# 國立交通大學

## 網路工程研究所

## 碩士論文

CDRE - 利用網路編碼以降低封包錯誤率的機制

CDRE - Reducing Packet Error Rate by Utilizing Network Coding

研 究 生： 曾翊峰

指導教授： 趙禧綠 教授

中華民國九十七年八月

# CDRE - 利用網路編碼以降低封包錯誤率的機制
# CDRE - Reducing Packet Error Rate by Utilizing Network Coding


研 究 生 ：曾翊峰　　Student: Yi-Fong Tseng

指導教授：趙禧綠 教授　　Advisor: Prof. Hsi-Lu Chao

國 立 交 通 大 學
網 路 工 程 所
碩 士 論 文


A Thesis
Submitted to Institute of Network Engineering
College of and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

August 2008

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 七 年 八 月

# 摘　　要

無線訊號具有強烈的衰減和干擾效應，相對於有線訊號來說，顯得不穩定且不可靠。傳輸時的位元錯誤率會隨著訊噪比的下降而增加，過去實體層發展出來的各種頻道編碼技術，則是在訊噪比無法提升時用來降低位元錯誤率。而在 IEEE 802 家族中，位元錯誤率和封包長度完全左右了封包本身的錯誤率，也就是說，當位元錯誤率和封包長度固定時，封包錯誤率也隨之固定了。使用較長的封包在高速網路裡有更佳的效能，但是，較長的封包會讓封包錯誤率提高，為了不讓封包錯誤率太高，IEEE 標準裡制定了各種網路技術的封包長度上限，這限制了可用的封包長度。

此篇論文提出一個基於網路編碼的傳輸率增強機制，此機制可在位元錯誤率固定時再降低封包錯誤率，在連線品質惡劣時，可以用來提升傳輸率和產量，而且對於即時或非即時性的傳輸都有效。然而，這方法需要額外的運算和記憶體作為代價。透過模擬，顯示此機制在實體層沒有頻道編碼的情形下，傳輸時的訊噪比需求可降低約 2.4dB；搭配迴旋編碼的情況下則可以降低 1.8dB。

# Abstract


Wireless signals have strong fading and interference effects, therefore they are unstable and unreliable compared with wired signals. The bit error rate (BER) grows when the signal-to-noise ratio (SNR) gets low. Various channel coding technologies in physical layer were designed to reduce BER under the same SNR. In IEEE 802 families, the packet error rate (PER) of MAC layer is decided by BER and packet length. That means PER will be fixed if BER and packet length are both decided. Larger packet length brings better performance in high speed networks. However, the PER grows with the increasing of packet length. To against PER to be high, IEEE standards specified the max packet lengths of various network technologies. This restricts the raising of packet length.

In this thesis, we proposed a flexible coding scheme for delivery rate enhancement (CDRE) based on network coding, which brings a lower PER than before with the same BER. This scheme can be used to improve the delivery rate and throughput under a poor link quality, and applied to both traffic type of realtime and best-effort. Of course, it needs some additional memory and computations in return. Through the simulation, we will show that this scheme can reduce the SNR requirement of about 2.4dB for transmitting without any channel coding in physical layer, or 1.8dB in case of convolutional coding.



Index Terms — wireless, delivery rate, network coding, linear combination

# 致　謝

兩年的碩士生活就這樣過去了，回想兩年前剛完成大學學業的我，從課堂上學到的是幾十年前的理論、技術，進入研究所後，才開始去探索目前世界上最新的研究資料。從最初摸索各種領域，不斷找不同的文獻來研讀，這期間也見識到許多令人拍案的巧妙技術；花了大半年時間，才總算找到自己的興趣所在。接著才針對喜歡的領域下手，嘗試去結合不同的應用或服務。也許是因為前面花了許多時間研讀不同領域的文獻，所以正式進入這個領域時，不久就找到這個領域所缺乏的項目，然後就有了這一篇。

這兩年讓我得到的不只是知識上的見聞，更重要的是如何自己蒐集資料、發掘問題，並進而研究出解法的一連串能力，這大概也是碩士和學士最大的差別吧。也很慶幸自己能遇到一個很好的指導教授，能提供這樣的環境培養我。反觀同屆的另一個同學，他們實驗室裡大部分時間就只忙著趕計畫，兩年下來是讀了不少規格書，但到了現在卻對研究方面完全提不起興趣了。

這篇論文的完成，首先要感謝指導教授的鼓勵、提攜與包容。同時也沒有給我太多外務干擾，讓我可以專心在研究上。在進行此題目之前還有過幾段小插曲，可惜之前那些研究並沒有做出很好的結果，覺得辜負了指導教授的期望。還好最後總算在這個題目上得到了不錯的結果。另外感謝實驗室裡的夥伴陪我走過了這兩年，雖然我在熱鬧的場合會感覺較不自在，不過在實驗室的深夜裡，有你們的陪伴真好。還有要感謝發明網路編碼的大師們，讓我可以站上你們的肩膀，眺望得更遠。最後要多謝我的家人，尤其是老媽，一直在背後支持我，讓我不用分心顧慮其它事。唯一遺憾的是最掛心我的老爹在去年四月離開了，沒能看到我完成學業。

有耐心讀到這邊的人，請接下去閱讀這篇觀念簡單易懂的正文部份。

# Contents

# List of Tables

# List of Figures

# Chapter 1. Introduction

Radio provides a convenient way for the long distance and mobile communications. Many wireless communication technologies were used among our life, such as remote controller, cellular phone, GPS and satellite TV. People enjoy these technologies for a long time. Besides, scientists also keep on developing new technologies of longer range and more capacity communications.

The radio was initially used on transmitting analog messages. People found the analog signal through radio has strong effects of fading and interference. The Analog messages were often distorted after transmission. Thus they tried to use digitalization to lighten the signal attenuation. Besides, digital signals can be restore and regenerate at the intermediate nodes. As the digital signals were used, people want to improve link quality further. Then the channel code was invented.

## 1.1. Channel Code

When the radio is traveling over a long distance, it can be affect by many factors such as wind, buildings, cars and magnetic field. The signal may fade or distort during the traveling. The engineers often use forward error correction (FEC) to detect and correct errors in communications. We can say that FEC is used to provide a not bad BER for poor SNR. FEC contains two main categories, block code and convolutional code. The general procedure of FEC is to insert some redundant bits into the transmitted messages. These bits are calculated by a predetermined algorithm using the original messages. This allows the receiver to detect error and correct without retransmissions (within the limitation itself).

In general, more redundancy achieves higher reliability, and lower proportion of bandwidth for utilization. Different coding technologies have different characteristics and can be applied to distinct applications.

People use channel code in communication to get a low BER. Although BER is small, but not zero. Bit error always occurs when transmitting a great amount of data. To avoid the whole data from damage, people split the transmitting data into small packets. Each packet carries a little data, only one packet of data will loss when bit error occurs. The sender may re-transmit single packet only if it loss. How long of a packet is reasonable? The statistics of packet length at the network layer on Ethernet showed that the length of 1500 bytes was best. Nevertheless, some people oppugned that the length limitation of 1500 bytes was not suitable for high speed modern network. Video and audio data are extensively used. The behavior pattern of network today is very different from the past. They are trying to raise the packet length for a better network performance.

## 1.2. Packet Length

When our data are being delivered over network. They are actually splited in many small packets at the network layer. Then the routers store and forward these packets one by one. The packet length we use today is called the Maximum Transmission Unit (MTU), which is defined fixed by standard or decided at connect time (like pppoe). As we know, the Ethernet MTU is 1500 bytes, and that in 802.11 standard is 2272 bytes. The original MTU was designed as above because of high error rate and low speed communication in the past, but it is not so suitable for the high speed link (1Gbps or higher) today.

A higher MTU brings higher bandwidth efficiency. The Jumbo frames were proposed and used in Ethernet, they can carry up to 9000 bytes of payload. Jumbo frames spend less proportional of header overhead relative to the standard MTU 1500, and provide more payloads in a Round Trip Time (RTT). Thus Jumbo frames actually have better performance in high speed link.

As the packet size increases, PER grows too. Therefore we can say that, the MTU in wireless communication cannot be enlarged because of the high error rate of the medium. What about if we can reduce further the PER under the same link quality? Can we choose a larger MTU for higher performance communications? We'll discuss that in the next sections.

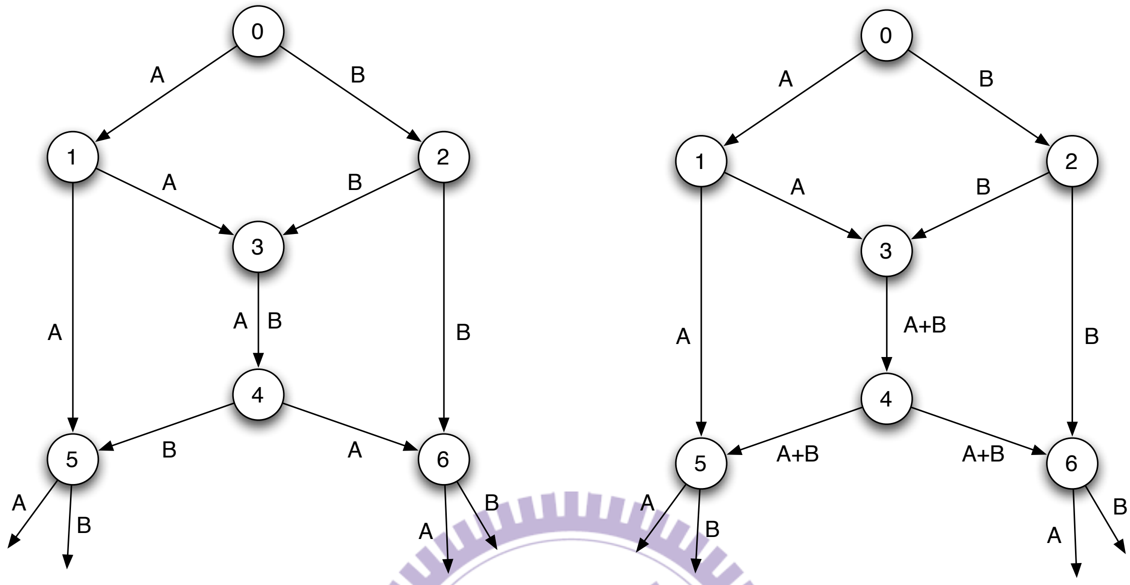
## 1.3. Network Coding

Network coding is a wonderful invention in communication area. It can improve network efficiency and reliability. It transfers the evidence of information instead of information itself. The core notion of network coding is contrary to our intuition, but really useful.

Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li and Raymond W. Yeung are the pioneers in this area [1, 2]. They thought about this idea 7 years ago, and they knew that a receiver will deduce the original information when enough evidence were collected. The receiver doesn't need to know all evidence in the network, and different pieces of evidence can replace each other. However, the most important of all is, the original information must be correctly restored by those collected evidence pieces.

Claude E. Shannon proved that each channel has its capacity (Shannon–Hartley theorem). A channel can provide reliability when the traffic in this channel is less than its capacity. For example, data transmission in a channel is like cars on a road. Cars from one side of road can safely outgo another side if the traffic is less than the capacity of road. Thus a channel provides limited traffic. However data are not cars, they can be combined in some way.

Fig.1-1 is the famous butterfly network. This easy example tells the superiority of network coding. A, B are data bit sent by node 0. Node 5 and node 6 are two sinks, which need to receive both A and B. The capacity of all edges are the same, which can deliver 1 bit in a time unit. From Fig.1-1(a), we know the edge between node 3 and node 4 is a bottleneck, because only this edge need to transfer 2 bits. In Fig.1-1(b), Network coding mixes A and B at node 3. Thus the data transmission between node 3 and node 4 is reduce from 2 bits (A and B) to 1 bit (A+B). Node 5 and node 6 can deduce the original information by the combination bit and the other bit they received. The link capacities are the same, but the traffic limitation is enlarged by network coding.




(a) Traditional routing (b) Network coding

Figure 1-1. Butterfly network

By such a notion, network coding replaces routers by encoders on the intermediate nodes. Relay nodes mix the received data and forward to the next. Finally, the destination deduces the original information by these mixed evidence. The computations network coding needs are only addition and multiplication. Thus, network coding provides larger traffic under fixed link capacity. Lots of evidence assures higher reliability, and only simple computations as requirement.

## 1.4. Motivation and Objective

As we described ahead, The wireless signals does highly fade and be easily distorted. People used various channel codes to provide low BERs in physical layer. Besides, they tried to use a large MTU in network layer for better performance. However the PER increases if the packet length grows. If we can keep the PER low with large MTU, the network performance will be improved more and more in modern high speed link.

Our target is to design a new coding scheme. This scheme provides a low PER and preliminarily decides the correctness of data. In addition, it must can be applied to various traffic type, simultaneously low computations are required.

## 1.5. Organization

The organization of this thesis is as follows: In chapter 2, we review the researches about network coding and some correlative applications. Then we will show CDRE in chapter 3. We also explain why and how this scheme can reduce PER in the same chapter. Then we run some simulations to confirm CDRE in chapter 4. Finally, we discuss the scope of application of CDRE in chapter 5.

# Chapter 2. Related Work

## 2.1. Network Coding

There are two common topics of network coding — XOR and linear combination. The XOR operation is usually used on physical layer, because of its properties of easiness, simple and high speed operation. It can be implement as hardware circuit. Contrarily, the linear combination is most used on application layer, which is a little more complex and needs more computations.

### 2.1.1. XOR Operation

The XOR of network coding was first described in [1]. To explain briefly by a simple equation: $A \oplus (A \oplus B) = B$. When I owned data $A$, I will get data $B$ if I receive data $(A \oplus B)$. This simple characteristic let different nodes to get different data from the same broadcast message, if they had different data before. Several transmissions would be done by one broadcast. According to this, many researches about XOR were made [6-8]. In 2006, S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft successfully implemented the practical wireless network coding [5]. They proposed COPE to realize the XOR operation of network coding, and run it on the 802.11 roofnet. The routers in COPE mix packets based on four topologies. The COPE really gave us an impressive performance improvement.

Fig.2-1 shows the scenario of data exchange. Data $x$ is sent from node $A$ to node $B$ via node $R$, $y$ is sent from $B$ to $A$ via $R$. In traditional routing as Fig.2-1(a), $A$ will send $x$ to $R$, then $R$ forward it to $B$. Similarly, $B$ send $y$ to $R$, then $R$ forward it to $A$. This exchange will take 4 transmissions.

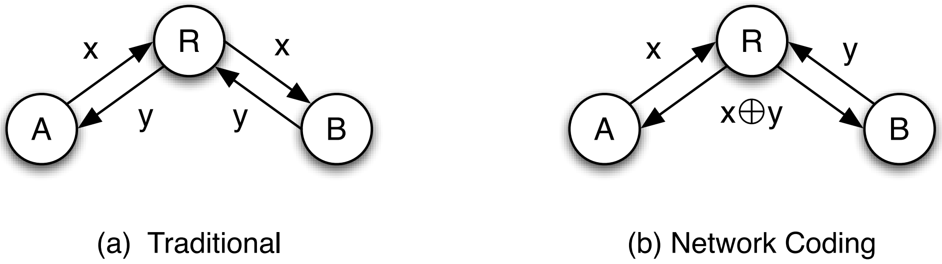



Figure 2-1. Application of XOR

In network coding as Fig.2-1(b), both *A* and *B* will send data to *R* first, then *R* mix the data *x* and *y* into $(x \oplus y)$. Node *R* consequentially broadcast $(x \oplus y)$. When *A* receive $(x \oplus y)$, it will get *y* easily if it takes XOR operation again on the *x* and $(x \oplus y)$. Similarly, *B* can get *x* from *y* and $(x \oplus y)$. Such that, two transmissions are complete in a broadcast. The exchange in network coding takes only 3 transmissions. This easy example shows that network coding can reduce the number of transmissions in the same situation.

### 2.1.2. Linear Combination Operation

The Linear combination of network coding was usually used on large data file. The most common applications is P2P sharing [9, 10]. A source node creates a lot of linear combinations and sends to different destinations. The destinations share all linear combinations each other. The advantage of linear combination is that destinations collect data in high speed. The source creates an unique linear combination for each destination, such that each destination can poll data from all nodes. In other word, all nodes will become source, any pieces of linear combination is useful. In traditional way, destinations can only poll data from nodes who has the different data blocks. The collecting speed falls down as the percentage of collection increasing. Because the nodes who have the remainder data blocks are fewer and fewer. Linear combination has a main problem: the collector cannot know any info of the data before completion. Thus it doesn't support preview before the whole data are completely restored.

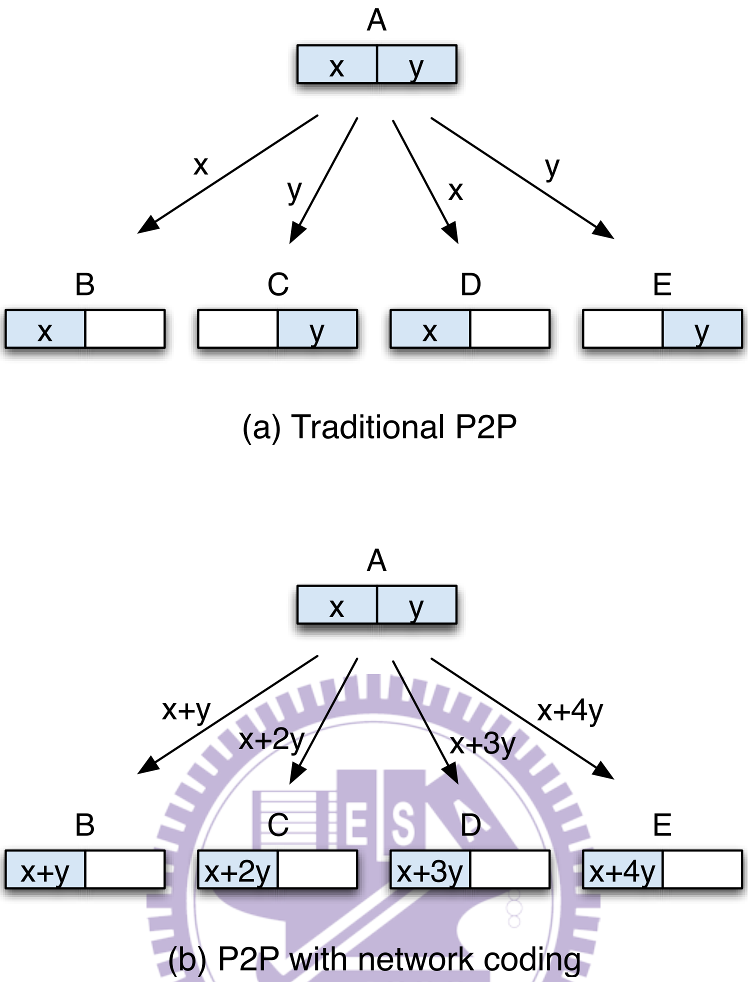



Figure 2-2. Application of linear combination

Fig.2-2 show a scenario of P2P sharing. Host *A* is the source, the others are destinations. In traditional P2P, host *A* splits the sharing data and randomly sends a piece of data to a destination. Assume the host *B*~*E* have received a piece from *A*, as shown in Fig.2-2(a). Host *B* has *x* now, so it can only get new data from the hosts who have *y*. The target host set of *B* is {*A*, *C*, *E*}. Host *D* owns *x* as *B* does. *B* cannot get new data from *D*, so *D* is an useless host for *B*.

In case of network coding, host *A* sends a linear combination to each destination. Each destination has received an unique data from *A*, as drew in Fig.2-2(b). Let's take a look on host *B*, *B* has a data piece *(x+y)*, no other hosts have the same data as *B*. *B* can get new data from all other hosts, and it will restore the mixed data to the original if it collects enough data pieces. The source host set of $B$ is $\{A, C, D, E\}$. P2P with network coding allows the peers to get new data from more sources than the traditional one. It can certainly improve the sharing efficiency.

## 2.2. Delivery Rate

The researches about delivery rate were almost focus on physical layer. Only a few people develop delivery rate on MAC layer. Chi-hsien Lin, Hui Dong, Upamanyu Madhow and Allen Gersho proposed an interesting method in 2004 [11]. Their research was to combine two little data fragment, then sent redundancy for a higher delivery rate. The data loss occurs only if two continuous packets were both loss. This easily modification makes the loss rate from p to $p^2$. Besides, the intermediate nodes could regenerate the loss packet by its previous and next packets. These two graceful schemes keep the loss rate much lower than the original. However, the combination of fragments in their research can only apply on small data fragments (real-time voice). The overhead will be twice if it is applied to the general size packets.

Szymon Chachulski, Michael Jennings, Sachin Katti and Dina Katabi proposed MORE in 2007 [13]. MORE is an improved version of multi-path routing protocol based on ExOR [12]. It transmits a lot of packets in a batch. The packets in the same batch will be encoded by linear combination. These coded packets were sent to destination via different paths. The relay nodes will forward the packets they received correctly. By the characteristic of linear combination, MORE decreases the number of retransmission of ExOR. The delivery rates of a packet in MORE and ExOR are the same, but MORE improves the delivery ratio of a batch of packets indeed.

The MORE encodes a batch of packets. The data can only be decoded when the whole batch was received. Thus MORE cannot support real-time traffic. Besides, the MORE improves the delivery ratio of a batch, which may over several hops. We want to designed a method which improves the delivery rate in single hop for all traffic types.

# Chapter 3. CDRE

We adopt the notion of network coding, but not to mix the data from different sources. We mixed the data of the same frame. According to the characteristic of linear combination, we can solve a number of unknown variables by the same number of linear independent equations [3]. Thus we design a scheme based on linear combination, which can reassemble correct pieces to the original message. Describing more exactly, the sender sends redundancy sectors of the same frame to prevent frame broken from a few error bits during transmission. When the frame is received, the receiver may restore the original message if the frame is lightly damaged.

This method can be implemented between the PHY layer and MAC layer. We insert a sub-layer to handle the processing of network coding. This NC-layer converts the native frames into coded ones when sending, and reverts to the original frames when receiving. The detail is shown below.

## 3.1. Encoding

When a MAC frame is formed, the NC-layer divides the frame into $n$ sectors of equivalent length. We can treat a sector as a very long positive integer. Each sector is multiplied by a random coefficient. Then we sum up all the products. The summation is called a linear combination (LC) of the frame. Repeat the same operation for $(n+j)$ times. Here the $n$ is the minimum number for restoring data, and $j$ is the number of redundancy used to improve delivery rate. In addition, we use a predefined coefficient matrix, which is mainly generated by random number. This matrix must satisfy the following three conditions.

i. The elements in the first row must be all *1*s.

ii. Each coefficient in matrix is an integer of size *1* bytes.

iii. All rows must be linear independent.

The coefficient matrix has the size of $n \times (n+j)$. This matrix was agreed by both nodes when they were building connection. Then all packets on this connection use the same matrix for encoding. Different sender-receiver pairs may use different matrixes for security.

The encoding equation is shown as equation (1). $A_1 \sim A_n$ are sectors of the original frame. $LC_1 \sim LC_{n+j}$ are linear combinations to form a new coded frame.

$$\begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & \cdots & C_{1,n} \\ C_{2,1} & C_{2,2} & C_{2,3} & \cdots & C_{2,n} \\ C_{3,1} & C_{3,2} & C_{3,3} & \cdots & C_{3,n} \\ & \vdots & & & \vdots \\ C_{n,1} & C_{n,2} & C_{n,3} & \cdots & C_{n,n} \\ C_{n+1,1} & C_{n+1,2} & C_{n+1,3} & \cdots & C_{n+1,n} \\ & \vdots & & & \vdots \\ C_{n+j,1} & C_{n+j,2} & C_{n+j,3} & \cdots & C_{n+j,n} \end{bmatrix} \times \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_n \end{bmatrix} = \begin{bmatrix} LC_1 \\ LC_2 \\ LC_3 \\ \vdots \\ LC_n \\ LC_{n+1} \\ \vdots \\ LC_{n+j} \end{bmatrix} \qquad (1)$$

Assume the length of a MAC frame is $L$ bytes, then the length of each sector is $\left\lceil \frac{L}{n} \right\rceil$ bytes. We control every coefficient to be an 1-byte integer. Then the product of a sector and coefficient has a length of $\left\lceil \frac{L}{n}+1 \right\rceil$ bytes. In addition, 1-byte is reserved for the add carry, therefore the length of an LC should be $\left\lceil \frac{L}{n}+2 \right\rceil$ bytes of maximum. Besides we append a CRC-16 to check the correctness of LC, then the max length of an LC-sector becomes $\left\lceil \frac{L}{n}+4 \right\rceil$ bytes. All LCs of the same frame are concatenated together to form a new coded frame. We also need to identify how this coded frame is formed, therefore a coded header is needed. We define a 4-bytes header to identify the frame. This header include three fields ($L$, $n$, $j$). Therefore, the length of a coded frame is actually $4+\left\lceil \frac{L}{n}+4 \right\rceil (n+j)$ bytes. We spend $4+\left\lceil \frac{L}{n}+4 \right\rceil (n+j) - L$ of overhead in payment for reducing PER. Besides, ($n$, $j$) is set to (1, 0) in order to run as the original mode under good SNR. The whole encoding process in NC-layer is shown in Fig.3-1.

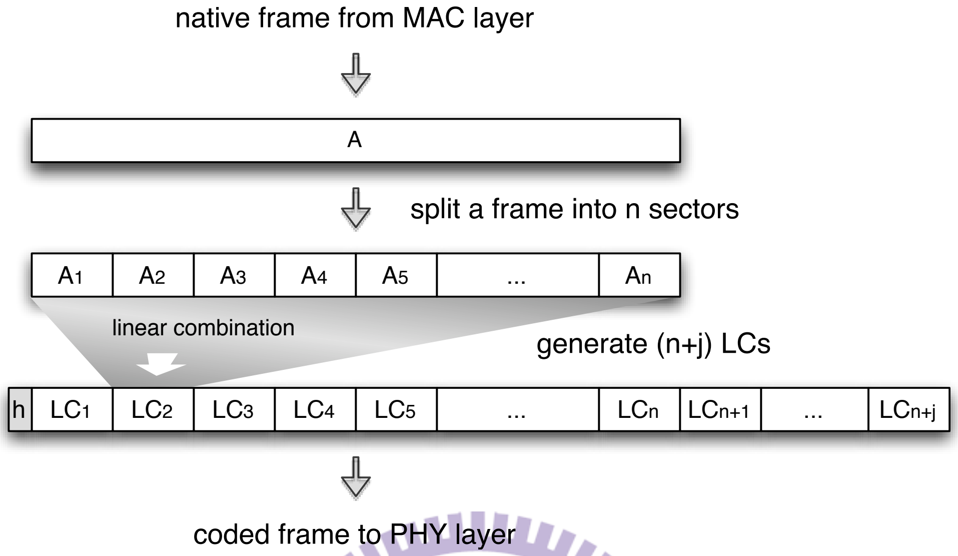



Figure 3-1. Encoding process

## 3.2. Decoding

When a coded frame comes from physical layer. The NC-layer first checks the header for its organization. If the size of this frame can not fit to the information in header, then the header is possible to be broken. The frame must be discarded if its header is broken. As the frame had passed the first header check, we need to check the correctness of each LC part by its CRC following up. If the number of damaged LCs was larger than $j$, then this frame must be unusable. Otherwise, If the number of correct LCs was $n$ or more, then we will be able to decode it by Gaussian elimination.

Suppose the set $LC_X$ contains only correct LCs, and the size of $LC_X$ is $n$. Of cause, $LC_X$ is a subset of all LCs transmitted by sender. We can trace the rows used in coefficient matrix by $LC_X$. Assume the collection of these corresponding rows of coefficient matrix is $M_X$. The encoding process can be denoted as $LC_X = M_X \times A$, or detail as equation (2). We can easily multiply both sides of this equation by $M_X^{-1}$, the inverse matrix of $M_X$, then we will get the original message A. This decoding process can be denoted as $M_X^{-1} \times LC_X = M_X^{-1} \times M_X \times A = A$, or detail as equation (3). Rows in predefined coefficient matrix are all linear independent, therefore $M_X$ always has an inverse matrix, the original message A can always be able to decode from $LC_X$. The whole decoding process is shown in Fig.3-2.

$$\begin{bmatrix} LC_1 \\ LC_2 \\ LC_4 \\ \vdots \\ LC_{n+j} \end{bmatrix} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & \cdots & C_{1,n} \\ C_{2,1} & C_{2,2} & C_{2,3} & \cdots & C_{2,n} \\ C_{4,1} & C_{4,2} & C_{4,3} & \cdots & C_{4,n} \\ & \vdots & & & \vdots \\ C_{n+j,1} & C_{n+j,2} & C_{n+j,3} & \cdots & C_{n+j,n} \end{bmatrix} \times \begin{bmatrix} A_1 \\ A_2 \\ A_4 \\ \vdots \\ A_n \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & \cdots & C_{1,n} \\ C_{2,1} & C_{2,2} & C_{2,3} & \cdots & C_{2,n} \\ C_{4,1} & C_{4,2} & C_{4,3} & \cdots & C_{4,n} \\ & \vdots & & & \vdots \\ C_{n+j,1} & C_{n+j,2} & C_{n+j,3} & \cdots & C_{n+j,n} \end{bmatrix}^{-1} \times \begin{bmatrix} LC_1 \\ LC_2 \\ LC_4 \\ \vdots \\ LC_{n+j} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ A_4 \\ \vdots \\ A_n \end{bmatrix} \tag{3}$$

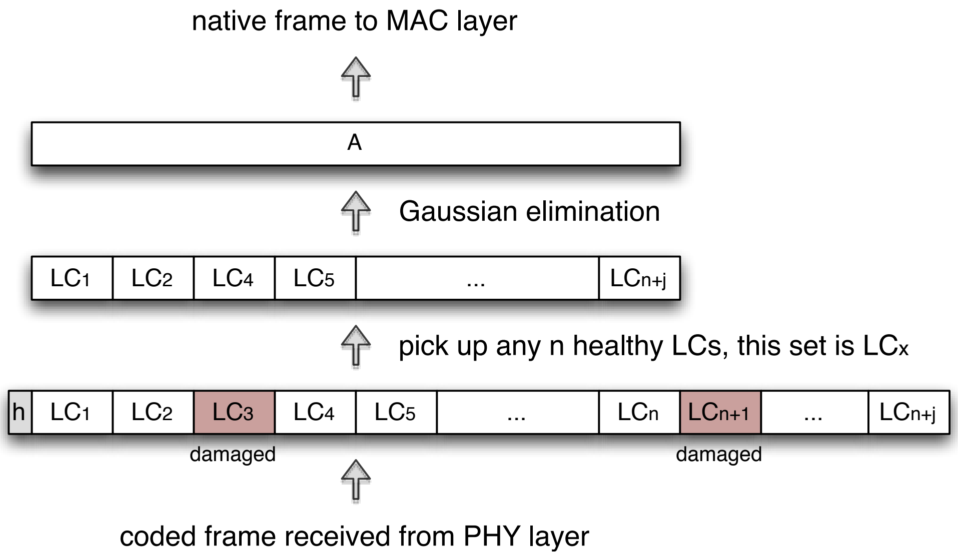



Figure 3-2. Decoding process

Let's take a look at the Gaussian elimination. The computational complexity of Gaussian elimination is $O(n^3)$, which is really big. We proposed another method to reduce the computational complexity. We use a cache to save the inverse matrixes of possible $M_X$, and we call it Gaussian elimination cache (GE cache). Because all frames are encoded by the same coefficient matrix, we can reuse the $M_X^{-1}$ calculated by previous frames to decode new frames if they have the same correct sectors. The coefficient matrix is fixed, then the inverse matrixes are fixed too. This will turn the decoding process of Gaussian elimination into a simple matrix multiplication. Thus the computational complexity can be downgraded to $O(n^2)$.

## 3.3. Analysis

Why the method we proposed has a higher delivery rate than the standard in IEEE 802.x? We derive the delivery rate equation for single packet in two schemes, these equations will show us clearly.

### 3.3.1. Comparison of Native and Network Coding PERs

First, let's take a look at the native transmission. When a MAC frame is transmitted from sender to receiver, the frame gets broken if any bit error occurs. The MAC layer often use CRC to check the correctness of a packet. The CRC failure probability is no worse than $\frac{BER}{2^k}$ (k-bit CRC), which is very small and can be negligible [4]. We assume the CRC check is always right. Therefore the error probability of a native frame, $P_{native}$, is shown as equation (4). Where $BER$ is the bit error rate, $L$ is the length of a native frame (in the unit of bytes).

$$PER_{native}=1-(1-BER)^{L\cdot 8} \tag{4}$$

Then take up our method. We divide a frame into $n$ sectors, so the sector error rate ($SER$) is defined as equation (5). There are $(n+j)$ LCs in a coded frame. The frame can be restored if the number of damaged LCs is less than $j$. However, it cannot be restored if the header is broke, even all LCs are correct. Considering the both cases and we will get the error rate of a coded frame as equation (6). Where $h$ is the length of a coded header ($h$ is 4 bytes here as we defined).

$$SER_n=1-(1-BER)^{\left\lceil \frac{L}{n}+4 \right\rceil 8} \tag{5}$$

$$PER(n,\, j)=[1-(1-BER)^{h\cdot 8}] + (1-BER)^{h\cdot 8}\cdot\left[1-\sum_{k=0}^{j}\binom{n+j}{k}\cdot(1-SER_n)^{n+j-k}\cdot SER_n^{k}\right] \tag{6}$$

Let's compare the $PER_{native}$ and $PER(n, j)$ by drawing out their curves together. Fig.3-3 shows the theoretical value of PER to SNR. (The $BER$ we used here was calculated by the function of berawgn in MATLAB®.) We can see that coding PERs will be always smaller than native one. The more redundant LCs were sent, the lower PER in return, and absolutely more overheads were paid. This payment is worthy for throughput improvement under poor link quality.

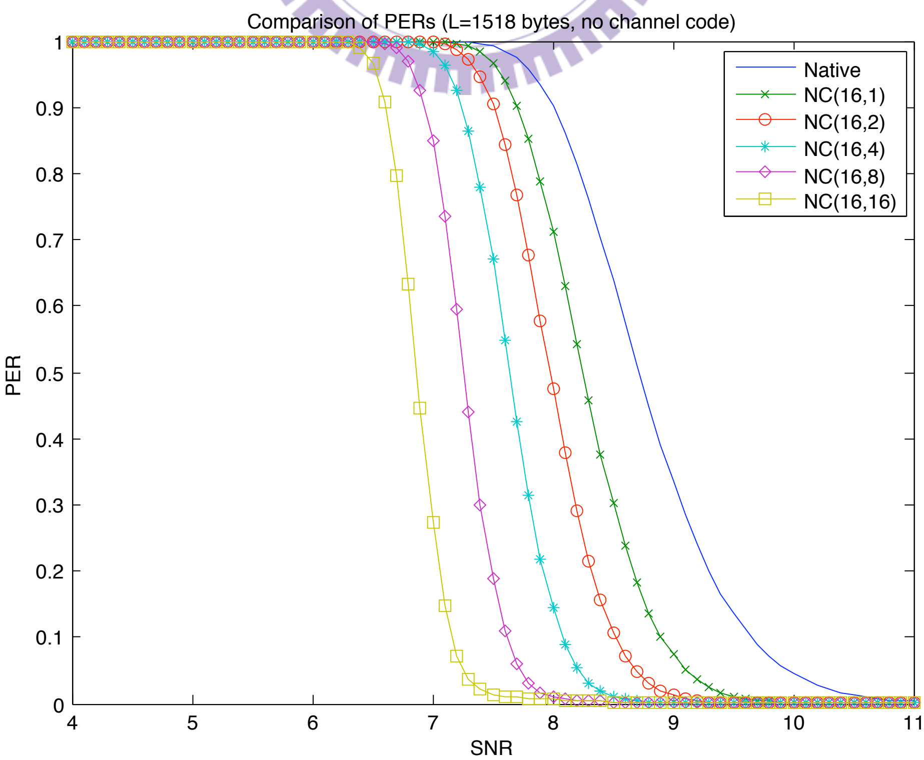



Figure 3-3. Comparison of native and network coding PERs with individual $j$

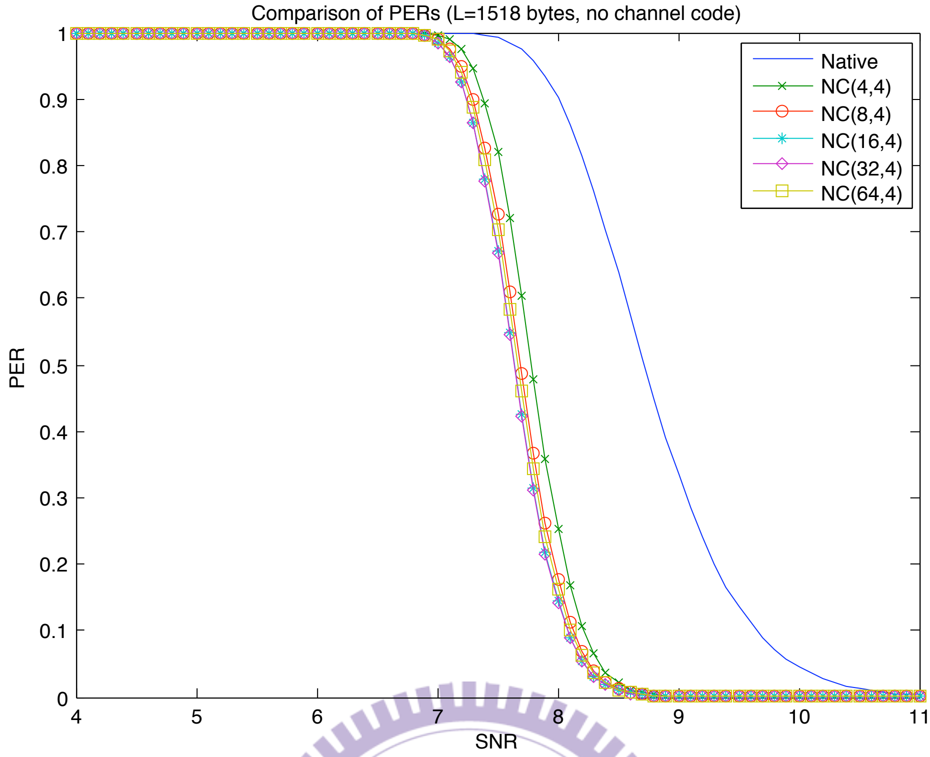



Figure 3-4. Comparison of native and network coding PERs with individual $n$

In Fig.3-4, we fix the variable $j$ to see the influence of different $n$. We observe that the coding PERs are almost the same. This is because the individual SERs in this case are very close. We need to know some thing first. A greater $n$ takes more computations and more overhead of CRCs, but it simultaneously brings a shorter sector length, which means lower SER. This is a trade-off between overhead and computations.

### 3.3.2. Retransmission

A sender often resends the previous frame if it didn't receive an ack. After a max number of retry, it will drop the frame if still no ack returns. This is the most common way for wireless technologies to improve reliability at MAC layer.

Assume the PER of a native frame is $P_{native}$. A frame will be dropped after $R$ failures, thus the overall failure probability is $P^{R}_{native}$. We draw the curve for $R$={1, 2, 3, 4}, as Fig.3-6.

Through Fig.3-6, we observe the delivery rate of $PER(16,2)$ being better than the Native(R=4). This result is obvious. The value of $P_{native}$ is 0.8988 at the SNR=8, such that the failure probability is still 0.6526 after 4 sending attempts. However, with more redundancy, the $PER(16,8)$ can easily reduce the failure probability to 0.0098 at SNR=8. The overhead of retransmission is 100% at each retry, but only about $\frac{j}{n}$ in CDRE.

The retransmission mechanism is to improve delivery rate under the same PER. CDRE directly reduce the PER. With a well-designed ($n$, $j$) pair, we even don't need any retransmission at MAC layer in CDRE.

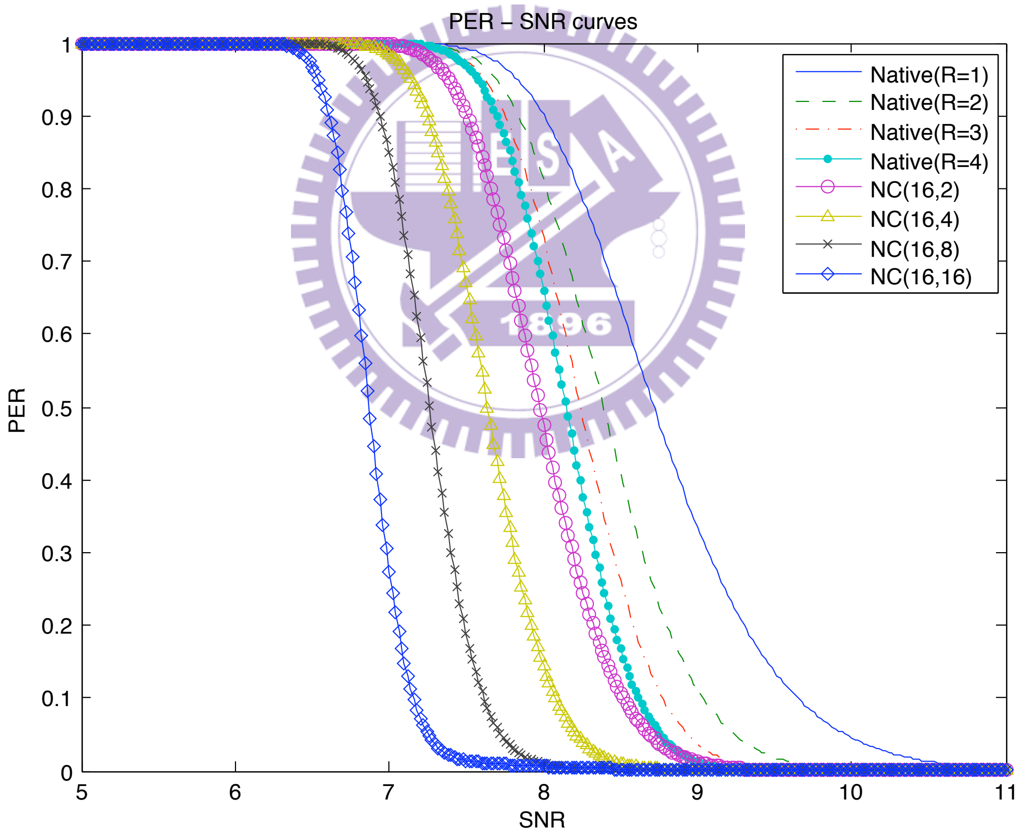



Figure 3-6. The effect of retransmission.

### 3.3.3. PER with Large MTU

Then we consider the case of jumbo frames. The size of jumbo frames is 9KB of maximum. The jumbo frames have higher PERs than that of original MTU in the same environment. According to equation (4) and (6), we can also draw out the figure as Fig.3-5.

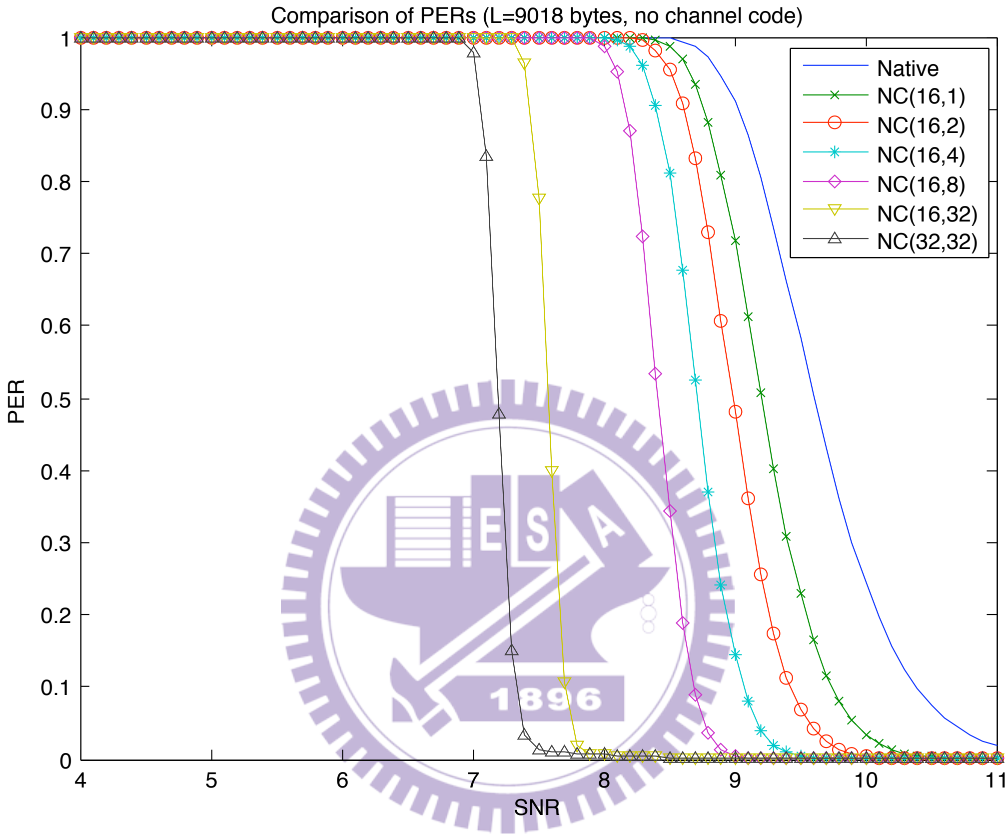



Figure 3-5. Comparison of the PERs of jumbo frames

Jumbo frames are 6 times the size of the original. We may further increase the values of $n$ and $j$ in order to get a smaller sector and a lower PER. Generally, increasing $j$ to get a lower PER, and increasing $n$ to reduce the proportion of redundancy overhead. However, the amounts of computation must not exceed that node can handle.

## 3.4. Transmission Example

We use a simple example to tell the superiority of our method in another point of view. We set the SNR to two states, as shown in Fig.3-7. The SNR changes sharply when state is transitting, but gradually in the same state.

Fig.3-8(a) recorded the SNR history in some place. We calculated the corresponding BER of each time slot, such as shown in Fig.3-8(b). According to the BER, we run a program to randomly decide the error bits on the time line. The result is shown in Fig.3-8(c). Each line in Fig.3-8(c) represents an error bit. This is the setting of environment so far.

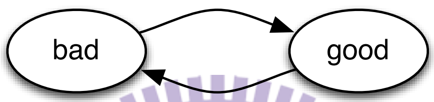
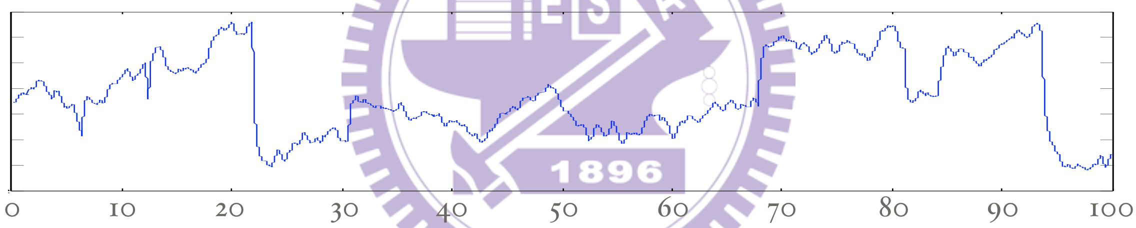
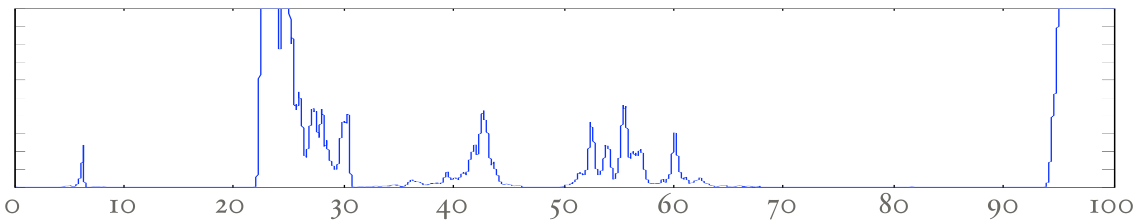


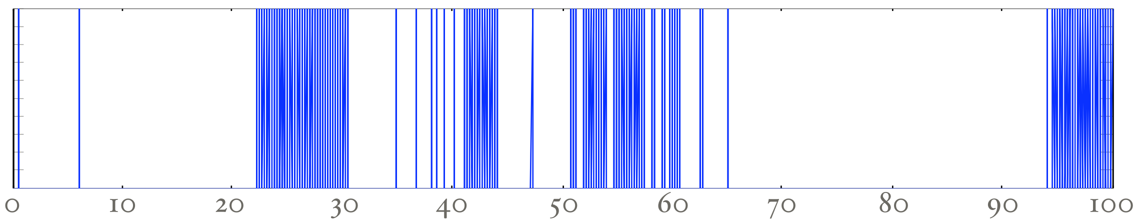

Figure 3-7. SNR state transition diagram




(a) SNR history




(b) BER history

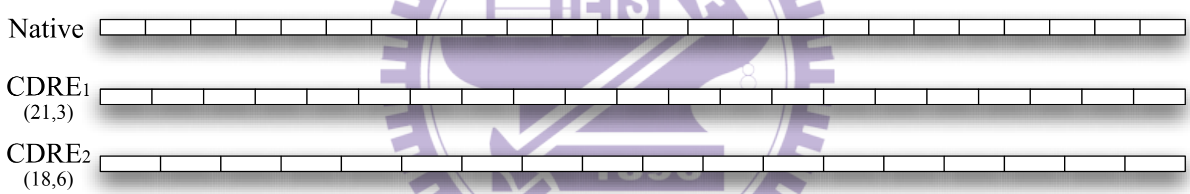



(c) Error bits history, each line represents an error bit

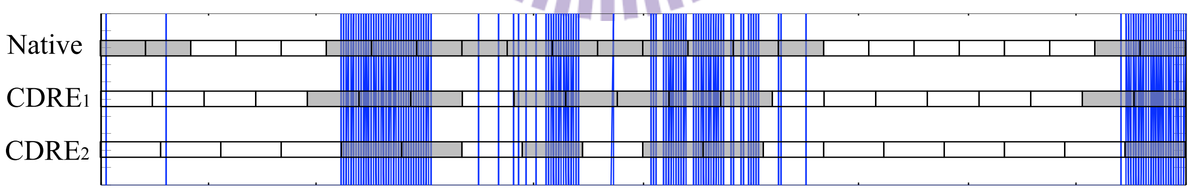
Figure 3-8. Environment example

Then let's take a comparison of the native and CDRE transmissions. Suppose we have a sequence of packets to be sent, as shown in Fig.3-9(a). We chose $(n, j)$=(21, 3) for the $CDRE_1$, (18, 6) for $CDRE_2$. The overhead of $CDRE_1$ is about 1/7 of packet length to the native, and 1/3 in $CDRE_2$. (The packet size is actually not so long in the schematic diagram, but the same principle.)

The native packets will be broken if any bit error occurs, but the coded packets have a tolerance of three broken LCs. Then the result of transmissions is shown as Fig.3-9(b). White cells are successful transmissions, and gray ones are failure. In this example, the native scheme has 9 successful transmissions. Such as 11 in $CDRE_1$, 12 in $CDRE_2$. It shows that CDRE achieves high delivery rate indeed. The comparison of delivery ratio and overhead in this example is shown in Table 3-1.




(a) Packets streams of two schemes




(b) Transmissions in two schemes

Figure 3-9. Comparison of transmissions

Table 3-1. Comparison of traditional and CDRE

| Scheme | Total TX | Successful TX | Delivery Ratio |
|---|---|---|---|
| Native | 24 | 9 | 37.5% |
| $CDRE_1$ | 21 | 11 | 52.4% |
| $CDRE_2$ | 18 | 12 | 66.7% |

## 3.5. Discussion

The increment of $n$ will reduce SER, but simultaneously bring a little more CRC overhead. The increment of $j$ will improve delivery rate, but bring more redundancy overhead. Both values of these two variables are trade-offs, and need to be designed to fit to the particular network. In general, the PER will be minimized if both $n$ and $j$ are maximized. However, large $n$ and $j$ will need huge computations. Our suggestion is to choose by the limitation according to the node's abilities of computation and memory. Such that it will work in an acceptable workload to improve network performance.

The GE cache, we used for high speed decoding, will take huge memory space if both $n$ and $j$ are large. Besides, it seems no way for the GE cache to have a high hit rate with less memory. The broken LCs are randomly distributed, we cannot determine which inverse matrix is more useful or not. Such that, GE cache cannot work in this situation.

Therefore we plan to design a Hierarchical-CDRE (H-CDRE) to reduce the computation complexity in the future. For example, if we divide directly the packet into 25 sectors. The computation complexity of Gaussian elimination is a degree of ($25^3$). However, if we divide it by two hierarchies, of which 5 sectors, as Fig.3-10. The computation complexity will become a degree of ($6\times5^3$). Thus, the computation requirement of H-CDRE will be downgrade to an acceptable level. Of course, the memory requirement will also downgrade, GE cache can be applied to large ($n$, $j$) pair via H-CDRE. Besides, we also consider applying pipeline to the encoding and decoding process in H-CDRE. Such that, the coding speed will be improved further.

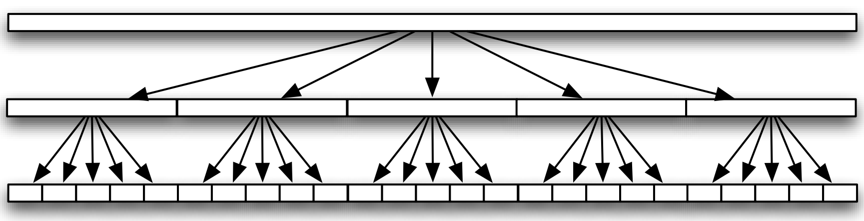


Figure 3-10. Hierarchical-CDRE diagram

# Chapter 4. Simulation

## 4.1. Environment Setting

We want to compare the delivery rate and throughput of our method with the original. The simulation environment parameters are listed in Table 4-1. The setting is mainly based on 802.11 standard. We only concern single-hop transmission, so we command the sender to transmit data forever during the simulation time.

Table 4-1. Simulation environment setting

| | |
|---|---|
| channel model | AWGN |
| SNR range | bad state: 0~5, good state: 5~10 |
| SNR state transition period | 10 seconds in average |
| medium bandwidth | 2 Mbps |
| frame length | 2304 bytes (MSDU of 802.11) |
| simulation time | 1000 seconds |

Table 4-2. Mode setting for different channel code.

| no channel code | |
|---|---|
| **(n, j)** | **SNR** |
| (1, 0) | >9.85 |
| (16, 1) | 9.25~9.85 |
| (16, 2) | 8.75~9.25 |
| (16, 4) | 8.25~8.75 |
| (16, 8) | 7.75~8.25 |
| (16, 16) | <7.75 |

| convolutional code | |
|---|---|
| **(n, j)** | **SNR** |
| (1, 0) | >7.35 |
| (16, 1) | 6.75~7.35 |
| (16, 2) | 6.45~6.75 |
| (16, 4) | 6.05~6.45 |
| (16, 8) | 5.75~6.05 |
| (16, 16) | <5.75 |

## 4.2. Simulation Result

We first ran all $(n, j)$ pairs in full SNR range. Then we chose the best mode in each strait SNR range. The mode setting we used is shown in Table 4-2. This is optimized for highest throughput.

Fig.4-1 shows the SNR history of the whole simulation time. The SNR hopped between bad and good states about every 10 secs. According to Fig.3-3, we realized that the original scheme would transmit successfully only when the SNR is higher than 7.5dB. (The $PER_{native}$ approximates to zero when SNR < 7.5dB) Our method is 6.3dB relatively.

Let's compare the Fig.4-1 with Fig.4-2. At time interval 100~200 sec, the original scheme only had successful transmissions at the time of 150 sec. However, The successful transmissions grew almost all the time interval in our method. This simulation result verifies the analysis in section 3.3.1. That is, our method certainty reduces the requirement of SNR for transmission.

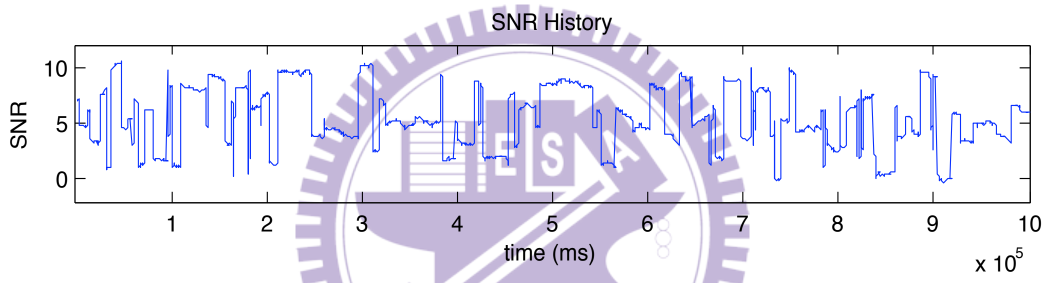



Figure 4-1. SNR history (no channel code)

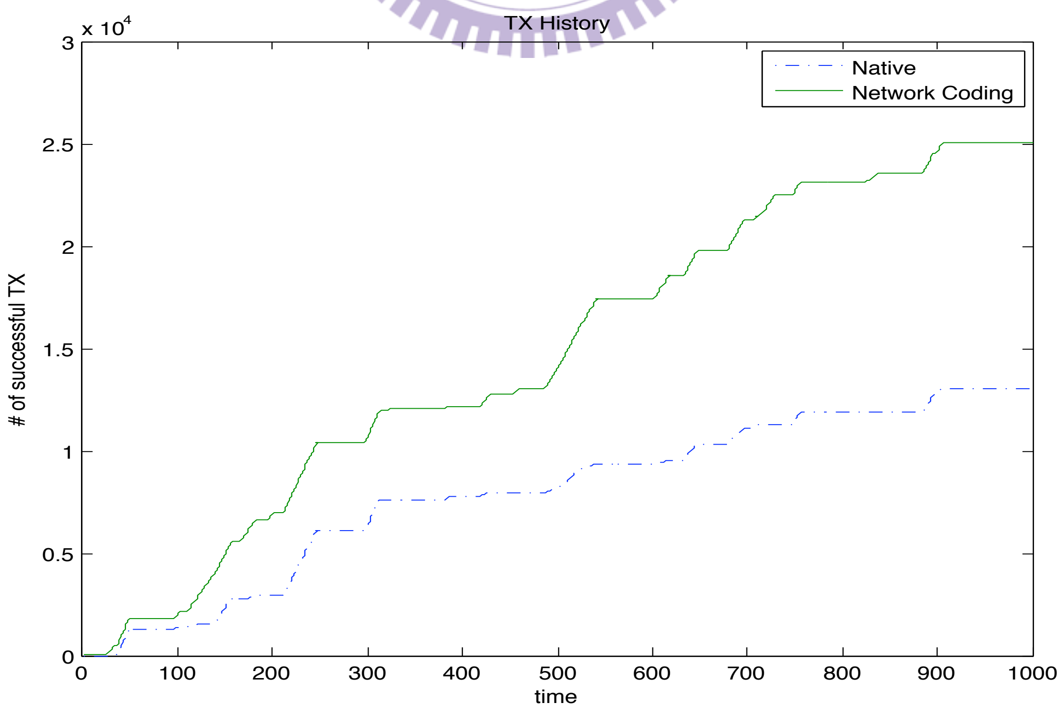



Figure 4-2. Accumulative number of packets (no channel code)

The accumulative throughput of this simulation is in Fig.4-3. We can observe the throughput of our method is always higher than the original. The transmission became stable after about 600 secs.

Fig.4-4 shows that overhead we used to reduce PER of successful transmissions. Total amounts of data transmitted in native scheme is 239 Mbits, and that in our method is 461 Mbits. In these successful transmissions, the overhead of redundancy is 130 Mbits. That means, our method actually transmitted a amount of total 591 Mbits for these successful transmissions in order to improve the delivery ratio.

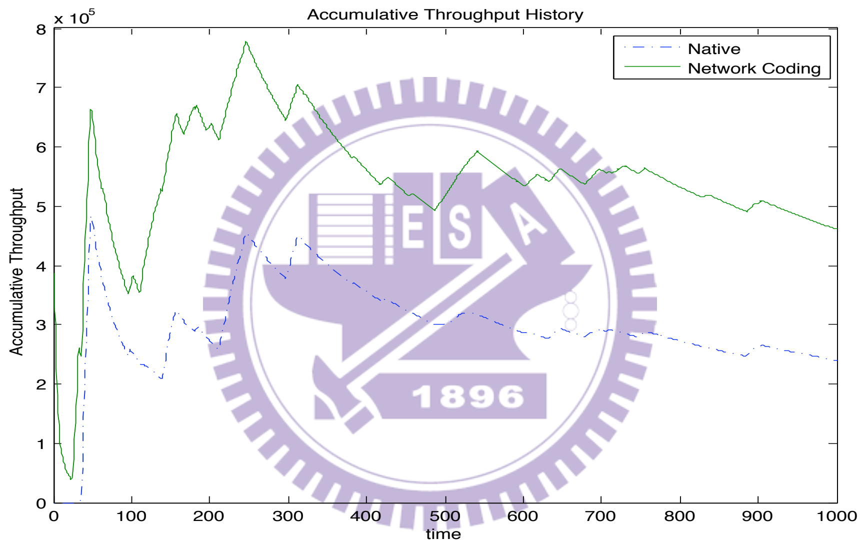


Figure 4-3. Accumulative throughput history (no channel code)

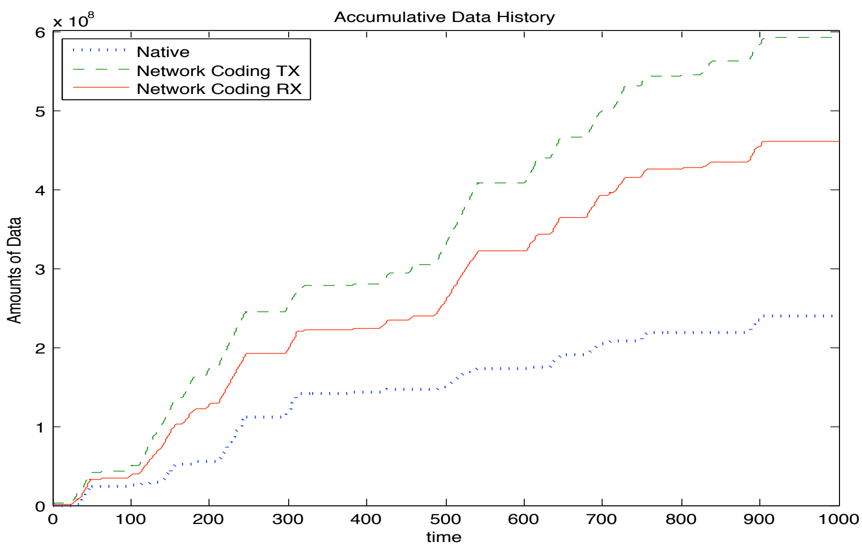


Figure 4-4. Proportion of bandwidth used by redundancy (no channel code)

Fig.4-5 and Fig.4-6 are the simulation results with convolutional code. The convolutional code provides a lower BER under the same signal quality. The requirement of SNR for transmission had been reduced by convolutional code. Our method can be still applied to it, and reduce the SNR requirement further.

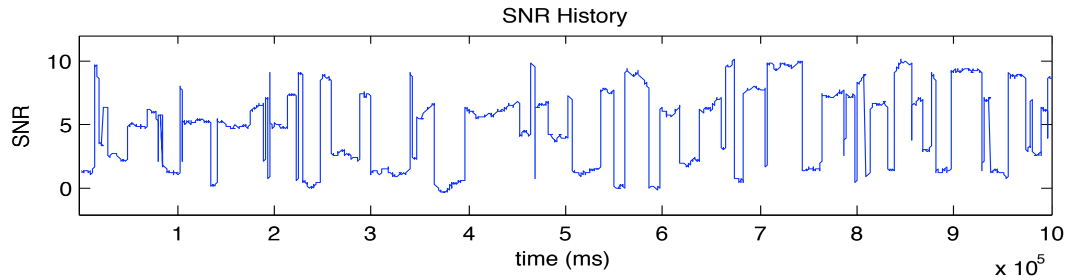



Figure 4-5. SNR history (convolutional code)

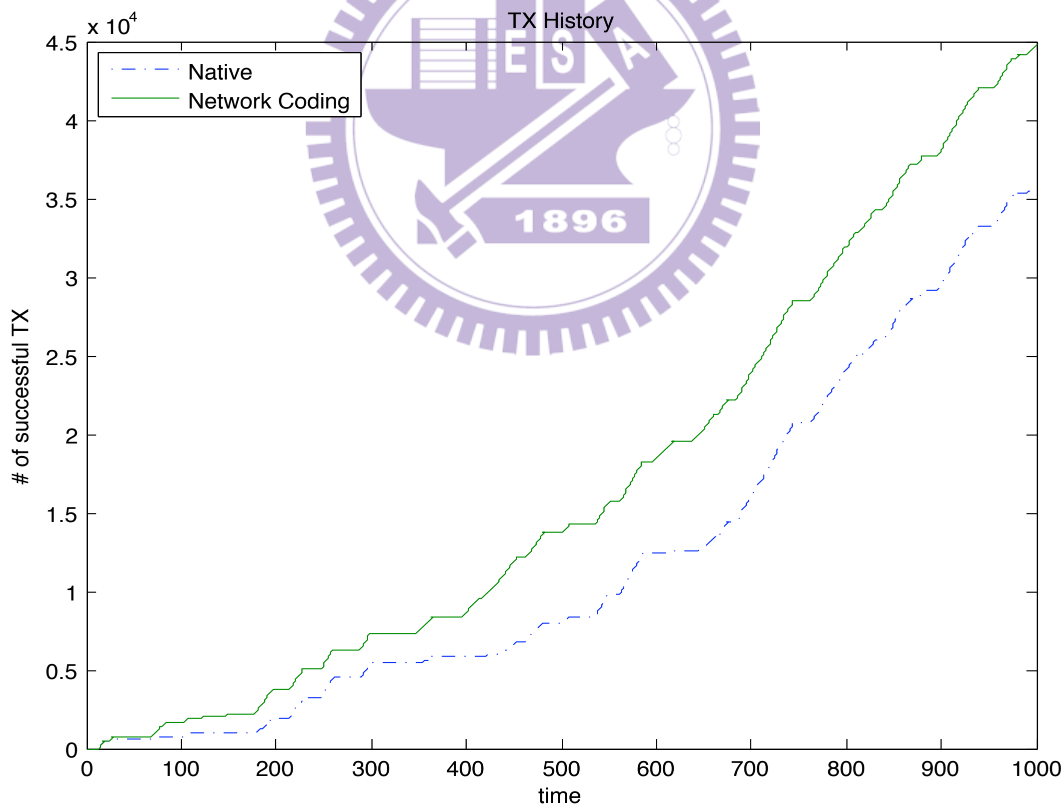



Figure 4-6. Accumulative number of packets (convolutional code)

# Chapter 5. Conclusion

In this thesis, we proposed a flexible coding scheme to reduce the packet error rate. This method can be applied to any kind of network, both real-time and best-effort traffic. The transmission overhead of a packet in CDRE is almost $\frac{j}{n}$ times to the original length, and the end-to-end delay of CDRE is approximately $\frac{n+j}{n}$ times the original delay. Large number of splitting sectors will decrease SER, but increase CRC overhead. Large number of redundancy brings higher delivery ratio, but more transmission overhead and longer delay. These are all trade-offs. It needs to tune the values of both variables to fit to different services.

As the SNR gets lower, the BER may grow to an unacceptable value. We may choose using CDRE to reduce PER in place of directly reducing BER in some cases. This will provide a better transfer rate. In another aspect, with long packet length comes high PER. we can also choose large MTU with CDRE in order to increase network performance.

# References


[1] R. Alshwede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow", *IEEE Trans. Inform. Theory*, vol:46, iss:4, pages:1204-1216, Jul 2000.

[2] S.-Y. R. Li and R. W. Yeung, "Linear network coding", *IEEE Trans. Inform. Theory*, vol: 49, iss:2, pages:371-381, Feb 2003.

[3] Anton, Howard, Elementary Linear Algebra (9th ed.), Wiley, Dec 2004.

[4] Jonathan Stone, Michael Greenwald, Craig Partridge and James Hughes, "Performance of Checksums and CRC's over Real Data", *IEEE/ACM Trans. Networking*, vol:6, iss:5, pages:529-543, Oct 1998.

[5] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard and Jon Crowcroft, "XORs in the air: practical wireless network coding", *SIGCOMM'06*, vol.36, issue:4, pages:243-254, Oct 2006.

[6] Ralf Koetter and Muriel Médard, "An Algebraic Approach to Network Coding", *IEEE/ACM Trans. Networking*, vol:11, iss:5, pages:782-795, Oct 2003.

[7] Abhinav Kamra, Vishal Misra, Jon Feldman and Dan Rubenstein, "Growth Codes: Maximizing Sensor Network Data Persistence", *ACM SIGCOMM'06*, vol:36, iss:4, pages:255-266, Aug 2006.

[8] Anwar Al Hamra, Chadi Barakat and Thierry Turletti, "Network Coding for Wireless Mesh Networks: A Case Study", *WOWMOM '06*, pages:103-114, Jun 2006.

[9] Uichin Lee, Joon-Sang Park, Joseph Yeh, Giovanni Pau and Mario Gerla, "CodeTorrent: Content Distribution using Network Coding in VANET", *MobiShare '06*, pages:1-5, Sep 2006.

[10] Shabbir Ahmed and Salil S. Kanhere, "VANETCODE: network coding to enhance cooperative downloading in vehicular ad-hoc networks", *IWCMC '06*, pages:527-532, Jul 2006.

[11] Chi-hsien Lin, Hui Dong, Upamanyu Madhow and Allen Gersho, "Supporting real-time speech on wireless ad hoc networks: inter-packet redundancy, path diversity, and multiple description coding", *WMASH '04*, pages:11-20, Oct 2004.

[12] Sanjit Biswas and Robert Morris, "ExOR: opportunistic multi-hop routing for wireless networks", *SIGCOMM '05*, vol:35, iss:4, pages:133-144, Aug 2005.

[13] Szymon Chachulski, Michael Jennings, Sachin Katti and Dina Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing", *SIGCOMM'07*, vol:37, iss:4, pages: 169-180, Aug 2007.