August 2012

Master's Degree Thesis

An Efficient Implementation of LT Codes Based on Its Optimized Degree Distribution Function

Graduate School of Chosun University

Department of Information and Communication Engineering

Muhammad Asim

An Efficient Implementation of LT Codes based on its Optimized Degree Distribution Function

최적화된 정도분포함수를 근거로한 LT 부호의 효율적인구현

August 24, 2012

Graduate School of Chosun University

Department of Information and Communication Engineering

Muhammad Asim

An Efficient Implementation of LT Codes based on its Optimized Degree Distribution Function

Thesis Advisor: GoangSeog Choi, PhD

This Thesis is submitted to Graduate School of Chosun University in partial fulfillment to the requirements for a Master's degree in Engineering

April 2012

Graduate School of Chosun University

Department of Information and Communication Engineering

Muhammad Asim

아씸무하메드 석사학위 논문을 인준함

위원장 조선대학교 교수 신영숙 印

위 원 조선대학교 교수 김영식 印

위 원 조선대학교 교수 최광석 印

2012 년 5 월

조선대학교 대학교

Contents

List of Figures

List of Tables

ABSTRACT

An Efficient Implementation of LT Codes based on its Optimized Degree Distribution Function

Muhammad Asim
Advisor: Prof. GoangSeog Choi, Ph.D.
Department of Information and Communication Engineering
Graduate School of Chosun University

본 논문은 네트워크를 통한 대량의 데이터 전송 시 발생할 수 있는 문제점을 채널 코딩의 관점에서 검토한다. 채널 코딩은 통신 채널 상 데이터 전송 시 발생할 수 있는 통신에러를 줄이기 때문에 통신 채널에서 하나의 패러다임으로 제기 되고 있다. 최근에는 새로운 채널 코딩 방식이 소실 채널 상에서도 사용되어 지고 있다.

본 논문에서는 LDPC 코드와 파운틴 코드를 포함하고, 그 중 보다 더 정확한 파운틴 코드에 초점을 맞추고 있다. 이전 연구에는 정확한 파운틴 코드의 개념을 내포하는 코드로 Reed Solomon 과 LDPC 코드 등의 방법이 있었다. 몇 년 전에는 새로운 코딩 방식으로 LT 코드가 제안되었다. LT 코드는 정확한 파운틴 코드이고, 이득이 높고 효율이 좋은 엔코더와 디코더 알고리즘이다. 또한 LT 코드는 첫 번째의 Rateless 코딩 방식이며, 그 중요성은 통신 표준들의 도입시 중요하게 여겨진다.

이에 본 논문을 통해 정도 분호 함수로 알려진 이러한 코드를 이용하여, 성능 지향적인 부분 중에 가장 중요한 한 부분을 최적화하였다. 우리는 정도 분포 함수의 최적화를 위한 새로운 분석을 제시하고, 이전에 정의된 학위 분포함수 또한 우리의 분석을 따르는 것을 보여준다.

분석을 바탕으로, 스케일이 없는 LT 코드와 랜덤 LT 코드로 총 두가지의 카테고리로 LT 코드를 분류한다.

두 카테고리 코드는 모두 Power Law 로 연결된다. 그리고 본 논문에서 스케일이 없는 LT 코드를 위한 두 가지 새로운 정도 분포를 제안했고, 그것은 이러한 코드의 성능을 대폭 향상시킬 수 있다. 그리고 우리는 또한 패이딩 채널을 통한 LT 코드의 성능평가를 위한 작업을 제시하였다. 또한 CFC(Concatenated Fountain Code)로 알려진 LT 코드의 향상된 버전을 제안하고 평가한다. 또한 기존의 LT 코드와 CFC 의 성능을 비교한다.

마지막으로, LT 코드를 위한 하드웨어 구조를 제안하는데, 그것은 정도 분포 함수를 사용하지 않고, 직접적으로 하드웨어에 매핑하는 노드와 에지 사이의 연결 정보를 포함하고 있다. 이러한 접근 방식은 LT 코드에 대해 서로 다른 최적의 정도 분포 함수 활용에 도움이 된다. 그리고 하드웨어를 통한 최적의 정도분포 함수를 구현하고 그 함수에 대한 영향을 확인하였다.

Chapter 1
Introduction

In this thesis we consider and analyze the problems related to one of the famous channel coding scheme known as fountain codes. These codes are extensively useful in different areas of data distribution and communication networks. Initially, we have addressed a core issue, known as Degree Distribution function that enhances the performance of these codes. Also, the performance of these codes are evaluated for fading channels and an enhance version of these codes are proposed that significantly improve the performance of these codes. The objective of this chapter is to introduce the problems addressed in this thesis, motivation for the new approach and the main contribution and organization of the thesis.

1.1 Motivation

The development of applications for internet, which can securely transfer large amount of data from one point to the others or some heterogeneous structures, which can meet the demand of excessively increasing bandwidth requirements are one of the addressable challenges in the communication networks. Despite the rapid advancement of the internet technologies allowing higher capacities and speed, the excessive increase in the size of data utilization and an increase in the number of subscriber turn the design mechanism in to an active research area aiming the reliable and robust distribution of digital data to a high number of heterogeneous and autonomous subscribers.

A common approach in data distribution network (e.g. Internet), the source information is divided in to a finite number of data chunks known as packets. These packets have the additional information about source and destination address despite the data itself. Usually, the intermediate router uses the additional source and destination information for routing the packet to a correct desired destination. In particular, if packets of data are transmitted in noisy environment, then there is probability that some of the packets will be lost or received with errors. Thus, causing reliability and robustness issues in the network by requesting the same packets again from the source. Also, this approach adds in an increase in the congestion of the network. Traditionally, reliable communication on the Internet is provided by the Transmission Control Protocol (TCP). This protocol keep tracks of the sent packets with in a variable size time slot and waits for an acknowledgement (ACK) from the receiver for reception of each transmitted packet and retransmit those packets for which an ACK are lost during the transmission process. This method is reliable when there is communication between a sender and receiver. But this scheme is not applicable for broadcast or multicast scenario, when there is a single transmitter and multi-receivers and if one receiver lost the desired packets. Then, missing an ACK from one receiver will imply the resending of the same packet to all receivers and as a result a waste of bandwidth occurs. These ACK based protocols offers poor performance for longer distance between the source and destination, due to idle times waiting for ACKs. Also, the synchronization between the transmitter and receiver play a significant role in ACK based protocols and incurs additional resources for its management. Hence, more efficient techniques are required that can transfer the data efficiently and solving the problems of ACK based algorithms in the new emerging network scenario.

In 1948, Shannon in his work [1] proposed a new paradigm for digital communication and set the basis of new fields known as “Information, Coding and Information Theory”. Shannon proposed two main things in the communication model, which includes: Source Coding and Channel Coding. Source coding removes redundancy from the transmitted information message to compress the source message to increase the bandwidth efficiency. The channel coding adds redundancy to eliminate the errors caused by the noise introduced by the channel, thus protecting the information against errors. In a mathematical way the information source is modeled as a stochastic process and the channel as a probabilistic mapping. Shannon proved that reliable communication is possible, when the rate does not exceed a channel parameter known as the Shannon capacity of the channel. At rates exceeding the channel capacity, reliable communication is not possible. However, he did not provide any algorithm or method explaining how to achieve this optimum rate. From that moment, efforts are being made to develop and design codes that can offer reliable communication at rates near the Shannon capacity at a low complexity.

Channel coding provides an alternative way to schemes based on retransmission and address the problems related to the distribution of large amount of data over networks. In this coding scheme, usually the redundant information is added in the original information followed by an algorithm or scheme before sending the information over the channel. This redundant information helps to detect errors caused by the noisy channel during the transmission. A systematic algorithm is used at the recipient side to use this redundant information and detect the original information from the information received from the channel. This mechanism ends the need of a feedback channel and uses the available bandwidth efficiently. The redundancy added in the original information at the transmission side incurs a price in bandwidth consumption and time because of the redundant information and coding and decoding operations that need to be done at source and destination respectively. Consequently these codes must be designed carefully to fulfill the application requirements. There are three different techniques that can be used for error handling, which includes error detection, error correction and erasure correction. Usually, in networks the received data are either considered as lost or received without errors, hence erasure correcting codes are used in such situations.

Currently much researched alternative for the traditional transmission techniques are the error correction schemes also known as forward error correction (FEC) schemes, and the transfer protocols are supporting these coding schemes. Traditionally erasure codes like Reed Solomon (RS) Codes [2] are employed at low levels, by correcting errors in physical media and the error correction is usually carried out directly in the hardware. These codes have a cubic decoding complexity which is unacceptable for some applications and thus are seldom used these days in real time applications. Low Density Parity Check (LDPC) Codes also sometimes known as Gallager Codes proposed by R. Gallager in [4] are also another type of erasure correcting codes having fixed code rates and are usually optimal for channels with known statistics. Practically, however the channels statistics are usually unknown to the transmitter and thus these codes may fail to come up to the theoretical expectations.

Digital Fountain codes are rateless erasure correcting codes that were first introduced by Byers et al. in [8, 9] and were mainly used for the robust and scalable transmission of data over a heterogeneous environment. These codes were introduced to address the problem of fixed rate codes. These erasures correcting codes provide protection from the affect of lost that is caused during transmission in the environment, and they ensure the recovery of lost data without requesting retransmission from the sender. These codes provide reliability in various network applications such as broadcasting, parallel downloading, video streaming etc.

Digital fountain codes can generate a limitless amount of encoded packets from the information packets. Initially, information source (Transmitted data) is divided into a defined number of pieces and different subsets of these pieces are then chosen randomly to form encoding symbols. The original information is then recovered when a sufficient amount of encoded packets is received. With good fountain codes, the total number of packets needed for decoding is close to the size of the original input packets, although some overhead is necessary because of the nature of these codes. The important characteristic of these codes are that they are nonsystematic codes, i.e., no specific order is required for decoding. As soon as a certain amount of encoded symbols is received, the original information is recovered with maximum probability. It should be noted that LDPC and RS codes possess the property of digital fountain to some extent but they have fixed code rates. But the most successful examples of rateless codes are Luby Transform (LT) proposed by Michael Luby in [10] and Raptor codes [11] by Shokrollahi and have the benefits of efficiency and low computational complexity in the encoding and decoding process as compared to the previous schemes.

1.2 Challenges in Digital Fountain

Ideally an efficient digital fountain code can be characterized by having a property called rateless, i.e., it can provide an infinite number of encoded symbols on the fly continually. Also it has an efficient encoding and decoding processes, which increases linearly with an increase in the size of the source packets. Last but not the least; it should have the ability to recover the original information, once a fixed number of encoded symbols are received. Some real implementation of fountain concept in the hardware environment can only approximate its characteristics.

An LT code is the first practical realization of the Digital Fountain concept and is proposed by Michael Luby in his famous paper [10]. In this coding scheme, the encoding and decoding depends on the probability distribution. The probability distribution function helps the encoder to select randomly the input symbols to generate the encoded symbol. The decoder also uses the same distribution functions to select the received symbols from the channel and recover the original information, although some overhead is necessary to recover the complete information symbols. The probability distribution can be termed degree distribution and is a crucial part of the LT codes design. Usually, Sum Product Algorithm (SPA) also known as Belief Propagation (BP) Algorithm [13] is used to recover the original information. Raptors

Codes [11] are concatenated codes in which LT codes are concatenated with a pre-codes usually LDPC codes. They offer linear decoding complexity at the decoder. Fountain codes are asymptotically optimal codes providing good performance over wireless channels with an increase in the size of the input symbols.

The sizes of the input information symbol have greater impact on the performance of the channel coding. For example in multimedia communication (audio or video), the latency should be kept low and the encoding and decoding time of fountain codes for longer message size are high. The small symbol size on the other hand increases the performance of the LT codes, as it takes less time in the encoding and decoding process and helps in an increase in the throughput. The importance of the fountain codes are improving steadily for data distribution networks and it's used in several other technologies. The standardization of Multimedia Broadcast and Multicast Service (MBMS) [15] and Digital Video Broadcasting (DVB) [16] turned significant interest in the research community in the fountain codes. 3GPPP standard support a message of length among 4 to 8192 and the number of frames in the Group Of Pictures (GOP) uses 10 to 100 frames in video streaming application. There are efforts to improve the state of the art Fountain Codes with the smaller and larger number of input symbols (Source Information). There are two main optimization ways, which are either by improving the degree distribution or by improving the decoding processes of the LT codes.

1.3 Outline

In this thesis, an optimization framework for improving the degree distribution function of the LT codes has been carried out by considering the LT process. The objective of the optimized degree distribution function is to enhance the efficiency of the LT codes by reducing the overhead, so that the decoding process can recovers the original information earlier with the same BP decoding algorithm. This optimization will help in reducing the hardware complexity and also aids in the latency at the encoding and decoding process. Also, a new system model using fountain code is also presented for AWGN and fading channels, which evaluates and compare the performance of the modified fountain codes with traditional fountain code. Finally, the impact of optimal degree distribution function in the hardware has been evaluated. The overview of this thesis is presented below:

Chapter 2: Erasure Correcting Codes

In this chapter, we have introduced some basics about the erasure correcting codes and then traditional erasure correcting schemes are presented, which includes: Reed Solomon, LDPC (Low Density Parity Check) codes and finally Digital Fountain codes are briefly explained. Note that, in Digital Fountain section of this chapter, we have discussed only Tornado Codes which are part of the digital fountain codes family. The main focus of this thesis is on efficient implementation of LT codes and the degree distribution. Thus, next chapter is devoted to the LT codes and some basics about the Raptor codes.

Chapter 3: Luby Transform Codes

This chapter contains detail information about LT codes which are also a part of the Digital Fountain family. Initially, we have presented an overview of the LT encoding and decoding. Also LT decoding using Belief Propagation (BP) algorithm is presented. The crucial part in the design of the LT codes is also discussed which is known as degree distribution functions. The symbol release probability and a framework to optimize the degree distribution functions are also presented. Finally, some basic study about the Raptor Code is also carried out.

Chapter 4: Optimized Degree Distribution

This chapter contains the main contribution of this thesis. In the first section of this chapter, a review of the previously proposed schemes for optimized degree distribution is carried out. In the next section, an analysis for LT process and some new assumptions are considered as a platform for the proposed degree distribution. Two new degree distribution functions are proposed on the basis of our analysis and their evaluation over an ideal channel is considered. A comparison for overhead and the input symbols and the performance of the encoding and decoding performance for bit error rate (BER) versus overhead is considered.

Chapter 5: Concatenated Fountain Codes

In this chapter, we proposed a new system for wireless broadcast using concatenated fountain codes. We evaluate the system model for AWGN and slow fading channel using Rayleigh flat fading and frequency selective fading channels. The Maximum Achievable Rate versus Signal to Noise Ratio (SNR) is used as a performance metric to evaluate the performance of the proposed system model.

Chapter 6: Hardware Implementation of LT Codec

In this chapter, we propose a new architecture for LT Codec that uses a novel approach in which the connection information between the nodes and edges are directly mapped on the hardware instead using the degree distribution function in hardware. Note that, the connection information used between the nodes and edges are derived from the degree distribution function and eliminate the need of any random number generator in the hardware. A comparison has been made for conventional and optimal degree distribution and their affect in the hardware has been considered.

Chapter 7: Conclusions and Future Research Work

This chapter concludes the main ideas presented in this thesis and also some future research issues that can be helpful are also presented.

Chapter 2
Erasure Correcting Codes

The approach to channel coding taken by modern digital communication system started in the late 1940's with the ground breaking work of Shannon [1], Hamming and Golay. Shannon introduced the theoretical basis for coding which has come to be known as information theory. He showed mathematically that by using the entropy of an information source and the capacity of a communications channel, it is possible to achieve reliable communications over a noisy channel provided that the source's entropy is lower than the channel's capacity. In this chapter, an introduction about the erasure channel is presented. Also, some commonly used erasure correction schemes are discussed in the next sections, which includes Reed Solomon (RS), Low Density Parity Check (LDPC) and Digital Fountain Codes.

2.1 Erasure Correcting Codes

The Erasure channel also known as non-trivial channel model and can model data networks, where packets either arrive correctly or are lost due to any kind of problem e.g. buffer overflow or excessive delays. The information may be lost in the erasure channel but is never corrupted. Consider a simple case of single bit transmission, in which the bits will be either received correctly or lost. The decoding problem is to find the values of the bits given the locations of the erasures and the non-erased part of the code word. The Binary Erasure Channel (BEC) is depicted below in Figure 2.1. The time is indexed by t and the transmitter and receiver are synchronized with the time.
The channel input at time t is given by $X_t \in \{0\ ,\ 1\}$ and the corresponding output Y_t have the values $\{0\ ,1,\ ?\}$, where $?$ shows an erasure which occurs for each independent t. Each transmitted bit is either erased with a probabilityε, or received correctly: Y_t

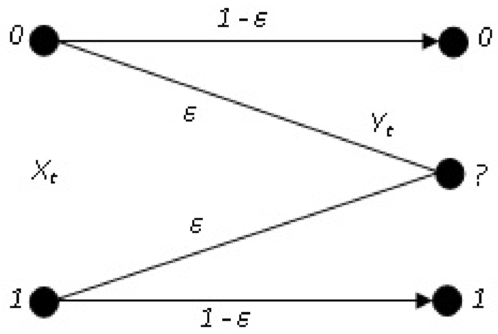

Figure 2.1 Binary Erasure Channel

$\in\{X_t, ?\}$ and $P\{Y_t == ?\} = \varepsilon$.

The capacity of the BEC (ε) is $C_{BEC}(\varepsilon) = 1 - \varepsilon$ bits per channel use. It is easy to see that $C_{BEC}(\varepsilon) \leq 1 - \varepsilon$: if k bits are transmitted then, on average, $(1 - \varepsilon)k$ bits are received (and they are received correctly). For large k, with high probability the actual number of (correctly) received bits is close to the average. Thus, even if the transmitter and the receiver knew in advance which bits will be erased, information can be transmitted reliably at a rate of at most $1 - \varepsilon$ bits per channel use. The channel coding schemes proposed for such a channel is known as erasure correction codes. Some commonly used erasure corrections codes are explained in the next sections.

2.2 Reed-Solomon Codes

Reed-Solomon codes are introduced in 1960 by I. S. Reed and G. Solomon [2] and are efficient in correcting both random and bursty errors. These codes are widely used in numerous consumer products and scientific applications which include: mass data storages (e.g. CDs and DVDs), networks (DSL modem and WiMax) and deep space communications. For t-errors correcting RS code of length $n = 2^m - 1$, the generator polynomial is constructed as follow:

$$g(X) = g_0 + g_1X + g_2X^2 + \dots\dots\dots + g_{2t-1}X^{2t-1} + X^{2t} \qquad (2.1)$$

$$= (X + \alpha^1)(X + \alpha^2)\dots.(X + \alpha^{2t})$$

where α is a primitive element of Galois Field GF(2^m) and $\alpha^1, \alpha^2, \ldots \alpha^{2t}$ are elements of the same field. Note that *g(X)* is of degree exactly *2t*, all *(n, k)* RS codes satisfy the following equation:

$$n - k = 2t \tag{2.2}$$

For systematic encoding of a message polynomial $m(X)$ with k coefficients being the k message symbols over $GF(2^m)$, the parity-check polynomial $\mu(X)$ is the remainder of the shifted message polynomial $X^{n-k} m(X)$ divided by the generator polynomial $g(X)$, that is:

$$\mu(X) = X^{n-k} m(X) mod\ g(X) \tag{2.3}$$

The resulting code polynomial is:

$$c(X) = \mu(X) + X^{n-k} m(X) \tag{2.4}$$

These codes have fixed transmission rate and usually it is less than one i.e., there is some redundant information that can help in recovering the original information. The main drawback is their decoding complexity which when solved using a system of equations leads to a cubic complexity with the size of the information.

One method to decode the encoded message of the RS encoder is proposed in [2] using polynomial encoding schemes. If there are no errors in the transmitted codeword, then it is easy to recover the original message by solving and t of 2^m equations present in the codeword c and if there are errors in the received codewords, majority vote method is used. Generally, this decoding approach is inefficient especially for longer codes length, which have higher error rate. There are several algorithms proposed for efficient decoding of the RS codes which includes: Berlekamp-Massey Algorithm which is also used for BCH codes, Euclidean Algorithms and soft decoding algorithm that performs a polynomial time soft decision algebraic list decoding which is based on the work of [3].

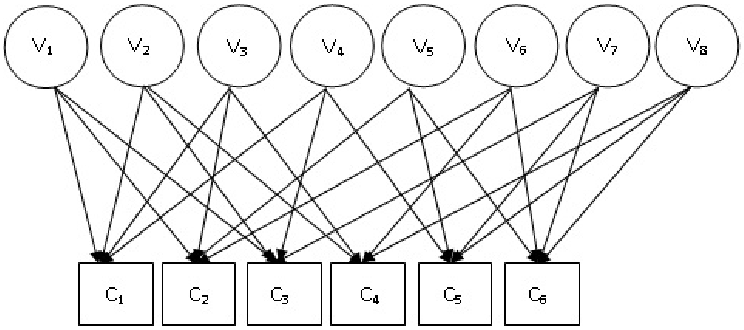

Figure 2.2 Regular (8, 6) LDPC code

2.3 Low Density Parity Check Codes

Low Density Parity Check (LDPC) Codes also referred as Gallager Codes were presented by Gallager in 1960 [4]. These codes were neglected by researcher for almost four decades. Also these codes need more computation speed and power consumption at the time of their invention. These codes were re-discovered by D. Mackay and R. Neal in 1995 [6]. Performances of these codes are evident from there used in numerous applications and standards which include: deep space satellite communication, DVB-S2 and ITU-T G.hn standard. Also these codes are adopted in 10GBase-T Ethernet and 802.11n WLAN. These codes are considered as a good alternative for Reed-Solomon codes because of their channel capacity performance and efficient decoding algorithm.

Tanner in his work [7] generalized the concept of LDPC codes and showed that these codes can be efficiently represented by bipartite graphs also known as Tanner graph. There are two types of nodes in Tanner graphs which include: Variable nodes and Check Nodes. The edges in the Tanner graph connect the two different types of nodes. In LDPC code, the Tanner graph of a code is drawn such that the check nodes j is connected to the variable node i, whenever the h_{ji} in H is a 1, where H us the sparse low density matrix. There are m rows and n columns in H and refers to the m check nodes and n variable nodes connections respectively. There are $m = n - k$ check nodes, one for each check equation and n variable nodes i.e., one for each code bit c_i. The k is the length of original information, where n is the encoded information generated from k length information. Thus the code rate can be given as $R = k/n$.

In the parity check matrix H of the LDPC code, it has exactly w_c 1's in each column and $w_r = w_c\ (n/m)$ 1's in each row, where $w_c << m$ is known as regular LDPC codes. Note that w_c is the number of columns and w_r is the of number rows of the matrix H respectively. If the number of 1's in each column or row is not constant in the parity matrix H then the code is irregular LDPC code. The bipartite graph of a regular LDPC codes is shown in Figure 2.2.

It can be noticed that the variable nodes in the Figure 2.2 are connected to three check nodes; similarly each check node is connected to four variable nodes. So, its a representation of a regular LDPC code as both variable nodes and check nodes are regular. It is observed that an LDPC codes having irregular bipartite graphs i.e., Irregular LDPC codes have better performance than the Regular LDPC codes. Calculation of check nodes can be read directly from the graph: Check node $C_2 = V_1 \oplus V_3 \oplus V_5 \oplus V_6$, where $\oplus$ denotes the exclusive-or operation. The encoding and decoding of LDPC codes can be performed in number of ways. A source message of length equal to the number of variable nodes in the sparse matrix is taken or simply the numbers of symbols in message are passed through the variable nodes. Then, the message nodes and check nodes are sent to receiver and the sent symbols are commonly known as codewords.

When the LDPC decoder receives the codeword from an ideal channel i.e., we are neglecting the errors caused by the noise and interference of the channel, a check is made if some other node can be decoded. For example in Figure 2.2, if we received set of symbols that contain nodes C_1, C_2 and C_4, then variable node V_3 can be recovered by calculating $V_3 = C_1 \oplus C_2 \oplus C_4$. Noted that the required number of symbols is not known at the decoder, and the process continues until all the variable nodes have been recovered. This implies that an LDPC codes require more than *n* of the original symbols, resulting in overhead greater than one, thus having greater overhead then RS codes. The most popular decoding algorithm for decoding the LDPC code is known as Sum-Product Algorithm (SPA) also some time refers to as Belief Propagation (BP). A good tutorial for LDPC decoder is presented by W. E Ryan in [5] and covered iterative decoding algorithms which include Probability-Domain SPA, Log-Domain SPA decoders. Also some reduced complexity decoders are also presented which includes Min-Sum and Min Sum plus Correction Factor Decoders.

Theoretical results on the LDPC codes give asymptotical results when the length of the message *n* approaches infinity. The performance of the LDPC code for longer message length is optimal and have significantly better overhead (lower) compare to the one when the length of the message size is smaller.

2.4 Digital Fountain Codes

Digital Fountain codes are rateless erasure correcting codes that were first introduced by Byers et al. in [8, 9] and were mainly used for the robust and scalable transmission of data over a heterogeneous environment. These erasure correcting codes provide protection from the effect of lost that is caused during transmission in the environment, and they ensure the recovery of lost data without requesting retransmission from the sender. These codes provide reliability in various network applications such as broadcasting, parallel downloading, video streaming etc. Digital fountain codes have the ability to generate a limitless amount of encoded packets from the information packets. Initially, information data is divided into a defined number of pieces and a different subset of these pieces are then chosen randomly to form encoding symbols. The original information is then recovered when a sufficient amount of encoded packets is received. With good fountain codes, the total number of packets needed for decoding is close to the size of the original input packets, although some overhead is necessary because of the nature of these codes. The important characteristic of these codes is that they are non-systematic codes, i.e., no specific order is required for decoding. As soon as a certain amount of encoded symbols is received, the original information is recovered with maximum probability.

Consider a fountain code with parameters (k, ρ), is any linear application that maps the binary input information of length k with random independent elements distributed over a Field F_2^k according to a probability distribution function ρ into the set of all possible sequences over binary field F, producing a potentially infinite stream of output symbols. Then, the encoding for such coding scheme can be given by the following steps:

1. The output symbol can be chosen by sampling some weighted vector value from the probability distribution function.
2. This vector should be chosen from field F_2^k in a random and independent way.
3. The output symbol is generated adding the input symbols selected by that vector.

Note that some kind of synchronization is necessary between the sender and the receiver because of the random nature of these codes that will help the receiver in identifying the correct symbols of a particular packet. This kind of information can be easily generated at source and destination using a random number generator with the same seed or using information in the header of each output symbol.

Ideally the decoding algorithm should be able to recover the input symbols from any k number of output symbols to be a perfect fountain code but this is not practical. Usually, fountain code needs some additional number of output symbols to recover the original information. This can be represented by an overhead $E = (1+\alpha).k.$

2.4.1 Tornado Codes

The first attempt to approximate the fountain code was carried out in [14] with the name of the codes known as Tornado code. These codes can be described in terms of bipartite graphs and are class of LDPC codes. The operations of these codes consist of bipartite graphs, which depend on the different layers of nodes. Two layers of nodes are sufficient to run the operation of the tornado codes but it is not necessary and it may have the many layers of nodes.

Similar to LDPC codes in which the each variable node represents the message node, each one node in the first layer of the tornado code is represented with symbol in the message, where the symbols are the number of bits in the packet. Second layer consists of some redundant nodes (additional bits), and the nodes on the first layer are connected by edges to the second layer. These connected edges of the first layer with the second layer shows the relation between the two layers. Usually exclusive OR operations is performed at the corresponding node of the second layer for each of its neighbor.

A recursive process can be used to derive more layers from the second layers, and so on. In this graph structure, the nodes in the first layer are variables nodes and the nodes on the subsequent layers are constraints (Check Nodes) on these variables. Probability distribution functions are used in a random fashion for these selecting these constraints properly, so that at the end of the decoding procedures the message will be decoded with higher probability when enough nodes information is available.

Figure 2.4.1 shows a typical example of tornado codes in which the left side of the figure shows an n bits of message bits $m_1, m_2, \dots m_n$. There are $\frac{n}{2}$ nodes at the right side of the figure that shows the parity bits $p_0, p_1, p_2, \cdots, p_{\frac{n}{2}-1}$ based on a distribution function. Each parity bit is an exclusive or (XOR) function of its neighbors on the left. For example, parity bit p_0 can be defined as:

$$p_0 = m_1 \oplus m_3 \oplus m_4 \tag{2.5}$$

These codes are sparse, systematic codes and due to a pre-defined bipartite graph structure these are not rateless codes with a linear complexity $\mathrm{O}(nd)$ at the encoder.

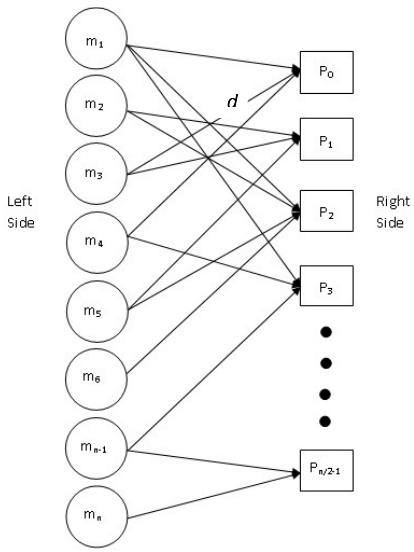

Figure 2.4.1 Tornado Codes

Note that the codeword generated from the bipartite graphs are related to each other such that in the codeword of some length K for n message bits and $\beta*n$ parity bits.

$$K = (1+\beta)*n, \qquad where\ 0<\beta<1 \tag{2.6}$$

The encoding of the message essential depends on the probability distribution function that helps in connecting the two nodes between the message and parity layers. The decoder also needs the probability function information to recover the original information from the erroneous bits received from the channel. Note that, as the encoding procedure consists of a bipartite graph, thus soft decoding can also be used in the tornado decoder. As from the procedures of the encoder clearly, the output of the tornado encoder is systematic i.e. all the input symbols that constitute the data are directly included among the encoding symbols, thus the parity information with the partial erasure information of the received codeword, the original information can be recovered.

In RS codes, every constraint depends on the message symbol and in tornado codes each constraint depends on a few message symbols. So, the encoding and decoding is fairly expensive in RS codes, where as in Tornado codes only a constant number of exclusive-or operations are required, thus it has fast encoding and decoding time.

The two other coding schemes includes: Luby Transform (LT) and Raptor Codes. As LT codes are one of the central part of this thesis work, thus we have dedicated the next chapter for stating the introduction of the LT and Raptor codes and the main performance parameters which includes: probability of success, overhead and complexity are addressed.

Chapter 3
Luby Transform Codes

The LT code also known as universal erasure codes are the first practical realization of the digital fountain concept and are proposed by Michael Luby in his famous paper [10]. These are the first codes that hold the property of rateless codes i.e., their coding rate is not fixed a priori (The relation between the code length and the dimensions are not fixed).The main advantages of LT codes are mentioned below:

1. The number of encoding symbols that can be generated from the data on the fly is potentially limitless i.e., they are rateless in nature and closely approximate the fountain codes.
2. These codes are optimal for almost every erasure channel independently from its erasure channel probability because the decoder can recovers the original data from any set of a fixed number of encoded packets.
3. Also these codes have lower complexity for both encoding and decoding processes, and therefore they are very suitable for hardware implementations and time constraint applications.

3.1 LT Encoding

In this coding scheme, the original information symbols N, is divided into k information symbols such that $k = N/l$, where l is the packet payload size. From these k information symbols, a potentially infinite amount of encoded symbols (codeword) is generated. These encoded symbols are uniformly, randomly chosen and are XOR combinations of the input symbols. The relation between the information symbol and the encoded symbols can be modeled as a sparse bipartite graph as shown in Figure 3.1, which is based on the degree distribution. The degree distribution is the key innovation in this coding scheme, and the performance of this coding scheme is associated with a

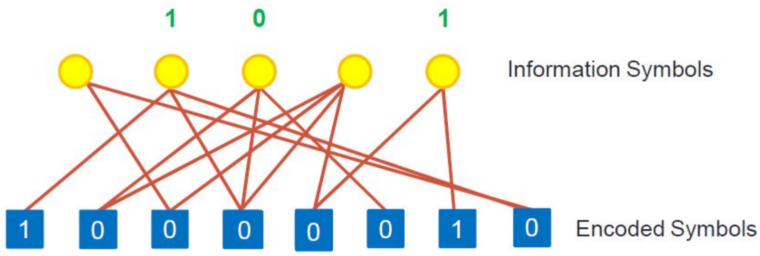

Fig.3.1 Tanner graph of an LT code.

well designed degree distribution. The degree is associated with an encoded symbol, and the distribution from which it is sampled is called degree distribution. The information symbols contained in an encoded symbol are called neighbor of the encoded symbols. The encoding process for LT codes can be described as follow:

1. Randomly draw a degree d according to the degree distribution function.
2. Choose uniformly at random d distinct input symbols.
3. Compute the encoding symbols as the bit wise exclusive-or (XOR) of these d input symbols.

3.2 LT Decoding

The goal of the decoder is to recover the original information symbols from the received information. In LT codes; this can be achieved using Gaussian elimination as well as belief propagation (BP). However, Gaussian elimination is prohibitively expensive due to its computational complexity; therefore, belief propagation is a kind of de facto decoding algorithm for LT codes.

At the receiver side, when K encoding symbols have arrives, where K is usually greater than k, the belief propagation algorithm is used. Initially, all the encoding symbols of degree one cover their unique neighbor, and the set of covered input symbols that are not currently processed is called ripple. At each subsequent step, one input symbol is processed, which is then removed as a neighbor from all other encoding symbols. In order to get the decoding algorithm started, at least one encoding symbol of degree one is required. The XOR operation is performed on the encoded symbols to recover the original input bits. The decoding process works as follow:

1. Identify the degree one encoded symbols, and if there are no degree one encoded symbols, then the decoding process will halt at this point.
2. After identifying the encoded symbol, find the corresponding neighbor input symbols.
3. Once the symbol is identified, it is removed as a neighbor from the other encoding symbols and the degree of all the encoding symbols will be reduced by one.
4. Repeat steps 1 to 3, until all the input symbols are decoded.

The decoding is successful when all input symbols are recovered and the ripple is empty at the end of the decoding process. In any other case, the decoding of LT code fails. Maintaining the ripple size during the decoding process is very important because the decoding process can fails at any stage if the ripple size becomes zero, when some input symbols remain unprocessed.

It is also notable that both encoding and decoding of the LT codes need the XOR operation and degree distribution. The XOR operation can calculates the computation complexity of LT codes. In a preferred case, few released encoding symbols cover the input symbols in the ripple to minimize the number of encoding symbols. The degree distribution ensures that the encoding symbols are incrementally released to cover the input symbols. Also, it keeps the size of the ripple small to avoid the redundant coverage of input symbols in the ripple by multiple encoding symbols. The tradeoff for the size of the ripple ensures the efficiency of the decoding process and consequently the LT codes.

As from the above analysis, it is clear that encoding and decoding depends on the probability distribution. The probability distribution function helps the encoder to select randomly the input symbols to generate the encoded symbol. In case of the decoder, it also uses the same distribution functions to select the received symbols from the channel and recover the original information, although some overhead is necessary to recover the complete information symbols. The probability distribution can be termed degree distribution and is a crucial part of the LT codes. The degree distribution determines the random behavior of the LT process. The LT process can be seen as a generalization of the classical process known as bins and balls. Input symbols are analogous to balls and encoding symbols are analogous to bins and k balls are randomly thrown into N bins.

Similarly, opposite of this also complies for the decoding process in which, the encoding symbols can be thought of as balls and input symbols as bins. The LT process succeeds if at the end all balls are recovered from the bins. Some characteristics of LT codes are explained in the upcoming sections.

3.3 Soft Decoding of LT Codes

The soft decoding in the LT codes yields significant improvement in the output. Belief Propagation (BP) algorithms is used to recover the information from the encoded symbols. The BP algorithm used in the decoding of LT code is similar to the LDPC codes decoder. The log-likelihood ratio (LLR) or L-value of a binary random variable $x \in \{0,1\}$ can be defined as:

$$L(x) = \log(\frac{p(x=0)}{p(x=1)}) \tag{3.1}$$

An exclusive-or operation is performed over two random variable x_1 and x_2 and the corresponding value can be calculated as:

$$L(z) = L(x_1 \oplus x_2) = 2ar\tanh(\tanh(\frac{L(x_1)}{2}).\tanh(L(\frac{x_2}{2}))) \tag{3.2}$$

The original message is then calculated by passing the message between the variable node and check node in BP algorithm. The Figure 3.2 shows that a received encoding

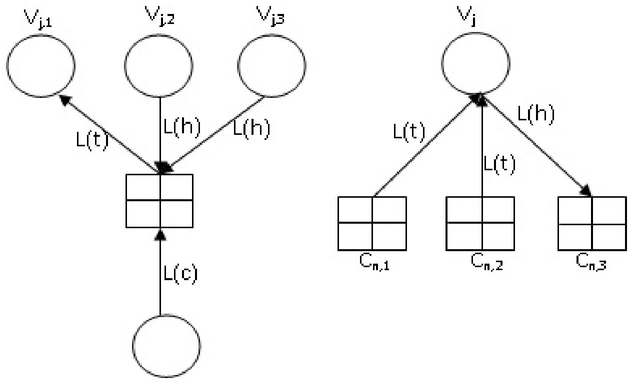

Figure 3.2 Message Passing on the Tanner Graph for Soft Decoding

symbol C_n with its neighbor V_i is represented in the tanner graph. The *L(t)* denotes an L-value message from the check node *n* to the variable node *j*. whereas, the *L(h)* is obtained from the bitwise exclusive-or operation according to degree distribution function and equation 3.2.

$$L(t) = 2\arctan h(\tanh(\frac{L(c)}{2}).\prod \tanh(\frac{L(h)}{2})) \qquad (3.3)$$

Note that *L(c)* denotes the soft information received by the decoder from the channel. The *L(h)* can be obtained by the sum of L-values passed to variable node *i*. The decoding decision *L(v)* is performed by using hard decision decoding using *L(t):*

$$L(v) = \sum L(t) \qquad (3.4)$$

The decoding is performed in an iterative manner and the performance of the decoder increases as the number of iterations increases. The decoding process will be attempted only when sufficient information is available at the decoder input.

3.4 Symbol Release Probability

The ripple size also plays a vital role in the decoding of the LT codes, and the evolution of the ripple is determined by the degree distribution. A relation between the degree of an encoded symbol and the point where the symbol is reduced to one of its input symbols is proposed in proposition 7 of [10]. Let the degree distribution *ρ(i)* be

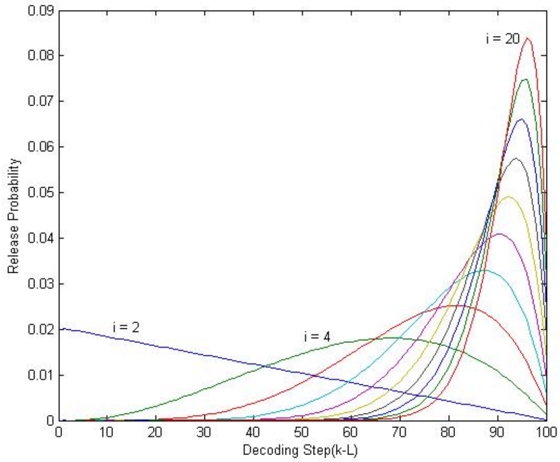

Fig.3.3 Release Probability as a function of decoding step for fixed degrees (d).

the probability that an output symbol has the degree *i*. Let the degree release probability *q(i, L)* be the probability that an output symbol of degree *i* is released exactly when *L* input symbols remain unprocessed. Then

$$q(i,k) = \begin{cases} 1 & for\ i = 1 \\ \frac{i(i-1)L\prod_{j=0}^{i-3} k-(L+1)-j}{\prod_{j=0}^{i-1} k-j} & for\ \ i = 2,\dots,k, for\ all\ L = k-i+1,\dots,1 \\ 0 & for\ all\ other\ i\ and\ L \end{cases} \tag{3.5}$$

Let *r(i, L)* = *ρ(i).q(i, L)* be the probability that an output symbol of degree *i* is released when *L* input symbols remain unprocessed i.e., *r(i, L)* = *ρ(i).q(i, L)*. Where *r(L)* be the overall probability that an encoding symbol is released when *L* input symbols remain unprocessed i.e., $r(L) = \sum_i r(i,L)$. In [12] is presented a finite length analysis of LT codes providing expressions for the probability of decoding success. Figure 3.3 shows a plot of the release probability, which expresses the release probability as a function of unprocessed input symbols and the original degree *d*.

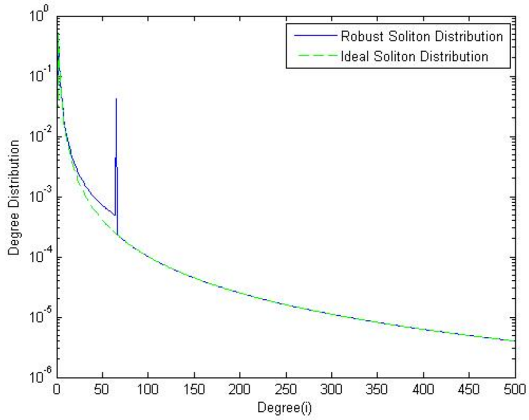

Figure3.4 Soliton and Robust Soliton Distribution Functions

3.5 Degree Distribution Functions

The performances of these codes are associated with a well designed degree distribution, and with such a degree distribution, the decoder needs less encoding symbols to recover all the information symbols. The use of degree distribution in LT codes induces a measured randomness and is one of the key aspects in the design of these codes, making these codes ideal for use in the heterogeneous environment. The performance of LT codes is closely associated with the degree distribution. There are mainly two objectives of the degree distribution. The first is to reduce the number of encoding symbols required to cover all the input symbols to ensure a minimum overhead and computational cost. The second is to ensure the success of the LT process with minimum number of encoding symbols. The overhead in the LT process shows the performance of LT codes and can be represented as $\alpha = K/k$. In [10], two degree distributions are proposed for the success of the LT process and are discussed below:

3.5.1 Ideal Soliton Distribution (ISD)

The ISD $\rho(d)$ shows the ideal behavior in terms of the expected number of encoding symbols needed to recover the input symbols. However, it is not possible that k

encoding symbols cover the k input symbols, as per the characteristics of the rateless codes. Also, as the ripple size is one, so after exactly k steps, the original information symbols should be recover and any variation in the ripple size will lead the decoding towards failure.

$$\rho(d)=\begin{cases}\frac{1}{k} & \text{for} \quad d=1\\ \frac{1}{d(d-1)} & \text{for} \quad d=2,3,.k\end{cases} \tag{3.6}$$

3.5.2 Robust Soliton Distribution (RSD)

The RSD is μ(d) defined by $R = c\ ln(k/\delta)\sqrt{k}$ for some suitable constant c>0: defined as:

$$\tau(d)=\begin{cases}\frac{R}{d.k} & \text{for} \quad d=1,2..k/R-1\\ \frac{R\ln(\delta)}{k} & \text{for} \quad d=k/R\\ 0 & \text{for} \quad d=k/R+1,..k\end{cases} \tag{3.7}$$

The $\rho(d)$ is added to τ (d) and normalized to obtain RSD.

$$\beta=\sum_{d=1}^{k}\rho(d)+\tau(d)$$

$$\mu\ (d)=\frac{\rho(d)+\tau(d)}{\beta} \qquad \text{for all } d= 1, 2,\ldots k \tag{3.8}$$

There are mainly two tuning parameters in RSD, which are c and δ, where c controls the mean of the degree distribution and δ estimates that there are $\ln(k/\delta)\sqrt{k}$ expected ripple sizes. The ISD and RSD are plotted on a log-linear plot in Figure 3.4.

The ISD is optimal in terms of overhead when all the random variables follow expected behavior. The optimal behavior of ISD is achieved when the ripple size is kept constant, but fails with a very small variation in the ripple. The performance of

RSD is better only when the size of the input symbol approaches infinity. In practice, only a limited size of information symbols can be transmitted and thus the complete information is divided into a finite number of input symbols. Thus, the performance of RSD is practically viable but not optimal. Therefore, the research community is searching for an optimal degree distribution that can enhance the performance of LT codes, irrespective of the length of the information symbols.

3.6 Optimal Degree Distribution

Mainly there are two ways, in which the performance of the LT codes can be improved. These includes: Optimizing the degree distribution function or either improving the BP decoding algorithms for LT codes. In this section, we present some preliminary study work that we have carried out to optimize the degree distribution function. The detail work about optimizing the degree distribution is presented in the next chapter.

Some different strategies are proposed previously to optimize the degree distribution. Like in [17], an algorithm for iterative optimization of the degree distribution is proposed, using important sampling approach. In [18, 19] some new degree distributions are proposed which are based on the varying ripple size R. Both degree distributions demonstrate the efficiency of their proposed schemes. In our proposed work we consider the equation (3.1), which is related to the released probability function that was proposed in [10] for LT codes. It was observed that with increase in the degrees, the probability value increases significantly which is also shown in the Figure 3.3. So, keeping this situation in mind, we have proposed another degree distribution function that follows the behavior of the release degree distribution; which is an increase in the value of degree values as the degree value increases at the end of the total degree distribution values.

The proposed degree distribution is based on a parabolic distribution. The parabolic distribution function is given below.

$$\phi(d) = \frac{6((d_o+1)d_o - d^2)}{d_o(d_o+1)(4d_o-1)}, \; d \in \{1, ..., d_o\} \tag{3.9}$$

From maximum to minimum values of d, the above function gives a monotonic parabolic shape. This function has a maximum return at $d = 1$ and minimum at $d_o \leq k$. The "d_o" is the free parameter (in general a function of k) in the above distribution, which is responsible for fine tuning. Also the sum of all $\phi(d)$ for values of d_o is equal to one. The proposed degree distribution is based on equation (1) i.e. parabolic distribution function and is given by:

$$\theta(d) = \begin{cases} \frac{2R}{k} & for \; d = 1 \\ \frac{1}{(1-\emptyset(d))} & for \; d < \left[\frac{4k}{5}\right] \\ \theta(d-1) + c.\theta(d-1) & for \left[\frac{4k}{5} \leq d \leq k - R + 1\right] \end{cases} \tag{3.10}$$

where the value of the d is chosen such that the above degree distribution is nearly equal to one and $c > 0$ is some suitable constant. The spectrum of degree distribution is divided in to three sections; the first section was defined for $\theta(1)$ to ensure that the ripple starts off at a reasonable size. The middle section is associated to the parabolic distribution and the last section ensures that there are some high degree symbols available at the end of the decoding. The histogram of the proposed degree distribution and Robust Soliton Distribution (RSD) is shown in the Figure 3.5.

We have analyzed the performance of the proposed degree on two different performance metrics. As we know that with an increase in the number of encoded symbols, the output of the decoder i.e., is the correctly received input symbols increase. Thus, we have analyzed both the degree distribution functions on the Bit Error Rate

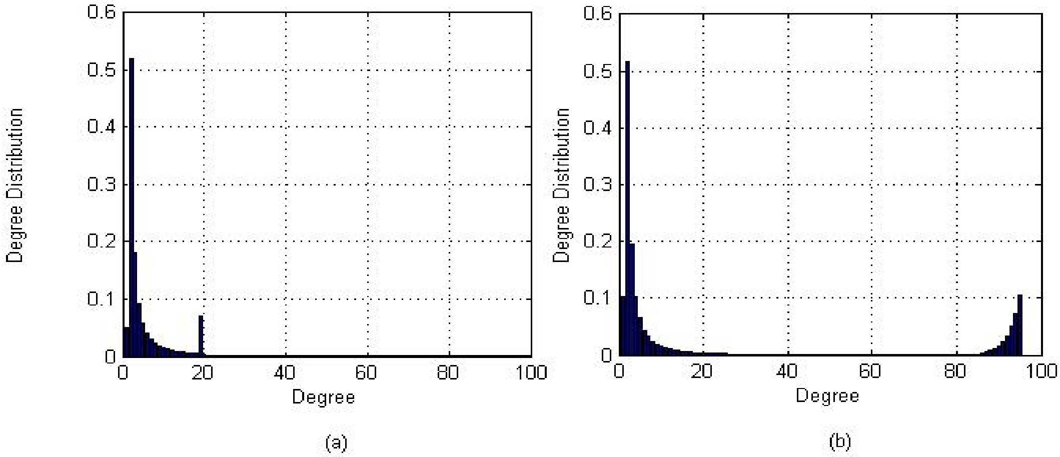

Fig.3.5 Histogram Degree distributions of (a) RSD and (b) Theta distributions

(BER) versus the overhead *(1+k)* symbols. Also a comparison of the required encoded symbols versus the known input symbols was carried out.

3.6.1 Simulation Parameters & Results

We have selected the following parameters for our simulation work. For RSD k = 100, $R = 2+\sqrt[4]{k}$ and $c = 0.1$, $\delta = 1$ and for optimized degree distribution $c = 0.045$, $d_o = 1.17$ is considered. The value of $k=1000$ and 50 randomize trials per point is considered for Figure 3.6. In Figure 3.6, we report the BER curves for LT codes based

on the respective distribution functions. Note that we have assumed an ideal channel i.e., there are no errors caused by the noise and interference in the channel. The errors curve shows the performance of the decoder for a particular overhead. The proposed degree distribution performs better than the RSD and converges to the input symbols with lower overhead. Figure 3.7 shows the comparison of the required encoded bits for a particular value of the input symbols. The proposed scheme outperforms the RSD. The parameter d_o in the parabolic distribution should be adjusted for different values of k that give us $\phi(d)$ distribution.

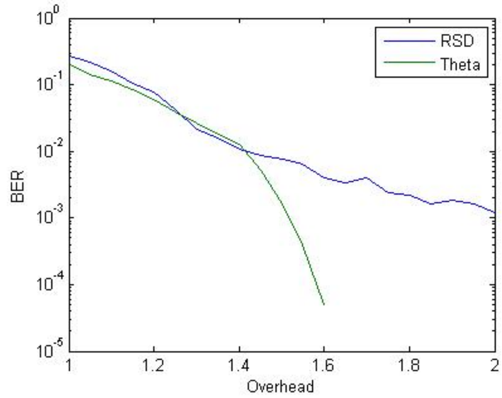

Fig.3.6 BER performances of LT codes with the RSD, the proposed *θ(d)*.

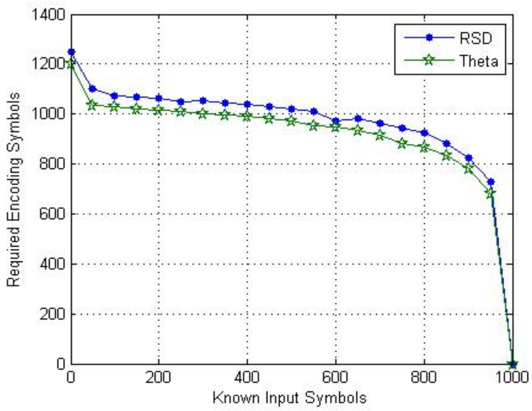

Fig.3.7 Performances of LT codes with the RSD and the proposed *θ(d)* degree distributions for k=1000

3.7 Raptor Codes

Raptor codes are part of Digital fountain family and are proposed by Amin Shokrollahi in [11]. The significance of these codes is clear from their adoption in 3 Generation Partnership Project (3GPP) and Multimedia Broadcast/Multicast Service (MBMS) for 3rd generation cellular networks and Digital Video Broadcast (DVB-H) standards [15, 16]. The Encoding and decoding complexities of these codes are linear, and thus have better efficiency than that of LT codes. These codes are an extension to LT codes in which original message *m(d)* is passed through a preliminary coding process called pre-coding. The pre-coding in the Raptor codes adds additional intermediate nodes in the encoding process and then these intermediate nodes with the original message are

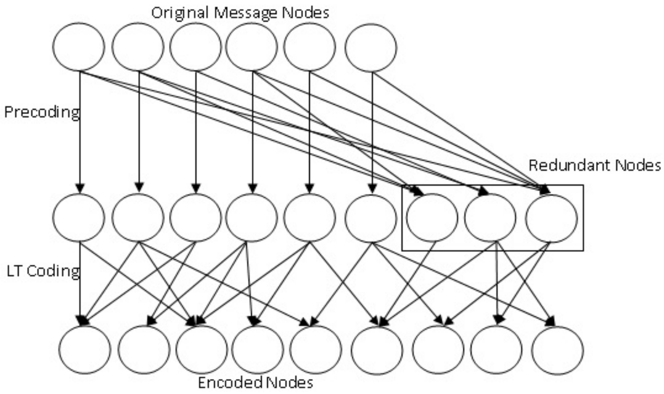

Figure 3.8 Tanner Graph of the Raptor Codes

passed to LT encoder as input nodes. The primary reason for adding additional intermediate node is to reduce error floor problem that exists in both LDPC and LT codes. The Raptor code can be represented as *(k, C,(d)),* where *C* is a linear code with block length *n* and dimension *k* called the pre-code and *d* is the degree distribution. An LT codes uses these *k* input symbols plus the redundant symbols produced by the linear code to generate encoded symbols. Figure 3.8 depict the tanner graph representation of the Raptor code.

The pre-coding in the Raptor codes can also be done in several steps, e.g., first using an LDPC erasure code to produce the intermediate symbols for an LT encoder. Then the original message and intermediate symbols are used to generate the encoded output symbols. Also codes without LT coding can be regarded as a subclass of Raptor codes, called pre-coding-only or PCO Raptor codes. By using some erasure correcting code as the pre-code, the requirement to recover all input symbols of the LT code is lifted: only a constant fraction of LT encoded symbols needs to be recovered and the original message can then be recovered by the erasure correcting property of the code used for the pre-coding. The decoding of Raptor code is done by first using the LT decoding process to produce the intermediate code and after that input nodes are recovered by using an algorithm for decoding of linear code C.

Chapter 4
Optimization of the Degree Distribution

In the last chapter we have discussed the Digital Fountain, more particularly, LT codes in detail and also we reviewed the Raptor Codes which are extensions of the LT Codes. The Significance of the Raptor Code is evident from their adoption in the MBMS and DVB-H standards. Also, it is notable that an efficient Raptor codes itself rely mainly on the LT Codes. Because Raptor codes use the LT Codes as their inner codes thus the performance of the LT codes directly influence its extended version of the code. Also, we have discussed that, in LT Codes, there are mainly two approaches which can enhances the performance of the LT codes that include: Optimizing the degree distribution function and secondly improving the decoding algorithm. For example: Tarus et al. in [20] proposed a joint decoding strategy based on belief propagation and Gaussian elimination to reduce the computational cost and overhead of LT codes. In this chapter, we will present our optimized degree distribution in detail. We will focus on the issue about selecting an appropriate and valid method/technique that can help in implementing an efficient LT codes. Our main objective for optimizing the LT code is to reduce the additional overhead that exists in the preceding degree distribution functions. That helps not only in the reducing the computational complexity of these codes but also reduce the time it takes to recover the original information symbols.

4.1 Review

Luby in his work [10] proposed two degree distributions to cooperate with the LT codes, which includes Ideal Soliton Distribution (ISD) and Robust Soliton Distribution (RSD). Figure 4.1 shows the plot of both the ISD and RSD on the logarithmic scale. The ISD is optimal in terms of overhead when all the random variables follow

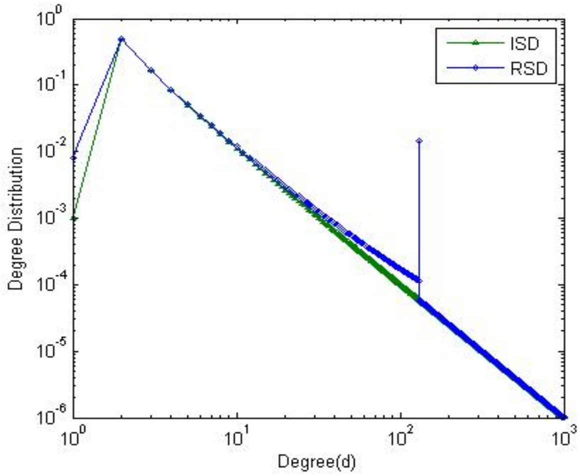

Figure 4.1 Degree Distribution function of ISD and RSD on the Logarithmic Scale

expected behavior. The optimal behavior of ISD is achieved when the ripple size is kept constant, but fails with a very small variation in the ripple. The performance of RSD is better only when the size of the input symbol approaches infinity. In practice, only a limited size of information symbols can be transmitted and thus the complete information is divided into a finite number of input symbols.

So, there should be some approach that will help the LT process to work better in this regard, irrespective of the size of the information symbol. ISD and RSD are plotted on the logarithmic scale in Figure. Luby in [10] used the LT process to design and analyze a good degree distribution for LT codes. An analysis of classical balls and bins argument was considered, which shows that $K=k.\ ln(k/\delta)$, balls are necessary on average to ensure that all k bins are covered with at least one ball with a probability of at least 1-δ, where as balls are considered as encoded symbols and input symbols are analogous to bins.

Therefore, there is still a search for an optimal degree distribution that can enhance the performance of LT codes, irrespective of the length of the information symbols. An optimization of the degree distribution function was considered in Hyytiä et al. in [17], which propose an algorithm for iterative optimization of the degree distribution, using

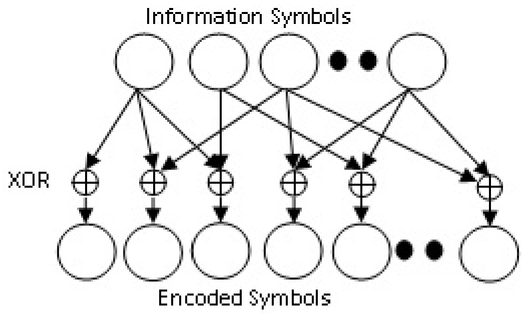

Figure 4.2 Tanner graph of an LT code.

an important sampling approach. In [18] and [19], some new degree distribution functions are proposed that consider a varying ripple size R, as compared to the proposed degree distribution in [10], where it is assumed to be a constant. In [21], a heuristically approach is used in which the multi-objective evolutionary algorithm (MOEA) is proposed in which two objectives are considered, including the operational cost and overhead are used to optimize the degree distribution of LT codes.

4.2 Analysis of the LT Process

This section provides details about the optimization frame work carried out for degree distribution by considering the power law as one of the main entity in the encoding and decoding process of LT codes. The analysis and role of the degree distribution have significant importance in LT codes. From the encoding and decoding procedure of LT codes, we can figure out some of the interesting and important characteristics of LT codes. As we know that the encoding and decoding of LT codes significantly depends on the degree distribution and follows a specific pattern.

We know that the encoding and decoding pattern of LT codes can be organized into a specific data structure called Tanner graph. Usually a Tanner graph is a bipartite graph where the nodes in the first set are the input symbols and the second sets are the encoded symbols, as is shown in Figure 4.2. Bipartite graph can also be considered as a network that consists of vertices: the nodes representing the input symbols and the information edges representing the encoded symbols. When the degree distribution follows the Power law, the network formed by the encoding and input symbols resembles the free scale network [21].

$$P\,(x) \sim x^{-\gamma} \tag{4.1}$$

Where γ is an exponent of the power law and x is a constant. For any fixed value of γ, $P\,(x)$ converges to 1. Usually a distribution following the power law can be easily identified by plotting the quantity on the logarithmic scale, which is a straight line. Similarly, when the degree distribution follows Poisson distribution, the resulting encoded and information symbols will resembles a random network. We observe that the ideal soliton distribution and robust soliton distribution when plotted on the logarithmic scale are a straight line, as is shown in Figure 4.1. Thus, these two distributions i.e., ISD and RSD follow the power's law. On the basis of this analysis, we categorize the LT codes into two different categories.

1. Free Scale LT codes
2. Random LT codes

There is a major topological difference between the random and scale free networks, as in random networks most nodes have approximately the same number of information edges. In contrast to that, a scale free network implies that the nodes with only a few information edges are numerous, but few nodes have a very large number of information edges. This non trivial development of this complex network have large applications in practical life, and a good example includes the World Wide Web, metabolic and protein network, sizes of earth quakes, etc. In the next section, we propose an optimized degree distribution based on the power law distribution, essentially a degree distribution for scale free LT codes.

4.3 Framework for Optimized Degree Distribution

There are two main mechanisms that are absent in a classical random network that give rise to the power law distribution. Most networks grow through the addition of new nodes that link through the information edges to the nodes already present in the system. Second, there is preferential attachment, in other words, the nodes with only a few information edges are numerous, but few nodes have a very large number of information edges. The growth and preferential attachment inspired the development of scale free LT codes to work better as it is more random then that of random LT codes. In this section, we propose two degree distributions that are inspired by the free scale LT codes. The second distribution is the modified form of the first degree distribution. The two distributions are described below:

4.3.1 Pareto's Distribution

The Pareto distribution *P(d)* is the statistical distribution of specific population, according to some pertinent size measurements of the individual entities that compose them, and follows the power law distribution. It was first proposed as a model for the distribution of income or wealth. The Pareto distribution is defined by the following functions:

In terms of cumulative density function:

$$P(d) = \left(\frac{d}{x}\right)^{-\alpha}, d \geq x > 0 \tag{4.2}$$

And the in terms of probability density function, it can be represented as:

$$p(d) = \frac{\alpha . x^{\alpha}}{d^{\alpha+1}}, d \geq x > 0 \tag{4.3}$$

Here, x>0 is a scale parameter, and α is a shape parameter, which usually measures the heaviness in the upper tail. The degree distribution function based on the Pareto distribution for LT codes is given below:

$$f(d) = \begin{cases} \frac{R}{n} & \text{for } d = 1 \\ p(d) & \text{for } d = 2,3,..k \end{cases} \tag{4.4}$$

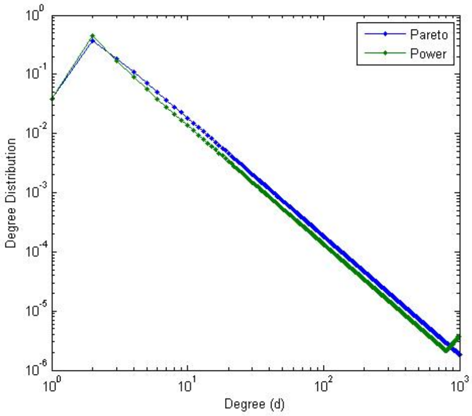

Figure 4.3 Degree Distribution function of Power and Pareto's Distribution on the Logarithmic Scale

4.3.2 Power Distribution

The second degree distribution is based on a heuristic study of our work and is based on the parabolic distribution function. The parabolic distribution function is given below.

$$\phi(d) = \frac{6((d_o+1)d_o - d^2)}{d_o(d_o+1)(4d_o-1)}, \; d \in \{1, \ldots, d_o\} \tag{4.5}$$

From the maximum to minimum values of d, the above function gives a monotonic parabolic shape. This function has a maximum return at $d = 1$ and minimum at $d_o \le k$. "d_o" is the free parameter (in general a function of k) in the above distribution, which is responsible for fine tuning. Also, the sum of all $\phi(d)$ for values of d_o is equal to one. The proposed degree distribution is based on equation (4.5), i.e., parabolic distribution function and is given by:

$$\theta(d) = \begin{cases} \dfrac{R}{n} & \text{for } d = 1 \\ \dfrac{1}{(1-\phi(d))} & \text{for } d < [4k/5] \\ \theta(d-1) + c.\theta(d-1) & \text{for } [4k/5] \le d \le k - R + 1 \end{cases} \tag{4.6}$$

Where the value of n is chosen such that the above degree distribution is nearly equal to one, and $c > 0$ is some suitable constant. The spectrum is divided into three sections; the first section was defined for $\theta(1)$ to ensure that the ripple starts off at a reasonable size. The middle section is associated with the parabolic distribution. In Figure 3.3, we observed that as the degree distribution increases, the release probability distribution function becomes an increasing function of the degree *d*; thus, the last section ensures that there are some high degree symbols available at the end of the decoding. Figure 4.3 shows the Power and Pareto distribution function on the logarithmic scale.

4.4 Performance Evaluation

In this section, we provide the experimental results to validate the performance of our proposed schemes.

4.4.1 Objectives

In this paper, degree distribution is mainly optimized to reduce overhead α and improve the bit error rate performance for a particular overhead. The redundancy in LT code is traded to provide better convergence of the received symbols towards the input symbols. Some extra redundant symbols increase the overhead in LT codes. An ideal case will be that the encoded symbols become equal to the input symbols; however, it seems to be difficult. Initially, we evaluate the performance of BER versus overhead to observe the convergence of the input symbols for a particular overhead. Also, it is noticed that there is no need to bin the data at all to calculate the distribution function. By the definition of probability distribution function, it is well defined for every value of *k* [23]. Also, the overhead is a function of input symbols and can be represented as (1+α).k. We can observe the convergence of the received encoded symbols towards the input symbols; however, it does not ensure the efficiency of the proposed scheme in terms of overhead. For this reason, we carry out another experiment, in which we compare the average overhead with input symbols at different values of *k*. Note that the results are the average of 50 different simulation runs.

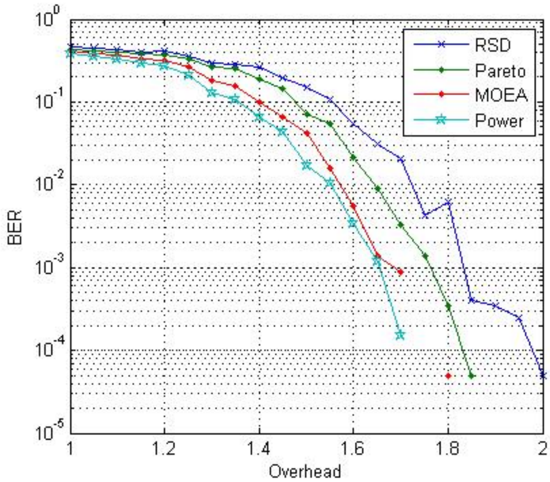

Figure 4.4 BER versus Overhead for Different Degree Distribution Functions.

4.4.2 Simulation Parameters & Results

The simulations parameters considered in this work are described in this section. We apply an LT code with intermediate symbols for calculating the bit error versus the overhead for k=1000 as shown in the Figure 4.4. Other values are α=2 and x = 0.96 for *f(d)* and d_o = 1.112 and c = 0.003 for $\theta(d)$. RSD is simulated with parameters c=0.1 and δ=1, since they provide the smallest average overhead [24]. $R = 2 + \sqrt[4]{k}$ is consider as suggested in [18]. The Figure 4.5 shows the comparison of the average overhead and input symbols k. The degree distributions are simulated at k equal to 256, 512, 768, 1024, 1280, 1536, 1792 and 2048 with the same parameters as consider for $k=1000$.

In Figure 4.4, the bit error rate (BER) of input symbols for an increasing overhead is shown. It is noticeable that for a particular overhead, the proposed degree distributions performed significantly better than the RSD and MOEA (Multi-objective Evolution Algorithm). Note that we have used the intermediate sample proposed in [21] for comparing the proposed scheme with MOEA, and we found that the proposed scheme works better and converges earlier for the input symbols. Figure 4.5 shows a comparison of average overhead and input symbols which clearly demonstrates the performance of the proposed degree distribution functions. It is notable that the

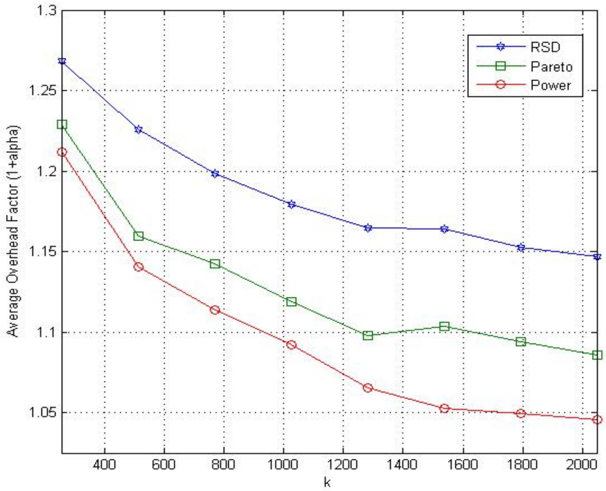

Figure 4.5 Performance of LT codes by the Proposed Schemes.

overhead decreases faster as the value of *k* increases for all degree distribution functions, giving rise to the argument that when the size of k approaches infinity the degree distribution functions performs better. On average the *f(d)* decreases 0.0585 and *θ(d)* by 0.0965 compare to the RSD and hence there is a decrease nearly 31% and 48% respectively in the average overhead.

A comparison of the degree values for a particular input symbols for three different degree distribution for encoding and decoding side of the LT codes has been shown in Figure 4.6 and Figure 4.7 respectively. Both Figure 4.7 and Figure 4.6 are identical, as the degree values at the decoder is a function of the code rate which is not considered at the encoding side. The degree values presented in the above figures are the average of 50 different frames of data that are generated randomly and independently from each other. We observe that there is a significant improvement in the degree values which accounts for the number of exclusive-or operation at the encoding and decoding side of LT codes. These exclusive-or operations in the LT codes represents there computational complexity. In Figure 4.7, the decoder performs on average 2669 operations per iteration to recover the original information for RSD for an input

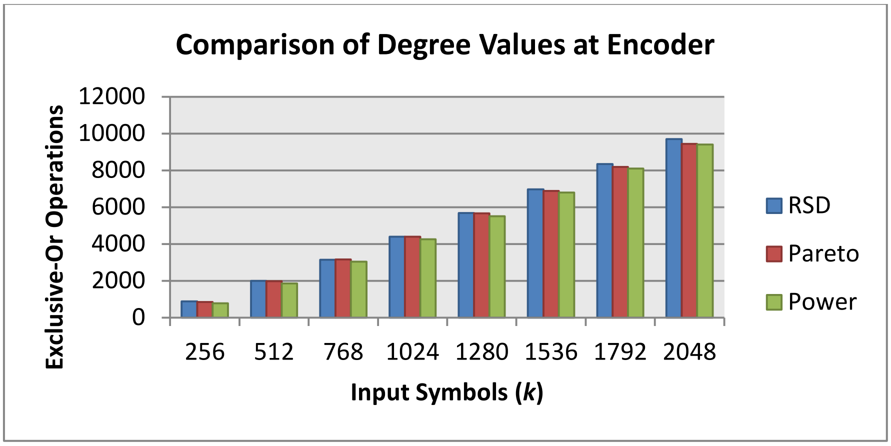

Figure 4.6 Comparison of Degree Values for different degree distribution at Encoder side.

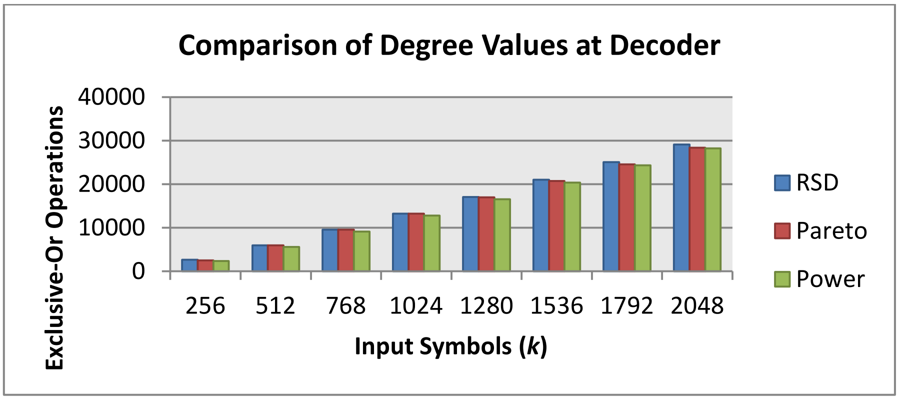

Figure 4.7 Comparison of Degree Values for different degree distribution at Decoder side.

symbols of k = 256. Whereas for Pareto and Power distribution functions, the number of operations reduces to 2537 and 2330 for a same input symbols size and thus an improvement of 5.2% and 12.7% in the performance of the LT codes has been observed by decreasing the number of exclusive-or operation for a particular input symbols. For the observed input symbols in the experiment having a range from 256 to 2048, the degree values for the proposed degree distribution functions are lower for a particular input symbol than that of RSD. Also, it is observed that, the percentage of degree values increases as the size of the input symbols increase and the percentage difference between the proposed degree distribution function and RSD reduces i.e. for example: At k = 2048, the RSD performs 29132 operations per iteration to recover the input symbols. Similarly, for Pareto and Power distribution functions, they perform 28375 and 28219 operations per iteration and thus having a net performance improvement of 2.5% and 3.23% respectively.

4.5 Conclusions

In this chapter, an optimization framework for LT Code crucial part i.e., degree distribution functions has been proposed for optimal degree distribution. From our analysis, we have categorized the LT codes into two different categories that include: Scale Free LT codes and Random LT codes, and this classification are based on the degree distribution function. In Figure 3.3, we notice that as the degree increases the release degree becomes a much faster increasing function of degree distribution. Also, we have defined the two degree distribution functions, which follow the Power Law. The simulation results show that the proposed degree distribution functions significantly outclasses the robust soliton distribution. The proposed work can also aid Raptor codes designers to customize the weakened LT code for the use of different pre-codes.

Chapter 5
Concatenated Fountain Codes

Fountain Codes ensure reliability and robustness for time varying channels in wireless communication and the performance of these codes is one of the interesting areas in the research community. In this chapter, we review the performance of LT Codes that is a part of the digital fountain family over fading channels. The performance of the LT codes and the modified version of the LT codes which we termed as concatenated codes are presented in the next sections.

5.1 Introduction

Broadcasting or multicasting is a natural solution to distribute large amount of data over the internet to millions of its concurrent subscriber. The standardization of the Multimedia Broadcast and Multicast Services (MBMS) [15], Digital Video Broadcasting (DVB) [16] and its introduction in to wireless cellular networks turned significant interests in the reliable broadcast. However the transmission in the wireless broadcast faces problems of different receiving conditions which are usually experienced by different loss rates or Signal to Noise (SNR) for individual receivers. The goal of any wireless broadcast network is reliable data transmission having low network overhead and minimum service infrastructure with the support of maximum number of receivers. Some broadcast schemes are proposed to solve the reliable broadcast problems like Automatic Repeat Request (ARQ). But these schemes are not feasible in such environments when there is no feedback with high data rates. Alternatively, Forward Error Correction (FEC) was used for reliable wireless broadcast and to some extend fulfills the wireless broadcast requirements. A channel code with potentially limitless redundancy is able to solve the problem of reliable broadcast and asynchronous data access and such a concept was introduced by Byers et al in [8,9] and is known as Digital Fountain (DF). The first practical realization of the digital fountain was proposed by Luby in [10] and known as Luby Transform (LT) codes and was extended to Raptor Codes [11] by Shokrollahi.

Fountain codes are erasure correcting codes having the ability to generate potentially limitless amount of encoded packets from the source packets. The original information is then recovered; when sufficient amount of encoded packets are received. With good fountain codes the total number of packets needed for decoding is close to the size of the original input packets, although some overhead is necessary because of the nature of these codes. These codes were initially proposed for the reliable multicast problems over the wired internet and have mostly investigated for the erasure channels. Some researcher investigated the LT and Raptor codes for Binary Symmetric Channel (BSC) and AWGN channel in [25]. In [13], a generalized frame work is presented for wireless broadcast system using LT codes and its performance is evaluated using AWGN and fading channels. Also in [26], rateless codes are analyzed from cross layer perspective by considering the reliability and efficiency in order to optimize the system throughput.

In this chapter a new system model for wireless broadcast is presented. The system model consists of concatenated fountain codes with Binary Phase Shift Keying (BPSK) modulation over fading channels. The system model is evaluated for AWGN channel and Rayleigh Flat and Frequency Selective channels.

5.2 Concatenated Fountain Codes System Model

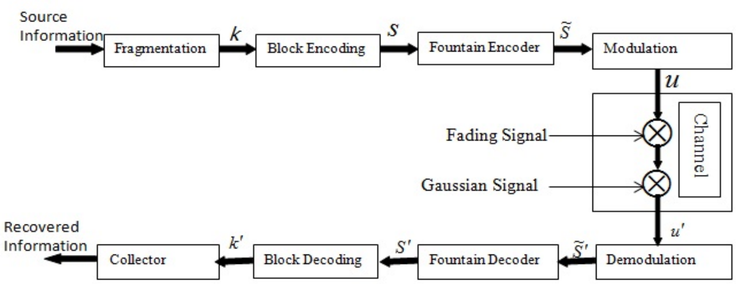

Figure 5.1 Concatenated Fountain Codes System Model for Wireless Broadcasting System

The system model of the wireless broadcast system based on the concatenated fountain code is shown in the Figure 5.1. The aim of this frame work is to send the source information reliably over the fading channel. We consider a simple broadcast system with a single transmitter and multiple receivers. The receivers are independent to receive and reconstruct the available information from the faded channel and able to receive the message at any arbitrary time. Also it's assumed that there are no feedback channels available for retransmission requests for the lost of data.

In the system framework, initially the source information is divided in to *k* number of source packets of finite length using the fragmentation process. These *k* numbers of finite length source packets are then encoded using block coding scheme. In general, the Cyclic Redundancy Check (CRC) scheme is used to identify the erroneous packets. Instead using CRC we use two different block coding schemes, which includes: Bose Chaudhuri Hocquenghem (BCH) and Reed Solomon (RS). These block codes act as a concatenated code with the fountain codes and their main advantage is their error detection plus error correction. The source packets are encoded using block coding scheme and redundant bits are appended to the original information after in the encoded packet and forwarded to the fountain decoder. The fountain encoder generates a sequence of potentially infinite encoded packets of finite length. These infinite encoded packets are modulated to be transmitted in the broadcast environment.

We consider that channel state information is not available at the receiver but not at transmitter. Also, each transmitted encoded packets are transmitted sequentially and the decoding process is attempted sequentially. The receiver tunes to the broadcast session at any arbitrary time, without any acknowledgment mechanism. The smallest entity a receiver can receive is a radio block (transmitted packet). The receiver will receive the transmitted blocks in a form of a receiving patter when the receiver is tune to the broadcast session. After receiving enough information from the radio broadcast, the fountain decoder after decoding will produce the appropriate information that can be then decoded using by the block decoder to recover the original information. If there exists any error after the block decoding in the packet, that particular packet will be discarded.

5.3 Performance Evaluation

5.3.1 Channel Metrics

Transmitter with multiple receivers are considered to transmit the information symbols over fading channel such that the channels state information is available at the receiver and not at the transmitter. If the channel gain is by h_i, then the information transmitted over the fading channels is given by

$$Yi = h_i X_i + N_i \tag{5.1}$$

Where, at time i, Y_i and X_i are the received and transmitted complex symbols, respectively and N_i is the complex additive white Gaussian noise. We consider block fading channel model, where the fading remain constant during one block but changes over different blocks.

Also, as we are evaluating the performance of the proposed system model for both AWGN and fading channels so, there are some constraints related to the fading channel. We have considered a slow fading channel for both Rayleigh flat and frequency selective channels. So the coherence time T_c, was considered to be relatively large (i.e. slow fading). The P_{Block} is the probability of erasure refers to the block error probability. The average block error probability P_{Block} for a block of B decoded bits and for a given fading vector can be upper bounded as [11]

$$P_{Block} \leq 1 - \int_{\alpha} (1 - P_E(\alpha))^B f(\alpha) d\alpha \tag{5.2}$$

Where $P_E(\alpha)$ is the error event probability defined as

$$P_E(\alpha) \leq \sum_{d=df}^{\infty} a(d) P_2(d|\alpha) \tag{5.3}$$

Where $P_2(d|\alpha)$ is the conditional pair-wise error probability (PEP) and is defined as $P_2(d|\alpha)=Q(\sqrt{2d\alpha})$ and $a(d)$ is the number of error events with distance d.

The receiver collects blocks with certain channel states until sufficient information is available to decode the successfully the specific information. For successful decoding, the receiver needs at least a minimum number of block n_{min} with certain channel state. In other words, the maximum achievable rate $R_{max} = k/ n_{min}$. We evaluate the performance of the proposed scheme using an average number of blocks n, thus the maximum achievable rate can be given as $\underline{R}_{max} = k/(B.\underline{n}_{min})$.

5.3.2 Rayleigh Fading Channels

The channel is assumed to be a slow fading channel for both flat fading and frequency selective fading. Clarke's model [27, 28] is used for generating slow fading channel in the simulation environment. The model consider no line of sight environment between the transmitter and receiver and the signals are reflected and scattered due to different kind of obstructions occurred in its path. Also the model considers the Doppler Effect on the wave propagation due to the motion. Usually a random number is used to produce two independent Gaussian noise baseband line spectrum with a maximum frequency of f_m, the Doppler shifted frequency. The negative frequency components are obtained by conjugating the positive frequency components and then the random spectrum is multiplied by the discrete frequency representations $\sqrt{S_{E_z}(f)}$. Where $S_{E_z}(f)$ is the power spectral density for a quarter wavelength antenna and $p(\alpha)$ uniform over $(0, 2\pi)$ and is given by:

$$S_{E_z}(f) = \frac{1.5}{\pi f_m \sqrt{1 - \left(\frac{f - f_c}{f_m} \right)^2}} \tag{5.4}$$

Table 5.1 Primitive Elements for Block Codes

	n	***k***	***t***
BCH	255	223	4
	1023	923	10
RS	255	223	4
	1023	923	10

Where f_c and f_m are the carrier frequency and maximum Doppler shift respectively and $p(\alpha)$ is a fraction of the total incoming power. The purpose of using power spectral density is to shape the random signals in the frequency domain. Finally an IFFT is performed on the resulting frequency domain signals from the in-phase and quadrature arms to get two N-length time series and add the squares of each signal point in time to create an N-point time series. The square root of the N-point time series is the simulated Rayleigh fading signal [29].

5.4 Simulation Parameters & Results

To evaluate the performance of the investigated system, we consider the carrier frequency is 1.8*GHz* and a bandwidth of each channel is 200 *KHz*. To ensure that there is a slow fading channel, we consider $Tc >> Ts$, i.e., coherence time Tc is greater than the symbol period Ts. The essential condition for flat fading is that, the symbol period should be greater than the delay spread i.e. $Ts >> \sigma_\tau$ and for frequency selective fading channel $Ts << \sigma_\tau$ are considered. The velocity for the Doppler shift is considered as 120 *km/hr*. The packet length is considered according to the output of the block coding scheme as mentioned in table 1. The degree distribution used in the LT codes have parameters $\delta = 0.999$ *and* $c = 0.03$. A maximum of ten iterations are used in the decoder to recover the original information.

The simulation results of the system model are presented in Figure 2, 3 and 4. Figure 2 shows the simulation results of the AWGN channel for maximum achievable rate versus SNR. Conventional LT Codes are compared with the concatenated fountain codes using different packet sizes and coding rates. We used the packet size according to the output of the block coding scheme. The short packet length for transmission over the fading channel is 255 regardless of the coding rates used by the LT codes, while the longest packet length is given by 1023.

Three different variations of results are compared. The conventional LT code with smaller and long packet size shows the least performance compare to the concatenate fountain codes. The performance of the LT codes with block coding with long packet length significantly performs better when a coded LT codes are used. Note that we use a coding rate *R=1/2* for coded LT codes. Also the performance of the LT Codes with block code with un-coded LT code for same packet length constraint is also carried out, although its performance is better than the conventional LT codes but is significantly away from the coded LT codes. It is also observed that the performance of BCH and RS codes with LT codes (Concatenated Fountain Codes) is comparably same.

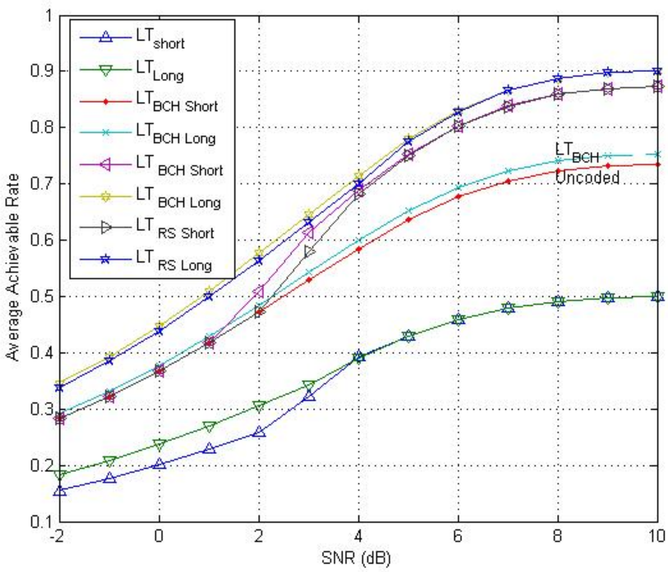

Figure 5.2 Maximum Achievable Rate vs. SNR(dB) over AWGN Channel

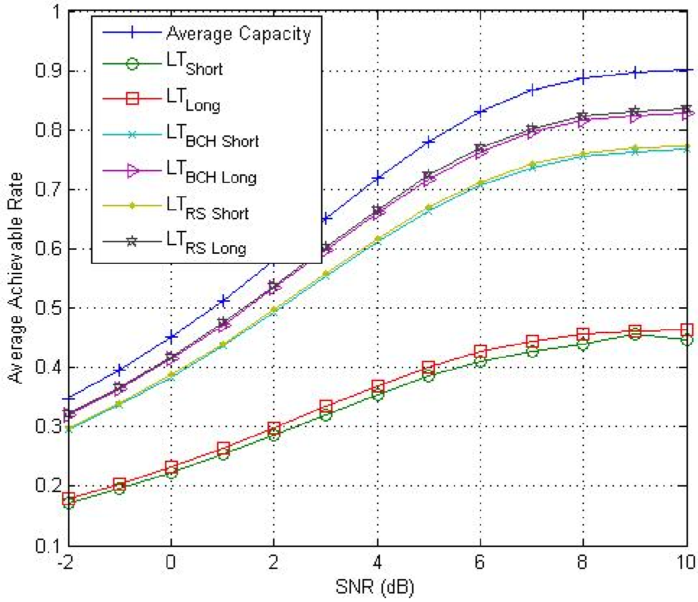

Figure 5.3 Maximum Achievable Rate vs. SNR(dB) over Flat Fading Channel

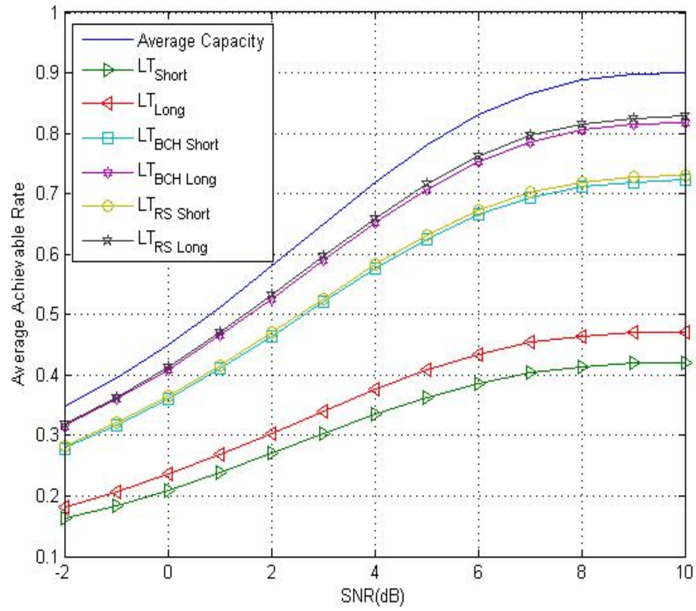

Figure 5.4 Maximum Achievable Rate vs. SNR(dB) over Frequency Selective Fading Channel

Although, the effect of different size of the packet is different. We can conclude that, with an increase in the packet length and signal to noise ratio (SNR) the performance of the proposed system model increases.

The maximum achievable rate versus SNR is also presented for Rayleigh flat fading and frequency selective fading channels in Figure 3 and Figure 4 respectively. The analysis for AWGN channel is also applicable for Rayleigh fading channels. It is obvious that because of multipath effect in the Rayleigh environment the performance of the system degrades. In the Rayleigh flat and frequency selective, the system performs better for concatenated LT codes with RS codes in long packet length scenario.

5.5 Conclusions

In this chapter, we investigated a system model of low complexity using concatenated fountain codes for wireless broadcast over the fading channels. We considered the slow fading channel for Rayleigh flat fading and frequency selective fading and AWGN channels in our simulation work. From our simulation work clearly when fountain codes are used with some concatenated block codes, there is a significant improvement in the performance. Note that around 10% additional symbols in each packet outclass the conventional system based on the LT codes.

Also we have selected different length of packets and observe that with an increase in the packet length and SNR, the performance of the system model increases significantly compare to those of smaller packet length. Finally, we can conclude that fountain code with some concatenated block codes are an ideal choice to be used in the wireless broadcast and multicast environment.

Chapter 6
Hardware Implementation of LT Codec

In this chapter we discuss the hardware architecture of LT encoding and decoding from the perspective of degree distribution function. Previously, there are many approaches which discussed the hardware architecture of LT Codec. But our approach is relatively a new approach for low power, efficient encoding and decoding of LT Codec. According to our limited knowledge from the literature, this is the first time an encoding and decoding approach is proposed for an LT codec in which the degree distribution is preserved and is used to exploit the benefits of the optimal degree distribution functions.

6.1 Introduction

In communication systems, some channel characteristics are usually not known as a priori or some may vary significantly during data transmission, such as the erasure probability in erasure channels, noise variance in Gaussian channels or the fading coefficients in the fading channels. Conventional solutions, such as data retransmission or changing the rate of block codes, are either wasteful or impractical. Therefore, an efficient technique is required to adjust adaptively the coding scheme depending on variable channel conditions. The rateless codes have no fixed rate assigned to them and thus they can solve this problem by adapting the output symbol length to the channel condition. There is always a need of some fixed specification for block length and code rate in practical scenario. However, a reconfigurable platform may solve the problem of fixed code rate and block length. In this thesis work, a fixed code rate and block length LT codec is proposed, which can uses the benefit of optimal degree distribution functions and also have a minimum number of iterations.

Luby Transform (LT) codes are the first and efficient class of rateless codes that have attracted a lot of attention in the channel coding schemes. The original data can be fully recovered from any set of received symbols with its length slightly larger than the original data. Furthermore, to achieve linear encoding and decoding time, Raptor codes were invented which concatenated the LT codes with a block code as their pre-codes [11]. Rateless codes, including LT codes and Raptor codes, show excellent performance in erasure channels. Research work also shows that their performances in other channels, such as Gaussian channels and fading channels are good [13, 25, 26]. As discussed in the previous chapters that these are non systematic codes and their crucial and performance oriented part is their degree distribution function. There is a high degree of randomness involve in the encoding and decoding of the LT codes. This randomness is measured randomness and which is usually obtained through a degree distribution function. Two different degree distribution functions are introduced by Luby in [10]. Like the low-density parity-check *(LDPC)* codes, LT codes can be effectively decoded using the iterative belief-propagation (*BP*) algorithm or *Log-BP* algorithm [30]. This similarity allows us to design LT decoder using similar architecture as that of LDPC decoder. The VLSI implementation of LT codes is very challenging due to its random code construction characteristic as well as the flexible output length.

Previously, some efficient LDPC decoders are presented; like in [31] fully parallel architectures are proposed. Although fully parallel architectures can obtain high throughput but they may have the complex interconnection between the check nodes units and variable node units. In [32]-[34] some partially parallel architecture are proposed which usually use some optimization techniques and is realized commonly in the research community. For example in [32], an overlapped message passing algorithm is proposed for quasi cyclic LDPC (QC-LDPC) codes to increase the utilization of the CNUs and VNUs by selecting a good column starting indices (CSI) and row starting indices (RSI). Some new schemes that reduce the number of messages to be ex changed between the CNUs and VNUs, and the multimode decoder design for

$$\begin{array}{c} \textit{Numbers of Nodes} \\ \textit{Number of Edges} \left\{ \begin{bmatrix} X_{1,1} & X_{1,2} & X_{1,3} & . & . & X_{1,j} \\ X_{2,1} & X_{2,2} & . & . & . & X_{2,j} \\ X_{3,1} & X_{3,2} & X_{3,3} & . & . & X_{3,j} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ X_{i,1} & X_{i,2} & X_{i,3} & . & . & X_{i,j} \end{bmatrix} \quad \begin{bmatrix} S_{1,j} \\ S_{2,j} \\ S_{3,j} \\ . \\ . \\ S_{i,j}. \end{bmatrix} \right\} \textit{Sum of Nodes/Edge} \\ \text{(a)} \qquad\qquad\qquad\qquad \text{(b)} \end{array}$$

Figure 6.1: (a) Edge Router (ER)   (b) Index table for Edge Router (IER).

various codeword lengths have been investigated in [33, 34].

In this chapter, a new hardware architecture based on finite state machine (FSM) to control the different computation operations in the LT codec is presented. The presented novel scheme offers to accommodate any optimal degree distribution functions. For partially parallel architecture, an internal memory is required that can store the extrinsic messages. The internal memory is divided into set of smaller memory blocks to increase the parallelism of data accesses and also helps in efficient usage of memory space. An efficient algorithm is used in which the memory access conflicts and their proper used in the operation of the LT codec. The proposed channel codec is configured for 128 bit input data with a code rate R = ½. It is notable that LT codes with an increasing code rate have better performance for different channel conditions. The three main entities in the architecture of LT Codec includes: Degree Distribution function, LT Encoder and LT Soft Decoder and are presented in detail.

6.2 Degree Distribution

The degree distribution functions for LT codes are first presented by Luby in [10]. Also, optimal degree distribution function and its performance have been evaluated in great detail in the previous chapters. In this section, we will discuss the degree distribution function from hardware perspective.

Consider the Robust Soliton distribution (RSD) function, which is usually a common choice for a degree distribution in the LT codes. From the RSD function, evidently it uses some complicated computational functions such as divisions and logarithmic functions which are usually expensive to implement directly in the hardware. Previously, some approaches are used to approximate the RSD and its implementation is carried out on the hardware. Like in [36], the some degrees according to intermediate symbols are mapped in the read-only memory (ROM) and random number generator is used to access the memory randomly. Degree distribution function provides the connecting information between the information bits k and the encoded bits K. It is the core component that generates the tanner graph and controls the computational complexity in the LT codes. Efficient degree distribution function can yield significantly better results and which can then adds in the performance of the Raptor codes. The degree distribution can be implied by some random number generator in the hardware with the degrees in the Lookup tables. These random number generators consist of Linear Feedback Shift Registers (LFSR). Different LFSR and their performance with degree distribution function are evaluated in [35]. Note that LFSR design adds additional computation and complexity in the in the LT Codec architecture. In this work degree distribution functions are preprocessed in the software and are then mapped directly in the hardware. Note that the degree distribution functions used here are intermediate degree values, as the degree distribution function is a probability distribution function and thus it is well defined for intermediate values. Figure 6.2 shows the framework of LT encoding.

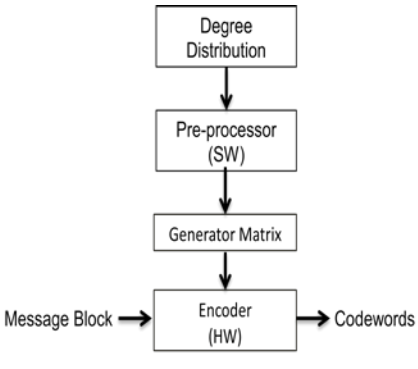

Figure 6.2: LT Encoding Framework

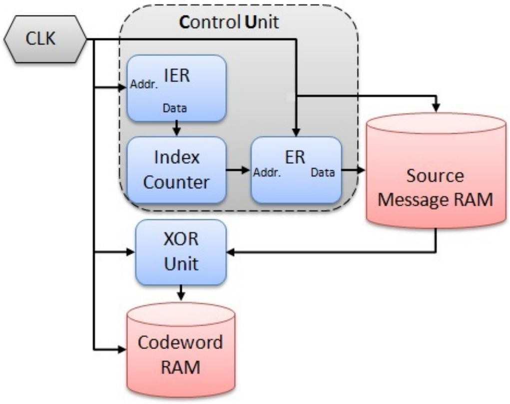

Figure 6.3: Block Diagram of LT Encoder

Two sparse matrices that are generated according to degree distribution functions are used in the encoding and decoding processes respectively. These two matrices are mapped in the hardware and are termed Edge Router (ER) and Node Router (NR). The ER contains the information about the nodes connected to each edge, where as NR is reverse of the ER and contains the information of number of edges connected to each node. Two additional index tables are used that provides the information about the number of total nodes or edges for each ER and NR respectively. Note that these index tables are solely used for efficient controlling of the ER and NR. Figure 6.1 shows the Edge Router (ER) and Index table for Edge Router (IER).

6.3 LT Encoder Architecture

The LT encoding procedure starts sequentially, and selects the respective message nodes symbols according to the ER from the message memory. And place the corresponding symbols in the codeword register after implementing the exclusive-or operation according to LT encoding procedure. Although some latency will be involved in the processing of the message bits for its desired codeword according to the degree distribution function.

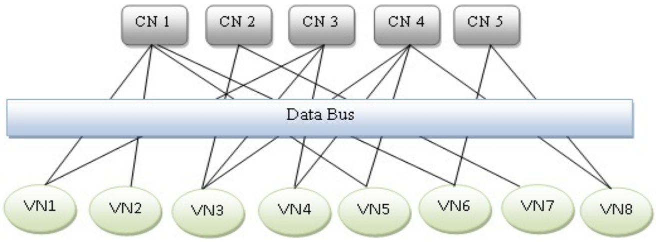

Figure 6.4: Factor graph representation of the LT codes

6.4 LT Decoder Architecture

Conventional Belief Propagation (BP) algorithm is used for decoding the encoded symbols in the LT codes by performing the sum-product message updates on the tanner graph generated using the degree distribution function of the LT codes as shown in Figure 6.4. The performance of BP algorithm is evident from their extensive used in the LDPC codes.

The algorithm is accomplish by passing message from check node *i* to variable node *j*. Initially, we define the channel soft information message received from the channel with each code bit by $2r_i/\delta^2$, where δ^2 is the noise variance of the channel and r_i is the received values. Let $L(r_i)$ represents the soft information on the received values of corresponding encoded symbols *r* from the channel. The iterations "it", according to BP algorithm then proceeds as:

$$L(t_{i,j}) = \tanh^{-1}(\tanh\frac{L(r_i)}{2}.\prod_{n\neq j}\tanh\frac{L(h_{n,i})}{2}) \tag{6.1}$$

$$L(h_{i,j}) = \begin{cases} 0 & , \ it = 0 \\ \sum_{e\neq j} L(t_{e,i}) & , \ it \geq 1 \end{cases} \tag{6.2}$$

$$L(u_i) = \sum L(t_{e,i}) \tag{6.3}$$

$L(t_{i,j})$ and $L(h_{i,j})$ are the L-value message passed from check node *i* to variable node j and variable node *i* to check node *j*, respectively as shown in equation (6.1) and (6.2). After some certain iteration, hard decision decoding is performed on the soft information $L(u_i)$ collected from the check nodes and variable nodes as shown in equation (6.3).

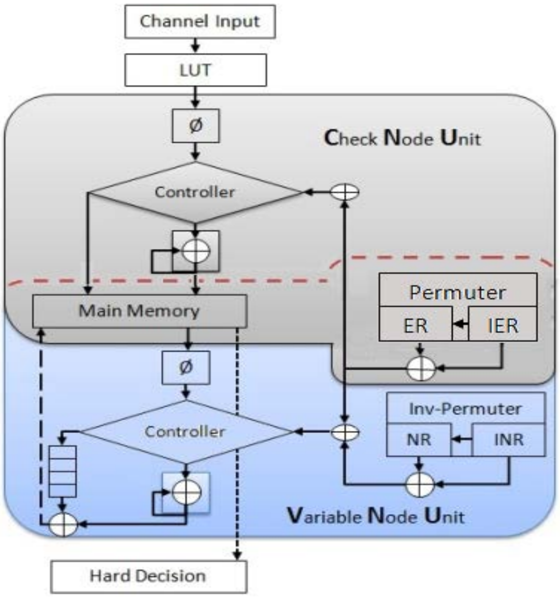

Fig.6.5. Serial Architecture of the (256,128) LT decoder.

To illustrate the hardware design of the decoder, irregular matrices formed by the degree distribution functions are used. These directly mapped matrices in the hardware work as parity check matrices for (256,128) LT codes and are accessed incrementally according to the soft decoding procedure of LT codes. A controller is used that controls the Permuter and Inv-Permuter function which consists of nodes and edges information provided by the degree distribution function. Figure 6.5 illustrates the architecture of the LT Code sum-product decoder. Two sets of memories are used in the mentioned main memory shown in the Figure 6.5. Each memory stores the messages from check to variable and variable to check nodes. Note that same parity check matrices are used simultaneously between the check node and variable node unit. In a check to variable node operation defined in equation (6.1), if the degree of that particular node is one, then the LLR according to the soft information is calculated and stores in the memory associated with the check node unit. Otherwise the marginalized values are calculated from the updated variable node and are then stored in the check node memory. In variable to check operation defined in equation (6.2), the variable node inside every processing unit accumulates check to variable messages serially. The sum is then stored in the particular memory associated with the variable node. The processing between the check node unit and variable node unit is executed for some particular number of iterations to recover the information successfully.

6.5 Simulation Results

To evaluate and compare the performance of optimal degree distribution functions, we implement a rate ½ 128 bits LT codes in Samsung 0.13um, 1.2v customized CMOS standard cell library with 6-layer metals. The Design Vision and Astro EDA tools are used to synthesize the logic components, place and route for the chip design.

It is observed that with an optimal degree distribution function in the hardware, the number of clock cycles per iteration in the LT decoder has been reduce by 16% compared with that of a conventional degree distribution function. The different parameters considered for the performance evaluation of the LT codes are mentioned in the Table 6.1. The layout of the LT codes for Robust Degree Distribution function is shown in the Figure 6.6. Also the layout of the LT codes for optimal degree distribution function is also demonstrated in Figure 6.7. It is observe that optimal degree distribution not only helps in reducing the computational cost of the hardware implementation but it also helps and enhances other essential features of the hardware implementation. It is clear that, the hardware implementation of the optimal degree distribution function occupies 16% less area for same dimension with better performance in

Table 6.1**.** Performance Comparison of LT Codes for Degree Distribution Functions

	Conventional Degree Distribution (Robust Soliton Distribution)	**Optimized Degree Distribution (Power Distribution)**	**% Difference**
Core Size(mm^2)	2.201	1.831	16.81
Chip Size(mm^2)	4.768	4.216	11.57
Cell Area(mm^2)	1.49	1.26	15.43
Throughput(Mbps)	18	22	22
Area (mm^2)	0.3358	0.2798	16.67
Net Interconnect Area(um^2)	3457.83	2839.30	17.89
Wire Length(mm)	7.7917	6.3576	18.4
Number of Nets	101565	87979	13.37
Power(uW)	830.6317	587.03	29.327

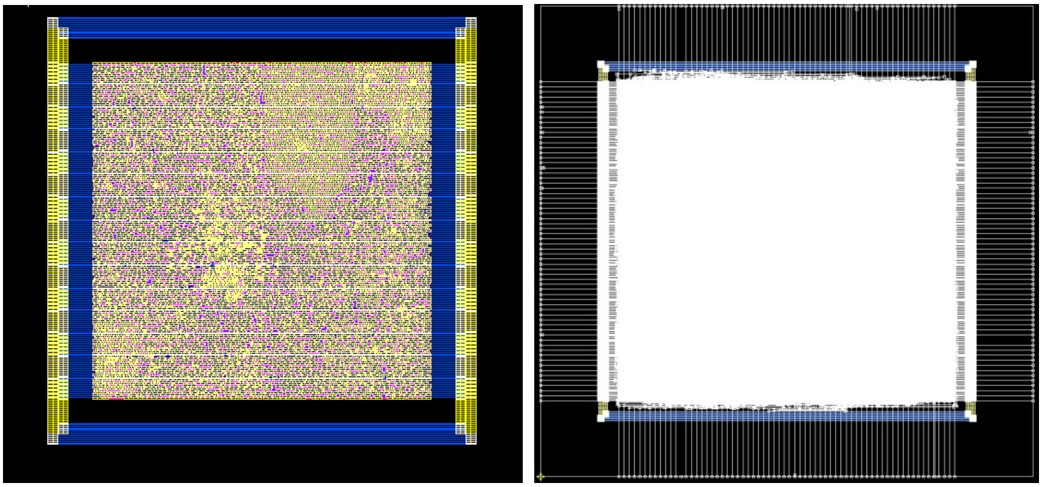

Figure 6.6 Layout of the LT codec for RSD Degree Distribution

terms of overhead. A throughput difference of 22% has been observed with a frequency of 300MHz for optimal and conventional degree distribution function with an added advantage of less chip size and core area for proposed scheme. The computational complexity of the hardware can be observed from their net interconnections; the proposed degree distribution function significantly outclasses the conventional degree distribution function in this area. Net interconnect area of 17.8%, wire length of 13.37% and with a decrease of 13.37% for number of nets have been observed for the optimal degree distribution function. This reduction in network complexity and area helps in reducing the power to almost 30% for the hardware.

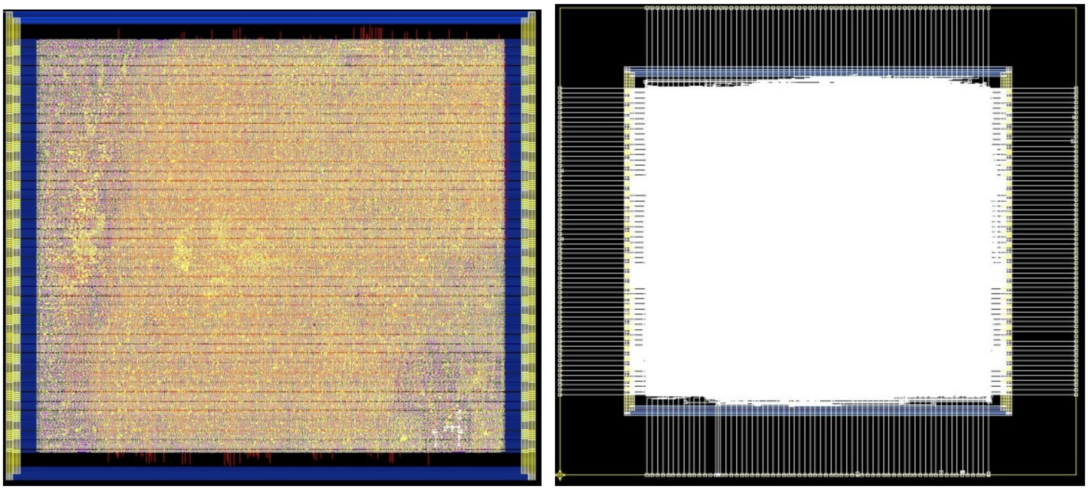

Figure 6.7 Layout of the LT codec for Optimal Degree Distribution

6.6 Conclusion

In this chapter, we have presented a new approach for the LT codec architecture. The proposed architecture uses directly mapped information of the degree distribution which is an essential part of the LT codec and provides the connecting information between the nodes and edges of the tanner graph. Also an optimized degree distribution proposed in the previous chapter is used to evaluate its impact in the hardware implementation. From the hardware implementation of the LT codes with conventional and optimized degree distribution function yields significant results of research interests. We observe that the optimized degree distribution function help in reducing the computation complexity in the hardware which usually appears in terms of nets connection. The reduction in computational complexity has impact on the size of the hardware and helps in increasing the throughput of the hardware. Note that we have evaluated the performance of the degree distribution for only 128 bits and its impact will be greater for higher number of input bits.

Chapter 7
Conclusion

This chapter concludes the work presented in this thesis work. Some discussion about the simulation work carried out in this thesis and some future issue that can help in improving the coding scheme and their application areas are suggested.

7.1 Summary

An Optimization work has been presented for degree distribution function of the LT codes. As we have seen, LT codes were categorized into two different categories: Scale Free LT codes and Random LT codes, and the basis of this classification were degree distribution function. It was also noticed that in Figure 3.3 as the degree value increases, the release degree becomes a much faster increasing function of degree distribution and that can also aid in the optimization of the degree distribution function. Two new degree distribution functions which follow the Power Law are proposed in this work which includes: Pareto's distribution and Power distribution. It is observed that there is a decrease of 31% and 48% in the overhead when the LT codes are used for the proposed schemes using an ideal channel. Also an evaluation of the degree values for LT codes using different degree distribution is considered. From the simulation results, it is evident that the proposed scheme uses less degree values compare to that of conventional degree distribution and thus contributes in a reduction of computational complexity of the coding scheme. The proposed work can also aids in Raptor codes designs to customize their weakened LT code for the use of different pre-codes.

Also a system model based on a concatenated fountain codes are proposed in this work and their performance is evaluated for AWGN and Rayleigh flat and frequency selective channel. The evaluation of the system was carried out with smaller and longer packet length. It was observe that the concatenated fountain codes outclass the conventional LT codes for both AWGN and fading channels. Also, it is observed that, system model performs better for longer packet lengths.

Finally, a hardware implementation of LT codec is presented in which optimized degree distribution functions are evaluated in the hardware. From the simulation results and hardware implementation of the LT codes, it is observed that the degree distribution function also have a greater impact in the hardware and it essentially adds in the different features of the hardware.

7.2 Suggestion and Future Work

Before this work, there was no such platform for future researchers to improve the degree distribution function of the LT codes and usually the researcher were focusing on heuristically work and interestingly they were getting better results. There is a need for a platform, such that in future work, different researchers can optimize the degree distribution in some specific way. In this thesis work, an effort has been made to create such a platform by categorizing the LT codes in two different categories that provides an opportunity to future researcher to focus and utilize the degree distribution function and generate the LT codes according to their requirements.

The evaluation of LT codes over fading channel shows that, they have poor performance and thus a concatenated version of the LT codes should be used for wireless channels. We have described a simple broadcast wireless model for concatenated fountain codes which have significantly better performance for wireless channels. In such channel model an optimal degree distribution can also play its role by removing the unnecessary symbols for an efficient LT codes. Also, in hardware implementation, the previously defined block coding schemes can be used with efficient hardware implementation for practical wireless communication scenarios.

The hardware implementation of LT codes is of greater significance. In our work, we have proposed a new way in which sparse matrix generated from the degree distribution function is mapped in the hardware and is used for an efficient hardware implementation. Such schemes are useful and can also be employed easily for irregular LDPC codes. Also, for heterogeneous environment, some sets of optimized degree distribution function can be used with some control logic that can exploit the random nature of the LT codes.

References

[1]. C. Shannon, “E.(1948) A Mathematical Theory of Communication,” Bell System Technical Journal, vol. 27, pp. 379–423.

[2]. I. Reed and G. Solomon, “Polynomial codes over certain finite fields,” Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, pp. 300–304, 1960.

[3]. R. Koetter and A. Vardy, “Algebraic soft-decision decoding of Reed Solomon codes,” Proceeding of the IEEE International Symposium on Information Theory, pp.61, 2000.

[4]. R. Gallager, “Low-density parity-check codes,” Information Theory, IRE Transactions on, vol. 8, no. 1, pp. 21–28, 1962.

[5]. W. E. Ryan, “An Introduction to LDPC Codes,” in CRC Handbook for Coding and Signal Processing for Recording systems, CRC Press, 2004.

[6]. D. Mackay, R. Neal,“Near Shannon Limit Performance of Low Density Parity Check Codes”, Electronics Letters, Volume 32, p. 1645, Aug. 1996.

[7]. R. M. Tanner, “A Recursive Approach to Low Complexity Code”, IEEE Transactions in Information Theory”, pp. 533-547, Sept 1981.

[8]. J. Byers, M. Luby, and M. Mitzenmacher, “A digital fountain approach to asynchronous reliable multicast,” IEEE Journal on Selected Areas in Communications, vol.20, pp.1528–1540, Oct.2002.

[9]. W. Byers, M. Luby, M. Mitzenmacher and A. Rege, “A digital fountain approach to reliable distribution of bulk data”, In Proceedings of the ACM SIGCOMM’98 Conference on Applications, technologies, architectures, and protocols for computer communication, pp. 56–67, ACM, 1998.

[10 M. Luby, “LT codes,” in Proceeding of 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[11] A. Shokrollahi, "Raptor codes," IEEE Transactions on Information Theory, Volume.52, no.6, pp. 2551-2567, June 2006.

[12]. R. Karp, M. Luby and A. Shokrollahi, "Finite length analysis of LT-codes," in IEEE International Symposium on Information Theory, pp. 39–39, 2004.

[13]. T. Stockhammer, H. Jenkac, T. Mayer and W. Xu, "Soft Decoding of LT Codes for Wireless Broadcast", Proceeding in IST Mobile, Dresden, Germany, pp. 262-264, Jun 2005.

[14]. M. Luby, M. Mitzenmacher, A. Shokrollahi and D. Spielman, "Efficient Erasure Correcting Codes", IEEE Transactions on Information Theory, Volume. 47, pp. 569-584, Feb 2001.

[15]. Technical Specification Group Services and System Aspects, Third Generation Partnership (3GPP) Multimedia Broadcasting/Multicasting Services (MBMS); Protocols and Codecs (Release 7), standard, June 2008.

[16]. ETSI, "Digital Video Broadcasting (DVB); IP Datacast over DVB-H; Content Delivery Protocols", TS 102.472, Dec, 2006.

[17]. E. Hyytiä, T. Tirronen, and J. Virtamo, "Optimizing the Degree Distribution of LT Codes with an Importance Sampling Approach," 6th International Workshop on Rare Event Simulation. Bamberg, Germany, 2006.

[18]. H. Zhu, G. Zhang and G. Li, "A Novel Degree Distribution Algorithm of LT Codes," in 11th IEEE International Conference on Communication Technology, 2008.

[19]. J. H. Sørensen, P. Popovski and J. Østergaard, "Design and Analysis of LT Codes with Decreasing Ripple Size," IEEE Transactions on Communications, 2010.

[20]. H. Tarus, J. Bush, J. Irvine and J. Dunlop, "Exploiting Redundancies to Improve Performance of LT Decoding ", in proceeding of the 6th Annual Conference on Communication Networks and Services Research (CNSR 2008),2008, pp.568-573.

[21]. C. M. Chen, Y. P. Chen, T. C. Shen and J. K. Zao, "Optimizing the Degree Distributions in LT codes by using the Multi-objective Evolutionary Algorithm Based on Decomposition," in proceeding of the IEEE Congress on Evolutionary Computation,2010, pp. 3635-3642.

[22]. A. L. Barabasi, Z. Dezso, E. Ravasz, S. H. Yook and Z. Oltvai, “ Scale Free and Hierarchical Structures in Complex Networks”, in proceedings on Complex Networks,2003.

[23]. M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law". Contemporary Physics Vol. 46 (5): 323–351, 2005.

[24]. F. Uyeda, H. Xia and A.A. Chen, "Evaluation of a High Performance Erasure Code Implementation”, Technical Report, University of California, San Diego, 2004.

[25]. R. Palanki and J. S. Yedidia, "Rateless codes on Noisy Channels," ISIT 2004, Proceeding in IEEE International Symposium on Information Theory, pp. 37, Jun. 2004.

[26]. X. Wang, W. Chen and Z. Cao, "Throughput Efficient Rateless Coding with Packet Length Optimization for Practical Wireless Communication System", Proceeding in IEEE Globecom, pp. 1-5, 2010.

[27]. S. Rappaport, “Wireless Communications: Principles and Practice”, Second Edition, 2002.

[28]. J. I. Smith, “A Computer Generated Multipath Fading Simulation for Mobil Radio”, IEEE Transactions on Vehicular Technology, vol. 24, no. 3, pp. 39-40, August 1975.

[29]. R. H. Clarke, “A Statistical Theory of Mobile-Radio Reception,” The Bell Systems Technical Journal, vol. 47, no. 6, pp. 957-1000, July-August 1968.

[30]. J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” IEEE Transactions on Information Theory, vol. 42, pp. 429–445, Mar. 1996.

[31]. A J. Blanksby and C. J. Howland, “ A 690-mW 1-Gb/s 1024b, Rate ½ Low Density Parity Check Code Decoder,” IEEE Journal of Solid-State Circuits, vol. 37, no. 3, pp. 404-412, March 2002.

[32]. Y. Cheng and K. K. Parhi, “Overlapped Message Passing For Quasi-Cyclic Low Density Parity Check Codes,” IEEE Transactions on Circuits and Systems 1, Regular Papers, vol. 51, no 6, pp. 1106-1113, June 2004.

[33]. S. H. Kang, and I. C Park, "Loosely Coupled Memory Based Decoding Architectures for Low Density Parity Check Codes," IEEE Transactions on Circuits and Systems 1, Regular Papers, vol. 53, no 5, pp. 1045-1056, May 2006.

[34]. X. Y. Shih, C. Z. Zhan, and A. Y. Wu, "An 8.29 mm^2 52 mW Multi-Mode LDPC Decoder Design for Mobile Wi-MAX System in 0.13μm CMOS Process," IEEE Journal of Solid- State Circuits, vol. 43, no. 3, pp. 672-683, March 2008.

[35]. H. Wang, "Hardware Design for LT decoding", MSc. Thesis, Delft University of Technology, 2006.

[36]. K. Zhang, X. Huang and C. Shen, "Soft Decoder Architecture of LT codes," IEEE Conference Signal Processing System, pp.210-215, 2008.

ABSTRACT

An Efficient Implementation of LT Codes based on its Optimized Degree Distribution Function

Muhammad Asim
Advisor: Prof. GoangSeog Choi, Ph.D.
Department of Information and Communication Engineering
Graduate School of Chosun University

This thesis examines the problems associated to transfer of bulk amount of data in the networks from the perspective of channel coding. As channel coding is one paradigm in the communication channels, which help in reducing errors when information symbols are transmitted over communication channels. Recently, some new channel coding schemes are employed for erasure channels, these includes: LDPC codes and Fountain codes. The focus of this thesis is the codes that approximate the fountain codes.

Previously, there were some schemes that approximate the Fountain codes concept, the famous among which includes: Reed Solomon and LDPC codes. Few years back a new coding scheme is proposed known as LT codes that approximates the fountain codes and have the benefit of robust and efficient encoding and decoding algorithms. Also these are the first rateless coding scheme and its importance is evident from their adoption in the communication standards. In this thesis work, we optimized one of the crucial and performance oriented part of these codes known as degree distribution functions. A well defined degree distribution function in the LT codes increases its

performance to many folds. We have presented a new analysis for optimization of the degree distribution function, and shown that the previously defined degree distribution functions also follow our analysis. Based on the analysis, we have categorized the LT codes in to two categories that include: Free Scale LT codes and Random LT codes. Both the categories of the codes are associated with Power Law. We have proposed two new degree distributions for Free Scale LT codes that significantly enhance the performance of these codes.

We have also presented an evaluation work that evaluates the performance of the LT codes over fading channels. An enhance version of the LT codes also known as Concatenated Fountain Codes (CFC) are proposed and evaluated. The performance of the conventional LT codes and CFC are also compared.

Finally, hardware architecture for LT Codec is proposed in which the connection information between the nodes and edges are directly mapped on the hardware instead using the degree distribution function in hardware. This approach helps in utilizing the different optimal degree distribution functions for LT codes. The impacts of optimal degree distribution function on different features of the hardware are considered.

Acknowledgments

I would like to begin by expressing my sincere gratitude towards my advisor and mentor, Professor GoangSeog Choi for his valuable guidance and advice throughout my research at System On Chip (SoC) Design Lab and especially for the opportunity to pursue research under his tutelage. His enthusiasm, discussions, and guidance were of greater help in the realization of this thesis. I am also thankful to all my faculty members of Department of Information and Communication Engineering, who are always supportive and helpful in understanding and enhancing my knowledge. I am forever indebted to my lab colleagues especially Kang Gyungsik, Kim Ilyoung and Jung Joo Koo for their support and guidance in different matters throughout my stay at SoC design Lab.

I am also thankful to all my family members for their utmost support, encouragement and guidance. Finally, I would like to thank the National IT Promotion Agency (NIPA) for their Global Talent Scholarship and grants for my research activity and other living expenditure.

Muhammad Asim