

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

可分離之非均等錯誤保護 LT 編碼

Research On Separable UEP-LT Code

學生：顏國光

指導教授：張錫嘉 助理教授

中華民國九十六年九月

可分離之非均等錯誤保護 LT 編碼

Research On Separable UEP-LT Code

研究生：顏國光

Student: Guo-Guang Yan

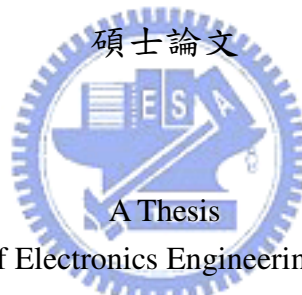
指導教授：張錫嘉

Advisor: Hsie-Chia Chang

國立交通大學

電子工程學系

電子研究所碩士班



Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

In partial Fulfillment of the Requirements

for the Degree of Master of Science

In

Electronics Engineering

September 2007

Hsinchu, Taiwan, Republic of Chian

中華民國 九十六 年 九月

可分離之非均等錯誤保護 LT 編碼


學生：顏國光

指導教授：張錫嘉

國立交通大學

電子工程學系 電子研究所碩士班

摘要

The logo of National Central University (NCU) is a circular emblem. It features a central figure of a person holding a torch, surrounded by the university's name in Chinese and English. The year '1896' is inscribed at the bottom of the emblem.

由於 RS 編碼複雜度與資料量呈平方倍的關係，相較之下 LT 碼擁有無比例碼的特性及線性編解碼複雜度，特別適合應用在無線網路的多重播送和廣播；此外，對於多媒體傳輸，由於資料擁有不同重要性，必須給予不同比例的保護，因此我們提出將 LT 碼結合非均等錯誤保護的功能，稱之為可分離之非均等錯誤保護 LT 編碼，根據不同資料所要求的錯誤保護率去調整編碼的連結關係，模擬結果顯示，我們所提出的方法只需改變編碼過程，且在不增加編解碼複雜度的情況下，達到比先前被提出的非均等錯誤保護 LT 編碼更好的結果。

Research On Separable UEP-LT Code

Student: Kuo-Kuang Yeng

Advisor: Dr.Hsie-Chia Chang

Department of Electronics Engineering

Institute of Electronics

National Chiao Tung University

ABSTRACT

With the properties of rateless codes and the complexity of encoding and decoding almost linear to input symbol number k , LT code is especially suitable for multicast and broadcast in cellular network since the complexity of RS code is quadratic to k . For multimedia application, data with different importance must be unequally protected which is named UEP (Unequal Error Protection). As a result, we propose a method that combines UEP and LT code called separable UEP-LT code. According to required error probabilities of data with different importance, we can adjust connections between codeword packets and input symbols. Simulation results show that comparing to traditional LT code, with packet loss rates of most important symbols 10^{-6} and less important symbols 10^{-2} lower than the UEP-LT code, we only have to change encoding process but keep the same decoding process as traditional one. Note that, no more complexities are increased in both encoder and decoder.

謝誌

本論文的完成，最感謝張錫嘉教授的用心指導及修改，其亦師亦友的領導方式，給我學業上很大的助力及發揮空間，生活上倍感溫馨；感謝邵家健老師、彭文孝老師、王忠炫老師在口試時給我的指導與建議，讓我的論文更加完備，其中邵老師詳盡的意見，更補足我思慮不周之處。

除了老師之外，朋友們是我精神上很大的支柱，當我遇到挫折時有人鼓勵，也和我分享喜悅；修齊、博元、佳瑋、義凱、俊閔、Martin、冷王、企鵝、永裕、阿樸、SPICE、俊男、俊誼、晉欽、建君、宜欣、小VAN、篤雄、偉磬、小龍、JUJU、喬凱有你們真好，不論將來各自如何發展，我們的友誼長存。

最後，我想將完成碩士學位的光榮獻給我的父母，感謝他們的教誨及信任，無悔的付出讓我沒有後顧之憂，盡全力發揮。

國光

謹誌於 新竹

2007 九月

CONTENTS

CHAPTER 1 INTRODUCTION.....	1
CHAPTER 2 LT CODES.....	6
2.1 RATELESS CODES	8
2.2 DEGREE DISTRIBUTION.....	10
2.2.1. <i>DERIVE OF DEGREE DISTRIBUTION.....</i>	<i>11</i>
2.2.2. <i>THE IDEAL SOLITON DISTRIBUTION.....</i>	<i>13</i>
2.2.3. <i>THE ROBUST SOLITON DISTRIBUTION</i>	<i>13</i>
2.3 LT CODES.....	14
2.3.1. <i>THEORETICAL LT DECODING</i>	<i>19</i>
2.3.2. <i>REGULAR LT DECODING.....</i>	<i>25</i>
2.3.3. <i>DECODING LATENCY.....</i>	<i>29</i>
2.4 SIMULATION RESULT	30
CHAPTER 3 UEP-LT CODES.....	32
3.1 UNEQUAL ERROR PROTECTION	32
3.2 THE UEP-LT CODE.....	33
3.3 PROPOSED SEPARABLE UEP-LT CODE.....	36
3.4 UNEQUAL RECOVERY TIME AND DISTRIBUTED DECODING	43
CHAPTER 4 SIMULATION RESULTS	45
CHAPTER 5 CONCLUSION.....	47
BIOBIBLIOGRAPHY	48

List of Figures

Figure 1. With good error-correcting codes followed by Cyclic Redundancy Check (CRC), noisy channels can be considered as erasure channels.....	2
Figure 2. The disadvantages of Automatic Repeat reQuest (ARQ), traditional block code and the reason why unequal error protection is needed is depicted, too.	5
Figure 3. A example to show relationship between noisy channel and erasure channel	7
Figure 4. Input symbols versus codeword packets matrix of rateless codes through erasure channel.....	9
Figure 5. The decoding procedure of a codeword	11
Figure 6. LT-encoding process	15
Figure 7. The transmission of connection structure between input symbols and codewords by encapsulating degrees and codeword neighbors in packets	18
Figure 8. Encoder and decoder use the identical uniform random number generator to produce the same connection structure between input symbols and codewords	19
Figure 9. Theoretical LT decoding.....	25
Figure 10. Regular LT decoding process.....	28
Figure 11. decoding time (sec) versus data size (Mbytes).....	30
Figure 12. Simulation results of LT codes with $K=100$ and $K=1000$ respectively. Y-axis and X-axis stand for packet loss rate and codeword overhead $(1+\epsilon)$ respectively.....	31
Figure 13. The UEP-LT code structure	34
Figure 14. Separable UEP-LT code structure	37
Figure 15. Simulation result shows packet loss rate of MIS and LIS using the UEP-LT code and proposed separable UEP-LT code. Parameters of the UEP-LT code are $K=10000$, $\alpha=0.1$, $K_m=2$ and parameters of proposed separable UEP_LT code are $K=10000, \alpha=0.1$, $P_{MIS}=0.2$. Y-axis and X-axis stand for packet loss rate and codeword overhead $(1+\epsilon)$ respectively.	38
Figure 16. Histogram of robust soliton distribution	39
Figure 17. the relationship between parameter c and the ratio of packet operations.....	40

Figure 18. Ratio of decoding latency versus parameter c ($k = 1024$, symbol length = 1k bytes)42

Figure 19. Flow chart of designing a separable UEP-LT code.....43

Figure 20. Simulation result shows packet loss rate of MIS and LIS using the UEP-LT code and proposed separable UEP-LT codes with single degree distribution and separated degree distribution. Parameters of the UEP-LT code are $K=10000$, $\alpha=0.1$, $K_m=2$ and parameters of both proposed separable UEP_LT code with single degree distribution and separated degree distribution are $K=10000$, $\alpha=0.1$. Y-axis and X-axis stand for packet loss rate and codeword overhead $(1+\epsilon)$ respectively.46



CHAPTER 1

INTRODUCTION

Recently, wireless video and audio broadcasting services are getting popular. Moreover, Multimedia Broadcasting and Multicast Services (MBMS) [1] [2] [3] has been standardized and introduced into wireless cellular network. As a result of one-to-many services such as broadcasting and many-to-many services such as video conferencing, we must transmit data to every user through distinct channel which has its own error probability or may be time variant. The same situation is also encountered on internet when data are delivered from one point to many points. How to reliably deliver large files to many users through different channels and bandwidth-limited networks becomes a difficult problem. Since the most applications are IP-based, data are chopped into many packets before transmission. With good error-correcting codes such as LDPC codes, Turbo codes, Viterbi codes and so forth followed by Cyclic Redundancy Check (CRC), noisy channels can be considered as erasure channels in **Figure 1**.

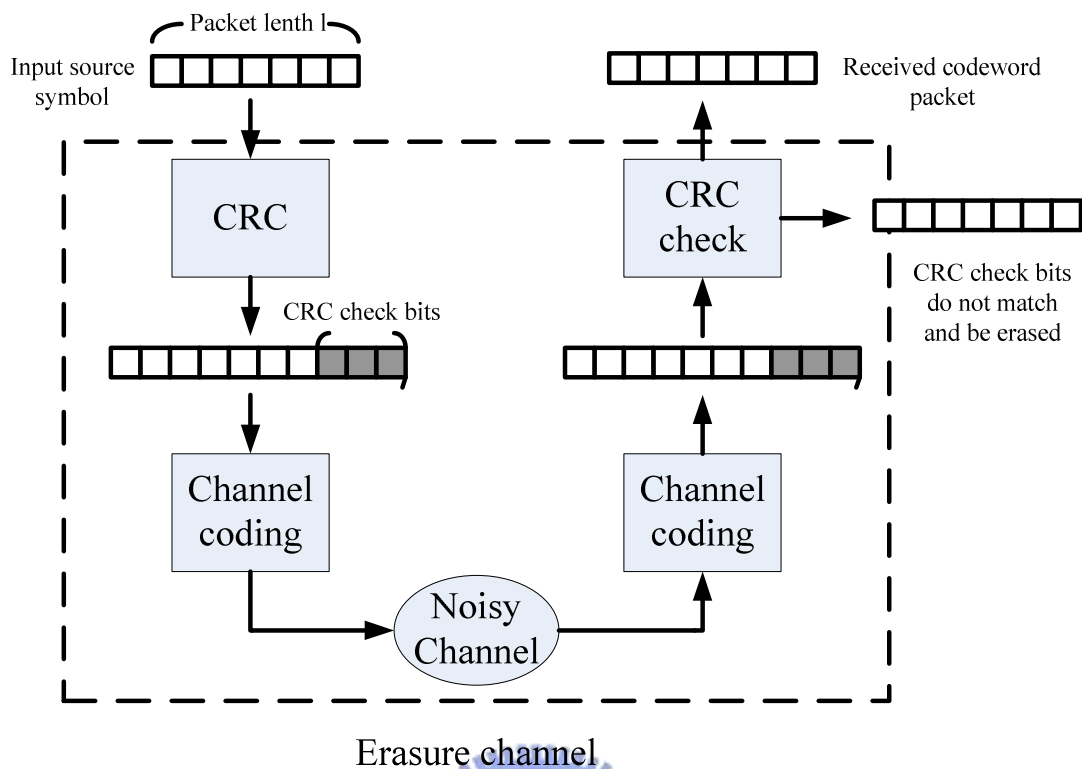


Figure 1. With good error-correcting codes followed by Cyclic Redundancy Check (CRC), noisy channels can be considered as erasure channels

When the decoder fails, it reports some packets are lost, otherwise, packets can be received correctly.

Traditionally, we can divide the transmission over erasure channel into two principle classes: Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC) schemes. ARQ scheme allows receivers use a feedback channel to send retransmission requests for lost packets. For example, the receiver might send back messages to inform transmitter which packets are lost and then retransmitted. The receiver might send back messages to acknowledge the received packets; the

transmitter keeps on transmit the following packets and retransmit those packets which are lost until they are acknowledged. ARQ scheme will not work well when feedback channel does not exist such as wireless network or transmitting data to many users that every user would request different retransmission, too many retransmission requests will jam the channels and makes data transmission impossible. In addition, according to Shannon, channel capacity does not change whether or not we have feedback and Reliable communication is possible when the transmission rate is under channel capacity which implies that.

So, FEC using erasure-correcting codes which require no feedback are better choice. The classic block codes for erasure channel are Reed-Solomon codes (RS codes) [4]. An (N,K) RS code (with packet length $q=2^l$) can recover K original source symbols if receiver received any K of the N transmitted symbols (For $N < q$ there exist RS codes). However, RS codes have high packet operation complexity of order $K(N-K)\log_2 N$. Its high packet operation complexity makes RS codes with large K inefficient which follows that RS codes are not on-the-fly and are not suitable for broadcasting. In addition to the high packet operation complexity of RS codes, there are more disadvantages of RS codes. An RS code, just like other block codes, must be designed according to the erasure probability $P(e)$ of channel which is estimated. If data is transmitted through different channels or a time variant channel, erasure probability f of channels is probably larger than expected one, RS decoder will introduce more errors into received packets, on the other hand, if erasure probability f of channels is smaller than expected one, though all the original source packets can be recovered, receiver received more redundancies than which are necessary. Another drawback of RS codes is that every receiver has to receive a copy of original data which is inefficient.

So far, we have considered transmitted data with equal importance. And the

corresponding technic used to protect the transmitted data is equal error protection (EEP) which ensures that every transmitted packet has the same probability to be recovered in receiver. Nevertheless, in many applications, different portion of data has different importance and requires distinct probability of recovery. For examples, in an video stream, I-frames need more protection than P-frames, when transmits photos, data could roughly divided into two parts one is header which contains more important information such as photo format, size, and so forth, another is values of pixels which need less protection. In some other applications, data may have different portion with different priority to be recovered such as video-on-demand services [5], in which the stream should be reconstructed in sequence. These applications need codes with unequal error protection (UEP) which supports different probability of recovery for data with distinct importance, or unequal recovery time (URT) which provides several recovery time priorities for different data portions. Prior descriptions are depicted in **Figure 2**. Recently, UEP codes have been designed with some block codes. But with the disadvantages of RS codes mentioned above, block codes with UEP or URT are not suitable for cellular network, accordingly, rateless codes with UEP or URT are more appropriate choices. Theoretical and simulation results also shows that rateless codes can satisfy the requirements of cellular network and achieve UEP or URT.

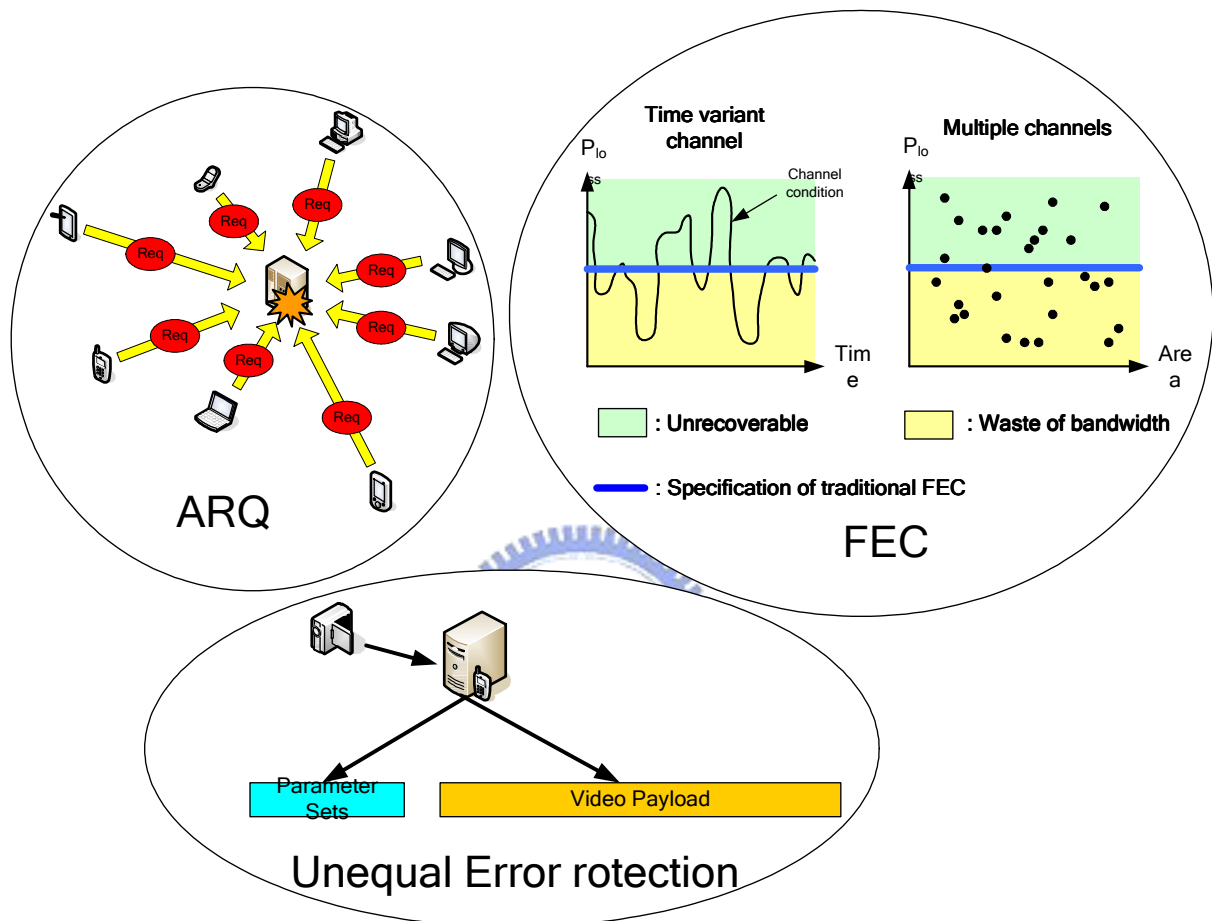


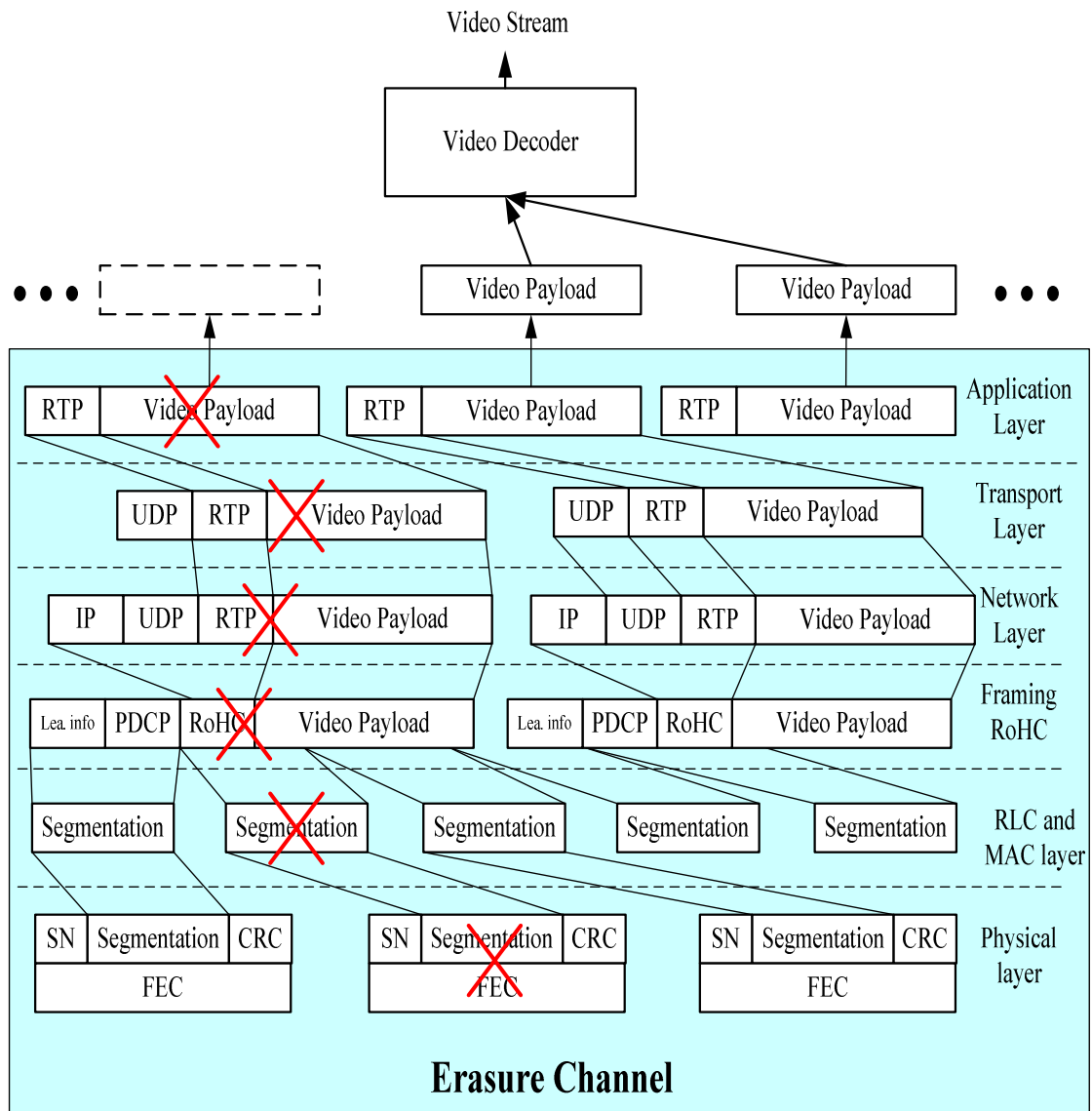
Figure 2. The disadvantages of Automatic Repeat reQuest (ARQ), traditional block code and the reason why unequal error protection is needed is depicted, too.

CHAPTER 2

LT CODES

In application layer, channel can be considered as erasure channel that the packets are received without error bits or totally lost as depicted in Figure 3. Many researches indicate that there are numerous advantages to apply forward error correction in the application layer. These approaches are especially suitable for multicasting and broadcasting. Because most of computers today have ability to operate instruction at the order of 10^9 per second and additional power consumption is much lower than the total cost of a web server. Additional instruction per communicated byte is allowed to be executed with a little CPU load and processing power. Thus Exclusive-or based FEC can easily be setup to the most modern CPU without affecting memory limitation. Moreover, those static services such as video on demand can pre-code and restore their data in advance therefore data can be transported without encoding delay. Besides, the use of FEC in application layer needs no changes in hardware. There are no extra processing added to routers and switches in network and only end clients are responsible for recovering input data from codewords. Also, it allows gradual incorporation into the network, by adding a possible negotiation option to the TCP connection and requiring no changes in network elements along the communication path between the two endpoints. Efficient and simple channel codes are able to be applied in other communication layer and the simple implementation in ether software or hardware is possible because of the

transparency of this change.

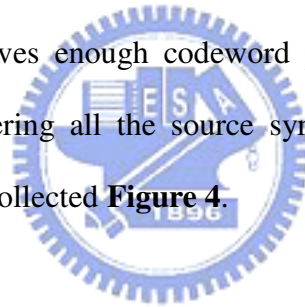


X : Erased

Figure 3. A example to show relationship between noisy channel and erasure channel

2.1 RATELESS CODES

Since cellular network data transmission is more and more important, traditional block codes are no more fitted for multi-channel, time variant channel and on-the-fly data receiving because we can not know situation of channels before or in data transmission. We need a new approach satisfied the requirements of cellular network. The essence of rateless codes [6] [7] [8] is as follows. The original source symbols are input into encoder and then a potentially infinite amount of codeword packets with the same size as source symbols are generated. Now, encoder acts as a fountain that produces infinite water drops, every drop represents an codeword packet. Every receiver, just like a bucket, collects drops (codeword symbols) from encoder until the bucket is full (encoder receives enough codeword packets for decoding). At last, decoder is capable of recovering all the source symbols completely regardless of which codeword packets are collected **Figure 4.**



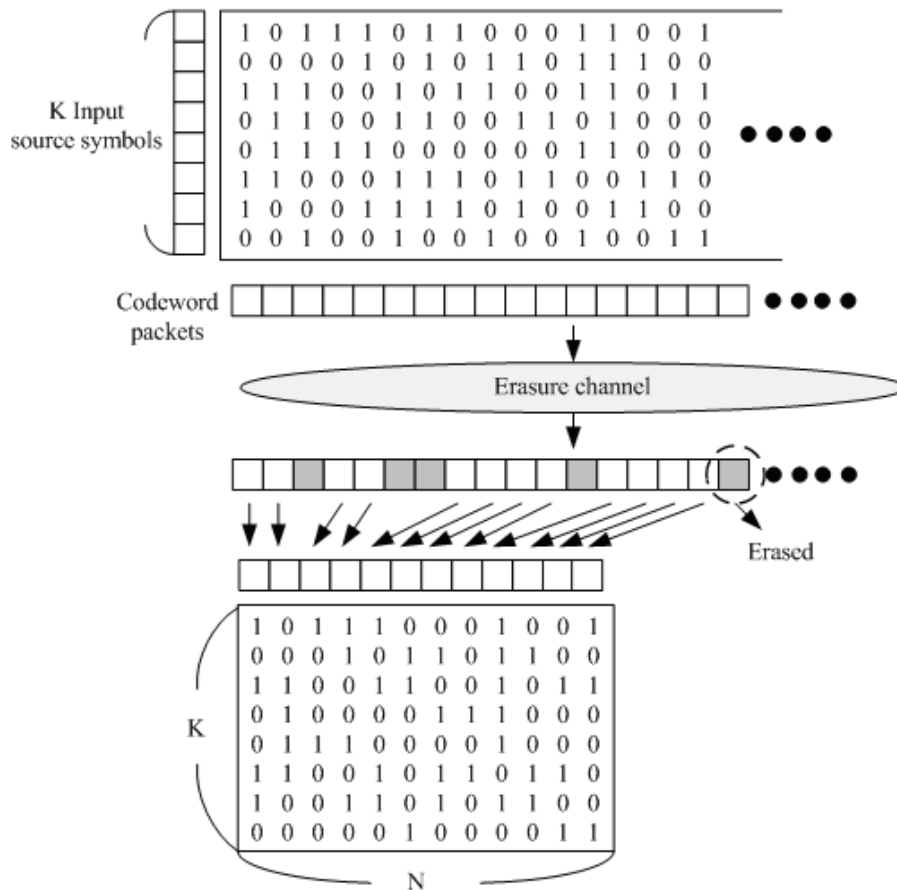


Figure 4. Input symbols versus codeword packets matrix of rateless codes through erasure channel

Properties of rateless codes are optimal for erasure channel and cellular network application even time variant channel because rateless codes are channel independent, we have not to estimate channel erasure probability in priority, the receivers only need to collect enough codeword packets to decode the source symbols completely with high probability. The number of codeword packets needs to be collected is $N = (1 + \epsilon)K$, where K is the number of source symbols and ϵ is the overhead. It is shown that for $K \rightarrow \infty$ there exist codes with $\epsilon \rightarrow 0$. For actual rateless code implementations, K is limited, which implies that ϵ is possibly increased. A good rateless code is designed with ϵ closer to 0 and guarantees high probability of source symbol recovery. In addition, low packet operation complexity is another substantial advantage of rateless

codes. As the development of rateless codes, packet encoding and decoding complexity is getting lower and close to the order of $O(k)$ which is far smaller than the complexity of the order of $O(K^2 \log_2 K)$ of RS codes.

LT code [9] [10] [11] is the realization of erasure codes. A LT code can be described by $LT(K, \Omega(x))$ where K is the input symbols with symbol length l -bits, l is an integer equal or larger than 1 and $\Omega(x)$ is the degree distribution which determines the number of edges connected between one codeword packet and input source packets. On average, $O(\ln(K/\delta))$ packet operations for each codeword packet should be taken to encode K input symbols. Every codeword packet is generated independently and theoretically we can produce infinite codeword packets, though in reality, only finite codeword packets are generated and its number depending on applications. At receiver, receiving $K + O(\sqrt{K \ln 2} (K/\delta))$ of codeword packets, no matter what codeword packets are collected, K original source symbols can be recovered with on average probability $1 - \delta$ after $O(K \ln(K/\delta))$ packet operations. As a result, encoding and decoding times are efficient asymptotically as a function of input symbol number K .

2.2 DEGREE DISTRIBUTION

For LT codes, the probability of a input symbol to be recovered is absolutely determined by degree distribution $\Omega(x)$ and K the number of input symbols. Degree distribution should be design to fit the following three goals:

- Receiver are required to collect as few codeword packets as possible on average to guarantee that input symbols can be recovered completely since the number of collected codeword packets affects the success probability of LT decoding process.

- The average degree of codeword packets should be as low as possible that can decrease LT encoding and decoding times because the complexity of LT encoding and decoding process is proportioned to the average degree of codeword packets.
- Input symbols are added to the ripple at the same rate as they are processed

2.2.1. DERIVE OF DEGREE DISTRIBUTION

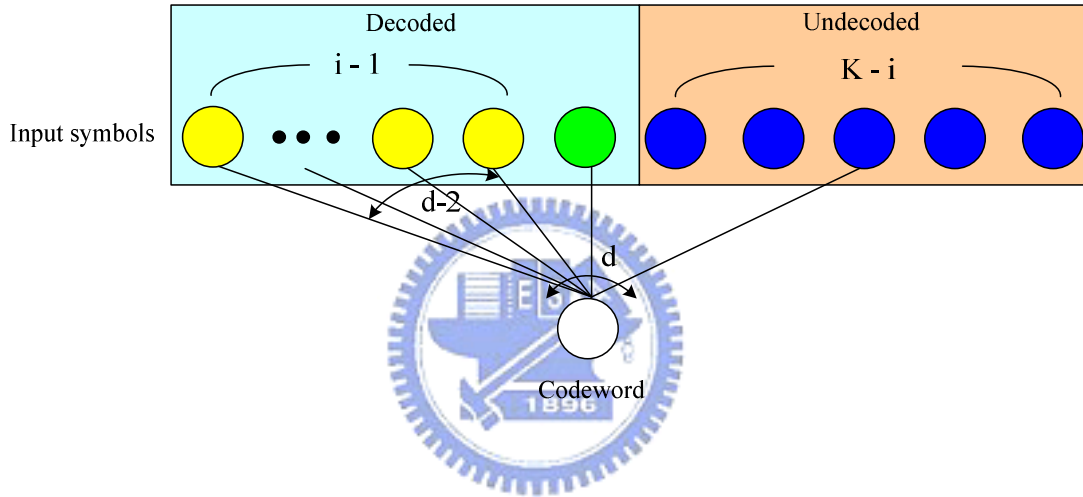


Figure 5. The decoding procedure of a codeword

The concept of deriving degree distribution is based on analysis of decoding process. The analysis starts from focusing on processing of a single codeword as depicted in **Figure 5**. Firstly, the probability of a codeword with degree d released at iteration i is concerned.

$$pr[\text{codeword released at iteration } i \mid \text{degree} = d] = \frac{\binom{k-i}{1} \times \binom{i-1}{d-2}}{\binom{k}{d}}$$

Extension of this result to entire degree is as follows (Ω_d is the probability of degree d):

$$pr[\text{codeword released at iteration } i] = \sum_d \Omega d \times \frac{\binom{k-i}{1} \times \binom{i-1}{d-2}}{\binom{k}{d}}$$

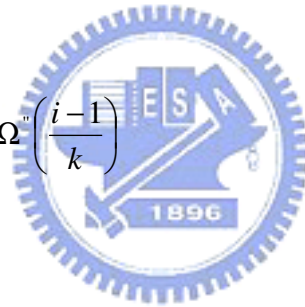
Note that number of codeword with degree one at iteration i must not be too small or too large. This property follows that only one codeword is expected to be released at each iteration. Because k and i is much greater than d , the formula is simplified as following:

$$pr[\text{codeword released at iteration } i \mid \text{degree} = d] = d \times \left(1 - \frac{i}{k}\right) \times \left(\left(\frac{i}{k}\right)^{d-1} - \left(\frac{i-1}{k}\right)^{d-1}\right)$$

$$pr[\text{codeword released at iteration } i] = \left(1 - \frac{i}{k}\right) \times \left(\Omega \left(\frac{i}{k}\right) - \Omega \left(\frac{i-1}{k}\right)\right)$$

Moreover

$$\Omega \left(\frac{i}{k}\right) - \Omega \left(\frac{i-1}{k}\right) = \frac{1}{k} \times \Omega \left(\frac{i-1}{k}\right)$$



Then

$$pr[\text{codeword released at iteration } i] = \frac{1}{k} \times \left(1 - \frac{i}{k}\right) \times \Omega \left(\frac{i-1}{k}\right)$$

Then the number of codeword collected by receiver N is multiplied by the probability of a codeword released at iteration i to obtain the average number of codewords leased at iteration i .

$$E[\text{number of codewords released at iteration } i] = \frac{N}{k} \times \left(1 - \frac{i}{k}\right) \times \Omega \left(\frac{i-1}{k}\right)$$

Assume that N is close to k

$$E[\text{number of codewords released at iteration } i] = \left(1 - \frac{i}{k}\right) \times \Omega \left(\frac{i-1}{k}\right)$$

Finally, let the average number of codewords released at iteration i equal to one and we can reach the goal.

$$(1-x) \times \Omega''(x) = 1$$

$$\Omega(x) = c_0 + c_1 \times x + \sum_{d \geq 2} \frac{x^d}{d \times (d-1)}$$

Degree distribution can be described as:

$$\Omega_1, \Omega_2, \dots, \Omega_k \quad \text{and} \quad \Omega_1 = \frac{1}{k}$$

This result can be proved as a valid probability distribution.

$$\frac{1}{k} + \sum_{d=2}^k \left(\frac{1}{d \times (d-1)} \right) = \frac{1}{k} + \sum_{d=2}^k \left(\frac{1}{d-1} - \frac{1}{d} \right) = \frac{1}{k} + 1 - \frac{1}{k} = 1$$

2.2.2. THE IDEAL SOLITON DISTRIBUTION

Theoretically, Soliton distribution behaves well with the expected number of codeword packets needed to recover the input symbol but it does not work in practice because it has too small expected number of degree 1. Soliton distribution is as follows:

- $\Omega_{\text{ideal Soliton}}(1) = 1/K$
- $\Omega_{\text{ideal Soliton}}(x) = \sum 1/i(i-1)x^i$, for $i=1,2,\dots,K$

2.2.3. THE ROBUST SOLITON DISTRIBUTION

With large enough expected number of degree 1, the robust Soliton distribution guarantees that after collecting $K + O(\sqrt{K \ln^2(K/\delta)})$ codeword packets, receiver can

recover input symbols completely with δ the allowable failure probability and the average degree of codeword packets is $O(\ln(K/\delta))$. The robust Soliton distribution Ω

$(x)_{\text{Robust Soliton}}$ is as follows:

Let $R = c \ln(K/\delta) \sqrt{K}$ for some suitable constant $c > 0$. Define :

$$\tau(i) = \begin{cases} (1) R/iK & \text{for } i=1, \dots, K/R-1 \\ (2) R \ln(R/\delta)/K & \text{for } i=K/R \\ (3) 0 & \text{for } i=K/R+1, \dots, K \end{cases}$$

$$\beta = \sum \Omega_{\text{ideal Soliton}}(i) + \tau(i)$$

$$\Omega(x)_{\text{Robust Soliton}} = (\Omega_{\text{ideal Soliton}}(i) + \tau(i)) / \beta \quad \text{for } i=1, \dots, K$$



2.3 LT CODES

LT codes have simple encoding process as follows:

- Step I For each codeword packet, randomly choose a degree d from degree distribution.
- Step II Uniformly choose d distinct input symbols at random as neighbors of the encoding codeword packet.
- Step III Operate exclusive-or on the d neighbor and the result is the value of the encoding codeword packet

Figure 6 is a simple example of LT decoding. There are five input symbols in the upside of **Figure 6** and we will generate seven codewords (codeword 1~7) as depicted

in the lower side of **Figure 6**. First, at Step1, degree 3 of codeword 1 is chosen based on degree distribution. Then three distinct input symbols (input symbol 1, 3, 5) are selected randomly at Step2. At step3, these input symbols (1, 1, 0) are operated exclusive-or and assigned to codeword 1 that codeword 1 is completely encoded and ready to be sent to channels. At step4, degree of codeword 2 are chosen 5. Five input symbols are chosen as the neighbors of codeword 2 at step5. Afterward, values of these neighbors of codeword 2 are exclusive-ored and the result is assigned to codeword 2. Until now, there are two codewords are encoded and the remaining codewords repeat the same procedure to finish the full encoding process.

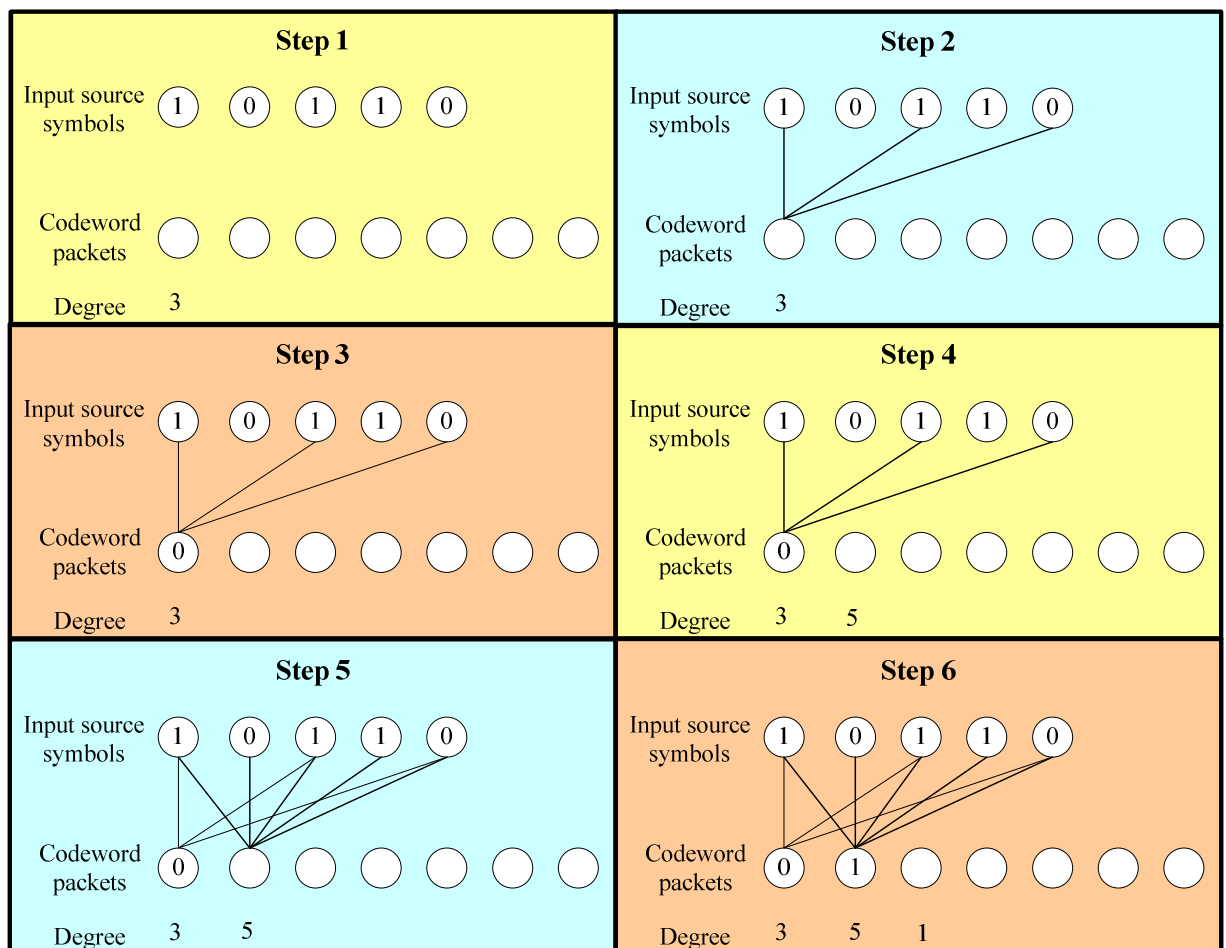


Figure 6. LT-encoding process

It is a significant issue to ensure that both transmitter and receiver know the identical connection structure between input symbols and codewords. Mainly, there are two methods for synchronization of codeword connection structure. First, a codeword is encapsulated into a packet with its degree and the IDs of input sources connected to it. Obviously, additional information bits will increase sizes of codeword packets and lower the transmission efficiency. Codewords with different degrees have to record distinct numbers of input sources IDs that varies codeword packet lengths. Codeword packets with higher degrees need longer packet lengths having higher loss probabilities, For instance, with number of input symbols $k=1000$, symbol length $=100$ bytes, minimum degree $d_{min}=1$ and $d_{max}=67$, we need 7 bits and 10 bits at least to represent degrees and IDs of input sources respectively as shown in **Figure 7**. As a result, for receivers, optimal degree distribution alters that a codeword with higher degree has lower probability of appearance than it is designed and the average degree of entire codewords decreases leading to higher packet loss rate (input symbols have lower probability to be connected with lower average degree). Second, codeword packets are transported only with their codeword IDs depicted in **Figure 8**. For each codeword, either of transmitter and receiver generates both identical degree and the IDs of input sources connected to it according to its unique codeword ID. Compared with the first method, the second one is bandwidth efficient and has equal packet lengths that keep the degree distribution the same at both transmitter and receiver. Note that, in practical, packets may be transmitted through different path and suffering distinct latency. Though packets do not reach its destination in order, receivers reconstruct information of each codeword according to its codeword ID and reorder of codewords are not necessary. To implement second method, we design a uniform random number generator, `random()`, with its period much greater than the number of input symbols for both transmitter and receiver. With identical uniform

random number generator, transmitter and receiver can both recognize the neighbors to each codeword. The details of generating connection information of a codeword with k input symbols using method two is described as follows:

- Step I generate a number $a = \text{random}(\text{codeword ID})$
- Step II degree d is decided according to the interval of degree distribution which number a locates
- Step III IDs of input sources $E[i]$, $i = 0 \sim d-1$, connected to this codeword is produced after d iterations of operating $E[i] = \text{random}(a) \% k$, $a = \text{random}(a)$



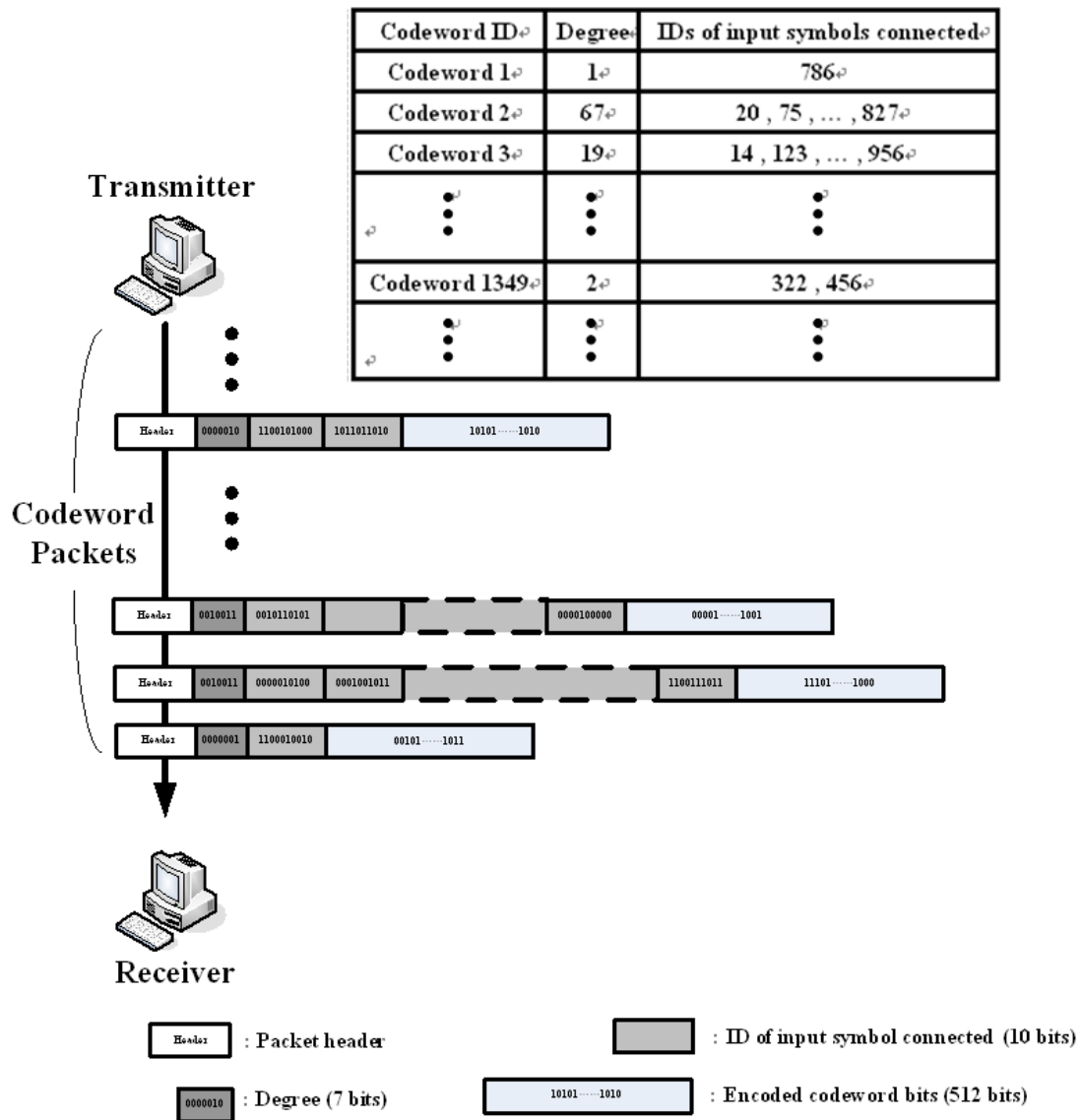


Figure 7. The transmission of connection structure between input symbols and codewords by encapsulating degrees and codeword neighbors in packets

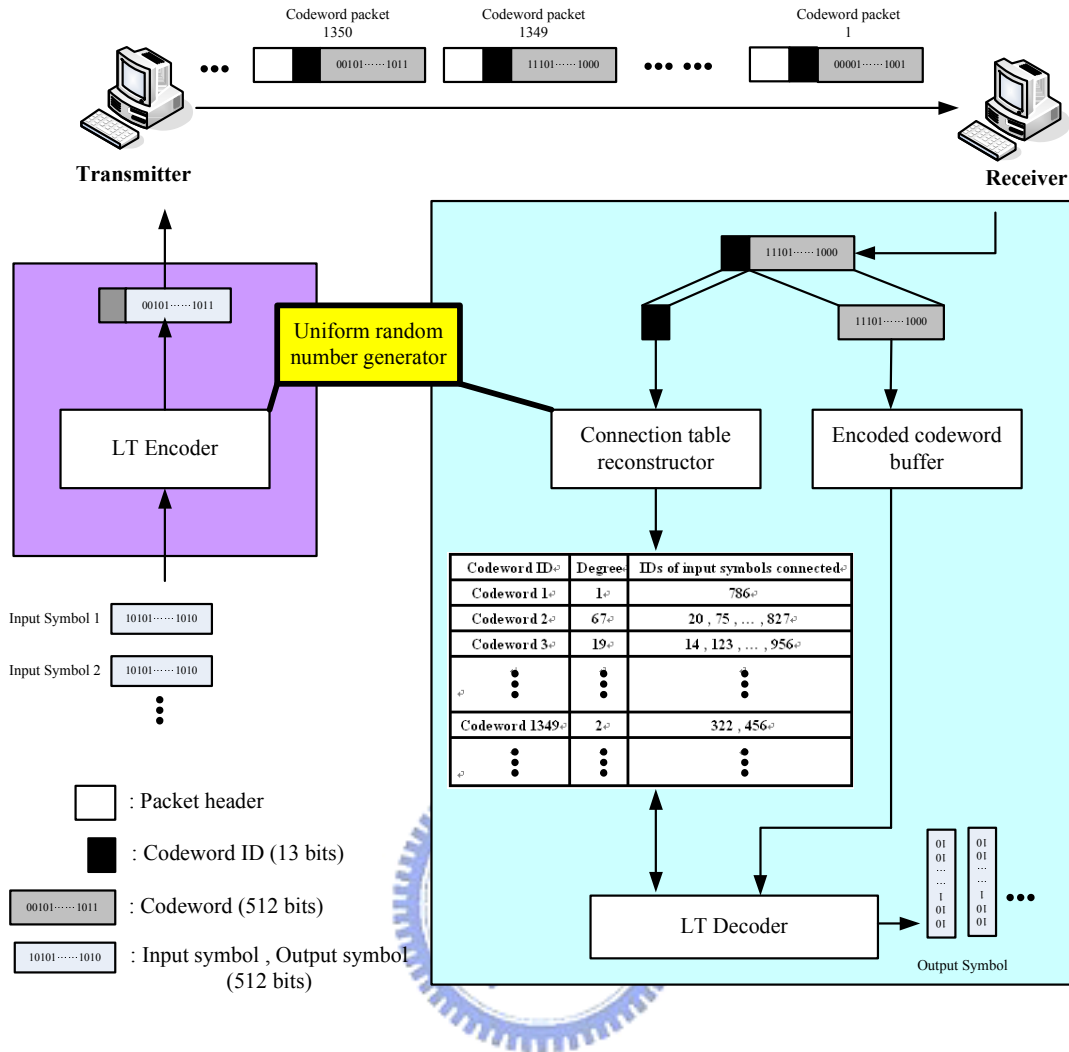


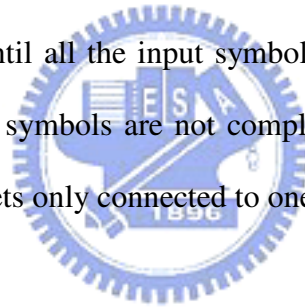
Figure 8. Encoder and decoder use the identical uniform random number generator to produce the same connection structure between input symbols and codewords

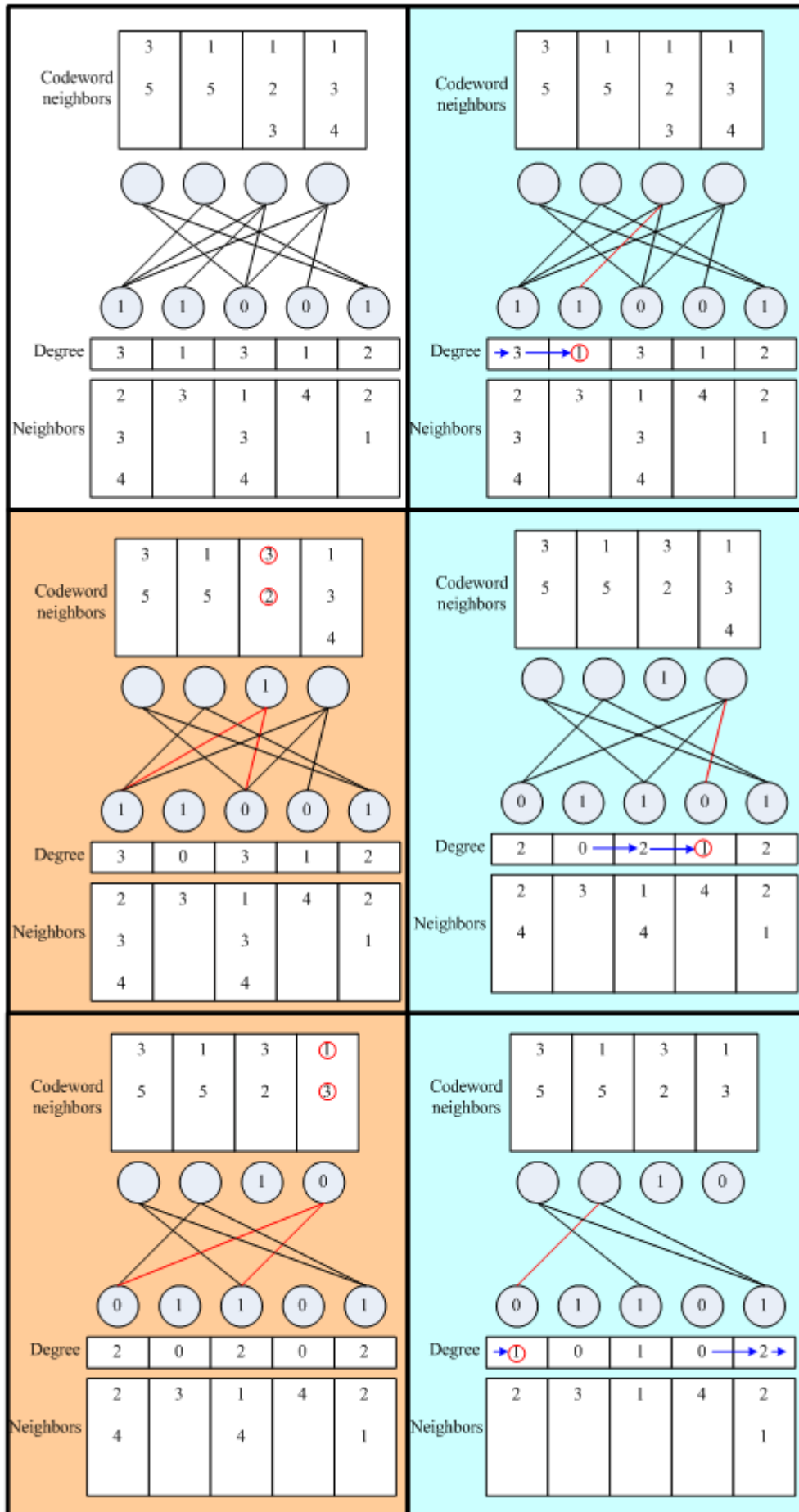
2.3.1. THEORETICAL LT DECODING

As receivers collect enough codeword packets, no matter what codeword set is, they can start decoding process respectively. Now, we can regard decoding process as solving a almost random matrix with its elements 0 or 1. There are two different method available, namely Gaussian elimination and believe propagation (BP). As far

as Gaussian elimination is concerned, it takes packets operation complexity of the order $O(K^2)$ to recover input symbols that it is inefficient and is almost impossible for implementation with K large. For BP, its low packet operation complexity and ease of realization are main reasons to be chosen as LT decoder. BP process is described as below:

- Step I Find a codeword packet that is connected to only one input symbol and set its value to the input symbol which is now decoded.
- Step II The value of this decoded input symbol is operated exclusive-or on those codeword packets of its neighbor.
- Step III Remove the decoded source symbol and all its connections.
- Step IV Repeat step I~III until all the input symbols are decoded or stop decoding process when input symbols are not completely recovered but there is not any codeword packets only connected to one input symbol.





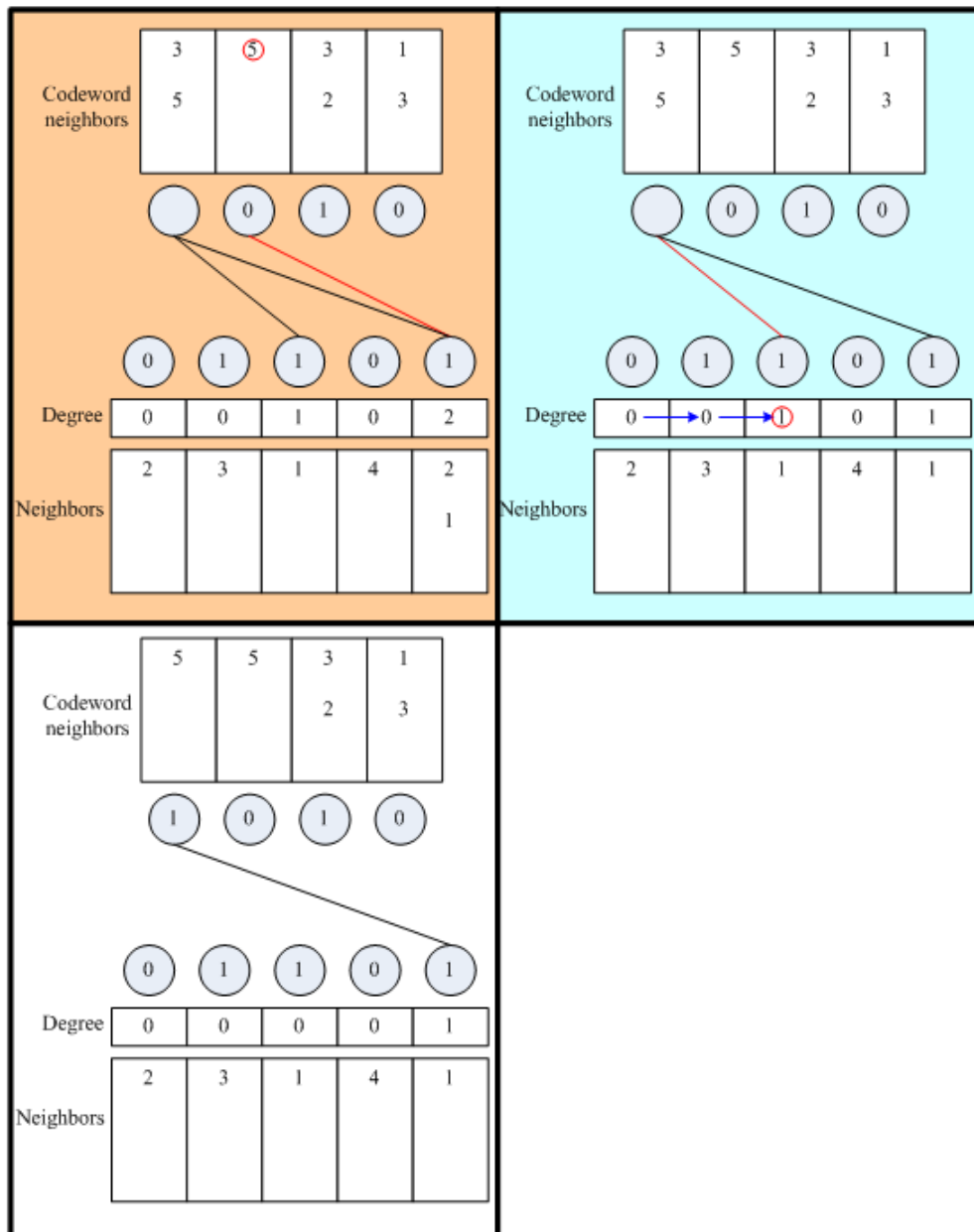
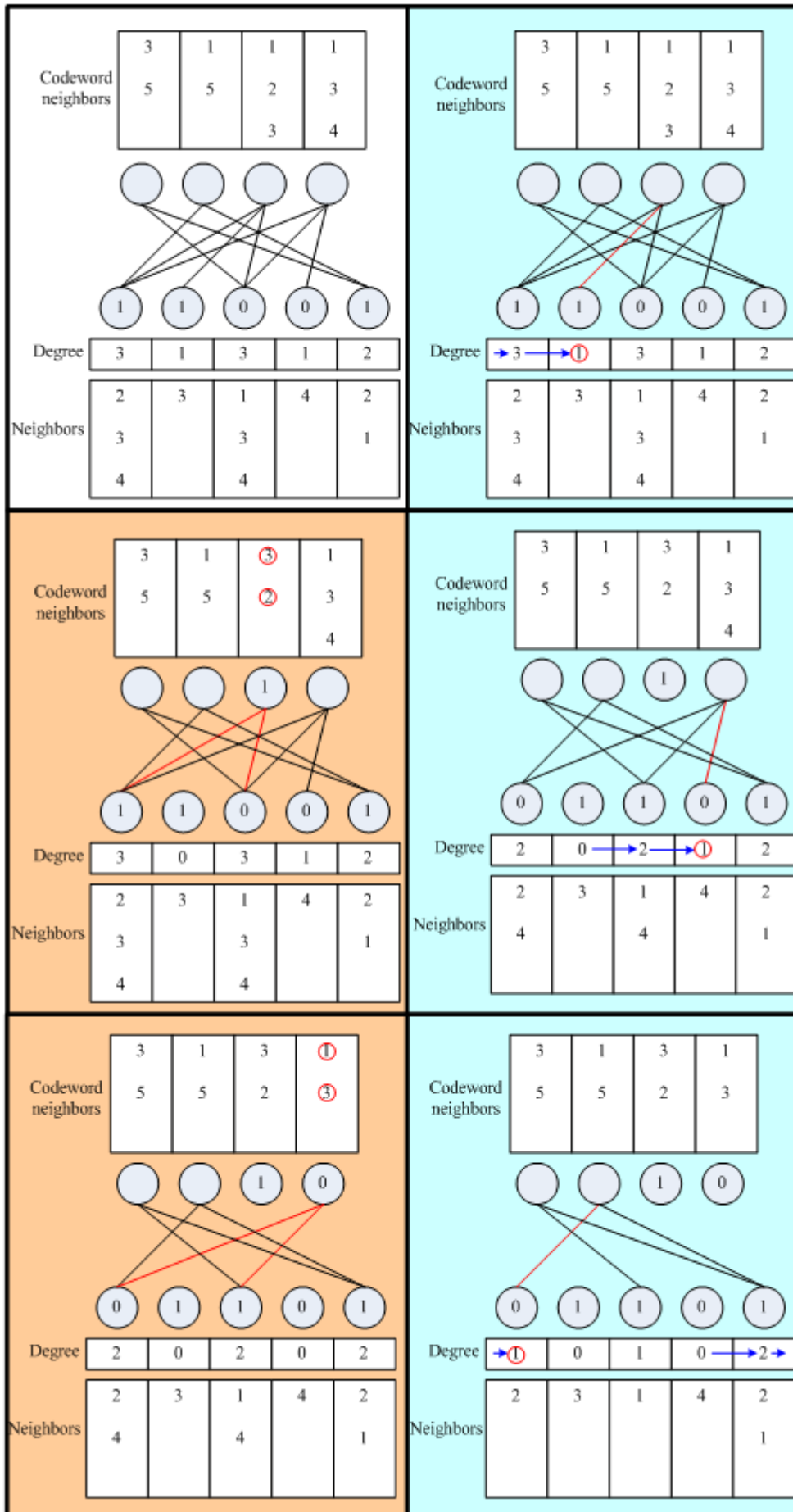


Figure 9 is a simple example to describe how is decoding process executed. Now, we have five codewords and four input symbols need to be recovered. First, all the degrees of codewords are scanned and codeword 2 is selected because of its degree is one. Value of codeword 2 is assigned to input symbol 3 and degree of codeword 2 is set zero. Input symbol 3 is now decoded and its value is operated exclusive-or to codeword 1, 3. Then input symbol 3, 4 are removed from connection table of codewords. Scanning continues and codeword 4 is selected. Input symbol 4 is specified the value of codeword 4. Neighbors (codeword 1, 3) of input symbol 4 is

exclusive-ored with value of input symbol 4 and this input symbol is removed from codeword connection table. Degrees are rescanned and codeword 1 is selected. Value of codeword 1 is assigned to input symbol 2. Only codeword 5 is connected to input symbol 2. After exclusive-or their value, input symbol 2 is removed from neighbors of codeword 5. Keep on scanning degrees, codeword 3 is chosen and its value is assigned to input symbol 1. Codewords are recovered completely and decoding process successes.





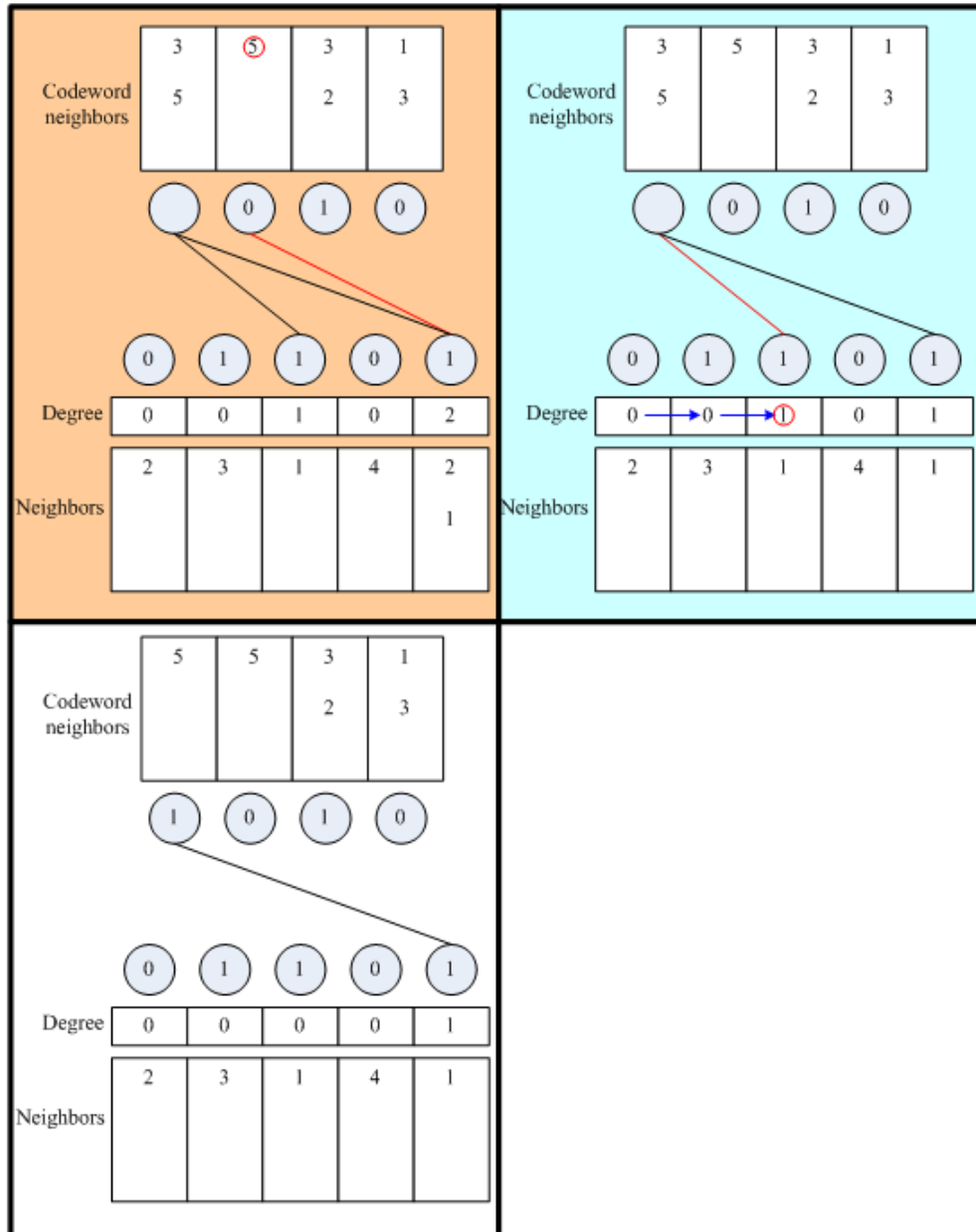


Figure 9. Theoretical LT decoding

2.3.2. REGULAR LT DECODING

Roughly, each step of conceptual decoding process described in 2.7.1 can be considered as a function (we have Function I, Function II and Function III). In Function II, the value of the decoded input symbol is operated exclusive-or on those codeword packets of its neighbor that besides of codeword connection table,

neighbors of all input symbols must be recorded leading to additional table usage (size of connection table of input symbol is the same as that connection table of output symbol has). Moreover, during this type of decoding process, Function I and Function II are frequently switched that decreases decoding efficiency. To accelerate the decoding speed and save the additional buffer usage, a regular decoding process with only a small table recoding if each codeword is decoded or not is necessary as follows:

Step I Scan entire codeword connection table. If a codeword has degree one exactly, its value is assigned to the corresponding input symbol. Then this codeword is removed. The size of codeword set is decreased by one. Record the number of decoded input symbols and set a flag to declare processable.

Step II If the flag declares processable. Scan whole neighbor input symbols of each codeword. If one or more neighbor input symbols of a codeword are marked decoded, their values are operated exclusive-or on this codeword and its degree is decreased by the number of neighbor input symbols operated. Reset the flag.

Step III If all the input symbols are decoded or the flag is not set, stop decoding process.

A simple example of regular decoding process is depicted in **Figure 10**. Before starting decoding process, we have five codewords and four input symbols which need to be recovered. At the beginning, as illustrated in step I above, degrees of codewords are scanned in receiving order and those values of codewords with degree one (codeword 2, codeword 4) are assigned to corresponding input symbols (input

symbol 3, 4) and these input symbols are set decoded. The codewords with degree one are updated to be degree zero and a flag are raised to declare preceivable. When there is not any codeword with degree one, decoder operates step II. Neighbors (input symbol 3, 4)of codewords with degree greater than one are checked in order and those value of neighboring input symbols which are marked decoded are operated exclusive-or to corresponding codewords (codeword 1, 3) and removed from connection table. Next, rescan degrees and find those codewords with degree one (codeword 1, 3) and assign their value to corresponding input symbols (input symbol 1, 2) and raised the flag for preceivable. Though the flag is raised, all the input symbols are recovered and the decoding process must end.



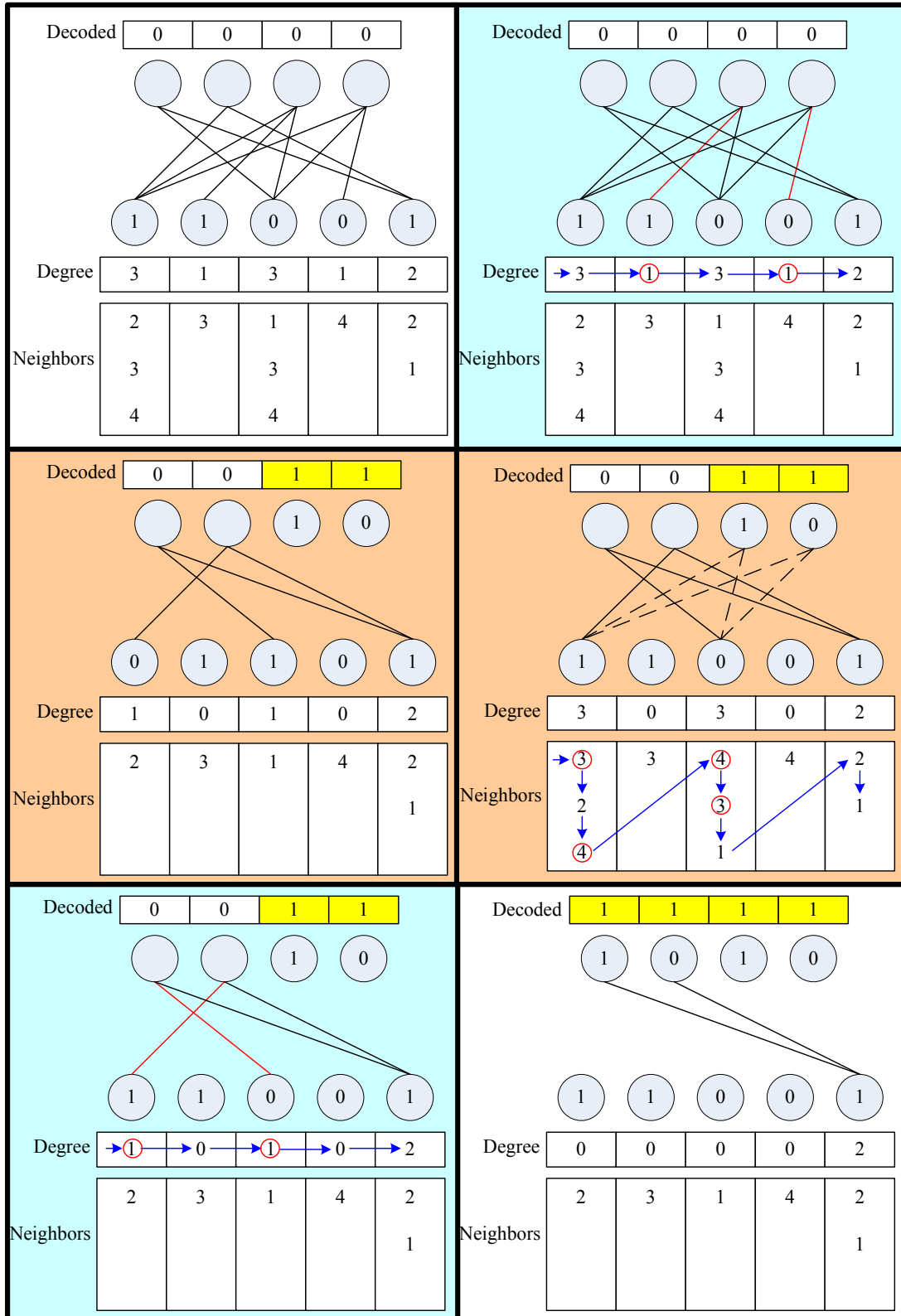
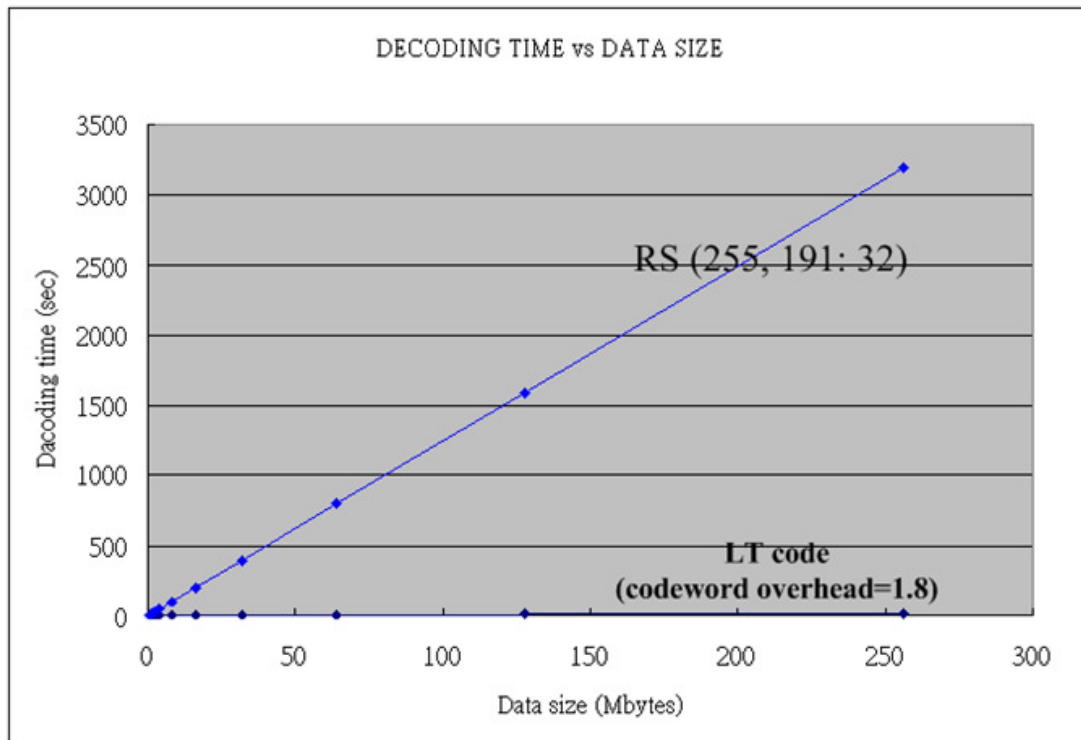


Figure 10. Regular LT decoding process

2.3.3. DECODING LATENCY

So far as real-time video transmission is concerned, delay time is critically constrained. Also, video stream must be segmented into packets which often contain a complete slice data over IP-based network. For example, an I-frame of video (QCIF) encoded with H.264 is about 2k bytes. If this I-frame is separated into two slices, each slice is about 1k bytes that if each packet contains one slice, 1k bytes of packet length must be chosen. With packet length about 1k bytes or longer, when implemented with software, delay time of RS codes, LDPC codes, and other block codes are extremely high, moreover, packet length may not be able to selected arbitrarily. The only solution to decrease delay time for block codes is to further separate each packet into smaller sizes and encoding them. Note that, codewords generated by block encoder must be interleaved and recapitulated as packets in order to avoid loss of whole slice. As a result, the low computational complexity and allowance of arbitrary input symbol lengths make LT codes much suitable for real-time video transmission compared with traditional block codes. Conceptually, decoding complexity of a LT code is proportional to $O(K \ln(K/\delta))$ where K is the number of input symbols and δ is the designed packet loss rate. But this conceptual result considers only the number of packet operations. In practice, decoding time depends on different decoding processes. Even for the same decoding process, in addition to packet operations, codeword table update, remove of codeword and the similar operations also affect decoding time. So decoding time is much different than that expected by theoretical results. **Figure 11** shows the relationship between different data sizes and decoding times with packet length equal to 1k bytes and the LT regular decoding process under steady simulation condition. Obviously, the LT code requires much shorter decoding time than RS code does. Moreover, decoding time of the LT code is proportional to k but not to

$O(K \ln(K/\delta))$. This result stands for that other operations but not packet operations dominates the decoding time and regular decoding process is simple enough to let decoding time linearly proportional to number of input symbols successfully.



Channel Erasure Probability = 10%
Decoding Packet loss rate = 10^{-12}

Figure 11. decoding time (sec) versus data size (Mbytes)

2.4 SIMULATION RESULT

The LT-code simulation result is shown in **Figure 12**, with $K=10000$, the number of collected codeword packets from K to $2K$, degree distribution chosen as in [12]:

$$\begin{aligned} \Omega(x) = & 0.007969x + 0.493570x + 0.166220x^3 \\ & + 0.072646x + 0.082558x + 0.056058x \\ & + 0.037229x + 0.055590x + 0.025023x \\ & + 0.003135x \end{aligned}$$

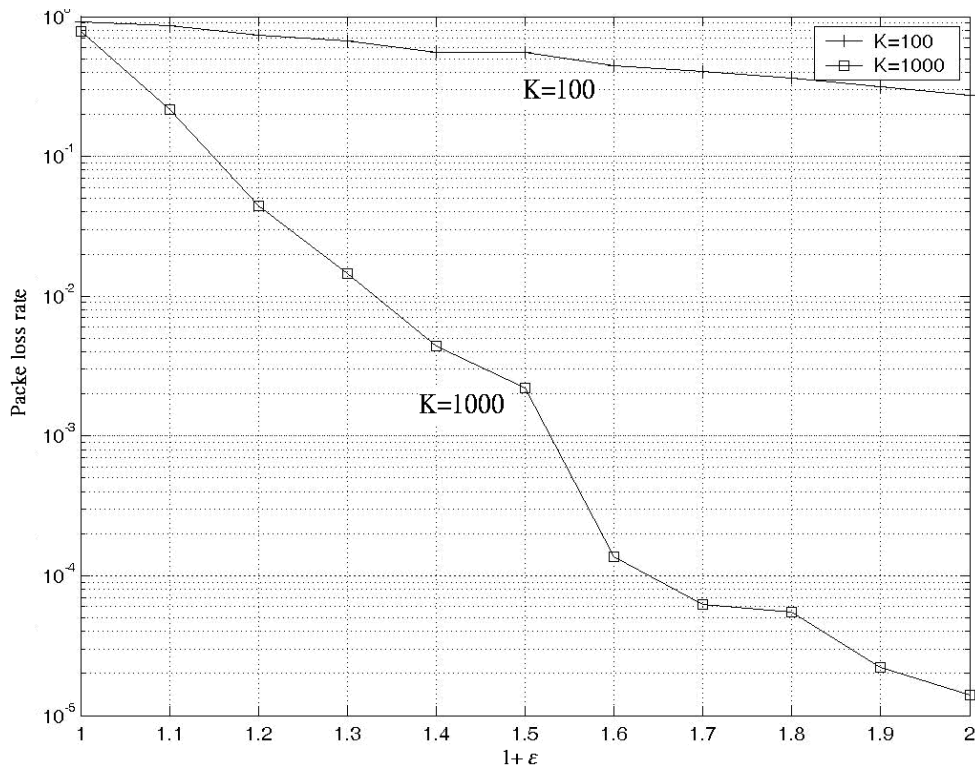


Figure 12. Simulation results of LT codes with K=100 and K=1000 respectively. Y-axis and X-axis stand for packet loss rate and codeword overhead ($1 + \epsilon$) respectively.

Obviously, LT code has better performance as K getting larger. Because of random connection between input symbols and codeword symbols, unlike block codes, we can consider that source information is equally spread on every codeword symbol and no matter how high the erasure probability is, with enough codeword packets, input symbols can be recovered completely. This property is opposite to the property of RS code (RS code performs better when K is smaller).

CHAPTER 3

UEP-LT CODES

Shannon's separation theorem is one of the foundations of information theory. It states that source coding and channel coding can be operated independently and then combined without end-to-end performance loss. Nevertheless, this statement only holds for specific conditions. For time variant channels (mobile communication) or multipoint communication (video conferencing), we can not know channel condition in advance so Shannon's separation theorem fails. Moreover, for multimedia data transmission such as H.264 video stream, data with different importance must have distinct protection so that we have to take both source coding and channel coding into consideration simultaneously to achieve optimal performance. Therefore, channel coding plays a significant role that requires unequal error protection and channel irrelevant abilities.

3.1 UNEQUAL ERROR PROTECTION

Recall those mentioned in introduction, rateless codes have unique advantages which can not be arrived by block codes on cellular network such as channel independent, on-the-fly, low encoding decoding complexity and bandwidth efficiency. Furthermore, proportion of video and audio transmission is increasing rapidly that makes UEP more and more important. Consequently, rateless codes combine with

UEP will be a trend in the future. Recently, LDPC with UEP is popular but its encoding and decoding complexity decreases heavily, in addition, it is not suitable for cellular network due to its essence of block codes. A rateless code with UEP is proposed based on LT codes and Raptor codes in [14]. According to the simulation results in [15], performance of LT codes and Raptor codes with UEP can perfectly match the requirements of transmission with low encoding and decoding complexity increasing.

3.2 THE UEP-LT CODE

In theory, for rateless codes, a input symbol connected to more codeword packets has higher probability to be recovered because more information of this codeword is transmitted. The UEP-LT code combined with maximum-likelihood decoding is proposed in [14]. Suppose the number of input symbols is K with two level of importance. Assume the number of more important bits (MIB) is $K_1 = \alpha K$, which are put in the head of whole input sequence, and $K_2 = (1 - \alpha)K$ is the number of less important bits (LIB). It is proposed to construct a UEP-LT code and UEP-Raptor code the same as traditional LT codes and Raptor codes except that the codewords select their neighbors nonuniformly at random. Take a single codeword with degree d for example, there are $d_1 = \min(\lceil \alpha dk \rceil, K_1)$ ($\lceil x \rceil$ means the nearest integer to x) neighbors from MIB (for some $km > 1$) and $d_2 = d - d_1$ neighbors from LIB as shown in **Figure 13**.

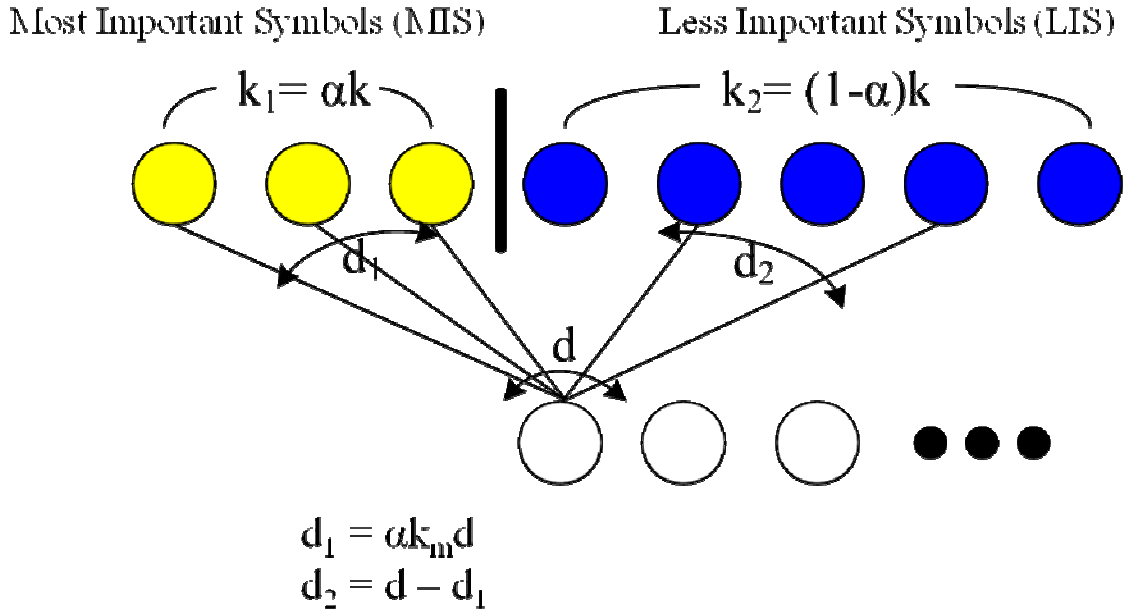


Figure 13. The UEP-LT code structure

Every neighbor in the same codeword is distinct that any sequence of d_1 (d_2) neighbors in MIS (LIS) is selected uniformly.

ML decoding of UEP-LT codes: Upper bounds of ML decoding is proposed in [14]. Consider a UEP-LT code with degree distribution chosen as in [12]:

$$\begin{aligned} \Omega(x) = & 0.007969x + 0.493570x^2 + 0.166220x^3 \\ & + 0.072646x^4 + 0.082558x^5 + 0.056058x^6 \\ & + 0.037229x^7 + 0.055590x^8 + 0.025023x^9 \\ & + 0.003135x^{10} \end{aligned}$$

and number of input symbols K , α , K_m , and overhead γ_L . The upper bounds on ML decoding BERs of MIS and LIS are given by

$$p_{b, MIB}^{ML} \leq \min \left\{ 1, \sum_{w=1}^k \sum_{w_1=1}^w \binom{n_1-1}{w_1-1} \binom{n_2}{w_2} \cdot \left(\sum_d \Omega_d \frac{\sum_{t=0}^1 \left(\prod_{r=1}^2 \left(\sum_{s=t, 2+t, \dots, 2\lfloor \frac{d_r}{2} \rfloor - t} \binom{w_r}{s} \binom{n_r-w_r}{d_r-s} \right) \right) \right)}{\binom{n_1}{d_1} \binom{n_2}{d_2}} \right)^{k\gamma_L} \right\}$$

and

$$p_{b, LIB}^{ML} \leq \min \left\{ 1, \sum_{w=1}^k \sum_{w_1=0}^{w-1} \binom{n_1}{w_1} \binom{n_2-1}{w_2-1} \cdot \left(\sum_d \Omega_d \frac{\sum_{t=0}^1 \left(\prod_{r=1}^2 \left(\sum_{s=t, 2+t, \dots, 2\lfloor \frac{d_r}{2} \rfloor - t} \binom{w_r}{s} \binom{n_r-w_r}{d_r-s} \right) \right) \right)}{\binom{n_1}{d_1} \binom{n_2}{d_2}} \right)^{k\gamma_L} \right\}$$

And lower bounds on ML decoding BERs of MIB and LIB are given by

$$p_{b, MIB}^{ML} \geq \left(1 - \sum_d \Omega_d \frac{d_1}{n_1} \right)^{k\gamma_L}$$

And

$$p_{b, LIB}^{ML} \geq \left(1 - \sum_d \Omega_d \frac{d_2}{n_2} \right)^{k\gamma_L}$$

In section 3.2, for the UEP-LT code, a codeword is connected to MIB and LIB in different ratio except those with degree 1 to achieve UEP and URT and the decoder uses Maximum-likelihood algorithm which can recover input symbols with high probability. But computational complexity of Maximum-likelihood algorithm raises drastically as bit lengths of input symbols increase that makes direct implementation of this UEP-LT code in application layer impossible. Substitution of ML decoding with believe-propagation decoding is the best solution to realize this UEP-LT code in application layer. Actually, separation of edges in codewords can be considered that input symbols with different importance are classified into distinct groups and each group has its own degree distribution which is distorted from that we designed in a

ratio. These distortions of degree distribution greatly increase packet loss rate of decoding. Moreover, if a codeword is connected with both MIB and LIB, its packet loss rate depends on it of LIB which is designed to have lower probability of recovery. In this way, packet loss rate of MIB extremely increases. In the following sections, a different UEP-LT method which retains the essence of rateless codes is proposed with much better performance.

3.3 PROPOSED SEPARABLE UEP-LT CODE

Suppose the number of input symbols is K with two level of importance. Assume the number of more important symbols (MIS) is $K_1 = \alpha K$, which are put in the head of whole input sequence, and $K_2 = (1 - \alpha)K$ is the number of less important symbols (LIS). It is proposed to construct a UEP-LT code the same as traditional LT codes except that the codewords are separated into two sub-groups, codewords in the first group only connect to MIS and codewords in the second group merely connect to LIS. There is probability P_{MIS} for codewords to be in the first group, P_{LIS} for codewords to be in the second group and $P_{MIS} + P_{LIS} = 1$. The proportion of P_{MIS} and P_{LIS} is the most important parameter of our UEP-LT code and how to decide them will be discussed in the later section. The codewords versus input symbols relationship is depict in **Figure 14**. This method does not distort the designed degree distribution in both MIS and LIS groups that the designed packet loss rate can be achieved. On the other hand, average degree does not change after regrouping codewords that computational complexity keeps the same in theory. Simulation of the first proposed UEP-LT code with BP decoding and the UEP-LT code we proposed are performed in identical condition with

$k=10000$, $\alpha=0.1$, $K_m=2$, $P_{MIS}=0.2$, and $P_{LIS}=0.8$ as shown in the **Figure 15**. Obviously, the method we proposed for UEP-LT code has much better performance when numbers of edges connected to MIS and LIS are the same in two method (Packet loss rate of our method is about 10^{-5} and 10^{-3} smaller in MIS and LIS respectively compared with the first proposed method). In this example, only connections between input symbols and codewords are changed.

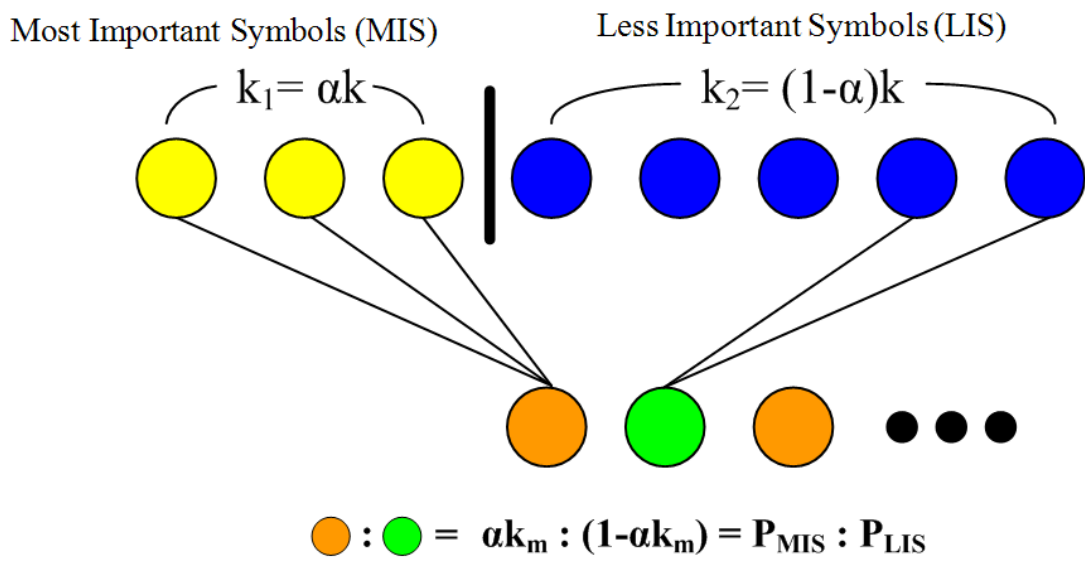


Figure 14. Separable UEP-LT code structure

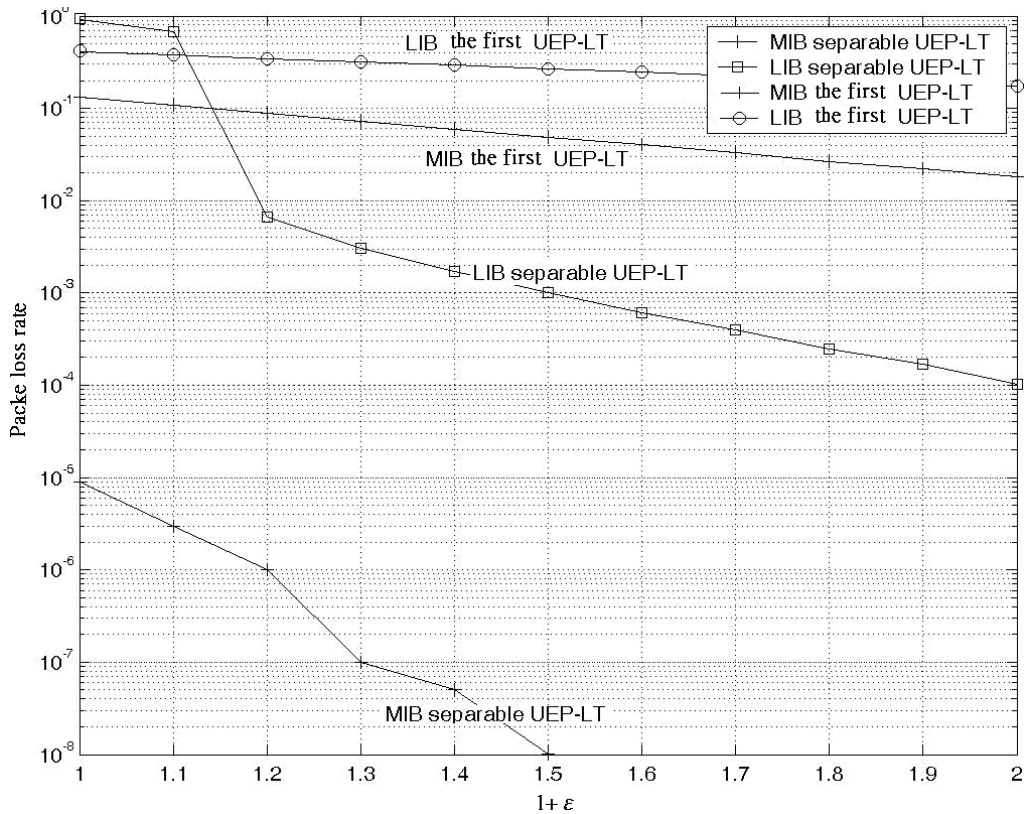


Figure 15. Simulation result shows packet loss rate of MIS and LIS using the UEP-LT code and proposed separable UEP-LT code. Parameters of the UEP-LT code are $K=10000$, $\alpha=0.1$, $K_m=2$ and parameters of proposed separable UEP-LT code are $K=10000$, $\alpha=0.1$, $P_{MIS}=0.2$. Y-axis and X-axis stand for packet loss rate and codeword overhead ($1+\epsilon$) respectively.

The main object of our UEP-LT code is to decide values of P_{MIS} and P_{LIS} . Recall that the core of designing a LT code bases on its degree distribution. Once the degree distribution is decided, the performance of this LT code is also decided. Similarly, if we consider P_{MIS} and P_{LIS} as the ratio of codeword overheads of MIS and LIS individually, we can find the relationship between these two parameters and the degree distribution which can help us to design an appropriate UEP-LT code. In

another word, MIS and LIS have different group sizes and requires distinct packet loss rates that each of them needs a unique degree distribution. More details will be discussed in the following sections.

Carefully analyzing the robust soliton distribution, we can find that there are three significant variables the number of input symbols k , the packet loss rate δ and a parameter c . Generally, we must set the packet loss rate δ as small as possible to achieve specified value. Also, the number of input symbols k depends on the size of source data. The only parameter must be adapted is c . **Figure 16** shows the histogram of degree distribution with $k=1000$, $c=0.01$ and $\delta=10^{-8}$. Effectively, if a robust soliton distribution is decided, it must be truncated as a sub-optimal degree distribution according to a acceptable probability threshold that increases the packet loss rate. For example, if 1000 input symbols must be encoded and codeword overhead is 1.2, those degrees with probability less than 0.000834 ($1/(1000*1.2)$) can be ignored because they hardly appear in receivers.

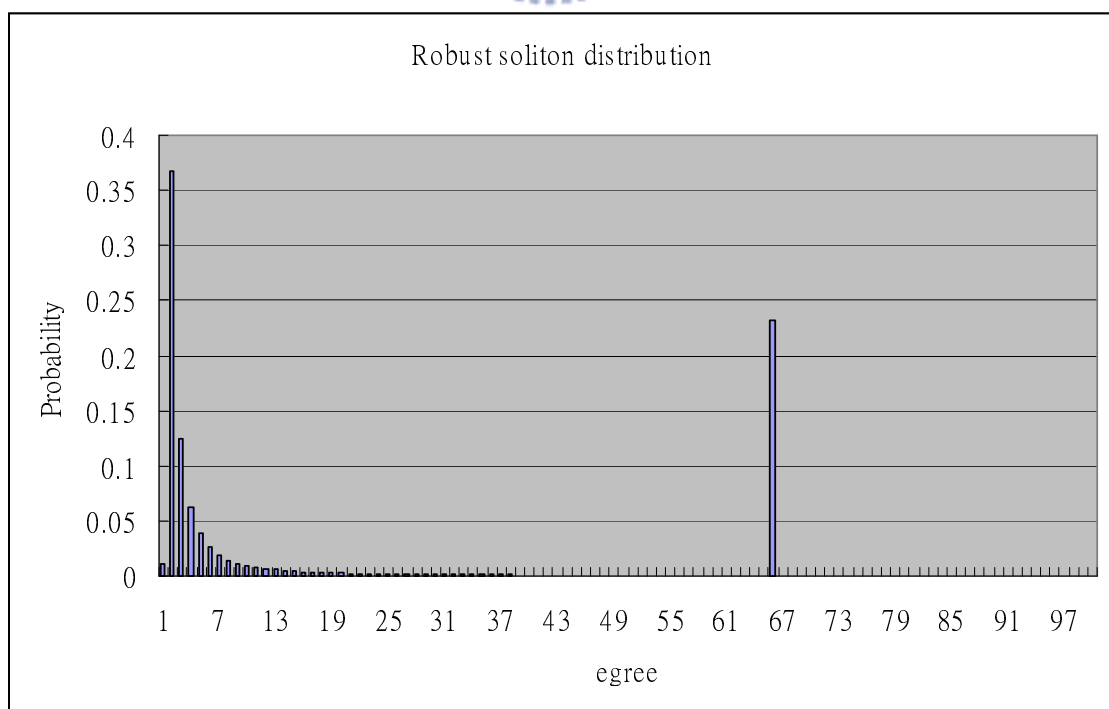
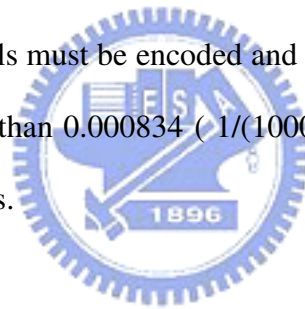


Figure 16. Histogram of robust soliton distribution

Generally, c is chosen to minimize the product P (Theoretically, LT encoder generates infinite number of codewords and channel bandwidth is not concerned) of the average degree d_{avg} and the number of codewords K' needed to achieve packet loss rate. The product P of d_{avg} and K' means the total number of packet operations which is linearly proportion to decoding time in theory. **Figure 17** shows the relationship between parameter c and the ratio of packet operations. In this case, c must be 0.02 to achieve minimum number of packet operations.

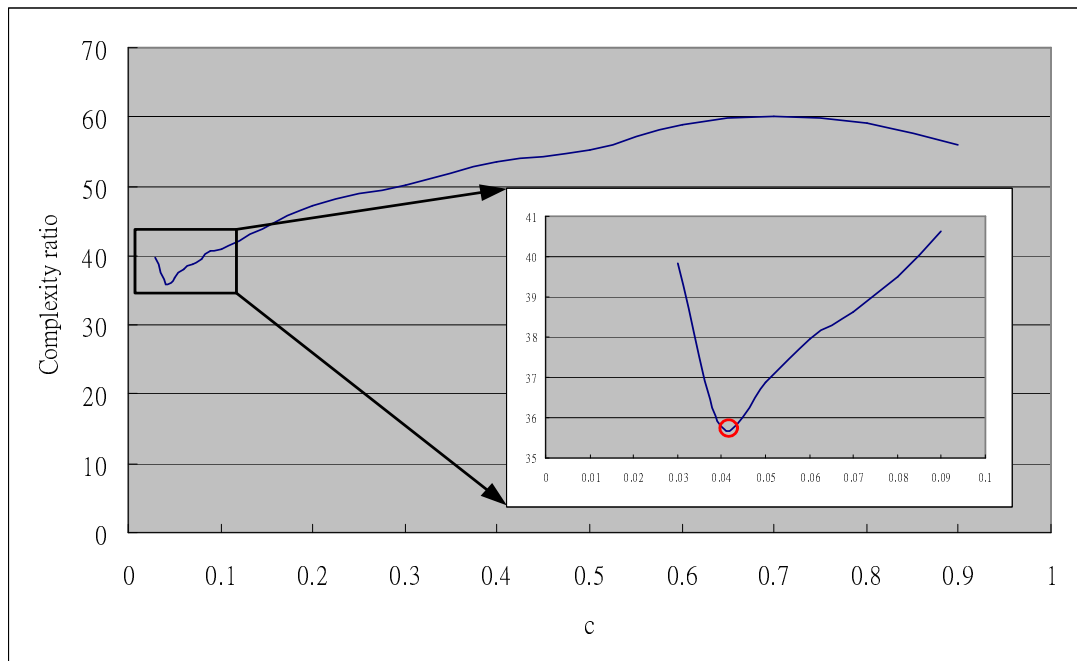


Figure 17. the relationship between parameter c and the ratio of packet operations

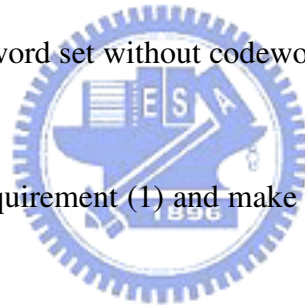
Practically, in addition to decoding time, we must take codeword overhead into consideration which also affects end-to-end delay. Furthermore, besides of packet operation, decoding process includes memory access, degree scan, and so forth. These operations influence decoding time, too. **Figure 18** shows the relationship between

different choice of c and decoding time of regular decoding process. It is clear that decoding time is almost the same even c changes greatly. This result describes that decoding time isn't dominated by number of packet operations but other procedures. In other words, we do not have to care about total number of exclusive-or operations in substance when we design the value of c . In real world, unlike the assumption in theory, channel bandwidth is limited that we must concern about codeword overhead and minimize it. An optimal selection of c must fit the requirement of packet loss rate and reduce codeword overhead as many as possible. Besides, we have to make sure that appearance probability of codeword with degree one (A codeword set without codeword only connecting to one input symbol fails to start decoding) must be large enough. Complete design criterions of c are as follows:

(1) The probability of a codeword set without codeword having degree one is (1 -

$$\Omega_1) k \cong 0.01 \times \delta$$

(2) The value of c must fit requirement (1) and make codeword overhead as small as possible



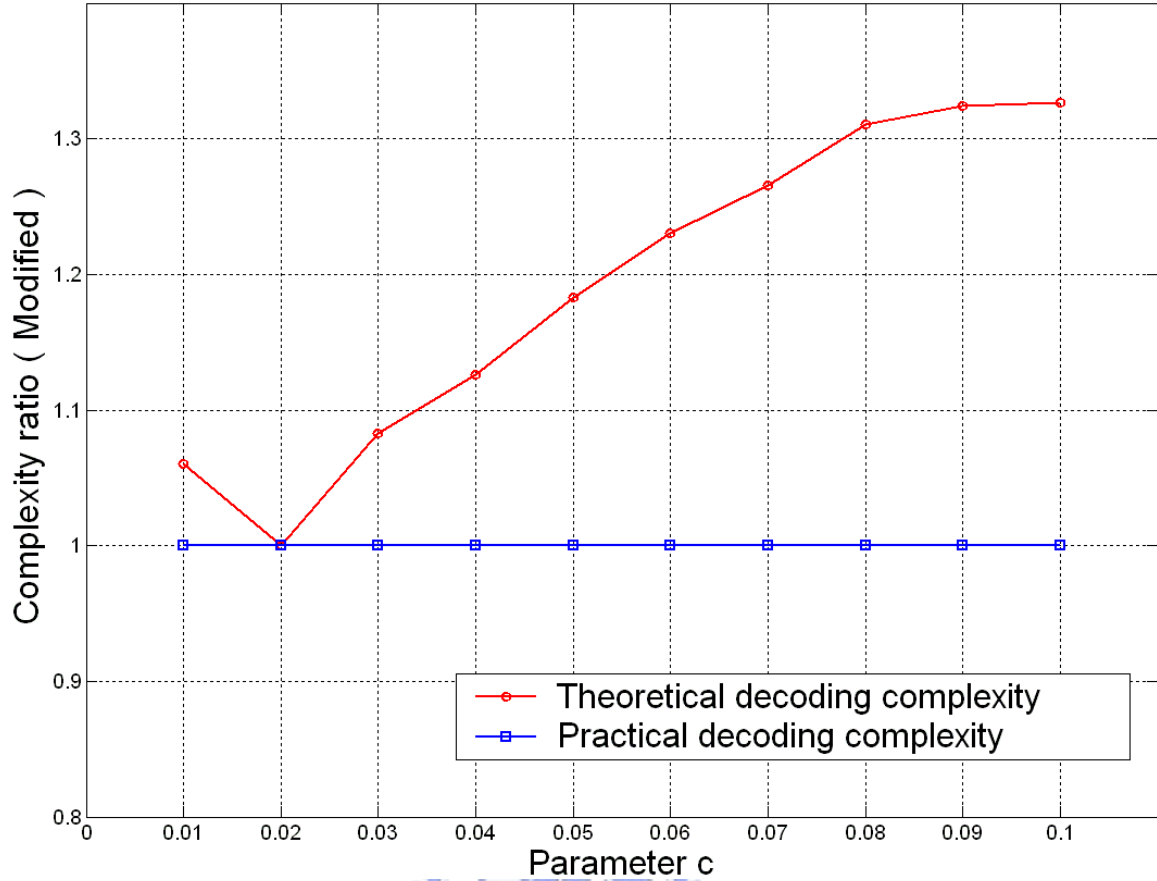


Figure 18. Ratio of decoding latency versus parameter c ($k = 1024$,
symbol length = 1k bytes)

All characteristics of proposed separable UEP-LT code are described in prior sections. To sum up, we can represent the design procedure of this code in **Figure 19**. as the following steps:

- Step I Decide the packet loss rates (P_1, P_2, \dots, P_N) for distinct groups of input symbols (The numbers of input symbols in groups are k_1, k_2, \dots, k_N)
- Step II Generate the robust soliton degree distribution according to $P_1 \sim P_N$ and $k_1 \sim k_N$ for each group but leave parameter c undecided.
- Step III Choose each parameter c which minimizes codeword overhead and satisfies $(1 - \Omega_1) \times k \leq 0.01 \times P_i, i=1 \sim N$
- Step IV Adapt the degree distribution of each group

Step V Decide groups of codewords according to designed ratio

Step VI Generate degrees of codewords

Step VII Codeword are connected to their neighbors in their groups and transmitted.

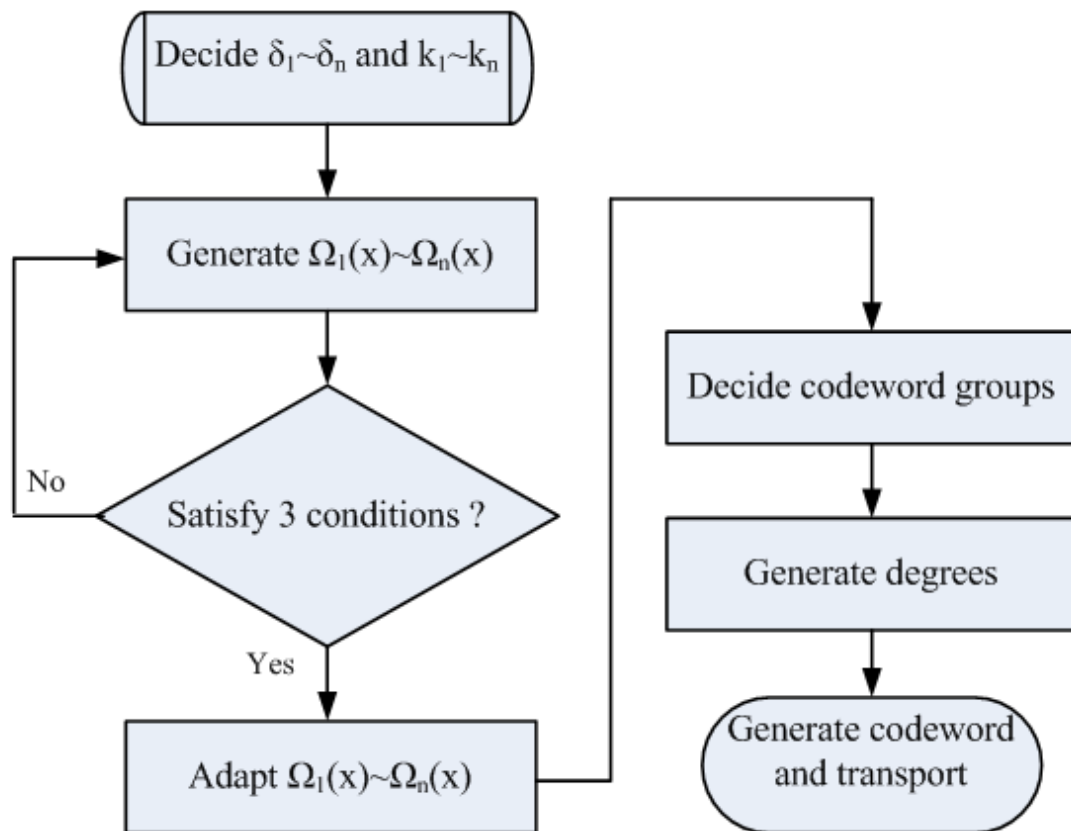


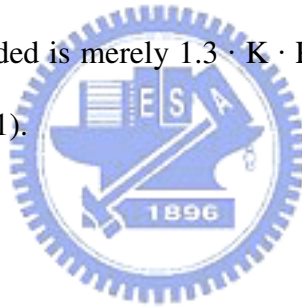
Figure 19. Flow chart of designing a separable UEP-LT code

3.4 UNEQUAL RECOVERY TIME AND DISTRIBUTED DECODING

In addition to functionality of unequal error protection, the separable UEP-LT we proposed has another capability of unequal recovery time. This capability is useful when each receiver requires distinct packet loss rates of MIS and LIS instead of the

designed ones (Receivers may have different size of displays, allow different qualities or resolutions ...). Taking the simulation result in **Figure 15** as an example, if δ_{MIS} and δ_{LIS} are designed to be 10^{-8} and 10^{-3} , we can recover MIS and LIS with packet loss probability 10^{-6} and 10^{-2} after collecting $1.2 \cdot k = 12000$ codeword packets for some receivers requiring lower packet loss rate. In this way, receivers can just collect appropriate number of codewords they need that decreases end-to-end latency and buffer size.

Because codewords connect only to MIS or LIS, we can decode them individually. This is efficient for receivers demanding different parts of data. In this way, the number of receiver buffer can be decreased and decoding complexity is also reduced. For example, if a receiver only needs information in MIS with $\delta_{\text{MIS}} = 10^{-7}$, the number of codewords needed is merely $1.3 \cdot K \cdot P_{\text{MIS}} = 2600$. And the number of packet operations is $2600 \cdot \Omega(1)$.



CHAPTER 4

SIMULATION RESULTS

Figure 20 shows simulation results of the proposed separable UEP-LT code with each group of codewords having its own degree distribution. Simulation condition is the same as before (parameters $k=10000$, $\alpha=0.1$, and $K_m=2$). Obviously, packet loss rate of MIS and LIS are about 10^{-2} and 10^{-1} smaller compared to separable UEP-LT code with single degree distribution.



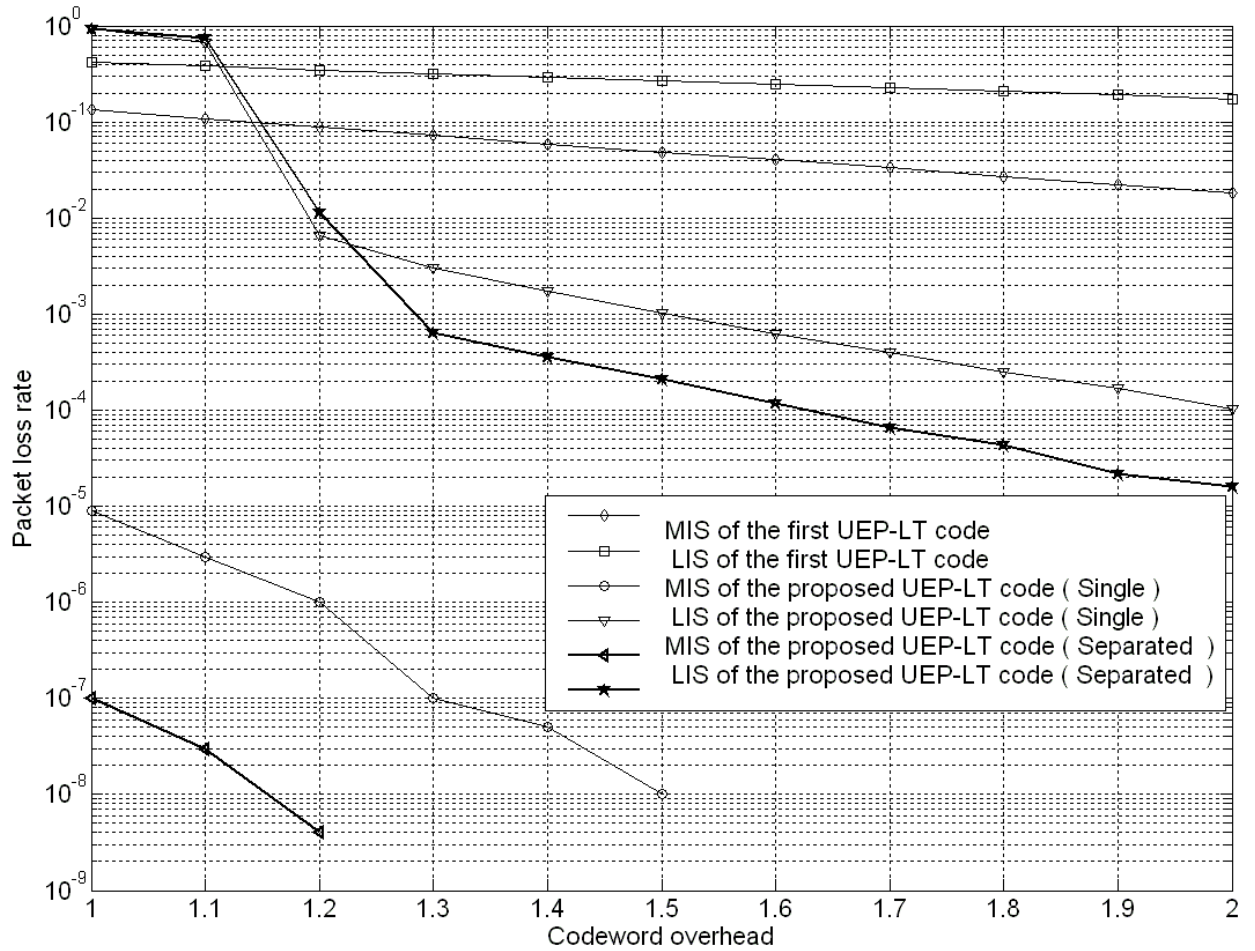
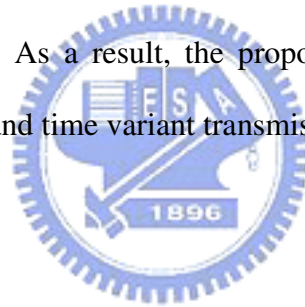


Figure 20. Simulation result shows packet loss rate of MIS and LIS using the UEP-LT code and proposed separable UEP-LT codes with single degree distribution and separated degree distribution. Parameters of the UEP-LT code are $K=10000$, $\alpha=0.1$, $K_m=2$ and parameters of both proposed separable UEP-LT code with single degree distribution and separated degree distribution are $K=10000$, $\alpha=0.1$. Y-axis and X-axis stand for packet loss rate and codeword overhead ($1+\epsilon$) respectively.

CHAPTER 5

CONCLUSION

We propose separable UEP-LT code that combines LT code and UEP in application layer but still retain the essence of rateless code. Besides, suggested decision of parameter c in robust soliton distribution can minimize codeword overhead and increase bandwidth efficiency. There are also additive capabilities such as unequal recovery time (URT) and distributed decoding that allows receivers only collect codewords they need. As a result, the proposed separable UEP-LT code is suitable for multiple channel and time variant transmission in application layer.



BIOBIBLIOGRAPHY

[1] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable Multimedia Download Delivery in Cellular Broadcast Networks" in *IEEE TRANSACTIONS ON BROADCASTING*, vol. 53, no. 1, Mar 2007.

[2] T. Gasiba, T. Stockhammer, J. Afzal, and W. Xu, "System Design and Advanced Receiver Techniques for MBMS Broadcast Services" at *the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings*.

[3] T. Gasiba, T. Stockhammer, J. Afzal, and W. Xu, "System Design and Advanced Receiver Techniques for MBMS Broadcast Services" in [2006 IEEE International Conference on](#) vol. 12, June 2006 pp:5444 - 5450 digital object identifier 10.1109/ICC.2006.255527

[4] U. DEMIR and O. AKTAS "Raptor versus Reed Solomon Forward Error Correction Codes" in *the Seventh IEEE International Symposium on Computer Networks (ISCN'06)*

[5] H. Jenkac, T. Stockhammer and W. Xu "Asynchronous and reliable on-demand media broadcast" in [Network, IEEE](#) vol 20, issue 2, Mar-Apr 2006 pp:14 - 20 digital object identifier 10.1109/MNET.2006.1607891

[6] D.J.C Mackay, "Fountain codes" in *IEE Proc.-Commun.*, vol. 152, no. 6, Dec 2005

[7] M. Mitzenmacher, "Digital Fountains: A Survey and Look Forward"

- [8] D. Vukobratovic, "On the Packet Lengths of Rateless Code" in [*Computer as a Tool, 2005. EUROCON 2005. The International Conference on*](#) vol 1, 2005 pp:672 - 675 digital object identifier 10.1109/EURCON.2005.1630019
- [9] M. Luby "LT Codes" in *the 43 rd Annual IEEE Symposium on Foundations of Computer Science (FOCS' 02)*
- [10] E. Maneva and A. Shokrollahi "New model for rigorous analysis of LT-codes" in [*Information Theory, 2006 IEEE International Symposium on*](#) July 2006 pp:2677 - 2679 digital object identifier 10.1109/ISIT.2006.262139
- [11] R. Karp, M. Luby, A. and Shokrollahi, "Finite Length Analysis of LT Codes" in [*Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*](#) 27 June-2 July 2004 pp:39 digital object identifier 10.1109/ISIT.2004.1365074
- [12] A. Shokrollahi, "Raptor Codes" in [*Information Theory, IEEE Transactions on*](#) vol 52, Issue 6, June 2006 pp:2551 - 2567 Digital Object Identifier 10.1109/TIT.2006.874390
- [13] P. Cataldi, M. P. Shatarski, M. Grangetto, and E. Magli, "Implementation and performance evaluation of LT and Raptor codes for multimedia applications" in *the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06)* 0-7695-2745-0/06
- [14] N. Rahnavard and F. Fekri, "Finite-length unequal error protection rateless codes: design and analysis" in [*Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*](#) vol. 3, 28 Nov.-2 Dec. 2005 pp:5 . Digital Object Identifier 10.1109/GLOCOM.2005.1577872
- [15] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Rateless Codes With Unequal Error Protection Property" in Nazanin Rahnavard; Badri N. Vellambi; Faramarz Fekri; *IEEE Trans Inf. Theory* , vol. 53, Issue 4, April 2007 pp:1521 - 1532 Digital Object Identifier 10.1109/TIT.2007.892814

作者簡歷

姓名：顏國光

出生地：台灣省彰化縣

出生日期：1983 年 3 月 27 日

學歷： 1989.9 ~ 1995.6 彰化縣鹿港鎮鹿港國小
1995.9 ~ 1998.6 彰化市私立精誠中學
1998.9 ~ 2001.6 彰化市私立精誠中學
2001.9 ~ 2005.6 國立交通大學 電子工程學系 學士
2005.9 ~ 2007.8 國立交通大學 電子研究所 系統組 碩士

