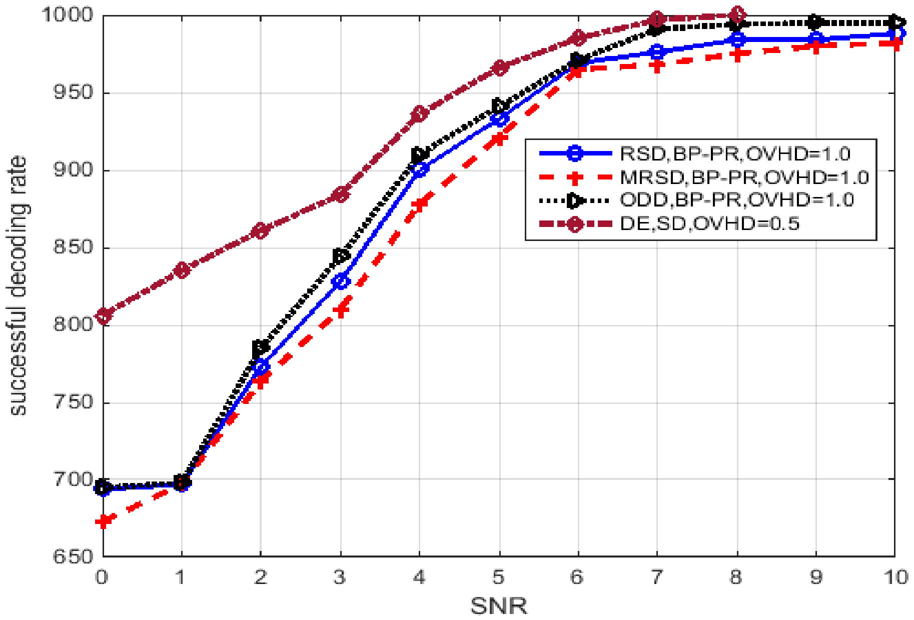


Figure 35 Successful decoding rates for LT code of length $k = 32$ using different degree generation RSD, MBRSD, ODD and DE over AWGN channel.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

Since the first steps for the invention of the rateless codes, a huge number of researches have been presented. One of the most important research was the practical implementation of the rateless codes theory that has been introduced by Luby [18] and his novel design of LT codes over BEC. In this thesis, we try to focus on the performance of such codes for an extremely short data length. Investigating the features of the degree distribution used to generate LT codes, we found that RSD is the optimal distribution not only for bulk data lengths but also for short data lengths. This has been reached by adding our proposed PR approach to the conventional BP decoding approach to form the new BP-PR which can remove one of the major article faced by the decoder which is the absence of degree one coded symbol. The new LT-PR successes in removing the decoder stuck by combining certain patterns which can produce a new degree one coded symbol and resume decoding again. We compare its performance with another powerful decoding approach called BP-GE which used mainly for short data length codes, and we found that our BP-PR has almost the same performance but it could need less decoding complexity. The new LT-PR approve its improvement for the LT-RSD as well as one of the state of art coding called LT-MBRSD. This improvement appears obviously in the records of BER floor and the successful decoding rates.

Motivating from the idea of BP-PR which use certain patterns to overcome the problem of losing degree one coded symbol and the idea of the MBRSD which assign the degree one coded symbol to the high degree data symbol, motivated from these two ideas, we present our DE approach. This deterministic approach is mainly designed for short length data LT codes. DE generates the degree (d) in a repeated manner from 1 up to a predetermined value called repetition period (R_p). Second, the data is truncated to segments of length (R_p). The coded symbol is generated by combining (d) data symbol serially from a random selected segment. Testing this new LT-DE against LT codes using RSD-BP-PR, MBRSD-BP-PR, and ODD-BP-PR in a BEC environment approve its superiority with respect to BER floor, successful decoding rates, overhead and symbol operations. The superiority of such LT-DE is mainly appearing for erasure probability ($\alpha < 0.1$), when the erasure probability increase such encoding faced some performance degradation. However, its encoding simplicity and high performance for rather low erasures could nominate it for certain applications which can fit its proprieties with very low decoding complexity. Another feature of such LT-DE is the sequential relation between the coded symbols that can be used to extract the data symbols. The LT-DE-SD has been tested in an AWGN channel against Lt codes using RSD-BP-PR, MBRSD-BP-PR, and ODD-BP-PR, and the simulation results approve the improvements of such encoding-decoding approach with respect all evaluation factors mentioned above. An interesting decoding feature for the DE-SD approach that it needs only $O(k)$ complexity cost to extract the source k data symbols.

6.2. Future work

Rateless codes are dedicated for implementation in many multicast communication systems. For certain application which need to transmit rather short number of packets, our improvement for the LT-RSD could nominate such codes for these applications. Several ideas may be listed below to be a set of future works:

1. More investigation for the remaining unprocessed coded symbols when no degree one coded symbol neither useful patterns are found to resume decoding, which could further enhance the BP approach.

2. More enhancement for the LT-DE by proper design for the random data selection.
3. Test the new LT-BP-PR as an outer code for the well-known raptor code [19].
4. Test the new LT-DE against systematic LT codes in wireless communication systems.
5. Utilizing all the tested degree generators in an adaptive LT decoder which could select the appropriate decoding approach according to the channel condition and the complexity cost.
6. Utilizing the new LT-DE-SD in an UWB-WBN for medical applications.
7. Testing the new LT-BP-PR in FSO communication system and WSN to improve the BER performance.

REFERENCES

1. **Shannon. C. E.**, **(1948a),** “*A mathematical Theory of Communication*”, Bell Syst. Tech. J., vol. 27, pp. 379-423. July.

2. **Shannon. C. E.**, **(1948b),** “*A mathematical Theory of Communication*”, Bell Syst. Tech. J., vol. 27, pp. 623-656. October.

3. **Shannon. C. E.**, **(1949),** “*Communication in the Presence of Noise*”, Proc. IRE, vol. 37, pp. 10-21, January.

4. **Shannon. C. E.**, **(1959a),** “*Coding Theorems for a Discrete Source with a Fidelity Criterion*”, IRE Nat. Conv. Rec., pt. 4, pp. 142-163, March.

5. **Proakis J. G.**, **(1998),** “*Digital Communications, fourth edition* (2001)”, McGraw-Hill, New York.

6. **Gilbert, E. N. (1952),** “*A Comparison of Signaling Alphabets*”, J. Bell Syst. Tec., vol. 31, pp. 504-522, May.

7. **Elias, P. (1954),** “*Error Free Coding*”, IRE Trans. Convention Record, vol. IT-4, pp. 29-37, September.

8. **Elias, P. (1954),** “*Coding for Noisy Channel*”, IRE Trans. Convention Record, vol. 3, part 4, pp. 37-46.

9. **Cover T. and Thomas J., (1991),** “*Elements of Information Theory*”, John Wiley and Sons.

10. **Blackwell D., Breiman L. and Thomasian A. J., (1959),** “*The capacity of a class of channels*”, Annals of Mathematical Statistics, 30, 1959, 1229–1241.

11. **Reed I.S. and Solomon G., (1960),** “*Polynomial codes over certain finite fields*,” J. of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, pp. 300-304.

12. **Byers J. W., Luby M., and Mitzenmacher M.**, **(2002),** “*A Digital Fountain Approach to Asynchronous Reliable Multicast*,” J. IEEE on selected areas in communications, vol. 20, no. 8, pp. 1528–1540.

13. **Byers J. W., Luby M., and Mitzenmacher M., (2002),** “*A digital fountain* approach to reliable distribution of bulk data”, in Proc. ACM SIGCOMM, Vancouver, BC, Canada, Aug. 1998, pp. 56–67.

14. **Luby M., Mitzenmacher M., Shokrollahi, A., and Spielman D**., **and V. Stemann, (2001),**“ Practical loss-resilient codes”, in Prc. ACM Symp. Theory Comp., El Paso, TX, 1997, pp. 150–159

15. **Luby M., Mitzenmacher M., Shokrollahi, A., and Spielman D**., **(2001),**“ *Efficient erasure correcting codes*”, IEEE finsaclions on Information

    Theory, vol. 47, no. 2 , pp. 669-584.

16. **Mitzenmacher M.**, **(2004),** “*Digital Fountains: A Survey and Look Forward*”, In IEEE information theory workshop, October 24 2004, pp.271-276.

17. **Richardson T. and Urbanke R..**, **(2001),** “*Efficient encoding of low-density*

parity-check codes.", IEEE Transactions on Information Theory, vol.47, no. 2, pp. 638-656.

18. **Luby M., (2002),** "*LT codes*", In Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 271-282.

19. **Shokmllahi A.**, **(2006),** "*Raptor codes*", IEEE transactions on information theory vol.52, no. 6, pp. 2551-2567..

20. **Byen J., Horn G., Luby M., Mitzenmacher M., and Shaver W.**, **(2002),** "*FLID-DL: congestion control for layered multicast*", J. IEEE on Selected Arras in Connumerations, vol. 20, no. 8, pp. 1558-1570.

21. **Byers J. W., Luby M., and Mitzenmacher M., (1999),** " *Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads.*", In Proceedings of INFOCOM, pp. 27j.283..

22. **Rost S., Byers J., and Bestamos A. (2002),** "*The Cyclone Server Architecture: streamlining delivery of popular content*", J. Computer Communications, vol.25, no. 4, pp. 403-412.

23. **Mahanti A., Eager D. L., Vernon M. K. and Sundaram-Stukel D.. (2003),** "Sealable *On-Demand Media Streaming with Packet Loss Recovery*", IEEE/ACM Transactions on Networking, vol.11, no. 2, pp. 195-209.

24. **Floyd S., Jacobson V., Liu C. G., McCanne S., and Zhang L. (1995),** "*A reliable multicast framework for light-weight sessions and application level framing*," in Proc. ACM SIGCOMM, Aug. 1995, pp. 342–356.

25. **Kasera S. K., Kurose J., and Towsley D. (1998),** "*A comparison of server-based and receiver-based local recovery approaches for scalable reliable multicast*, "in Proc. IEEE INFOCOM, Mar. 1998, pp. 988–995.

26. **McCanne S., Jacobson V., and Vetterli M. (1996),** "*Receiver-driven layered multicas*t," in Proc. ACM SIGCOMM, Aug. 1996, pp. 117–130.

27. **Paul S., Sabnani K., Lin J., and Bhattacharyya S. (1997)**, "*Reliable multicast transport protocol (RMTP)*," IEEE J. Select. Areas Commun., Special Issue on Network Support for Multipoint Communication, vol. 15, pp. 407–42.

28. **Yavatkar R., Griffoen J., and Sudan M. (1995),** "*A reliable dissemination protocol for interactive collaborative applications*," in Proc. ACM Multimedia, an Francisco, CA, 1995, pp. 333–344.

29. **MacKay D.J**. **(2004)**, "*Information theory, inference and learning algorithms*," Cambridge university press, pp. 591-596.

30. **Joshi G., Joong B. R., John S., and Da W**. **(2010), "***Fountain codes*," In Global telecommunications conference *(GLOBECOM 2010)*, pp. 7-12.

31. **MacKay D.J**. **(2005),** "*Fountain codes*," *IEE Proceedings-Communications* vol. 152, no. 6, pp. 1062-1068.

32. **Scott J. (1845),** Report on Waves. Report of the fourteenth meeting of the British Association for the Advancement of Science, York, September 1844 (London 1845), pp 311-390, Plates XLVII-LVII).

33. **Wang K., Chen Z. (2013),** "*A Novel Decoding Scheme for LT-Codes in Wireless Broadcasting Systems*," IEEE communications letters, vol. 17, no. 5, pp. 972-975**.**

34. **Gallager R. (1962),** "*Low-density parity-check codes*," IRE Trans. Inf. Theory. 8, no. 1, pp. 21–28.

35. **Gallager R. (1963),** "*Low-density parity-check codes*," MIT Press, Cambridge,
36. **Kschischang F., Frey B., Loeliger H.-A. (2001),** "*Factor graphs and the sum-product algorithm*," IEEE Trans. Inf. Theory, vol. 47, no. 2, pp.498–519, Feb. 2001.

37. **Hussain, I. (2014**), "*Analysis and Design of Rateless Codes*," (Doctoral dissertation, KTH Royal Institute of Technology).

38. **Etesami O. and Shokrollahi A. (2006),** "*Raptor codes on binary memoryless symmetric channels*," IEEE Trans. Inf. Theory, vol. 52, no. 5, pp.2033–2051.

39. **Hyytiä E., Tirronen T., Virtamo J. (2007),** "*Optimal Degree Distribution for LT Codes with Small Message Length*," In INFOCOM 2007. 26th IEEE International Conference on Computer Communications, pp. 2576-2580. IEEE.

40. **Ross S. M. (2000),** "*Introduction to Probability Models*," 7th ed. Academic Press.

41. **W. R. Inc.,** "*Mathematica*," http://www.wolfram.com/.

42. **Hyytiä E., Tirronen T., Virtamo J. (2006),** "*Optimizing the degree distribution of LT codes with an importance sampling approach*," In RESIM 2006, 6th International Workshop on Rare Event Simulation 2006 Oct 8, pp.64-73.

43. **Farrell P. and Moreira J. C. (2006),** "*Essentials of Error–Control Coding*," John Wiley and Sons.

44. **Hosseini S. M., Akbari, M., Falahati, A. (2010),** "*Modification of short length LT block coding technique by removing LT weaknesses with an equivalent matrix*," In Telecommunications (IST), 2010 5th International Symposium on, pp. 356-360.

45. **Chen C. M., Chen Y. P., Shen T. C., Zao J. K. (2010),** "*Optimizing degree distributions in LT codes by using the multi-objective evolutionary algorithm*

based on decomposition,” In IEEE Congress on Evolutionary Computation, pp. 1-8.

46. **GUO, C., & BI, X. (2010),** “*Performance analysis and parameter optimizing rules of lt codes*,”. China Communications J., vol. 7,no. 4, pp. 103-107.

47. **Chen C. M., Chen Y. P., Shen T. C., Zao J. K. (2010),** “*On the optimization of degree distributions in LT code with covariance matrix adaptation evolution strategy*,” In Evolutionary Computation (CEC), 2010 IEEE Congress on, pp. 1-8.

48. **Yilong X., Haibo J. (2011),** “*A new class of rateless codes*,” In Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on, pp. 2390-2393.

49. **Chen G. T., Cao L., Zhao F., Zheng H. F., Pan M. (2012),** “*Analysis of robust soliton distribution for LT code*,” In Signal Processing (ICSP), 2012 IEEE 11th International Conference on (Vol. 2, pp. 1546-1549). IEEE.

50. **Yen K. K., Chen C. L., Chang H. C. (2012),** “*An improved LT encoding scheme with extended chain lengths*,” In Information Theory and its Applications (ISITA), 2012 International Symposium on (pp. 11-15). IEEE.

51. **Yen K. K., Liao Y. C., Chen C. L., Chang H. C. (2012),** “*Non-repetitive encoding with increased degree-1 encoding symbols for LT codes*,” In Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on (pp. 655-658). IEEE.

52. **Tsai P. C., Chen C. M., Chen Y. P. (2012),** “*Sparse degrees analysis for LT codes optimization*,”. In Evolutionary Computation (CEC), 2012 IEEE Congress on (pp. 1-6). IEEE.

53. **Du Q., Song X., Liu Y., Zhao L. (2013)**. “*Implementation of LT Code with a Novel Degree Distribution*,” Open Journal of Applied Sciences, vol.2, no. 4, pp. 203.

54. **Yen K. K., Liao Y. C., Chen C. L., Chang H. C. (2013),** “*Adjusted robust Soliton distribution (ARSD) with reshaped ripple size for LT codes*,” In Wireless Communications & Signal Processing (WCSP), 2013 International Conference on (pp. 1-6). IEEE.

55. **Yen K. K., Liao Y. C., Chen C. L., Zao J. K., Chang, H. C. (2013),** "*Integrating non-repetitive LT encoders with modified distribution to achieve unequal erasure protection*," IEEE Transactions on Multimedia, vol. 15, no. 8, pp. 2162-2175.

56. **Zhang L., Liao J., Wang J., Qi Q., Xu T., Tian S., Liao M. (2013),** "*Diversified SLT codes based on feedback for communication over wireless networks*," In Global Information Infrastructure Symposium, 2013 (pp. 1-6). IEEE.

57. **Yen K. K., Liao Y. C., Chen C. L., Chang H. C. (2013),** "*Modified robust soliton distribution (MRSD) with improved ripple size for LT codes*," IEEE Communications Letters, vol. 17, no. 5, pp. 976-979.

58. **Hussain I., Land I., Chan T. H., Xiao M., Rasmussen, L. K. (2014),** "*A new design framework for LT codes over noisy channels*," In Information Theory (ISIT), 2014 IEEE International Symposium on (pp. 2162-2166). IEEE.

59. **Hussain I., Xiao, M., Rasmussen L. K. (2014),** "*Buffer-based distributed LT codes*," IEEE Transactions on Communications, vol. 62, no. 11, pp. 3725-3739.

60. **Alam S. S., Sultana S., Choi G., Iqbal S. (2015),** "*Performance evaluation of novel degree distribution on power law of LT codes for internet technology*,". In Consumer Electronics (ISCE), 2015 IEEE International Symposium on (pp. 1-2). IEEE.

61. **Cheong S. T., Fan P. (2016),** "*Novel degree function over finite field for LT codes*," In Electronics, Information, and Communications (ICEIC), 2016 International Conference on (pp. 1-5). IEEE.

62. **Li X., Jiang T. (2013),** "*Fountain codes over GF (q),*" Wireless Communications and Mobile Computing, vol. 13, no.16, pp. 1423-1434.

63. **Wiedemann D. (1986),** "*Solving sparse linear equations over finite fields*," IEEE transactions on information theory, vol. 32, no.1, pp. 54-62.

64. **Jun B., Yang P., No J. S., Park, H. (2015),** "*New rateless codes for receiver with limited memory*," In Information and Communication Technology Convergence (ICTC), 2015 International Conference on (pp. 482-484). IEEE.

65. **Nozaki T. (2014),** "*Fountain codes based on zigzag decodable coding*," In Information Theory and its Applications (ISITA), 2014 International Symposium on (pp. 274-278). IEEE.

66. **Yao W., Yi B., Huang T., Li W. (2016),** "*Poisson Robust Soliton Distribution for LT Codes*," IEEE Communications Letters, vol. 20, no.8, pp. 1499-1502.

67. **Sheng Z., Xie S. Q., Pan C. Y. (2008),** "*Probability theory and mathematical statistics*," (Version No.4). Zhejiang, China: Zhejiang Univ. Press, 2008.

68. **Li X. L., Shao Z. J., Qian J. X. (2002),** "*An optimizing method based on autonomous animats: fish-swarm algorithm*," System Engineering Theory and Practice, vol. 22, no. 11, pp. 32-38.

69. **Yen K. K., Liao Y. C., Chang, H. C. (2015),** "*Design of LT code degree distribution with profiled output ripple size*," In Signal Processing Systems (SiPS), 2015 IEEE Workshop on (pp. 1-6). IEEE.

70. **Palanki R., Yedidia J. S. (2004),** "*Rateless codes on noisy channels*,". In Information theory, 2004. ISIT 2004. Proceedings. International symposium on (p. 37). IEEE.

71. **Maneva E., Shokrollahi, A. (2006),** " *New model for rigorous analysis of LT-codes*," In Information Theory, 2006 IEEE International Symposium on (pp. 2677-2679). IEEE.

72. **Darling R. W., Norris J. R. (2005),** "Structure of large random hypergraphs," The Annals of Applied Probability, vol. 15, no. 1A, pp. 125-152.

73. **Karp R., Luby M., Shokrollahi A. (2004),** "*Finite length analysis of LT codes*," In Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on (p. 39). IEEE.

74. **Nguyen T. D., Yang L. L., Hanzo L. (2007, October),** "*Systematic Luby transform codes and their soft decoding*," In Signal Processing Systems, 2007 IEEE Workshop on (pp. 67-72). IEEE.

75. **Bioglio V., Grangetto M., Gaeta R., Sereno M. (2009),** "On the fly Gaussian elimination for LT codes," IEEE Communications Letters, vol. 13, no. 12, pp. 953-955.

76. **Kim S., Ko K., Chung S. Y. (2008),** "*Incremental Gaussian elimination decoding of raptor codes over BEC*," IEEE communications letters, vol. 12, no. 4.

77. **Hussien H., Shehata K. A. El Ramly, S., Tawfik N. M. (2012),** "*Design of a Merged Algorithm for Luby Transform Decoder*," International Journal of Computer and Communication Engineering, vol.1, no. 3, pp. 246-249.

78. **Sorensen J. H., Popovski P., Ostergaard J. (2012),** "*Design and analysis of LT codes with decreasing ripple size*," IEEE Transactions on Communications, vol. 60, no. 11, pp. 3191-3197.

79. **Wiedemann, D. (1986),** "*Solving sparse linear equations over finite fields*," IEEE transactions on information theory, vol. 3, no. 1, pp. 54-62.

80. **Lu H., Lu F., Cai, J., Foh C. H. (2013),** "*LT-W: Improving LT decoding with Wiedemann solver*," IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 7887-7897.

81. **Wang X., Willig A., Woodward, G. (2013, April);** "*Improving fountain codes for short message lengths by adding memory*," In Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on (pp. 189-194). IEEE.

82. **Jones A. L., Chatzigeorgiou I. Tassi, A. (2014, July),** "*Performance assessment of fountain-coded schemes for progressive packet recovery*," In Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2014 9th International Symposium on (pp. 679-684). IEEE.

83. **Hayajneh K. F., Yousefi S., Valipour M. (2015),** "*Improved finite-length Luby-transform codes in the binary erasure channel*," IET Communications, vol. 9, no. 8, pp. 1122-1130.

84. **Kharel A., Cao, L. (2015, April),** "*Decoding of short LT codes over BIAWGN channels with Gauss-Jordan elimination-assisted belief propagation method*," In Wireless Telecommunications Symposium (WTS), 2015 (pp. 1-6). IEEE.

85. **Kharel A., Cao, L. (2016, July),** "*On the performance of LT codes over BIAWGN channel*," In Computer, Information and Telecommunication Systems (CITS), 2016 International Conference on (pp. 1-5). IEEE.

86. **Hayajneh K. F., Yousefi, S. (2015, July),** "*Overlapped fountain coding for delay-constrained priority-based broadcast applications*," In Information Theory (CWIT), 2015 IEEE 14th Canadian Workshop on (pp. 79-82). IEEE.

87. **Albanese A., Blomer J., Edmonds J., Luby M., Sudan, M., (1996),** "*Priority Encoding Transmission*," IEEE Transactions on Information Theory, vol.42, no. 6, pp. 1737-1744.

88. **Rahnavard N., Vellambi B.N., Fekri F.,(2007),** "*Rateless Codes With Unequal Error Protection Property*," IEEE Transactions on Information Theory ,vol. 53, no. 4, pp. 1521-1532.

89. **Sandberg S., von Deetzen N. (2010),** "*Design of Bandwidth-Efficient Unequal Error Protection LDPC Codes*," IEEE Transactions on Communications. 58, no. 3, pp. 802-811.

90. **Abbas R., Shirvanimoghaddam M., Li, Y., Vucetic B. (2016, May),** "*Performance analysis and optimization of LT codes with unequal recovery time and intermediate feedback*," In Communications (ICC), 2016 IEEE International Conference on (pp. 1-6). IEEE.

91. **Aref V. (2015, May),** "*Rateless codes from spatially coupled regular-LT codes*," In Communication and Information Theory (IWCIT), 2015 Iran Workshop on (pp. 1-6). IEEE.

92. **Aref V., Urbanke, R. L. (2011, October),** "*Universal rateless codes from coupled LT codes*," In Information Theory Workshop (ITW), 2011 IEEE (pp. 277-281). IEEE.

93. **Cheong H., Eun J., Kim H., Kim K. J. (2016),** "*Belief propagation decoding assisted on-the-fly Gaussian elimination for short LT codes*," Cluster Computing Journal, vol. 19, no. 1, pp. 309-314.

94. **Ahmad S., Hamzaoui R., Al-Akaidi, M. (2008, October),** "*Practical channel-adaptive video streaming with fountain codes*," In Pervasive Computing and Applications, 2008. ICPCA 2008. Third International Conference on (Vol. 1, pp. 351-355). IEEE.

95. **Ahmad S., Hamzaoui R., Al-Akaidi, M. (2010)),** "*Adaptive unicast video streaming with rateless codes and feedback*," IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, pp. 275-285.

96. **Ahmad S., Hamzaoui R., Al-Akaidi, M. (November 2007),** "*Robust live unicast video streaming with rateless codes*," in Proc. 16th Int. Packet Video Workshop, Lausanne.

97. **Wagner J. P., Chakareski J., Frossard P. (2006, July),** "*Streaming of scalable video from multiple servers using rateless codes*," In Multimedia and Expo, 2006 IEEE International Conference on (pp. 1501-1504). IEEE.

98. **Xu Q., Stankovic V., Xiong Z. (2007),** "*Wyner–Ziv video compression and fountain codes for receiver-driven layered multicast*," IEEE transactions on circuits and systems for video technology, vol. 17, no. 7, pp. 901-906.

99. **Schierl T., Johansen S., Perkis A., Wiegand T. (2008),** "*Rateless scalable video coding for overlay multisource streaming in MANETs*," Journal of Visual Communication and Image Representation, vol. 19, no. 8, pp. 500-507.

100. **Li Z., Li Y., Chiang M., Calderbank A. R., Chen Y. C. (2009, January),** "*Optimal transmission scheduling for scalable wireless video broadcast with rateless erasure correction code*," In Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE (pp. 1-5). IEEE.

101. **Sejdinovic D., Vukobratovic D., Doufexi A., Senk V., Piechocki R. J. (2009),** Expanding window fountain codes for unequal error protection. IEEE Transactions on Communications, vol. 57, no. 9.

102. **Vukobratovic D., Stankovic V., Sejdinovic D., Stankovic, L., Xiong, Z. (2009),** "*Scalable video multicast using expanding window fountain codes*," IEEE Transactions on Multimedia, vol. 11, no. 6, pp. 1094-1104.

103. **Al-Jobouri L., Fleury M., Ghanbari M. (2012, January),** "*Multicast and unicast video streaming with rateless channel-coding over wireless broadband*," In Consumer Communications and Networking Conference (CCNC), 2012 IEEE (pp. 737-741). IEEE.

104. **Li Y., Soljanin, E. (2009, September)**, "*Rateless codes for single-server streaming to diverse users*," In Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on (pp. 1419-1426). IEEE.

105. **Du Toit, J.,Wolhuter, R. (2011),** "*A Practical Implementation of Fountain Codes over WiMAX Networks with an Optimized Probabilistic Degree Distribution*," In ICSNC Conference, 2011 (pp. 32-37).

106. **Hamada Y., Takizawa K., Ikegami T. (2012, January),** "*Highly reliable wireless body area network using error correcting codes*," In Radio and Wireless Symposium (RWS), 2012 IEEE (pp. 231-234). IEEE.

107. **Xu L. (2003),** "Resource-efficient delivery of on-demand streaming data using UEP codes," IEEE Transactions on Communications, vol. 51, no. 1, pp. 63-71

108. **Arslan S. S., Cosman P. C., Milstein, L. B. (2012),** " *Generalized unequal error protection LT codes for progressive data transmission*," IEEE Transactions on Image Processing, vol. 21, no. 8, pp. 3586-3597.

109. **Prakash G., Kulkarni M., Sripati U. (2013, August),** " *BER performance and Energy Efficiency of Luby Transform codes with varying BPSK thresholds over the Free Space Optical channel*," In Advances in Computing, Communications and Informatics (ICACCI), 2013 International

Conference on (pp. 1473-1478). IEEE.

110. **Yue J., Lin Z., Vucetic B., Xiao P. (2014, June)**, " *The design of degree distribution for distributed fountain codes in wireless sensor networks*,"In Communications (ICC), 2014 IEEE International Conference on (pp. 5796-5801). IEEE.

111. **Tang H., Zhao J., He Y., Tao L. (2014),** "*Cooperative rateless codes for NGB-W*,"In Proc. of 10th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2014).

112. **Sofia J. J., Jayakumari, J. (2015, April),** "*Application of LT code technology in deep space communication*," In Communication Technologies (GCCT), 2015 Global Conference on (pp. 458-463). IEEE.

113. **Kong B., Zhang G., Dong F., Cheng L., Zhou X. (2015, October),** "*Efficient distributed storage based on LT codes in wireless sensor networks*, "In Wireless Communications & Signal Processing (WCSP), 2015 International Conference on (pp. 1-5). IEEE.

114.**Ye X., Li J., Chen W. T., Tang F. (2015, May),** "*LT codes based distributed coding for efficient distributed storage in wireless sensor networks*," In IFIP Networking Conference (IFIP Networking), 2015 (pp. 1-9). IEEE.

115. **Al-Awami L., Hassanein H. H. (2015, December),** " *Energy Efficient Distributed Storage Systems with LT-Codes in Resource-Limited Wireless Systems*," In Global Communications Conference (GLOBECOM), 2015 IEEE (pp. 1-6). IEEE.

116. **Jin L., Li Y., Zhao C., Wei, Z., Li, B., Shi J. (2016),** "*Cascading polar coding and LT coding for radar and sonar networks*," EURASIP Journal on Wireless Communications and Networking, 2016(1), 254.

117. **Arikan E. (2009),** "*Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels*," IEEE Transactions on Information Theory, vol. 55, no. 7, pp. 3051-3073.

118. **Yang G., Zhuang M. (2016, September),** "*Achieving the secrecy capacity on strong security using LT code with polar code pre-coding*," In Anti-counterfeiting, Security, and Identification (ASID), 2016 10th IEEE International Conference on (pp. 53-57). IEEE.

119. **Janet J., Balakrishnan S., Somasekhara K. (2016, February),** " *Fountain code based cloud storage mechanism for optimal file retrieval delay*," In Information Communication and Embedded Systems (ICICES), 2016 International Conference on (pp. 1-4). IEEE.

120. **Zhiliang Z., Sha L., Jiawei Z., Yuli Z., Hai Y. (2012, October),** "*Performance analysis of LT codes with different degree distribution*," In Chaos-Fractals Theories and Applications (IWCFTA), 2012 Fifth International Workshop on (pp. 142-146). IEEE.

121. **Zao J. K., Hornansky M., Diao P. L. (2012, June),** "*Design of optimal short-length LT codes using evolution strategies*," In Evolutionary Computation (CEC), 2012 IEEE Congress on (pp. 1-9). IEEE.

122. **Wang X., Willig A., Woodward G. (2013, April),** "*Improving fountain codes for short message lengths by adding memory*," In Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on (pp. 189-194). IEEE.

123. **Richardson T. J., Urbanke R. L. (2001),** "*The capacity of low-density parity-check codes under message-passing decoding*," IEEE Transactions on information theory, vol. 47, no. 2, pp. 599-618.

124. **Chen S., Zhang Z., Zhu L., Wu K., Chen, X. (2013),** "*Accumulate rateless codes and their performances over additive white Gaussian noise channel*," IET Communications, vol. 7, no. 4, pp. 372-381.

125. **Shokrollahi A. (2003),** "*LDPC codes: An introduction*," Digital Fountain, Inc., Tech. Rep, 2.

126. **Ma X., Li C., Bai B. (2010, September),** “*Maximum likelihood decoding analysis of LT codes over AWGN channels*, “In Turbo Codes and Iterative Information Processing (ISTC), 2010 6th International Symposium on (pp. 285-288). IEEE.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Abdulkhaleq, Nadhir Ibrahim

Date and Place of Birth: 07 June 1966, Baghdad-Iraq

Marital Status: Married

Phone: +90 5380650219

Email: nadir66raouf@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Military Engineering College	1995
B.Sc.	Military Technical College	1990
High School	Al Karkh Secondary School	1984

WORK EXPERIENCE

Year	Place	Enrollment
2004- Present	Technology Institute - Middle Technical University	Lecturer
1991-2004	Military Institutes	Lecturer

FOREIN LANGUAGES

Advanced English, Beginner Turkish

PUBLICATIONS

1. Abdulkhaleq, N. I., & Gazi, O. (2016). Decoding of LT-like codes in the absence of degree-one code symbols. ETRI Journal, 38(5), 896-902.
2. Abdulkhaleq, N. I., & Gazi, O. (2017, April). A sequential coding approach for short length LT codes over AWGN channel. In Electrical and Electronic Engineering (ICEEE), 2017 4th International Conference on (pp. 267-271). IEEE.

HOBBIES

Soccer, Travel, Tennis.