AN ABSTRACT OF THE THESIS OF

Samir Elmougy for the degree of Doctor of Philosophy in Computer Science presented on April 28, 2005.

Title: Some Contributions to Asymmetric Error Control Codes

Redacted for privacy

Abstract approved: _____

Bella Bose

In some practical systems, most of the errors are of $1 \rightarrow 0$ type and $0 \rightarrow 1$ errors occur very rarely. In this thesis, first, the capacity of the asymmetric channel is derived. The capacity of the binary symmetric channel (BSC) and the Z-channel can be derived from this expression as special cases.

Second, the error detecting capability of Bose-Lin codes beyond the maximum designed error detection capability are described. Third, a new class of a systematic t-unidirectional error detecting codes over Z_m, $m \geq 2$ is designed. These codes can detect 2 errors using $r = 2$ check bits and up to $m^{r-2} + r - 2$ errors using $r > 2$ check bits. Some upper bound on the maximum number of detectable errors when using r check bits are given.

Finally, some analysis on the data throughput when using the following protocols over the m-ary Z-Channel, $m \geq 2$ are derived:

(1) ARQ protocols using t-Asymmetric Error Detecting (t-AED) codes.

(2) ARQ protocols using All Asymmetric Error Detecting ($AAED$) codes.

(3) Type-I Hybrid ARQ protocols using t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$EC|AAED$) codes.

(4) ARQ Protocols with diversity combining using t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$EC|AAED$) codes.

Finally, some open research problems are described.

Some Contributions to Asymmetric Error Control Codes

by

Samir Elmougy

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented April 28, 2005
Commencement June 2005

Doctor of Philosophy thesis of Samir Elmougy presented on April 28, 2005

APPROVED:

Redacted for privacy

Major Professor, representing Computer Science

Redacted for privacy

Associate Director of the School of Electrical Engineering and Computer Science

Redacted for privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon rec Redacted for p .

Redacted for privacy

Samir Elmougy, Author

ACKNOWLEDGMENT

First of all, I am grateful to the GOD, the most generous and the most merciful for helping me in my life, my research and search for the truth.

I would like to thank Prof. Bella Bose, my advisor, who shared with me very challenging and interesting problems to work on, with whom I had many helpful discussions and from whom I learnt a lot. Many thanks to him for his trust, patience, encouragement, helping me improve my writing skills, showing me how to do research and how to approach an open problem. I would like to thank him for supporting me as a teaching assistant and research assistant.

I would like to thank Prof. Paul Cull, my committee member, for his advice, comments, and his encouragement in my thesis. I would like to thank all my committee members: Prof. Paul Cull, Prof. Timothy A. Budd, Prof. Toshimi Minoura from the Department of Computer Science, and Prof. Huaping Liu from the Department of Electrical and Computer Engineering for their patience and help.

The financial support for my Ph.D. education is from a scholarship provided by the Egyptian Government, Mansoura University.

I thank Prof. Torleiv Klove and Dr. Steve Gorshe for their help and discussions. I thank Prof. Luca Tallini for his discussions, help, kind guidance and from whom I learned many things.

Many thanks to Prof. Bose's research group members: Prof. Badder Almohammad "Kuwait University", who introduce me to Prof. Bose, Dr. Paul Oprisan, Dr. J. H. Youn "Nebraska University" and Madhu Anantha Subramanian. I thank all the professors and staff in the Department of Computer Science.

I thank the professors in both Faculties of Science "Mathematics Department", and Computer Science, Mansoura University, Egypt.

I thank all of my friends and colleagues espicially Dr. Essam Sharaf, Elsayed F. Radwan, Dr. Mounir Louhaichi, Dr. Osama Abdel Salam, Dr. Ahmed Hassan, Dr. Ahmed El-Shafei, Dr. Emad El-Sebakhy, Dr. Osama Mohammed, Prashant Shah, Mansour Al-Mutairi, Benjamin Hermens, Parijat Naik, Dr. Doug Chow, Seikyung Jung, Abdel Hamid Elnaggar, Laura Beckwith, Madhu Srinivasan, Neville Mehta, Robin Abraham, and Terry David. Many thanks to Patricia Lacy who advised and helped me.

Finally, this work will not have been possible without the love and moral support of my parents, my wife, Suzan, my wonderful kids: Ahmed, Yousef and Hadi, my uncle, Mohsen Keshk, and all other members of my family for their support, love, patience and help.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF TABLES

DEDICATION

To my parents, my wife, Suzan, and my wonderful kids: Ahmed, Yousef and Hadi.

SOME CONTRIBUTIONS TO ASYMMETRIC ERROR CONTROL CODES

1. INTRODUCTION

1.1. Preliminaries and Foundations

Communication systems are designed to send information or messages from a specific source to one or more destinations. A typical communication system can be represented as shown in Figure 1.1. The main parts of the system are the transmitter, the channel, and the receiver. The *transmitter* converts the data source to a suitable format to be sent through the channel. The *channel* is the transmission media between the source and the destination. The channel might be a digital channel, a telephone line, optical fiber cables, etc. In most cases, the channel is noisy, and so the transmitted word may be corrupted. In this case, after receiving this corrupted word, the *receiver* tries to recover the original word. In the following paragraphs, we describe the components of Figure 1.1.

The *input* (data source) of the system is a source which provides a stream of information. This stream of information might be a sequence of images such as X-rays or pictures, or a sequence of symbols such as letters from the English language, a set binary symbols from a computer file, a waveform such as a voice signal from a microphone, etc. There may be a lot of redundancy in the source symbols. Thus, the *source* encoder compresses this data stream, meaning that

the data stream is represented with as few bits as possible without destroying its information content.

The *channel encoder* adds some appropriate redundant bits (called the check bits) to the compressed data and then sends it through the channel. At the receiver, these check bits are used to recover the original data word at the receiver by the *channel decoder.* Finally, the source decoder recovers the original source data by decompressing the data obtained from the channel decoder, it maps the resulting sequence of bits back to its original form by using the inverse 'mapping' of the source encoder.

As mentioned earlier, the channel is noisy and so the transmitted word may be corrupted - some bits may be lost or changed during the transmission. In 1948, Shannon published his famous paper "A Mathematical Theory of Communication" [45]. In this paper, he introduced the channel coding theory in which he showed that channel noise does not prevent error-free communication, i.e. he proved that information can be sent reliably over a channel at all rates (measured in bits per seconds) up to the channel capacity (also measured in bits per seconds). In other words, he proved that the channel capacity is the upper bound on the number bits that can be sent per unit time with almost zero bit error probability. Shannon's work provided only the theoretical model for the information capacity of the channel without discussing how to achieve this capacity. After this paper was published, researchers started working on designing codes which could achieve the capacity of the channel. These techniques are known as error control coding.

The errors that can occur because of noise are many and varied but can be classified into three main types, symmetric, asymmetric, and unidirectional errors. In symmetric errors, both of $1 \rightarrow 0$ and $0 \rightarrow 1$ errors can occur simultaneously in a data word, and this can be modeled using the Binary Symmetric Channel (BSC) as

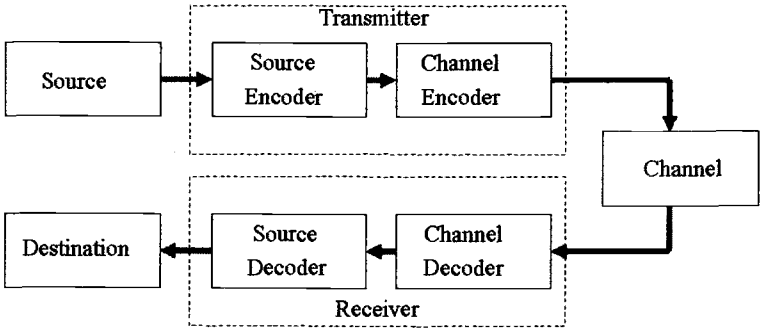

FIGURE 1.1. Typical Communication System.

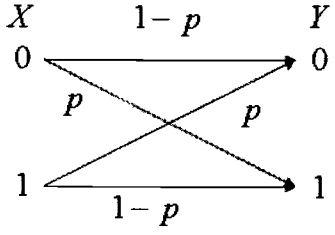

FIGURE 1.2. BSC Channel.

shown in Figure 1.2. Errors in many practical systems such as telecommunication systems, etc, can be described using the BSC model.

In asymmetric errors, only one type of errors, either $1 \to 0$ or $0 \to 1$, can occur in a data word and the other type does not occur in any data word. In this case, the decoder knows a priori the type of error. This type of error can be modeled using the Z or $\bar{Z}$ channel as shown in Figure 1.3. Errors in some practical systems can be characterized using this model. For example, in optical systems, the photons may decay or fade but no photos can be generated upon transmissions. In these systems, the presence of photos is represented by 1 and the absence by 0. Thus, the error characteristic of these systems can be modeled by the Z-channel.

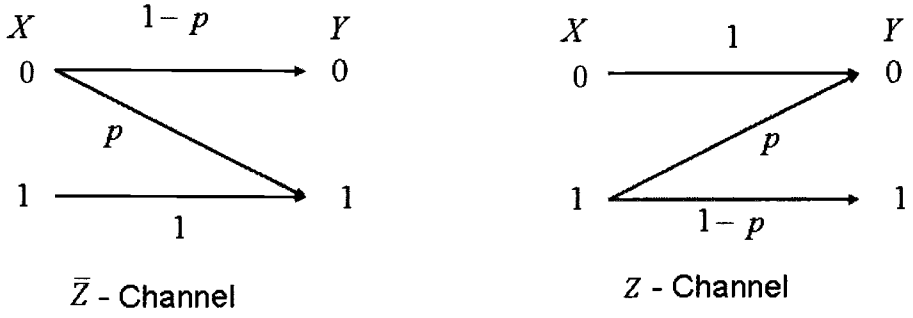

FIGURE 1.3. Z-Channel and $\bar{Z}$-Channel.

In unidirectional errors, both of $1 \rightarrow 0$ and $0 \rightarrow 1$ errors can occur but can not occur simultaneously in a data word. In this case, the decoder does not know a priori by which type of error can occur. The errors in some digital devices such as data transmission systems, shift-register and magnetic-recording mass memory, ROM and RAM memories, and interconnection networks can be modeled using unidirectional errors. For example, in a shift register memory, a stuck-at-1 (or a stuck-at-0), at the output of a register results in an all 1 (or all 0) output.

There are two major error control techniques used in practice - Forward Errors Control (FEC) which uses error correcting codes and Automatic-Repeat-Request (ARQ) which uses error detecting codes. Many times a combination of these two methods known as hybrid ARQ protocol, which uses error correcting and error detecting codes simultaneously, are also used. These methods are briefly explained in the next few sections.

1.2. Forward Error Control (FEC)

In these schemes, the transmitter encodes the information word into an error correcting code word and sends it to the receiver. If the receiver detects

errors in the received word, it attempts to determine the exact location of these errors, and then attempts to correct them using the parity bits. The amount of the added parity bits is expressed in terms of the code rate (R). The code rate of transmission is the ratio between the number of data symbols transmitted per code word to the total number of symbols transmitted per code word [52].

FEC schemes have bounded time delay equal to the processing time for encoding/decoding and have a constant throughput equal to the code rate regardless of the channel conditions. These schemes are suitable for those types of communications that require getting the message correct in the first transmission.

On the other hand, if the receiver fails to determine the exact locations of errors, then the received data will be incorrectly decoded and the user (or data sink) will receive erroneous data [33]. We can summarize the main disadvantages of using FEC techniques as [52, 29, 2, 33]:

(1) Hard to achieve high system reliability.

(2) FEC requires more coding processes than ARQ to achieve the same reliability.

(3) The decoding process is hard to implement and expensive.

Some of the widely used symmetric error correcting codes are Hamming codes [24], linear block codes [52, 40], Hadamard codes [4, 52], cyclic codes [41–43], BCH codes [14, 13], Reed-Solomon codes [44], and convolution codes [19, 53]. Some codes and implementation methods for asymmetric error correction are given in [47, 46, 9, 8, 1, 18, 36, 35].

1.3. ARQ Protocol

As we mentioned in the previous section, the data transmission takes place in only one direction when using FEC techniques, i.e. from the transmitter to the receiver. However, in ARQ techniques, data transmission is done in both directions. Further, FEC uses error correcting codes, whereas ARQ uses error detecting codes. The error correction codes mentioned in the previous paragraph can also be used for error detection. The codes given in [6, 5, 25, 31, 26, 37, 38, 40, 48, 28, 32, 51, 27, 23] are examples of t-unidirectional detecting codes. The transmitter starts sending the codewords and sets a timer for each one it transmits. If the received word is error free, and is correctly received by the receiver, then it will be delivered to the user or stored in a buffer. At the same time, a positive acknowledgment (ACK) is transmitted by the receiver through a return channel to notify the transmitter that the word has been successfully received. On the other hand, if the receiver detects one or more errors in the received word, it sends a negative acknowledgment (NAK) to the transmitter via a return channel requesting it to resend the word. The system continues this retransmission until the received word is correctly received, i.e. received without errors. This scheme is simple to implement and provides highly reliable data transmission.

In 1964, Benice and Frey [2] classified ARQ protocols into three main types: stop-and-wait ARQ, go-back-N ARQ, and selective-repeat ARQ. These types of protocols differ in the following [52]:

(1) Number of words that the transmitter can send without receiving acknowledgments from the receiver for the previous transmitted words.

(2) The buffering availability at the transmitter and the receiver.

Also, the performance of protocol is studied based on [33]:

(1) *Reliability* (Accepted packet Error Rate). This is the ratio of the number of accepted words that contain one or more bit/symbol errors to the total number of accepted words by the receiver.

(2) *Throughput* for the system: the average number of encoded data words accepted by the receiver in the time it takes the transmitter to send a single k-bit data packet.

In the following paragraphs, we briefly discuss these three protocols [52, 29, 34, 2].

1.3.1. Stop and Wait

This is the simplest of the ARQ protocols. After transmitting the code word, the transmitter will wait for an acknowledgment. If the transmitter receives ACK, it then sends the next code word. If either the timeout times expires without receiving an acknowledgment, or the transmitter receives NAK, the transmitter retransmits the same code word again. This procedure continues until ACK is received. So, buffering is not necessary at both the receiver and the transmitter. The main disadvantage of using this scheme is that the transmitter is idle while waiting for the acknowledgment resulting in a low throughput performance. Stop-and-wait is useful in some computer applications such as interprocessor transfer in multiprocessing systems, where the round trip delay is extremely low. Figure 1.4 explains this protocol.

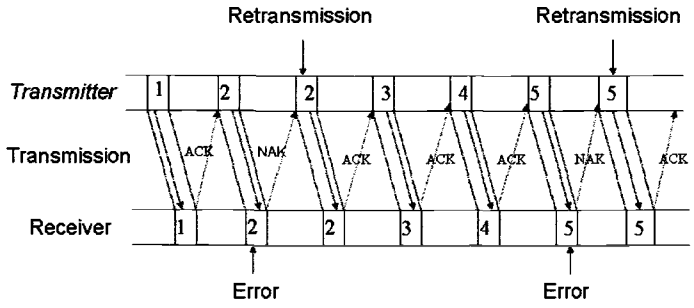

FIGURE 1.4. Stop-and-Wait Protocol.

1.3.2. Go-back-N ARQ protocol

If there is some buffering available in the transmitter side and not necessary at the receiver end, go-back-n ARQ protocol can be used. In this protocol, the transmitter sends the code words in a continuous stream without waiting for an acknowledgment from the receiver. If the receiver detects an error in a received word, it requests a retransmission for this word by sending NAK to the transmitter. At this point, all subsequent incoming words are ignored until the transmitter retransmits the requested word and the receiver receives it. Therefore, buffering is not necessary at the receiver. When the transmitter resends a word, it also resends all subsequent words (which were ignored at the receiver after detecting the first erroneous word). This makes buffering necessary at the transmitter. Figure 1.5 explains this protocol.

1.3.3. Selective-Repeat ARQ protocol

If some buffering is available at both the transmitter and the receiver, Selective-Repeat ARQ protocol can be used and implemented. In this protocol, the transmitter sends the words in a continuous stream without waiting for ac-

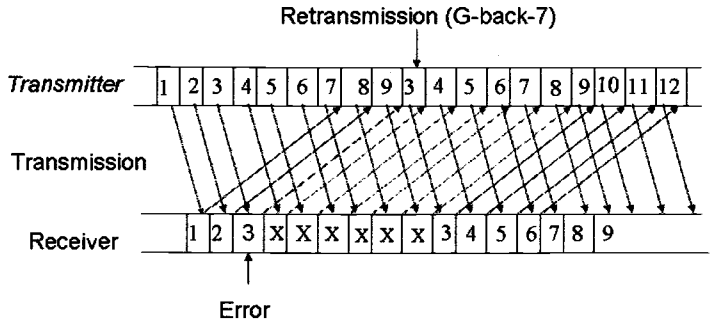

FIGURE 1.5. Go-Back-N Protocol.

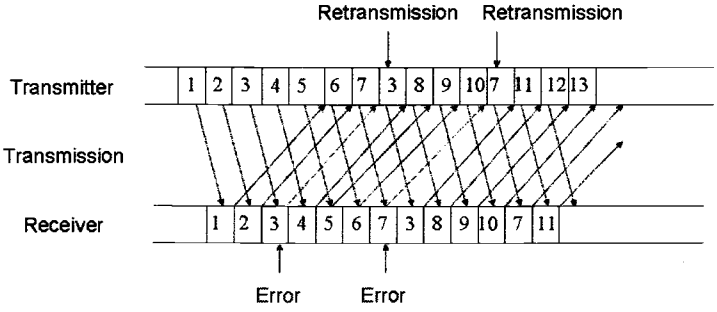

FIGURE 1.6. Selective Repeat Protocol.

knowledgment from the receiver. If the receiver detects an error in one of the received words, it requests a retransmission for this word by sending NAK to the transmitter. At this point, the transmitter resends the required word and then resumes transmitting the new code words. So, buffering is necessary at both sides. Figure 1.6 explains this protocol.

1.4. Hybrid ARQ protocol

In ARQ, as explained earlier, if the receiver detects one or more errors in the received word, it asks the transmitter to retransmit the same word again, and this procedure is repeated until the word is correctly received. This scheme is

simple to implement and provides high reliability but it has some disadvantages such as [52, 29, 2, 33]:

(1) It has a variable delay time.

(2) It is harder to implement when the round-trip delay increases.

(3) The throughput of the system will rapidly decrease when the channel error rate increases

To overcome the drawbacks of both of FEC and ARQ schemes, a combination of both schemes, called hybrid ARQ protocols, have been developed. There are two types of hybrid-ARQ protocols - type-I hybrid ARQ and type-II hybrid ARQ, which are described below.

1.4.1. Type-I Hybrid ARQ protocol

In this type of protocol, parity bits are included for both error correction and error detection. (Note that a code is capable of correcting t-errors and detecting d $(d \geq t)$ errors if and only if the minimum distance of the code is $t + d + 1$.) If the receiver detects an error in the received word, and the number of errors is within the error correcting capability of the designed code, then the errors will be corrected and ACK will be sent to the transmitter requesting a transmission of the next word. On the other hand, if the word is received with detectable but uncorrectable errors, then the receiver discards the erroneous word and sends NAK requesting a retransmission of the same word. This process continues until the word is successfully accepted or the maximum retransmission number has been reached [34, 52].

1.4.2. Type-II Hybrid ARQ protocol

The main way that the type-II hybrid ARQ protocol differs is that, it sends additional parity check digits for error correction to the receiver only if they are needed. The words that could not be successfully decoded at the receiver are saved. When the transmitter receives a request for a retransmission, it sends additional parity bits to the receiver. The receiver appends these additional parity bits to the saved (corrupted) words and attempts to correct the errors. This process is repeated until the word is successfully decoded. [33, 52, 34].

1.5. Diversity Combining for the Z-Channel

In the case of the Z-channel, the throughput of the ARQ system can be improved using a simple diversity combining technique without adding much to the hardware. The main idea is briefly explained below assuming that the system uses a t-AED code [30].

At the receiving end, the received word is combined with the previously combined word. This word combination is done by a bit-by-bit logic OR operation as shown in Figure 1.7. When the combined word is still in error, a NAK is sent to the transmitter requesting it to resend the same word. However, if the combined word is error free, the word is accepted and ACK is sent to the transmitter requesting it to send the next codeword. As an example, assume that the codeword $X = (0100111010101)$ is transmitted over the Z-channel and is received as (0100$\underline{0}$1$\underline{0}$010$\underline{0}$01), i.e. it suffers from three bit errors. Assume that the code can detect up to 3 errors. Now, the receiver requests the transmitter to resend the word and in each of the consecutive steps, the received word is bit-by-bit OR-ed with the previously stored word. Assuming that the sequence of the first three

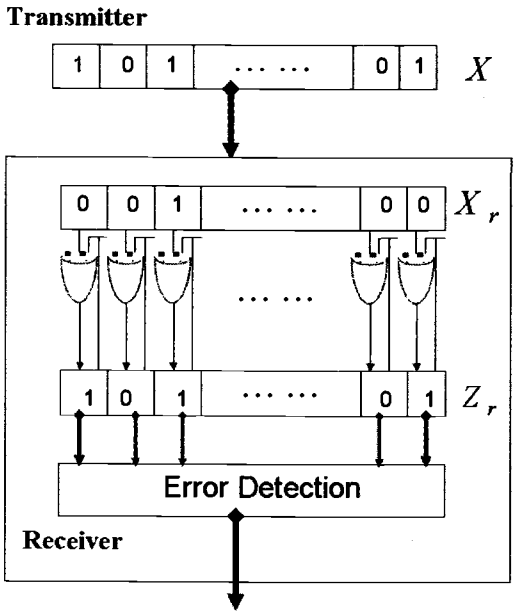

FIGURE 1.7. Packet reusing scheme for the Z-channel.

r	X_r	Z_r
0	–	0000000000000
1	0100$\underline{1}$0010$\underline{0}$01	0100$\underline{1}$0010$\underline{0}$01
2	0100$\underline{1}$01010$\underline{0}$01	0100111010$\underline{0}$01
3	0100$\underline{1}$010$\underline{0}$0101	0100111010101

TABLE 1.1. A sequence of transmissions for the codeword $X = 0100111010101$ over the Z-channel using diversity combining technique.

retransmissions yield the words given in Table 1.1, the codeword X is recovered after these three retransmissions. In this table, $Z_r = Z_{r-1} \vee X_r$ is the combined word at each step r.

1.6. Outline of the Dissertation

Since all the work in this thesis are for the asymmetric errors, we first study the capacity of the asymmetric channel. To the best of our knowledge, these result is not known till now. Bose and Lin [11] have designed systematic codes for detecting asymmetric errors. These codes can detect up to 2, 3, and 6 errors using 2, 3, and 4 check bits, respectively. On the other hand, two different methods are proposed for the cases of check bits $r \geq 5$. The codes designed based on Method 1 can detect up to $2^{r-2} + r - 2$ errors and based on Method 2 up to $5 \times 2^{r-4} + r - 4$ errors where $r \geq 5$. Table 1.2 shows the error-detecting capabilities of the Bose-Lin codes designed by both Method 1 and Method 2. In some applications, it may be important to apply the Bose-Lin to a larger block of data to reduce the number of times the check needs to be performed, at the risk of exceeding the maximum detected capabilities. In one recent example, a Bose-Lin code has been used in conjunction with a linear-feedback shift register (LSFR) multiple-input signature register (MISR) in order to decrease the probability of undetected faults. Hence, studying the performance of Bose-Lin codes when the errors are beyond the maximum designed error detection capabilities is worth.

Although there are some codes designed before to detect unidirectional errors over Z_m, $m \geq 2$, we design a new code but with few number of check bits.

In ARQ protocols, as mentioned before, it is important to improve the performance of the system by reducing the number of retransmissions needed to receive a correct code. In our work, we study the performance of some codes over a discrete memoryless m-ary asymmetric Z-channels, $m \geq 2$, by deriving an expression for the number (or the expected number) of retransmissions needed to receive a code correctly. First, we derive the expected number of retransmissions

for pure ARQ protocols. Then, we do the same for type-I hybrid ARQ protocols, and apply the derived expression to obtain the throughput of some special cases of the m-ary asymmetric channels.

Finally, we design a new *diversity combining* scheme to reduce the number of retransmissions. In this scheme, the correcting and detecting process are used based only on the combined word not on the received word. Due to the characteristics of the asymmetric errors and the way our scheme works, the number of errors in the combined word will be less than or equal to the number of errors in the previous one. Hence, the number of retransmissions needed to received a code is decreased. Later we give a numerical comparison between the performance of type-I hybrid ARQ and ARQ with diversity combining protocols.

The thesis is organized as follows. In Chapter 2, the capacity of the asymmetric channels is derived. Further, the capacity of the binary symmetric channel (BSC) and the Z-channel can be obtained as special cases of this formula.

In Chapter 3, some analysis of extended error detecting capabilities of Bose-Lin codes are described.

In Chapter 4, a new class of a systematic t-unidirectional error detecting codes over Z_m, $m \geq 2$ is designed. The codes can detect 2 errors using $r = 2$ check bits and up to $m^{r-2} + r - 2$ errors using $r > 2$ check bits. Some upper bound on the maximum number of detectable errors when using r check bits are described.

In Chapter 5, we analyze the throughput of the following ARQ schemes over the m-ary Z-Channel, $m \geq 2$:

(1) ARQ protocols using t-Asymmetric Error Detecting (t-AED) codes.

(2) ARQ protocols using All Asymmetric Error Detecting ($AAED$) codes.

r	**Number of Errors Detected**	
	*Method*1	*Method*2
5	11	11
6	20	22
7	37	43
8	70	84
9	135	165
10	264	326
11	521	647
12	1034	1288

TABLE 1.2. Maximum number of errors detected by Bose-Lin codes using Method 1 and Method 2.

(3) Type-I hybrid ARQ protocols using t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$AEC/AAED$) codes.

(4) ARQ Protocols with diversity combining using t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$AEC/AAED$) codes.

Conclusions and future works are given in Chapter 6.

2. CAPACITY OF THE ASYMMETRIC CHANNEL

2.1. Introduction

Asymmetric channel, as given in Figure 2.1, is the channel with $\{0, 1\}$ as input and output alphabets, and p_1 and p_2 as the probabilities of $0 \rightarrow 1$ and $1 \rightarrow 0$ bit errors, respectively. In many practical systems, such as optical fibers and disks, semiconductor memories, etc., the errors can be modeled using the asymmetric channel. In this chapter, the capacity of asymmetric channel is analyzed.

This chapter is organized as follows: fundamentals and definitions of asymmetric channel, and capacity are briefly given in Section 2.2. Analysis and achieving the capacity of asymmetric channel are described in Section 2.3.

2.2. Definitions and Fundamentals

In this section, we will briefly give some definitions from information theory. For a channel with input alphabet S_X and output alphabet S_Y, let $p(y_i|x_i)$ be the probability that the output from a channel is y_i given that the input to the channel is x_i. Then, we can represent the channel by a $(|S_Y| \times |S_X|)$ *transition probability matrix*, M, such that $M = [m_{ij}]$=$[p(y_i|x_j)]$, where i =0,1,2,$\cdots$,$|S_X| - 1$, and j = 0,1,2,$\cdots$,$|S_Y| - 1$, i.e.

$$M = \begin{pmatrix} p(y_0|x_0) & p(y_1|x_0) & \cdots & p(y_{|S_Y|}|x_0) \\ p(y_0|x_1) & p(y_1|x_1) & \cdots & p(y_{|S_Y|}|x_1) \\ \vdots & \vdots & \vdots & \vdots \\ p(y_0|x_{|S_X|}) & p(y_1|x_{|S_X|}) & \cdots & p(y_{|S_Y|}|x_{|S_X|}) \end{pmatrix}.$$

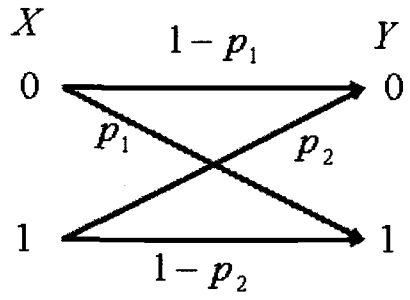

FIGURE 2.1. Asymmetric Channel.

A channel is said to be a symmetric channel if the rows of the transition probability matrix, M, are permutations of each other, and the columns are permutations of each other. For example, the binary symmetric channel (BSC) shown in Figure 1.2, in which p is the probability of the error, has the following transition probability matrix:

$$M = \begin{pmatrix} p(0|0) = 1-p & p(1|0) = p \\ p(0|1) = p & p(1|1) = 1-p \end{pmatrix}.$$

Entropy is a measure of the uncertainty in a random variable. It is the number of bits on the average required to describe the random variable. The entropy of a random variable X with a probability mass function $p(x)$ is defined as:

$$H(X) = -\sum p(x) \log_2 p(x).$$

The *conditional entropy*, $H(X|Y)$, is the entropy of a random variable X given another random variable Y. The *mutual information*, $I(X,Y)$, is the measure of the amount of information that one random variable Y contains about another random variable X. The mutual information for the two variables X, and

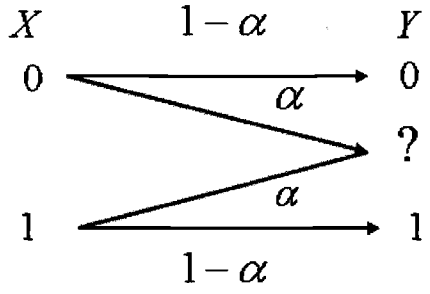

FIGURE 2.2. Erasure Channel.

Y is defined as:

$$I(X,Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) log_2 \left[\frac{p(x,y)}{p(x)p(y)} \right].$$

The *channel capacity*, C, of a discrete memoryless channel with input X and output Y is defined as the maximum mutual information, i.e.

$$C_{p_1,p_2} = \max_{p(x)} I(X,Y),$$

where the maximum is taken over all possible input distributions $p(x)$.

For example, the information capacity of a binary symmetric channel (BSC), in which p is the probability of the error, is $C(p,p) = 1 - h(p)$ bits [15], where $h : [0,1] \rightarrow R$ is the entropy function

$$h(x) = -\left[x \log_2 x + (1-x) \log_2 (1-x)\right].$$

Also, the information capacity of a binary erasure channel shown in Figure 2.2 is $C = 1 - \alpha$ bits [15], where α is the probability of occurrence of an erasure. A third example is that the capacity of Z-channel [49], in which the input and the output alphabets are {0,1}, and the probability of the crossover $1 \rightarrow 0$ errors is p, and the probability of the crossover $0 \rightarrow 1$ errors is 0 as shown in shown in Figure 1.3, is

$$C_{Z(p)} = \frac{h(p)}{2^{h(p)/(1-p)} + 1} - \frac{h(p)}{[2^{h(p)/(1-p)} + 1](1-p)}.$$

Definition 1 *An (M_1, n)* ***code*** *for the channel $(S_X, p(y|x), S_Y)$, where M_1 is the number of codewords with length n, consists of the following:*

(1) *An index set $\{1, 2, \cdots, M_1\}$.*

(2) *An encoding function $X^n : \{1, 2, \cdots, M_1\} \rightarrow S_X^n$, yielding codewords $X^n(1), X^n(2), \cdots, X^n(M_1)$.*

(3) *A decoding function*

$$g : S_Y^n \rightarrow \{1, 2, \cdots, M_1\},$$

which is a deterministic rule which assigns a guess to each possible received vector.

Definition 2 (Probability of error): *Let $\lambda_i = Pr(g(Y^n) \neq i | X^n = X^n(i))$ be the* ***conditional probability of error*** *given that index i was sent.*

Definition 3 ***The maximal probability of error*** *$\lambda^{(n)}$ for an (M_1, n) code is defined as:*

$$\lambda^{(n)} = max_{i \in \{1,2,\cdots,M_1\}} \lambda_i \ .$$

Definition 4 *The* ***rate****, R, of an (M_1, n) code is $R = log_2 M_1 / n$ bits per transmission, i.e. R is the ratio between the length of a source message and the length of an encoded message.*

A rate R is said to be achievable if there exists a sequence of $(2^{nR}, n)$ codes such that the maximal probability of error $\lambda^{(n)} \rightarrow 0$ as $n \rightarrow \infty$ [15]. The next theorem is the fundamental result in information theory which specifies the maximum number of codewords that we can define and maintain completely distinguishable outputs. In this theorem, *Shannon* proved that the information can be sent reliably over a channel at all rates up to the channel capacity [15, 45].

Theorem 2.2.1 (The Channel Coding Theorem) *All rates below capacity C are achievable. That is, for every $\epsilon > 0$ and rate $R < C$, there exists a sequence of $(2^{nR}, n)$ codes with maximum probability of error $\lambda^{(n)} \to 0$. Conversely, any sequence of $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R \leq C$.*

2.3. The Capacity of the Asymmetric Channel

In this section, we find the capacity of asymmetric channel shown in Figure 2.1. Let p_1 (p_2) be the probability that a 1 (0) is received when 0 (1) is transmitted. Let $p(y|x)$ be the probability that the output from a channel is y given that the input to the channel is x. Then, the following transition probability matrix, M, defines the asymmetric channel:

$$M = \begin{pmatrix} p(0|0) & p(1|0) \\ p(0|1) & p(1|1) \end{pmatrix},$$

i.e.

$$M = \begin{pmatrix} 1-p_1 & p_1 \\ p_2 & 1-p_2 \end{pmatrix}.$$

Let $X \in \{0, 1\}$ be a random variable with $p(X = 0) = 1 - q$, and $p(X = 1) = q$. Suppose that X is fed as input to the asymmetric channel, and Y is the output with the following probabilities:

$$p(Y = 0) = (1 - q)(1 - p_1) + q\,p_2, \text{ and}$$
$$p(Y = 1) = (1 - q)p_1 + q(1 - p_2).$$

Define $I(X, Y)$ as the mutual information between the two random variables X and Y, i.e. $I(X, Y) = H(Y) - H(Y|X)$ as given in Figure 2.3. Thus, the *capacity*

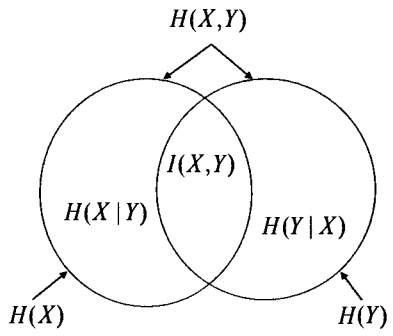

FIGURE 2.3. Mutual Information.

of the channel is defined as:

$$C_{p_1,p_2} = \max_{q\in[0,1]} I(X,Y) = \max_{q\in[0,1]}[H(Y) - H(Y|X)].$$

Let $h : [0,1] \to R$ be the entropy function as defined before,

$$h(x) = -\left[x \log_2 x + (1-x)\log_2(1-x)\right],$$

then, the entropy of Y is

$$\begin{aligned} H(Y) &= [(1-q)p_1 + q(1-p_2)] \log_2 \frac{1}{[(1-q)p_1 + q(1-p_2)]} \\ &+ [(1-q)(1-p_1) + q\,p_2] \log_2 \frac{1}{[(1-q)(1-p_1) + q\,p_2]} \\ &= h\left[(1-q)p_1 + q(1-p_2)\right]. \end{aligned}$$

For the entropy function of Y given X, we have

$$\begin{aligned} H(Y|X) &= \sum_x p(x)H(Y|x) \\ &= p(X=0)H(Y|X=0) + p(X=1)H(Y|X=1) \\ &= (1-q)H(Y|X=0) + q\,H(Y|X=1), \end{aligned}$$

where

$$H(Y|X=0) = p(Y=0|X=0)\log_2 \frac{1}{p(Y=0|X=0)} + p(Y=1|X=0)\log_2 \frac{1}{p(Y=1|X=0)} = (1-p_1)\log_2 \frac{1}{1-p_1} + p_1 \log_2 \frac{1}{p_1} = h(p_1),$$

and similarly,

$$H(Y|X=1) = p(Y=0|X=1)\log_2 \frac{1}{p(Y=0|X=1)} + p(Y=1|X=1)\log_2 \frac{1}{p(Y=1|X=1)} = p_2.\log_2 \frac{1}{p_2} + (1-p_2)\log_2 \frac{1}{1-p_2} = h(p_2).$$

Hence,

$$H(Y|X) = (1-q)h(p_1) + q\,h(p_2).$$

$$\Rightarrow I(X,Y) = H(Y) - H(Y|X) = h[(1-q)p_1 + q(1-p_2)] - (1-q)h(p_1) - qh(p_2)$$

$$= h[p_1 + q(1-p_1-p_2)] - (1-q)h(p_1) - q\,h(p_2).$$

Consider the above function as a function of $q \in [0,1]$, f_{p_1,p_2}, as:

$$f_{p_1,p_2}(q) = h[p_1 + q(1-p_1-p_2)] - (1-q)h(p_1) - q\,h(p_2).$$

For all $p_1, p_2 \in [0,1]$, $f_{p_1,p_2}(q)$ is continuous for all $q \in [0,1]$, and derivable for all $q \in (0,1)$. Thus,

$$f'_{p_1,p_2}(q) = (1-p_1-p_2)\;\; h'\,[p_1 + q(1-p_1-p_2)] + h(p_1) - h(p_2)$$

Since $h'(x) = \log_2\,[(1-x)/x]$, then we have,

$$f'_{p_1,p_2}(q) = \log_2 \frac{1-[p_1+q(1-p_1-p_2)]}{p_1+q(1-p_1-p_2)}(1-p_1-p_2) + h(p_1) - h(p_2).$$

Now, let $f'_{p_1,p_2}(q) = 0$. Thus, we will have

$$\log_2 \frac{1-[q(1-p_1-p_2)+p_1]}{q(1-p_1-p_2)+p_1} \geq \frac{h(p_2)-h(p_1)}{1-p_1-p_2}.$$

$$\Rightarrow \frac{1}{q(1-p_1-p_2)+p_1} - 1 \geq 2^{\left[\frac{h(p_2)-h(p_1)}{1-p_1-p_2}\right]}.$$

$$\Rightarrow q(1-p_1-p_2)+p_1 \leq \frac{1}{2^{\left[\frac{h(p_2)-h(p_1)}{1-p_1-p_2}\right]+1}}.$$

$$\Rightarrow q \leq \left[\frac{1}{2^{\left[\frac{h(p_2)-h(p_1)}{1-p_1-p_2}\right]+1}} - p_1\right] / [1-p_1-p_2] = q_{max}(p_1,p_2).$$

$$\begin{aligned}
\Rightarrow C_{p_1,p_2} &= max_{q\in[0,1]} I(X,Y) \\
&= h\left((1-q_{max})p_1 + q_{max}(1-p_2)\right) - (1-q_{max})h(p_1) - q_{max}h(p_2) \\
&= h\left(q_{max}(1-p_1-p_2)+p_1\right) - (1-q_{max})h(p_1) - q_{max}\, h(p_2) \\
&= h\left(q_{max}(1-p_1-p_2)+p_1\right) - q_{max}\left(h(p_2)-h(p_1)\right) - h(p_1)
\end{aligned}$$

$$= h\left[\frac{1}{2^{(h(p_2)-h(p_1))/(1-p_1-p_2)}+1}\right]$$

$$- \left[\frac{1}{2^{(h(p_2)-h(p_1))/(1-p_1-p_2)}+1} - p_1\right]\left[\frac{h(p_2)-h(p_1)}{(1-p_1-p_2)}\right] - h(p_1).$$

To conclude this section, we give three special cases:

Case 1:

When both of p_1 and p_2 are small, we will have

$$C_{p_1,p_2} \approx h\left(\frac{1}{2}\right) - \frac{1}{2}\left[h(p_2)-h(p_1)\right] - h(p_1)$$

$$\approx 1 - \frac{1}{2}\left[h(p_1)+h(p_2)\right].$$

Case 2:

When $p_1 = p_2 = p$, we will have

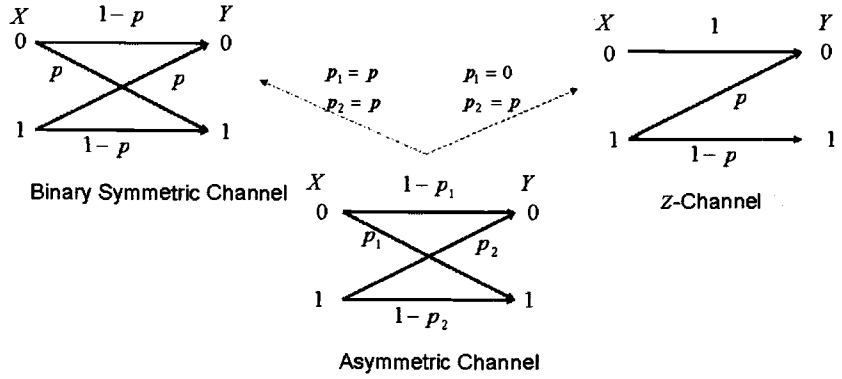

FIGURE 2.4. BSC, Asymmetric and Z channels.

$$C_{p_1,p_2} = C_{p,p} = h\left(\frac{1}{2^0+1}\right) - \left(\frac{1}{2^0+1} - p\right)\left(\frac{1}{1-2p}\right)[h(p) - h(p)] - h(p)$$
$$= h\left(\frac{1}{2}\right) - h(p) = 1 - h(p).$$

This is the same as the capacity of a binary asymmetric channel with bit error probability p.

Case 3:

When $p_1 = 0$ and $p_2 = p$, we will have

$$C_{0,p} = h\left[\frac{1}{2^{\frac{h(p)-h(0)}{1-0-p}}+1}\right] - \left[\frac{1}{2^{\frac{h(p)-h(0)}{1-0-p}}+1} - 0\right]\left[\frac{1}{1-0-p}\right][h(p) - h(0)] - h(0).$$

$$= \frac{h}{2^{h(p)/(1-p)}+1} - \frac{h(p)}{[2^{h(p)/(1-p)}+1](1-p)}.$$

This is the same as the capacity of a Z-channel with bit error probability p.

Figure 2.4 shows the BSC, Asymmetric and Z-Channels.

3. SOME ERROR DETECTING PROPERTIES OF BOSE-LIN CODES

3.1. Introduction

The error characteristic of some VLSI systems is unidirectional or asymmetric [5]. In the asymmetric case, all errors are of only one type, say $1 \rightarrow 0$, whereas in the unidirectional case, all errors within a word can be of the same type, but they can be $1 \rightarrow 0$ type in one word and $0 \rightarrow 1$ type in another word. From the error detecting point of view, these cases are equivalent, meaning that a code capable of detecting t-asymmetric errors is also capable of detecting t-unidirectional errors. Optimal all unidirectional error detecting codes, both systematic and non systematic, are given in [21, 3]. Optimal non-systematic t-asymmetric error detecting codes are given in [7]. Systematic codes, where check bits are separated from information bits, capable of detecting t-unidirectional errors regardless of data word length are given by Bose and Lin in [11].

Unidirectional and asymmetric errors are typical of stuck faults in VLSI circuits. These conditions can also be approximated by channels such as some fiber optic links with very asymmetrical error transition probability. Let $tmax$ be the maximum number of errors a Bose-Lin code is designed to detect. Typical applications for Bose-Lin codes to date (e.g., [22]) have been ones in which the results are checked often enough that no more than $tmax$ errors will occur in the code word.

In this chapter, the error detecting capabilities of Bose-Lin codes beyond these $tmax$-errors are analyzed [20]. An example application where a Bose-Lin code could encounter such large number of errors is if it were used to detect faults in blocks of data transferred over a bus with a data path stuck fault. If the

bus is narrower than the block length, multiple bits can be affected by the bus fault. In other applications, it may be beneficial to apply the Bose-Lin to a larger block of data (or to data passing through more processing) in order to reduce the number of times the check needs to be performed, at the risk of exceeding *tmax*. In one recent example involving a complex system-on-chip, a Bose-Lin code has been used in conjunction with a linear-feedback shift register (LSFR) multiple-input signature register (MISR) in order to decrease the probability of undetected faults.

The chapter is organized as follows: Bose-Lin code constructions are briefly given in Section 3.2. Some analysis of Bose-Line codes capable of detecting more than the *tmax* errors is described in Section 3.3.

3.2. Bose-Lin Codes

The Bose-Lin codes can detect up to 2,3 and 6 errors using 2,3, and 4 check bits, respectively. For all check bits $r \geq 5$, two methods are used. Using r check bits, the codes designed based on Method 1 can detect up to $2^{r-2}+r-2$ errors and based on Method 2 up to $5 \times 2^{r-4} + r - 4$ errors. In this chapter, some analysis of the codes for detecting more than these maximum designed error detection capabilities are given. Before describing the main results, some definitions and notations, which are useful in studying the error detecting capabilities of these codes, are given.

Let $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$ be any two n-tuples over $GF(2)$. Let $N(X, Y)$ denote the number of $1 \rightarrow 0$ crossovers from X to Y. For example, if $X = 1011$ and $Y = 0101$, then $N(X, Y) = 2$ and $N(Y, X) = 1$. In

general $N(X,Y) \neq N(Y,X)$. From this definition, we can express the Hamming distance between X and Y as $D(X,Y) = N(X,Y) + N(Y,X)$.

Two words X and Y are called *unordered* if $N(X,Y) \geq 1$ and $N(Y,X) \geq 1$. For example if $X = 11011$, $Y = 10001$, and $Z = 00111$, then X covers Y, which is represented by $Y \leq X$, whereas X and Z are unordered. Further $Z \not\leq X$ indicates that X does not cover Z.

The following theorem describes the unidirectional error-detecting capability of block codes [11].

Theorem 3.2.1 *A code C is capable of detecting t-unidirectional errors if and only if for all $X, Y \in C$, either X and Y are unordered or $D(Y,X) \geq t+1$.*

In the next few paragraphs, we briefly describe the Bose-Lin codes. We assume that $k > 2^r$; otherwise, we could use *Berger codes* to detect all errors. The design technique of these codes are described in the following three cases.

3.2.1. Double and Triple Error-Detecting Codes

Double and triple error-detecting codes require 2, and 3 check bits, respectively. In this case, the check symbol CS for each code word is generated as follows. Count the number of 0's, k_0, in the information part and take this modulo 2^r, i.e. $CS \equiv k_0 \pmod 4$ for the double error detecting codes, and $CS \equiv k_0 \pmod 8$ for the triple error-detecting codes.

3.2.2. Error-Detecting Codes with 4 Check Bits

In this case, the check symbol CS for each code word is generated as follows. Count the number of 0's, k_0, in the information part taken modulo 8,

convert the result to a binary number, and finally add 4, which in binary is 0100, i.e. $CS \equiv k_0 \pmod 8$+4, where k_0 mod 8, and 4 are 4-bit binary numbers. In other words, the Most Significant Bit (MSB) of the check bit is the complement of the second MSB. This can detect up to 6 errors.

3.2.3. Error-Detecting Codes with more than four check bits

3.2.3.1. Method 1

Divide the check bits into two parts. The first part contains the first two bits of the check part. The two most significant bits can take 01 and 10 only. The other part contains the remaining $r-2$ check bits which take all 2^{r-2} possible binary $(r-2)$ tuples. So, the number of check symbols will be $2 \times 2^{r-2} = 2^{r-1}$.

The least $(r-1)$ check bits are obtained by taking k_0 mod (2^{r-1}) in binary. The MSB of the check bits is then obtained by complementing the second MSB of the check. For example, when $r = 5$ there will be $2^{5-1} = 16$ check symbols where the repetitive check symbol sequence for the information symbols will be 10111, 10110, 10101, 10100, 10011, 10010, 10001,10000, 01111, 01110, 01101, 01100, 01011, 01010, 01001, 01000. These codes are capable of detecting up to $2^{r-2} + r - 2$ errors.

3.2.3.2. Method 2

Divide the check bits into two parts. The first part contains the first four bits of the check part which always take any one of the 2-out-of-4 vectors namely, 0011, 0101, 0110, 1001, 1010, or 1100. The other part contains the remaining $r-4$

check bits which take any one among the 2^{r-4} possible binary $(r-4)$ values. So, the number of check symbols will be $6 \times 2^{r-4}$.

To generate the check symbols, first count the number of 0's in the information part of the code word taken mod $6 \times 2^{r-4}$ and then express it in $(r-1)$- bit binary, i.e. the intermediate check symbol for the received word will be $CS' = k_0 \pmod{(6 \times 2^{r-4})}$ where k_0 is the number of 0's in the information part. The 3 most significant bits for CS' can be {000, 001, 010, 011, 100, 101}. Next, define a $1-1$ mapping, f, from these symbols to 2-out-of-4-words to get the check. These codes are capable of detecting $5 \times 2^{r-4} + r - 2$ errors.

Notice that when $r = 5$, both methods detect up to 11 errors. But, when $r > 5$, codes designed by Method 2 are superior since $5 \times 2^{r-4} + r - 4 > 2^{r-2} + r - 2$.

3.3. Error detecting properties

The following notation is used in this Section:

k: number of information bits,

k_0: number of zeros in the information part of the code word,

k_0': number of zeros in the information part of the received word,

r: number of check bits,

n: $= k + r$, length of the code,

x_{ch}: check value of the code word,

x_{ch}': check value of the received word,

E: number of errors in the received word,

e: number of errors in the check part,

$E - e$: number of errors in the information part,

In this section, first we describe the error detecting capabilities of codes using 2 (and 3) check bits, when the number of errors are more than 2 (and 3) respectively. Then, some rules to check whether a given number of errors greater than $2^{r-2}+r-2$ (or $5\times 2^{r-4}+r-4$) can be detected or not using Method 1 (or Method 2) are given. The analysis is done under the assumption of $0 \rightarrow 1$ errors. They are also valid for $1 \rightarrow 0$ errors.

For the codes with $r = 2$ or 3 check bits, using $k_0' = k_0 - (E - e)$, the syndrome, S, can be defined as:

$$\begin{aligned} S &\equiv (x_{ch}' - k_0')(\bmod 2^r) \equiv [x_{ch}' - (k_0 - (E-e))] \bmod 2^r) \\ &\equiv [E - e - (x_{ch} - x_{ch}')](\mathrm{mod} 2^r). \end{aligned}$$

If $S = 0$, then the decoder declares that there is no error in the codeword. On the other hand, if $S \neq 0$ then there must be some errors in the received word. Error detecting capabilities of these codes with $r = 2$ and 3 can be analyzed using this expression for S.

Theorem 3.3.1 *The two check bits code detects E errors if $E \equiv 1$ (mod 4) or $E \equiv 2$ (mod 4).*

Proof: Let $S \equiv [(E-e) - (x_{ch} - x_{ch}')] \pmod 4$. If $E = 4j+1$, $j \in N$, then the values for $(E-e)$ and $(x_{ch} - x_{ch}')$ are as follows:

(i) $e = 0 \Rightarrow E - e = 4j+1$ and $(x_{ch} - x_{ch}') = 0$.

(ii) $e = 1 \Rightarrow E - e = 4j$ and $(x_{ch} - x_{ch}') = -1$ or -2.

(iii) $e = 2 \Rightarrow E - e = 4j-1$ and $(x_{ch} - x_{ch}') = -3$.

In all these cases $S \not\equiv 0 \pmod 4$.

Similarly, when $E = 4j+2$, the possible values are:

(i) $e = 0 \Rightarrow E - e = 4j + 2$ and $(x_{ch} - x'_{ch}) = 0$.

(ii) $e = 1 \Rightarrow E - e = 4j + 1$ and $(x_{ch} - x'_{ch}) = -1$ or -2.

(iii) $e = 2 \Rightarrow E - e = 4j$ and $(x_{ch} - x'_{ch}) = -3$.

Thus, $S \not\equiv 0 \pmod 4$ in all these cases.

On the other hand, when $E = 4j + 3$ and $e = 1$, there exist values $E - e = 4j + 2$ and $(x_{ch} - x'_{ch}) = -2$. For these values $S = 0$. Similarly, when $E = 4j$, $j \geq 1$ with $e = 0$, $S \equiv 0 \pmod 4$. In these two cases, errors are not detectable.

Theorem 3.3.2 *The code with 3 check bits detects E errors if E ≡ 1,2,3, and 6 (mod 8).*

Proof: Since $E = 1, 2, 3$ errors can be detected using $r = 3$ check bits [11], then we have

$$\forall e = 0, 1, 2, 3, \; [(E - e) - (x_{ch} - x'_{ch})] \bmod 8 \not\equiv 0,$$

where x_{ch} and x'_{ch} are the check values of the code word and the received word respectively, and $E - e$, and e are respectively the number of errors in the received information and check parts. Thus,

$$\forall j \geq 1, \; [(E - e) - (x_{ch} - x'_{ch}) + 8j] \bmod 8 \not\equiv 0.$$

$$\Rightarrow \forall j \geq 1, \; [(E + 8j - e) - (x_{ch} - x'_{ch})] \bmod 8 \not\equiv 0.$$

$\Rightarrow \forall \; j \geq 1$, and $\forall \; E = 1, 2, 3$ errors, $E + 8j$ can be detected using 3 check bits. For $E = 8j + 6$, the following possibilities can occur:

(i) $e = 0 \Rightarrow E - e = 8j + 6$ and $(x_{ch} - x'_{ch}) = 0$.

(ii) $e = 1 \Rightarrow E - e = 8j + 5$ and $(x_{ch} - x'_{ch}) = -1, -2$, or -4.

(iii) $e = 2 \Rightarrow E - e = 8j + 4$ and $(x_{ch} - x'_{ch}) = -3, -5$, or -6.

(iv) $e = 3 \Rightarrow \ E - e = 8j + 3$ and $(x_{ch} - x'_{ch}) = -7$.

In all these cases, $S \not\equiv 0 \pmod 8$.

On the other hand,

(i) When $E = 8j+4$, there exists $e = 2$ such that $E-e = 8j+2$ and $(x_{ch}-x'_{ch}) = -6$. In this case $S \equiv 0 \pmod 8$.

(ii) When $E = 8j+5$, there exists $e = 1$ such that $E-e = 8j+4$ and $(x_{ch}-x'_{ch}) = -4$. In this case $S \equiv 0 \pmod 8$.

(iii) When $E = 8j + 7$, there exists $e = 1$ such that $E - e = 8j + 6$ and $(x_{ch} - x'_{ch}) = -2$. Again $S \equiv 0 \pmod 8$.

(iv) With $E = 8j$, $j \geq 1$ and $e = 0$, in this case $S \equiv 0 \pmod 8$.

In all these cases, the errors are not detectable.

Theorem 3.3.3 *The code with 4 check bits detects E errors if* $E \equiv$ *1,2,3,4,5 and 6 (mod 8).*

Proof: Since $E = 1, 2, 3, 4, 5$ and 6 errors can be detected using $r = 4$ check bits [11], then we have,

$$\forall\ e = 0, 1, 2, 3, 4, \text{ and } E = 1, 2, 3, 4, 5\ ,\ [(E - e) - (x_{ch} - x'_{ch})] \bmod 8 \not\equiv 0$$

$$\forall j \geq 1,\ [(E - e) - (x_{ch} - x'_{ch}) + 8j] \bmod 8 \not\equiv 0\ .$$

$$\Rightarrow \forall\ j \geq 1,\ [(E + 8j - e) - (x_{ch} - x'_{ch})] \bmod 8 \not\equiv 0\ .$$

$\Rightarrow \forall\ j \geq 1$, and $\forall\ E = 1, 2, 3, 4, 5$, and 6 errors, $E + 8j$ errors can be detected using 4 check bits.

On the other hand, when $E = 8j+7$, there exist values $E-e = 8j+5$ and $j \geq 1$ with $(x_{ch} - x'_{ch}) = -3$. For these values, $S = 0$. Similarly, when $E = 8j$, $j \geq 1$ with $e = 0$, $S \equiv 0 \pmod 8$. In these two cases, errors are not detectable.

For codes designed by Method 1, if there is an error in the first two bits of the check, then the decoder immediately detects this error. Thus, in the following analysis, it is assumed that there is no error in the two MSB of the check. For similar reason, in the analysis, it is assumed that there is no error in any of the four MSB of the check for the codes designed by Method 2.

Definition 5 *Let L_r be the designed maximum number of errors detected by Bose-Lin code when using r check symbols. Then for Method 1: $L_r = 2^{r-2} + r - 2$, and for Method 2: $L_r = 5 \times 2^{r-4} + r - 4$*

For codes designed by Method 1, the syndrome, can be defined as:

$$S \equiv (x'_{ch} - k'_0) \pmod{2^{r-1}} \equiv [x'_{ch} - (k_0 - (E-e))] \pmod{2^{r-1}}$$
$$\equiv [(E-e) - (x_{ch} - x'_{ch})] \pmod{2^{r-1}}.$$

Similarly, for Method 2, the syndrome is defined as:

$$S \equiv [(E-e) - (x_{ch} - x'_{ch})] \pmod{(6 \times 2^{r-4})}.$$

Lemma 1 *(i) For Method 1, $2^{r-1}j$ errors can't be detected using r check bits for some e errors in the check part, $e = 1, 2, 3, \cdots, r-2$ and $j \geq 1$.*
(ii) For Method 2, $6 \times 2^{r-4}j$ errors can't be detected using r check bits for some e errors in the check part, $e = 1, 2, 3, \cdots, r-4$ and $j \geq 1$.

Proof:

(i) $\forall r \geq 4$, when $e = 1$, $x_{ch} - x'_{ch}$ can be equal to 1. Thus

$$S \equiv [(E-e) - (x_{ch} - x'_{ch})] \pmod{(2^{r-1})} \equiv [(2^{r-1}j - 1) - (-1)] \pmod{(2^{r-1})} \equiv 0.$$

$\Rightarrow 2^{r-1}j$ errors can't be detected using r check bits, $j \geq 1$.

(ii) The proof is similar to the previous one except mod $(6 \times 2^{r-4})$ needs to be used instead of mod (2^{r-1}).

Lemma 2 *(i) For Method 1, $E + 2^{r-1}j$ errors can be detected using r check bits iff E errors can be detected using r check bits, $j \geq 1$.*
(ii) For Method 2, E can be detected using r check bits iff $E + 6 \times 2^{r-4}j$ can be detected using r check bits, $j \geq 1$.

Proof:

(i) E can be detected using r check bits iff

$$\forall e = 1, 2, \cdots, r-2, \ [(E-e) - (x_{ch} - x'_{ch})] \bmod (2^{r-1}) \not\equiv 0.$$

$$\Leftrightarrow \forall e = 1, 2, \cdots, r-2, \ [(E-e) - (x_{ch} - x'_{ch}) + 2^{r-1}j] \bmod (2^{r-1}) \not\equiv 0,$$

$j = 1, 2, 3 \cdots$.

$$\Leftrightarrow \forall e = 1, 2, \cdots, r-2, \ [((E + 2^{r-1}j) - e) - (x_{ch} - x'_{ch})] \bmod (2^{r-1}) \not\equiv 0,$$

$j = 1, 2, 3, \cdots$.

$$\Leftrightarrow E + 2^{r-1}j \text{ can be detected using } r \text{ check bits, } j \geq 1.$$

(ii) The proof is similar to the previous one except mod $(6 \times 2^{r-4})$ needs to be used instead of mod (2^{r-1}).

Lemma 3 *(i) For Method 1, not all $2^{r-1}j - 1$ errors can be detected using r check bits with e errors in the check part, $e = 1, 2, 3, \cdots, r-2$, and $j \geq 1$.*
(ii) For Method 2, not all $6 \times 2^{r-4}j - 1$ errors can be detected using r check bits with e errors in the check part, $e = 1, 2, 3, \cdots, r-4$, and $j \geq 1$.

Proof:

(i) $\forall r \geq 1$, when $e = 2$, $x_{ch} - x'_{ch}$ can be equal to 3. Then,

$$S \equiv [(E - e) - (x_{ch} - x'_{ch})] \pmod{(2^{r-1})}$$
$$\equiv [((2^{r-1}j - 1) - 2) - (-3)] \pmod{(2^{r-1})} \equiv 0.$$

Thus, some $2^{r-1}j - 1$ errors can't be detected using r check bits, $j \geq 1$.

(ii) The proof is similar to the previous one except mod $(6 \times 2^{r-4})$ needs to be used instead of mod (2^{r-1}).

Theorem 3.3.4 *For Method 1, let E' be the number of errors in the codeword and let $E \equiv E'(mod\ (2^{r-1}))$. Then E' can be detected using r check bits if E satisfies one of the following conditions:*

(1) $1 \leq E \leq L_r = 2^{r-2} + r - 2$.

(2) *For any E in the range $L_r = 2^{r-2} + r - 2 < E < B_r = L_{r-1} + 2^{r-2} - (r-5) = 2^{r-2} + 2^{r-3} + 2$, $E - (L_r - L_{r-1}) = E - (2^{r-3} + 1)$ errors can be detected using $r - 1$ check bits.*

(3) *For any E in the range $B_r = 2^{r-2} + 2^{r-3} + 2 \leq E \leq 2^{r-1}$, E errors can be detected using $r - 1$ check bits.*

Proof:

Case (1): Already proved in [11].

Case (2): Assume that $E_1 = E - (L_r - L_{r-1})$ errors can be detected using $r - 1$ check bits. Then $\forall e_1 = 0, 1, 2, 3, \cdots, r - 3$,

$$[(E_1 - e_1) - (y_{ch} - y'_{ch})] \text{mod}(2^{r-2}) \not\equiv 0, \tag{3.1}$$

where y_{ch}, and y'_{ch} are the check values of the code word and the received word respectively. Further, $E_1 - e_1$, and e_1 are respectively the number of errors in the received information and check parts.

Now, suppose that E errors can't be detected using r check bits. Then there exists at least one value of

$$e = 1, 2, \cdots, r-2 \text{ such that } [(E-e) - (x_{ch} - x'_{ch})] \bmod (2^{r-1}) \equiv 0 \ .$$

Since $E = E_1 + (L_r - L_{r-1}) = E_1 + 2^{r-3} + 1$, we will have

$$[(E_1 - e) - (x_{ch} - x'_{ch}) + (2^{r-3} + 1)] \bmod 2^{r-1} \equiv 0.$$

$$\Rightarrow [(E_1 - e) - [(x_{ch} - x'_{ch}) - (2^{r-3} + 1)]] \bmod (2^{r-1}) \equiv 0.$$

Since $E_1 < E < B_r < 2^{r-1}$, we will have

$$[(E_1 - e + 1) - [(x_{ch} - x'_{ch}) - 2^{r-3}]] \bmod (2^{r-2}) \equiv 0.$$

$$\Rightarrow [(E_1 - e + 1) - [(x_{ch} - x'_{ch}) - 2^{r-3}] - 2^{r-2}] \bmod (2^{r-2}) \equiv 0.$$

$$\Rightarrow [(E_1 - e + 1) - [(x_{ch} - x'_{ch}) + 2^{r-3}]] \bmod (2^{r-2}) \equiv 0.$$

When $e = e_1 + 1$, there exists at least one value of $e = 1, 2, 3, \cdots, r-2$ such that $(x_{ch} - x'_{ch}) + 2^{r-3}$ is equal to $(y_{ch} - y'_{ch})$ and in this case we will have

$$[(E_1 - e_1) - (y_{ch} - y'_{ch})] \bmod (2^{r-2}) \equiv 0.$$

$\Rightarrow E_1$ errors can't be detected using $r-1$ check bits which is a contradiction to the original assumption.

Hence, E can be detected using r check bits and so, E' can be detected using r check bits by applying Lemma (2).

Case (3): Assume that E can be detected using $r-1$ check bits. Then $\forall e_1 = 0, 1, 2, 3, \cdots, r-3$,

$$[(E - e_1) - (y_{ch} - y'_{ch})] \bmod (2^{r-2}) \not\equiv 0 \tag{3.2}$$

where y_{ch}, and y'_{ch} are the check values of the code word and the received word respectively. Further, $E - e_1$, and e_1 are respectively the number of errors in the

received information and check parts.

By using r check bits, there are $\binom{r-2}{e}$ different values for each$(x_{ch} - x'_{ch})$, $e = 1, 2, 3, \cdots, r-2$. The first $\binom{r-3}{e}$ values of $(x_{ch} - x'_{ch})$, (i.e. with the $(r-3)$rd check bit not in error), are the same values as the corresponding $(y_{ch} - y'_{ch})$ using $r-1$ check bits. Then, for all of these values, using $E \leq 2^{r-1}$, we obtain

$$[(E-e) - (x_{ch} - x'_{ch})] \text{mod}(2^{r-1}) \not\equiv 0, 1 \leq (x_{ch} - x'_{ch}) \leq 2^{r-3} \tag{3.3}$$

The remaining $\binom{r-3}{e-1}$ values of $(x_{ch} - x'_{ch})$ are equal to $-(2^{r-3} + A)$ where

$A = zero$ when $e = 1$, or

$A =$ summation of any $e-1$ distinct numbers from $\{2^0, 2^1, 2^2, \cdots, 2^{r-3}, 2^{r-2}\}$, $e = 2, 3, 4, \cdots, r-2$.

Now, for these values, we want to prove that:

$$[(E-e) - (x_{ch} - x'_{ch})] \text{mod}(2^{r-1}) \not\equiv 0 \ . \tag{3.4}$$

Assume that $M = (E-e) - (x_{ch} - x'_{ch})$. Since $2^{r-1} > E \geq B_r$, we will have

$$2^{r-1} > M \geq (B_r - e) - (x_{ch} - x'_{ch}) \ .$$

$$\Rightarrow M \geq [2^{r-3} + (r-3) + 2^{r-2} - (r-5)] - e - (x_{ch} - x'_{ch})] \ .$$

$$\Rightarrow M \geq [2^{r-3} + (r-3) + 2^{r-2} - (r-5)] - e + (2^{r-3} + A)] \ .$$

Since $2^{r-1} > E \geq M$, we get

$$M \bmod (2^{r-1}) \geq (2^{r-2} + 2 + 2^{r-2} - e + A) \bmod (2^{r-1}) \ .$$

$$\Rightarrow M \bmod (2^{r-1}) \geq (2^{r-1} + 2 - e + A) \bmod (2^{r-1}).$$

$$\Rightarrow M \bmod (2^{r-1}) \geq (2 - e + A),\ e = 1, 2, 3, \cdots, r-2.$$

When $e = 1$, $A = 0 \Rightarrow 2 - e + A = 1 \neq 0$.

When $e \neq 1$, the value of A can be $3 \leq A < 2^{r-1}$, and so $\forall e > 1$, $2 - e + A > 0$.

Thus, $\forall e$, $M \bmod (2^{r-1}) \not\equiv 0$.

$$\Rightarrow [(E - e) - (x_{ch} - x'_{ch})] \bmod (2^{r-1}) \not\equiv 0\,. \tag{3.5}$$

This implies that E errors can be detected using r check bits. Applying Lemma (2), we get E' errors can be detected using r check bits. ∎

Table 3.1 lists the errors that can be detected using two, three, and four check bits, and with $r \geq 5$ check bits using Method 1.

Example 1 *Let us check whether* $E = 110$ *errors can be detected or not using Method 1 with* $r = 8$ *check bits. Since* $r = 8$, *we have* $B_8 = 98$ *and* $L_8 = 70$. *Thus, we need to apply Rule (3) to check whether 110 errors can be detected using 7 check bits. Now, we can apply Lemma (2) to check this. Since* $110 \equiv 46 \bmod 2^6$, *we need to verify whether 46 errors can be detected using 7 check bits. In this case,* $B_7 = 50$ *and* $L_7 = 37$. *Since* $37 = L_7 < 46 < B_7 = 50$, *we need to apply Rule (2), i.e. need to check whether* $46 - (2^{7-3} + 1) = 29$ *errors can be detected using 6 check bits. Since* $29 \geq B_6 = 14$, *we need to apply Rule (3), i.e. need to check whether 29 errors can be detected using 5 check bits.*

Now, apply Lemma (2) again, i.e. need to check whether $29 \bmod 2^4 \equiv 13$ *errors can be detected using 5 check bits. Since* $11 = L_5 < 13 < B_5 = 26$, *we need to apply Rule (2), i.e. need to check* $13 - (2^2 + 1) = 8$ *errors can be detected using 4 check bits. Applying Lemma (1), we know that 8 errors can't be detected using 4 check bits for some errors. This implies that some* $E = 110$ *errors can't be detected using 8 check bits.*

Theorem 3.3.5 *For Method 2, let E' be the number of errors in the code word and let $E = E'(mod(6 \times 2^{r-4}))$. Then E' can be detected using r check bits if E satisfies one of the following conditions:*

(1) $1 \leq E \leq L_r = 5 \times 2^{r-4} + (r-4)$.

(2) *For any E in the range $L_r = 5 \times 2^{r-4} + (r-4) < E < B_r = L_{r-1} + 6 \times 2^{r-5} - (r-7) = 11 \times 2^{r-5} + 2$, $E - (L_r - L_{r-1}) = E - (5 \times 2^{r-5} + 1)$ errors can be detected using $r-1$ check bits.*

(3) *For any E in the range $B_r = 11 \times 2^{r-5} + 2 \leq E \leq 6 \times 2^{r-4}$, E can be detected using $r-1$ check bits.*

Proof:

Case (1): Already proved in [11].

Case (2): Assume that $E_1 = E - (L_r - L_{r-1})$ can be detected using $r-1$ check bits. Then $\forall e_1 = 0, 1, 2, 3, \cdots, r-5$,

$$[(E_1 - e_1) - (y_{ch} - y'_{ch})] \text{mod}(2^{r-2}) \not\equiv 0 \tag{3.6}$$

where y_{ch}, and y'_{ch} are the check parts in the code word and the received word respectively. Further, $E_1 - e_1$, and e_1 are respectively the number of errors in the received information and check parts.

Now, suppose that E errors can't be detected using r check bits. Then there exists at least one value of $e = 0, 1, 2, \cdots, r-4$ such that

$$[(E - e) - (x_{ch} - x'_{ch})] \text{ mod } (6 \times 2^{r-4}) \equiv 0.$$

Since $E = E_1 + (L_r - L_{r-1})$, we will have

$$[(E_1 - e) - (x_{ch} - x'_{ch}) + (5 \times 2^{r-5} + 1)] \text{ mod } (6 \times 2^{r-4}) \equiv 0.$$

$$\Rightarrow [(E_1 - (e_1 + 1)) - [(x_{ch} - x'_{ch}) - (5 \times 2^{r-5} + 1)]] \bmod (6 \times 2^{r-4}) \equiv 0.$$

$$\Rightarrow [(E_1 - e_1) - [(x_{ch} - x'_{ch}) - 5 \times 2^{r-5}] - 6 \times 2^{r-5}] \bmod (6 \times 2^{r-5}) \equiv 0.$$

$$\Rightarrow [(E_1 - e_1) - [(x_{ch} - x'_{ch}) + 2^{r-5}]] \bmod (6 \times 2^{r-5}) \equiv 0.$$

When $e = e_1 + 1$, there exists at least one value of $e = 1, 2, 3, \cdots, r-4$ such that

$$(x_{ch} - x'_{ch}) + 6 \times 2^{r-5} = (y_{ch} - y'_{ch}).$$

and in this case we will have

$$[(E_1 - e_1) - (y_{ch} - y'_{ch})] \bmod (6 \times 2^{r-5}) \equiv 0.$$

$\Rightarrow E_1$ can't be detected using $r-1$ check bits which is a contradiction to the original assumption.

Hence, E can be detected using r check bits and so, E' can be detected using r check bits.

Case (3): Assume that E can be detected using $r-1$ check bits. Then $\forall e_1 = 0, 1, 2, 3, \cdots, r-5$,

$$[(E - e_1) - (y_{ch} - y'_{ch})] \bmod [6 \times 2^{r-5}] \not\equiv 0\,. \tag{3.7}$$

where y_{ch}, and y'_{ch} are the check parts in the code word and the received word respectively. Further, $E - e_1$, and e_1 are respectively the number of errors in the received information and check parts.

By using r check bits, there are $\binom{r-4}{e}$ different values for each

$$(x_{ch} - x'_{ch}),\ e = 1, 2, 3, \cdots, r-4.$$

The first $\binom{r-5}{e}$ values of $(x_{ch} - x'_{ch})$ (i.e. with the $(r-5)$th check bit not in error), are the same values as the corresponding $(y_{ch} - y'_{ch})$ using $r-1$ check symbols. Then, for all of these values, using $E \leq 6 \times 2^{r-4}$

$$[(E-e)-(x_{ch}-x'_{ch})]\text{mod}(6\times 2^{r-4}) \not\equiv 0\,. \tag{3.8}$$

The remaining $\binom{r-5}{e-1}$ values of $(x_{ch}-x'_{ch})$ are equal to $-(2^{r-5}+A)$ where

A = zero when $e=1$, or

A = summation of any $e-1$ distinct numbers from $\{2^0, 2^1, 2^2, \cdots, 2^{r-5}, 2^{r-4}\}$, $e = 2, 3, 4, \cdots, r-4$.

Now, for these values, we want to prove that:

$$[(E-e)-(x_{ch}-x'_{ch})]\text{mod}(6\times 2^{r-4}) \not\equiv 0\,. \tag{3.9}$$

Assume that

$$M = (E-e)-(x_{ch}-x'_{ch}) \leq 6\times 2^{r-4}\,.$$

Since $E \geq B_r$, then we will have $M \geq (B_r - e) - (x_c h - x'_c h)$

$$\Rightarrow\ M \geq [5\times 2^{r-5}+(r-5)+6\times 2^{r-5}-(r-7)]-e-(x_{ch}-x'_c h)]\,.$$

$$\Rightarrow\ M \geq [5\times 2^{r-5}+(r-5)+6\times 2^{r-5}-(r-7)]-e+(2^{r-5}+A)]\,.$$

$$\Rightarrow\ M\text{mod}(6\times 2^{r-4}) \geq (6\times 2^{r-4}+2-e+A)\ \text{mod}(6\times 2^{r-4}).$$

$$\Rightarrow M\text{mod}(6\times 2^{r-4}) \geq (2-e+A), e=1,2,3,\cdots,r-4.$$

When $e=1$, $A=0 \Rightarrow 2-e+A=1\neq 0$.

When $e\neq 1$, the values of A can be $3 \leq A < 2^{r-3}$, and so $2-e+A>0$.

Thus, M mod $(6\times 2^{r-4}) \not\equiv 0$.

$$\Rightarrow \forall e, [(E-e)-(x_{ch}-x'_{ch})]\text{mod}[6\times 2^{r-4}] \not\equiv 0\,. \tag{3.10}$$

This implies that E errors can be detected using r check bits. Applying Lemma (2), we get E' errors can be detected using r check bits. ∎

Table 3.2 lists the errors that can be detected using Method 2.

Example 2 *Let us check whether* $E = 330$ *errors can be detected or not using Method 2 with* $r = 10$ *check bits. Since* $r = 10$, *then we have* $B_{10} = 354$ *and* $L_{10} = 326$. *Thus we need to apply Rule (2) to check whether* $330-(5\times 2^5+1) = 169$ *errors can be detected using 9 check bits. In this case, since* $165 = L_9 < 169 < B_9 = 178$, *we need to apply Rule (2), i.e. need to check whether* $169 - (5 \times 2^4 + 1) = 88$ *errors can be detected using 8 check bits. Since* $84 = L_8 < 88 < B_8 = 90$, *we need to apply Rule (2), i.e. need to check whether* $88 - (5 \times 2^3 + 1) = 47$ *errors can be detected using 7 check bits.*

Since $46 = B_7 \leq 47 \leq 6 \times 2^3 = 48$, *we need to apply Rule (3), i.e. need to check whether 47 errors can be detected using 6 check bits. Now, apply Lemma (2) to check this. Since* $47 \equiv 23 \bmod 6 \times 2^2$, *we need to verify whether 23 errors can be detected using 6 check bits. Applying Lemma (3), we know that all 23 errors can't be detected using 5 check bits for some errors. This implies that some 330 errors can't be detected using 10 check bits.*

$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$
1	1	1	1	1	1	1
$2=L_2$	2	2	2	2	2	2
———	$3=L_3$	.	.	.	.	.
5	6	.	.	.	.	.
6	———	$6=L_4$	$11=L_5$	$20=L_6$	$37=L_7$	$70=L_8$
———	9	———	$14=B_5$	23	40	73
↓	10	9	———	$26=B_6$	43	76
	11	10	17	27	44	77
	14	11	18	30	47	80
	———	12	.	———	$50=B_7$	83
	↓	13	.	33	51	84
		14	25	34	52	85
		———	26	.	55	88
		↓	27	.	58	91
			30	50	59	92
			———	51	62	95
			↓	52	———	$98=B_8$
				55	65	99
				58	66	100
				59	.	101
				62	.	104
				———	98	107
				↓	99	108
					100	111
					101	114
					104	115
					107	116
					108	119
					111	122
					114	123
					116	126
					119	———
					122	129

TABLE 3.1. The errors that can be detected using 2,3, and 4 check bits, and with $r \geq 5$ check bits using Method 1.

$r = 5$	$r = 6$	$r = 7$	$r = 8$	$r = 9$	$r = 10$
1	1	1	1	1	1
2	2	2	2	2	2
.	.	.	.	.	.
.	.	.	.	.	.
$11 = L_5$	$22 = L_6$	$43 = L_7$	$84 = L_8$	$165 = L_9$	$326 = L_{10}$
———	———	$46 = B_7$	87	168	329
13	25	———	$90 = B_8$	171	332
14	26	49	91	172	333
.	.	50	94	175	336
.	.	.	———	$178 = B_9$	339
23	46	90	97	179	340
———	———	91	98	180	341
↓	↓	94	.	183	344
		———	.	186	347
		↓	178	187	348
			179	190	351
			180	———	$354 = B_{10}$
			183	193	355
			186	194	356
			187	.	357
			190	.	360
			———	354	363
			↓	355	364
				356	367
				357	370
				363	371
				364	372
				367	375
				370	378
				371	379
				372	382
				375	———
				378	385
				379	386

TABLE 3.2. The errors that can be detected using Method 2.

4. SYSTEMATIC T-UNIDIRECTIONAL ERROR DETECTING CODES IN Z_M

4.1. Introduction

Errors correcting/detecting codes are used in providing protection against transient, intermittent, and permanent faults [5]. The errors that can occur because of the noise are many and varied. However, they can be classified into three main types, symmetric, asymmetric, and unidirectional errors.

Let $X = (x_{n-1}, x_{n-2}, \cdots, x_0)$ be a transmitted word through a noisy channel, and $Y = (y_{n-1}, y_{n-2}, \cdots, y_0)$ be the received word, where the symbols x_i's and y_i's are over Z_m. Define the error value $E = (e_{n-1}, e_{n-2}, \cdots, e_0) = (y_{n-1} - x_{n-1}, y_{n-2} - x_{n-2}, \cdots, y_0 - x_0)$ and $(y_i - x_i)$'s are over the integers. Based on the error value, E, the error types can be classified as asymmetric, unidirectional or symmetric. In the case of asymmetric type, at all time, the e_i's have values less than or equal to 0 (or at all time, the values of e_i's are greater than or equal to 0). In the case of unidirectional errors, again all the error values can be positive or all the error values can be negative within a word but this condition is not known a priori, i.e. when transmitting, the error values for one word can all be positive and for another word they can all be negative. Finally, in the case of symmetric errors, within a word the error values can be both positive and negative. If $E = \underline{0}$ then there is no error in the transmitted word. Further, the number of non-zero values in E gives the number of errors.

Given two words X and Y over Z_m^n, define $N(X, Y)$ as the number of positions at which $x_i > y_i$ for $i = 0, 1, 2, \cdots, n-1$. For example, if $X = (4321)$ and $Y = (1312)$, then $N(X, Y) = 2$, and $N(Y, X) = 1$. Note that the *Hamming distance* between X and Y is $D_H(X, Y) = N(X, Y) + N(Y, X)$. For X and Y, if

$N(X,Y) \geq 1$ and $N(Y,X) \geq 1$, then they are called *unordered* words. On the other hand, if $N(X,Y) = 0$, then it is said that Y *covers* X. For example, (4321) covers (2221).

A *chain* of length M over Z_m is defined as the sequence of M words such that successive words differ in one position by 1. For example, <0001, 0002, 0003, 0013, 0023, 0033, 0133, 0233, 0333> is a chain of length 9 over Z_4^4.

Theorem 4.1.1 *A code C is capable of detecting all unidirectional errors if and only if the code satisfies the following condition:*

$$\forall X, Y \in C,\ N(X,Y) \geq 1 \text{ and } N(Y,X) \geq 1.$$

In the case of binary, i.e. $m = 2$, by Sperner's theorem [17], the $\lfloor n/2 \rfloor$-out-of-n code is the optimal all unidirectional error detecting (AUED) code, i.e. this code gives the maximum number of unordered codewords for a given n. Further, the Berger-Freiman code [21, 3], is the optimal AUED systematic code. In a systematic code, the information digits are separated from the check digits.

For $m \geq 3$, as shown in [17], the set of n digits words S, such that for each $X = (x_{n-1}, x_{n-2}, \cdots, x_0) \in S$, $\sum_{i=0}^{n-1} x_i = \lfloor (m-1)n/2 \rfloor$ gives the maximum number of unordered codewords. For example, when $m = 5$ and $n = 3$, the unordered codewords are { (420),(411), (402), (330), (321), (312), (303), (240), (231), (222), (213), (204), (141), (132), (123), (114), (042), (033), (024)}, and there are 19 of them. The theorem given in [17] says that we can not have more than 19 codewords when $m = 5$ and $n = 3$.

Further, Bose and Pradhan in [12] have given optimal systematic AUED codes over Z_m. For a given k-digit information word $X = (x_{k-1}, x_{k-2}, \cdots, x_0)$, where x_i's $\in Z_m$, the check value is given by $\underline{v} = \sum_{i=0}^{k-1}((m-1) - x_i)$. The r- digit check is obtained by representing $\underline{v}$ in radix-m form, where $r =$

$\lceil log_m((m-1)k+1)\rceil$. For example, when $X = (3420)$ is the given $k = 4$ information digits over Z_5, the check value is $(4-3)+(4-4)+(4-2)+(4-0) = 7$ and the check is 12. As it is shown in [12], this is optimal systematic AUED code over Z_m.

In this chapter, systematic t-UED codes are described. It is assumed that the number of check digits r, satisfies the condition $r < \lceil log_m((m-1)k+1)\rceil$, where k is the number of information digits [10]; otherwise, one could use the AUED code given by Bose-Pradhan in [12]. In the case of binary, optimal non-systematic t unidirectional codes are given in [7] and systematic codes in [11].

4.2. Code Construction

The following theorem is proved in [7, 11] for binary. However, for completeness, the proof is given here and it is similarly to the one given in [7, 11].

Theorem 4.2.1 *A code C is capable of detecting t-unidirectional errors if and only if C satisfies the following condition:*

$$\forall X, Y \in C, \text{ either } X \text{ and } Y \text{ are unordered, or } D_H(X,Y) \geq t+1.$$

Proof: Consider any two codewords $X = (x_{n-1}, x_{n-2}, \cdots, x_0)$, and $Y = (y_{n-1}, y_{n-2}, \cdots, y_0)$ in C. If they are unordered, then there exists x_i, y_i, x_j, and y_j such that $x_i > y_i$, and $y_j > x_j$ where $0 \leq i, j \leq n-1$. Suppose X is transmitted word, and let the received word be $X' = (x'_{n-1}, x'_{n-2}, \cdots, x'_0)$ where $x'_p = x_p + e_p$ for all $p = 0, 1, 2, \cdots n-1$. If $e_p \geq 0$ for all $p = 0, 1, 2, \cdots n-1$, then X' differs from Y in the i-th position because $x_i > y_i$. Similarly, if $e_p \leq 0$ for all $p = 0, 1, 2, \cdots n-1$ then X' differs from Y in the j-th position because $y_j > x_j$. Further, if they are ordered pair and $D_H(X,Y) \geq t+1$, then X can not become Y due to t or

less unidirectional errors. Thus, if the code satisfies the conditions given by the theorem, it can detect up to t unidirectional errors.

Conversely, for $X, Y \in \mathcal{C}$, if they are ordered pair and $D_H(X,Y) = b \leq t$, then X can become Y due to b errors and so the code can not detect t unidirectional errors. ∎

From error detecting point of view, there is no difference between asymmetric and unidirectional errors as described in the following theorem.

Theorem 4.2.2 *A code $\mathcal{C}$ is capable of detecting t-asymmetric errors iff the code satisfies the following condition:*

$$\forall X, Y \in \mathcal{C}, \text{ either } X \text{ and } Y \text{ are unordered or } D_H(X,Y) \geq t+1$$

Now, we describe the code design scheme. As it was mentioned earlier, it is assumed that the number of information digits, $k > (m^r - 1)/(m-1)$.

(a) Code Design with $r = 2$ check bits:

Let $(a_{k-1}a_{k-2}\cdots a_0)$ be the given information word over Z_m, and let $b \equiv \sum_{i=0}^{k-1}((m-1) - a_i)) \bmod m^2$. Then, the check digits are the representation of b in radix-m system. This code is capable of detecting two errors.

Example 3 *Let $(44\cdots4230)$ be the given information word over Z_5, then we will have $b \equiv ((4-4)+(4-4)+\cdots(4-4)+(4-2)+(4-3)+(4-0))$ mod $25 = 7$. Hence, the check is* 12.

The error detecting capability of the code can be shown as follows. Let $XC = (x_{k-1}x_{k-2}\cdots x_0c_1c_0)$ be the transmitted word and $X'C' = (x'_{k-1}, x'_{k-2}\cdots x'_0c'_1c'_0)$ be the received word. Further, let $E = (e_{k-1}e_{k-2}\cdots e_0e'_1e'_0)$ where $e_i = x_i - x'_i$ for $i = 0, 1, \cdots k-1$ and $e'_i = c_i - c'_i$ for $i = 0, 1$. Let

$C'' \equiv \sum_{i=0}^{k-1}(m-1-x_i')$ mod m^2. If $C''-C' \equiv 0$ mod m^2, then the decoder declares that there is no error. We can consider the following three cases:

(1) Errors only in the information part:

In this case $C'' = \sum_{i=0}^{k-1}(m-1-x_i') = \sum_{i=0}^{k-1}(m-1-x_i+e_i') \equiv (C+E_1+E_2)$ mod m^2, where E_1 and E_2 are the error values and $0 \leq E_1, E_2 < m$ [When there is only one error, then E_2=0], then $C''-C' = (E_1+E_2) \not\equiv 0$ mod m^2. Thus, these errors can be detected.

(2) Errors only in the check part:

There may be one digit in error or both the digits may be in error. In any case $C'' = C$ and $C' = C - v$ where $1 \leq v \leq m^2-1$. Thus, $C''-C' \not\equiv 0$ mod m^2.

(3) Errors in both the information and check parts:

Now, $C'' = C + E_1$, and $C' = C - E_2$, where $1 \leq E_1 \leq m-1$ and $0 \leq E_2 \leq m(m-1)$. Thus, $C''-C' = E_1+E_2 \not\equiv 0$ mod m^2.

(b) Code Design with $r \geq 3$ check bits:

The proposed code can detect up to $m^{r-2}+r-2$ errors using r check digits. Let $X = (x_{k-1}, x_{k-2}, \cdots, x_0)$ be the given information word and let $\underline{v} \equiv (\sum_{i=0}^{k-1}(m-1-x_i))$ mod m^{r-1}. Represent $\underline{v}$ in radix-m system and let it be $\underline{v} = (c_{r-2}, c_{r-3}, \cdots, c_0)$. Thus, the check is given by $(m-1-c_{r-2})c_{r-2}c_{r-3}\cdots c_0$.

Example 4 *Let $m = 5$ and $r = 4$, suppose that the given information word is $X = (44\cdots442411)$. So, $\underline{v} \equiv [(4-4)+(4-4)\cdots(4-4)+(4-4)+(4-2)+(4-4)+(4-1)+(4-1)]$ mod $[5^3] \equiv 8$ mod 5^3. 8 in radix-5 system is* 013 *and hence, the check is* 4013.

Now, we can analyze the error detecting capabilities of these codes. Note that the first two digits of the check can be $\{(0, m-1), (1, m-2)(2, m-3) \cdots (m-1, 0)\}$ and these are unordered. The remaining $(r-2)$-check digits can take all possible m^{r-2} values. Now, consider a k-digit maximal chain, starting from $(000\ldots0)$ and ending with $(m-1, m-1, \cdots m-1)$. For example, when $m = 4$ and $k = 5$, <(00000) (00001) (00002) (00003) (00013) (0023) (00033) (00133) (00233) (00333) (001333) (02333) (03333) (13333) (23333) (33333)> is a complete chain. This chain can be divided into $k+1$ classes, with the first word which has all 0's in the zero-th class, the word $(i-1)m+1$ through $(i-1)m+m-1$ in the i-th class, $i = 1, 2, \cdots k$. Thus, each class contains $(m-1)$ words, except the zero-th class which has only one element. The Hamming distance between a word in the i-th class and another word in the j-th class is at least $|j-i|$. The Hamming distance between any two words within a class is 1. After appending the check symbols, if two words are within j position apart in this chain where $1 \leq j \leq m^{r-1}$, then the codewords are unordered. Further, if a codeword X covers another codeword Y in this chain, then

$$D_H(X, Y) \geq (m-1)\frac{m^{r-2}}{m-1} + r - 2 + 1 = m^{r-2} + r - 1.$$

Thus, the code is capable of detecting $m^{r-2} + r - 2$ errors.

A lower bound on the number of check bits required for detecting a given number of errors is given below.

Theorem 4.2.3 *A systematic code capable of detecting*

$$t > \left\lfloor \frac{m^r - m^{\lfloor \frac{r}{2} \rfloor} - m^{\lceil \frac{r}{2} \rceil} + 2}{m-1} \right\rfloor + r,$$

requires at least $(r+1)$ *check digits.*

Proof: Let us prove the following upper bound for a m-ary t-UED with r check digits and k information digits.

$$t > \left\lfloor \frac{m^r - m^{\lfloor \frac{r}{2} \rfloor} - m^{\lceil \frac{r}{2} \rceil} + 2}{m-1} \right\rfloor + r.$$

For all $X = (x_{k-1}, x_{k-2}, \cdots, x_0) \in Z_m^k$, let $W(X) = \sum_{i=0}^{k-1} x_i$ be the weight of X, where the sum is over the integers. For example, if $m = 4$ and $k = 6$ then $W(013021) = 7$. Consider the set $\mathcal{C} = \{X_0, X_1, \cdots, X_K\} \subset Z_m^k$, $K = (m-1)k$, of information words, where for $i \geq 1$,

$$X_i = \underbrace{00\cdots0}_{k-e_i-1}\ d_i \underbrace{(m-1)(m-1)\cdots(m-1)}_{e_i} = 0^{k-e_i-1} d_i\, (m-1)^{e_i} \in Z_m^k$$

with $d_i = (i-1) \bmod (m-1)+1 \in Z_m - \{0\}$, and $e_i = \lfloor (i-1)/(m-1) \rfloor$, and for $i = 0$, $X_0 = 000\cdots0 \in Z_m^k$. For example, if $m = 4$ and $k = 6$ then $K = 3 \times 6 = 18$ and consider the following code words $\mathcal{C}$ as:

$$\mathcal{C} = \left\{ \begin{array}{l} X_0 = 000000, \\ X_1 = 000001, \\ X_2 = 000002, \\ X_3 = 000003, \\ X_4 = 000013, \\ X_5 = 000023, \\ X_6 = 000033, \\ X_7 = 000133, \\ X_8 = 000233, \\ X_9 = 000333, \\ X_{10} = 001333, \\ X_{11} = 002333, \\ X_{12} = 003333, \\ X_{13} = 013333, \\ X_{14} = 023333, \\ X_{15} = 033333, \\ X_{16} = 133333, \\ X_{17} = 233333, \\ X_{18} = 333333. \end{array} \right\}$$

Note that $\mathcal{C}$ is a maximal chain such that for all $i, j = 0, 1, 2, \cdots K$, $i < j \Leftrightarrow X_i \subset X_j$, and $W(X_i) = i$.

Note that there always exists $X_i, X_j \in \mathcal{C}$ such that $i < j$ and $C_i \subseteq C_j$. This is because we assume that the number of possible weights for an information word in Z_m^k is $K + 1 = (m-1)k + 1 > m^r$ (otherwise the m-ary AUED code design in [12] can be used).

Hence, let $X_i, X_j \in \mathcal{C}$ be two different sequences such that $j - i$ is the smallest value for which $i < j$ ($\Rightarrow X_i \subset X_j$) and $C_i \subseteq C_j$. Since $X_iC_i \subset X_jC_j$, from Theorem 3.1 we must have

$$t + 1 \geq D_H(X_jC_j, X_iC_i) = W_H(X_j - X_i) + W_H(C_j - C_i) \leq W_H(X_j - X_i) + r, \tag{4.1}$$

where $W_H(X)$ denotes the Hamming weight of a word $X \in Z_m^k$. The following relation holds

$$W_H(X_j - X_i) \leq \left\lceil \frac{j-i}{m-1} \right\rceil + 1, \ \forall X_i, X_j \in \mathcal{C}. \tag{4.2}$$

In fact, if $i, j \geq 1$ then

$$i - 1 = (m-1)\left\lfloor \frac{i-1}{m-1} \right\rfloor + [(i-1) \bmod (m-1)] = (m-1)e_i + (d_i - 1),$$

and

$$j - 1 = (m-1)\left\lfloor \frac{j-1}{m-1} \right\rfloor + [(j-1) \bmod (m-1)] = (m-1)e_j + (d_j - 1)$$

$$\Rightarrow j - i = (m-1)(e_j - e_i) + (d_j - d_i).$$

$$\Rightarrow \frac{j-i}{m-1} = (e_j - e_i) + \frac{d_j - d_i}{m-1}.$$

$$\Rightarrow \left\lceil \frac{j-i}{m-1} \right\rceil + 1 = (e_j - e_i) + \left\lceil \frac{d_j - d_i}{m-1} \right\rceil + 1 \geq (e_j - e_i) + 1.$$

The last inequality follows because $d_j, d_i \in Z_m - \{0\} \Rightarrow \lceil (d_j - d_i)/(m-1) \rceil \geq 0$.

From the appropriate construction of $\mathcal{C}$, it follows $(e_j - e_i) + 1 \geq W_H(X_j - X_i)$ and so (4.2) is valid for $i \geq 1$. If $i = 0$ it can be easily checked that

$$W_H(X_j) \leq \left\lceil \frac{j}{(m-1)} \right\rceil + 1,$$

and so (4.2) is valid for all $i \geq 0$. Hence, from (4.1) and (4.2)

$$t \leq \left\lceil \frac{j-i}{m-1} \right\rceil + r = \left\lceil \frac{(j-i-1)+1}{m-1} \right\rceil + r. \tag{4.3}$$

At this point, note that if l is an index such that $i < l < j$ then C_l is unordered with C_i; otherwise $j - i$ would not be the smallest value for which $i < j$ and $C_i \subseteq C_j$. For the same reason, if l and p are two indices such that $i < l < p < j$ then $C_l \neq C_p$. Hence,the quality $|\{i+1, i+2, \cdots, j-1\}| = j - i - 1$ can be upper bounded by the number, $u(C_i)$ of different sequences $C \in Z_m^r$ which are unordered with C_i. Hence from (4.3), we have

$$t \leq \left\lceil \frac{u(C_i)+1}{m-1} + r \right\rceil \leq \left\lceil \frac{max_{C \in Z_m^r}(u(C)+1)}{m-1} \right\rceil + r\,.$$

Let $\bar{C} = 0^{\lfloor \frac{r}{2} \rfloor}(m-1)^{\lceil \frac{r}{2} \rceil} \in {Z_m}^r$ be the word containing $\lfloor r/2 \rfloor$ 0's followed by $\lfloor r/2 \rfloor$ $(m-1)$'s. Now,

$$\max_{C \in Z_m^r} u(C) = u(\bar{C}) = \left(m^{\lfloor r/2 \rfloor} - 1\right)\left(m^{\lceil r/2 \rceil} - 1\right) = m^r - m^{\lfloor r/2 \rfloor} - m^{\lceil r/2 \rceil} + 1.$$

Hence, the theorem follows.∎

Note 1 *Borden in [7] considers a slightly different error model, when a symbol changes from a to b, $b > a$, the number of errors is $b-a$. For example, suppose the transmitted and received words are (3421) and (1211) respectively. Then, under this model, the number of asymmetric errors is $2+2+1=5$, and the proposed code can detect $2(m-1)$ errors using 2 check digits and $(m-1)(m^{r-2}+r-2)$ errors using $r \geq 3$ check bits.*

5. TYPE-I HYBRID ARQ AND ARQ WITH DIVERSITY COMBINING OVER THE M-ARY ASYMMETRIC CHANNEL, $M \geq 2$

5.1. Introduction

Forward Error Control (FEC) and Automatic-Repeat-Request (ARQ) are the two main techniques used in data transmission systems for controlling transmission errors. In FEC, the system uses error correcting codes. In this case, the data transmission is done in only one direction, i.e. from the transmitter to the receiver. If the receiver detects errors in the received word, it attempts to correct them. But, if the receiver fails to correct these errors, erroneous data will be delivered to the destination.

On the other hand, in ARQ, error detecting codes are used. If the receiver detects errors in the received word, it sends a negative acknowledgment (NAK) to the transmitter requesting it to resend the data. This process continues until the received word is correctly received. As mentioned in Chapter 1, stop-and-wait ARQ, go-back-N ARQ, and selective-repeat ARQ are the three main types of ARQ protocols.

FEC is widely used in communication systems where it is required getting the message correct in the first transmission. Although FEC schemes have bounded time delay equal to the processing time for encoding/decoding data, and have a constant throughput equal to the code rate regardless of the channel conditions, it is hard to achieve high system reliability. Also, the destination might receive the data incorrectly if the receiver fails to correct the errors. On the other hand, ARQ scheme is simple to implement and provides highly reliable data transmission; however, it also suffers from some drawbacks such as variable

delay time, it is harder to implement when the round-trip delay increases, etc. Thus, the throughput of the channel will rapidly decrease when the channel error rate increases.

To improve the performance of a system, a combination of both FEC and ARQ techniques can be used. These techniques are known as hybrid ARQ techniques. The hybrid ARQ uses error correcting and error detecting (say t error correcting and d ($d > t$) error detecting) codes. At the receiver, if the number of errors in the received word is less than or equal to t errors, then the errors are corrected and a positive acknowledgment (ACK) is sent to the transmitter requesting it to send the next word. However, if the number of errors is greater than the error correcting capability but within the error detecting capability of the code (i.e. more than t but less than or equal to d) errors, the receiver sends the transmitter NAK requesting it to resend the word. This is called a type-I hybrid ARQ protocol.

To reduce the number of retransmissions and hence improve the throughput of the system for the Z-channel, another technique called *diversity packet combining* can be used as explained in Chapter 1 [30]. In these schemes, the number of retransmissions needed to receive the correct word is decreased by saving the corrupted word and combining it later with the subsequent received word. This process continues until the combined word is successfully accepted or the maximum number of the retransmissions has been exhausted.

As explained in Chapter 1, for the binary case, the diversity combining is done by using a bit-by-bit logical OR operation. In the case of the m-ary asymmetric Z channel, instead of using bit-by-bit OR operation of the received and the previous stored bits, we can use a digit-by-digit MAX operation of the received and the previously stored word in our new scheme. This operation guarantees us

that the number of errors at each step is always less than or equal to the number of errors in the previous saved word.

In this chapter, we analyze the throughput of both ARQ and type-I hybrid ARQ protocols by deriving an expression for the number of transmission (or the expected number of retransmissions) needed to receive the correct code. Also, we introduce our new scheme for ARQ protocols with diversity combining and then analyze the throughput of the proposed scheme and compare it with the type-I hybrid ARQ protocols [50]. All the analyses are over a discrete memoryless m-ary asymmetric Z-channel, $m \geq 2$. We assume that the general model for the m-ary asymmetric Z-channel is as depicted in Figure 5.1. As shown in this model, for all $p_{i,j} \in [0,1]$, $i,j = 0,1,\cdots,m-1$, the transition probabilities of such a channel satisfy:

$$P(y|x) = \begin{cases} 0 & \text{if } x < y, \\ 1 & \text{if } x = y = 0, \\ 1 - \sum_{j=0}^{y-1} p_{x,j} & \text{if } x = y \in \mathbb{Z}_m - \{0\}, \\ p_{x,y} & \text{if } x > y, \end{cases} \qquad \text{for all } x, y \in \mathbb{Z}_m.$$

In all the schemes, it is assumed that selective-repeat-ARQ (SR-ARQ) is used. Hence, the throughput efficiency is given by

$$\eta = \frac{k}{n} \frac{1}{R(\mathcal{C})}, \tag{5.1}$$

where $\mathcal{C} \subseteq \mathbb{Z}_m^n$ is the code used, and $R(\mathcal{C})$ is the number of retransmissions needed to accept all codewords correctly. To use Equation (5.1) to analyze the throughput of the system, we have to find an expression for $R(\mathcal{C})$ (or derive an expression for the average number of retransmissions needed to accept all codewords correctly, $\mathbb{E}\left[R^{(t)}(C)\right]$).

This chapter is organized as follows. In Section 5.2, we analyze the throughput of ARQ protocols using:

(1) t-Asymmetric Error Detecting (t-AED) codes.

(2) All Asymmetric Error Detecting ($AAED$) codes.

over the m-ary Z-channel, $m \geq 2$. In Section 5.3, we analyze the throughput of type-I hybrid ARQ protocols using t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$AEC/AAED$) codes over the m-ary Z-channel, $m \geq 2$. In Section 5.4, we introduce our proposal for ARQ protocols with diversity combining and then do the same analysis as in Section 5.3 but with our diversity combining scheme.

5.2. Analysis of ARQ Protocols using t-AED and $AAED$ Codes over the m-ary Z-Channel, $m \geq 2$.

Let $X = (x_1, x_2, \cdots x_n) \in \mathcal{C} \subseteq \mathbb{Z}_m^n$ be the transmitted codeword over the general m-ary asymmetric Z-channel shown in Figure 5.1. Upon receiving the word Y, if the receiver detects an error in the received word, it discards the erroneous word and sends NAK to the sender requesting it to resend X. This process continues until the word is successfully accepted or the maximum retransmission number has been reached. This is the main idea of the ARQ protocols. In the following, we analyze the throughput of the ARQ protocols using t-AED and $AAED$ codes over the m-ary Z-channel, $m \geq 2$.

Some known t-AED codes are the the non-systematic *Borden*'s codes [7] or the systematic codes given in Chapter 4. Further, $AAED$ codes are the systematic codes given by *Bose* and *Pradhan* in [12] and the optimal non-systematic codes given by *de Bruijn* in [17].

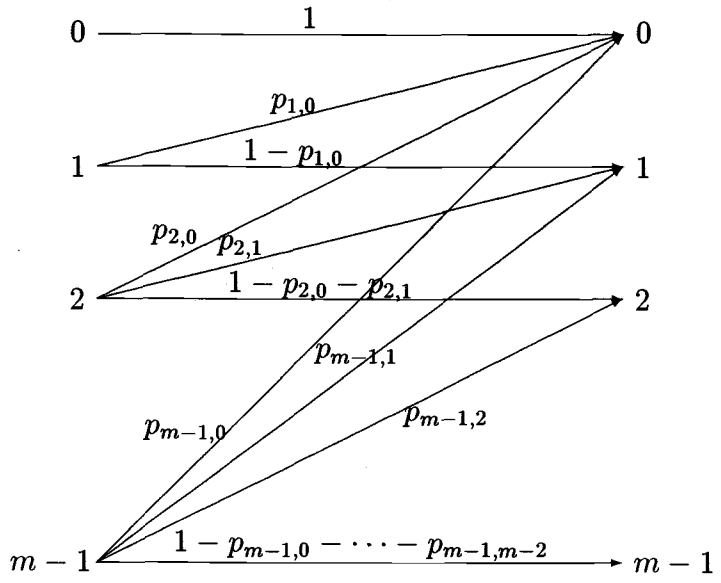

FIGURE 5.1. The general m-ary Z-asymmetric channel.

5.2.1. Analysis of ARQ Protocols using t-AED Codes over the m-ary Z-Channel, $m \geq 2$

Let $\mathcal{C}$ be a t-Asymmetric Error Detecting (t-AED) codes. Let $X = (x_1, x_2, \cdots x_n) \in \mathcal{C} \subseteq \mathbb{Z}_m^n$ be the transmitted codeword. The codeword will be accepted if there is no error in the received word, Y, or detected if the number of errors, e, is $\leq t$. On the other hand, if $e > t$, then the error may or may not be detected. Let $P_c(X)$ be the probability of receiving a correct codeword, $P_d(X)$ be the probability of detecting errors, and $P_u(X)$ be the probability of undetected errors where $P_c(X) + P_d(X) + P_u(X) = 1$. In this case, the number of retransmissions needed to receive the word X correctly is:

$$R^{(t)}(X) = 1\times(1-P_d(X))+2\times P_d(X)\times(1-P_d(X))+3\times P_d(X)^2\times(1-P_d(X))+\cdots$$

$$= \frac{1-P_d(X)}{(1-P_d(X))^2} = \frac{1}{1-P_d(X)} = \frac{1}{P_u(X)+P_c(X)}. \tag{5.2}$$

The complete analysis over Z channel is given in [39].

5.2.2. Analysis of ARQ Protocols using *AAED* Codes over the m-ary Z-Channel, $m \geq 2$

Assume that $\mathcal{C}$ is All Asymmetric Error Detecting (*AAED*) codes. Let $X = (x_1, x_2, \cdots x_n) \in \mathcal{C} \subseteq \mathbb{Z}_m^n$ be the transmitted codeword. In this case, either the received word is accepted with probability $P_c(X)$ or the error will be detected with probability $P_d(X)$ where $P_c(X) + P_d(X) = 1$. In this case, the number of retransmissions, $R(X)$, needed to receive X correctly is

$$\begin{aligned} R(X) &= 1 \times P_c(X) + 2 \times (1 - P_c(X))P_c(X) + 3 \times (1 - P_c(X))^2 P_c(X) + \cdots \\ &= \sum_{t=1}^{\infty} t\,(1 - P_c(X))^{t-1} P_c(X) \\ &= P_c(X) \sum_{t=1}^{\infty} t(1 - P_c(X))^{t-1} \\ &= \frac{P_c(X)}{[1 - (1 - P_c(X))]^2} = \frac{1}{P_c(X)}. \end{aligned}$$

In the rest of this section, we assume that the codewords are transmitted over the m-ary Z-Channel, $m \geq 2$, with error model as shown in Figure 5.2. This model is the same one that we used in Chapter 4. As shown in this model, for $p \in [0, 1]$, the transition probabilities of such a channel satisfy:

$$P(y|x) = \begin{cases} 0 & \text{if } x < y, \\ 1 & \text{if } x = y = 0, \\ 1 - x\,p & \text{if } x = y \in \mathbb{Z}_m - \{0\}, \\ p & \text{if } x > y, \end{cases} \qquad \text{for all } x, y \in \mathbb{Z}_m.$$

Further, given $X \in \mathbb{Z}_m^n$, let

$$\mathbf{w}(X) = (w_0(X), w_1(X), \ldots, w_{m-1}(X))$$

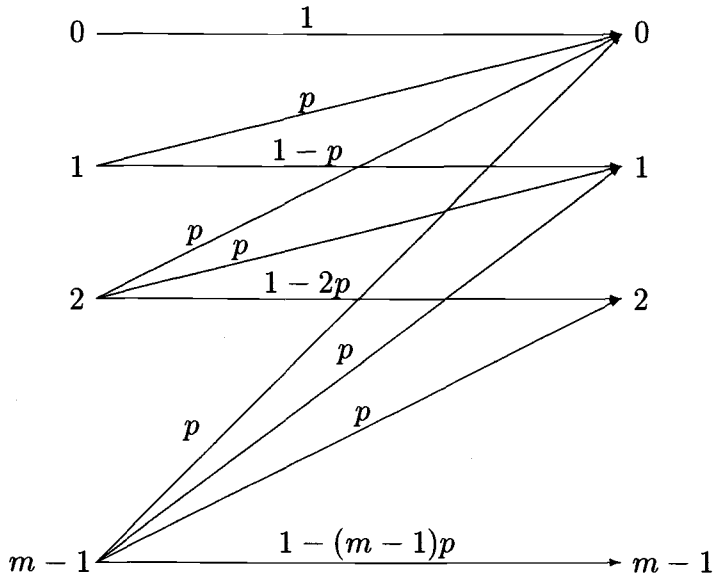

FIGURE 5.2. m-ary asymmetric Z-channel where every error is equally likely.

be the weight of X, where

$$w_a(X) = |\{j = 1, 2, \ldots, n : \ x_j = a\}|$$

indicates the number of occurrences of the symbol $a \in \mathbb{Z}_m$ in the word X. For example, if $m = 5$ then

$$\mathbf{w}(X = 1302043014) = (3, 2, 1, 2, 2).$$

Now, we want to find $P_c(X)$. Since the channel is a discrete memoryless channel (DMC) we have

$$\begin{aligned} P_c(X) &= P(X \text{ is received correctly}) \\ &= [1]^{w_0(X)} \ [1-p]^{w_1(X)} \ [1-2p]^{w_2(X)} \ \cdots \ [1-(m-1)p]^{w_{m-1}(X)} \\ &= \prod_{i=1}^{m-1} [1 - i \ p]^{w_i(X)} . \end{aligned}$$

Also, we have

$$P(X \text{ is received in error }) = 1 - P_c(X) = 1 - P(X \text{ is received correctly})$$
$$= 1 - P_c(X) = 1 - \prod_{i=1}^{m-1} [1 - i\,p]^{w_i(X)}\,.$$

If we assume that each symbol is equally likely in a codeword, then $\mathbb{E}[R(X)]$ can be calculated as follows. Define $R(X)$ as the expected number of retransmissions needed to receive X correctly. Define $R_i(X) = R(X)$ as the expected number of retransmissions needed to receive x_i correctly where $1 \leq i \leq n$. Thus, we have $R = R(X) = \max_{1 \leq i \leq n} R_i$. Let $P(R_i = j)$ be the probability that the expected number of retransmissions needed to receive x_i correctly is equal to j where $1 \leq i \leq n$ and $1 \leq j < \infty$. Thus,

$$P(R_1 = 1) = \sum_{x=0}^{m-1} P(R_1 = 1 | x_1 = x)\, P(x_1 = x)$$
$$= \frac{1}{m} \sum_{x=0}^{m-1} (1 - x\,p)\,.$$
$$P(R_1 = 2) = \sum_{x=0}^{m-1} P(R_1 = 2 | x_1 = x)\, P(x_1 = x)$$
$$= \frac{1}{m} \sum_{x=0}^{m-1} P(R_1 = 2 | x_1 = x)$$
$$= \frac{1}{m} \sum_{x=0}^{m-1} (x\,p)\,[1 - x\,p]\,.$$

In general,

$$P(R_1 = \tau) = \frac{1}{m} \sum_{x=0}^{m-1} (x\,p)^{\tau-1}\,(1 - x\,p)$$
$$= \frac{1}{m} \sum_{x=0}^{m-1} (x\,p)^{\tau-1} - \sum_{x=0}^{m-1} (x\,p)^{\tau}\,.$$

Special Case *($m = 2$)*:

$$P(R_i = \tau) = \begin{cases} \frac{1-p}{2}\, p^{\tau-1} & \text{if } \tau > 1, \\ \frac{1-p}{2} + \frac{1}{2} & \text{if } \tau = 1. \end{cases}$$

In this case,

$$R = \max\{R_1, R_2, \cdots, R_m\},$$

and

$$\mathbb{E}[R(X)] = \sum_{\tau=1}^{\infty} \tau \, P(R = \tau) \, .$$

From the above expression, to find $\mathbb{E}[R(X)]$, we have to find an expression for $P(R = \tau)$ where

$$P(R = \tau) = P(R \leq \tau) - P(R \leq \tau - 1) \, .$$

From the definition of R, we have

$$P(R \leq \tau) = P(R_1 \leq \tau \text{ and } R_2 \leq \tau \text{ and } \cdots \text{ and } R_n \leq \tau) = \prod_{i=1}^{n} P(R_i \leq \tau),$$

and

$$P(R_i \leq \tau) = \sum_{a=1}^{\tau} P(R_i = a) = \sum_{a=1}^{\tau} \frac{(1-p)}{2} \, p^{a-1} + \frac{1}{2}$$

$$= \frac{1}{2} + \frac{(1-p)}{2} \sum_{a=1}^{\tau} p^{a-1} = \frac{1}{2} + \frac{(1-p)}{2} \, \frac{1-p^{\tau}}{(1-p)}$$

$$= \frac{1}{2} + \frac{1-p^{\tau}}{2} = 1 - \frac{p^{\tau}}{2} \, .$$

$$\Rightarrow P(R \leq \tau) = \prod_{i=1}^{n} P(R_i \leq \tau) = \prod_{i=1}^{n} \left[1 - \frac{p^{\tau}}{2}\right]$$

$$= \frac{(2-p^{\tau})^n}{2^n} = \left[1 - \frac{p^{\tau}}{2}\right]^n \, .$$

$$\Rightarrow P(R = \tau) = P(R \leq \tau) - P(R \leq \tau - 1)$$

$$= \left[1 - \frac{p^{\tau}}{2}\right]^n - \left[1 - \frac{p^{\tau-1}}{2}\right]^n \, .$$

$$\Rightarrow \mathbb{E}\,[R(X)] = \sum_{\tau=1}^{\infty} \tau \left[1 - \frac{p^{\tau}}{2}\right]^{n} - \sum_{\tau=1}^{\infty} \tau \left[1 - \frac{p^{\tau-1}}{2}\right]^{n} = S_1 - S_2\,. \tag{5.3}$$

Now, we will find expressions for S_1 and S_2.

$$S_1 = \sum_{\tau=1}^{\infty} \tau (1 - \frac{p^{\tau}}{2})^{n} = (-1)^{n} \sum_{\tau=1}^{\infty} \tau \left[\frac{p^{\tau}}{2} - 1\right]^{n}$$

$$= (-1)^{n} \sum_{\tau=1}^{\infty} \tau \left\{(-1)^{n} + \sum_{k=1}^{n} \binom{n}{k} \left[\frac{p^{\tau}}{2}\right]^{k} (-1)^{n-k}\right\}$$

$$= (-1)^{n} \sum_{\tau=1}^{\infty} \tau (-1)^{n} + (-1)^{n} \sum_{\tau=1}^{\infty} \tau \left\{\sum_{k=1}^{n} \binom{n}{k} \left[\frac{p^{\tau}}{2}\right]^{k} (-1)^{n-k}\right\}$$

$$= \sum_{\tau=1}^{\infty} \tau + (-1)^{n} \sum_{k=1}^{n} \left\{\binom{n}{k} (-1)^{n-k} \sum_{\tau=1}^{\infty} \tau \left[\frac{p^{\tau}}{2}\right]^{k}\right\}$$

$$= \sum_{\tau=1}^{\infty} \tau + (-1)^{n} \sum_{k=1}^{n} \left\{\binom{n}{k} (-1)^{n-k} \frac{p^{k}}{2^{k}(1-p^{k})^{2}}\right\}.$$

On the other hand,

$$S_2 = \sum_{\tau=1}^{\infty} \tau (1 - \frac{p^{\tau-1}}{2})^{n} = (-1)^{n} \sum_{\tau=1}^{\infty} \tau \left(\frac{p^{\tau} - 1}{2} - 1\right)^{n}$$

$$= (-1)^{n} \sum_{\tau=1}^{\infty} \tau \left\{(-1)^{n} + \sum_{k=1}^{n} \binom{n}{k} \left[\frac{p^{\tau} - 1}{2}\right]^{k} (-1)^{n-k}\right\}$$

$$= (-1)^{n} \sum_{\tau=1}^{\infty} \tau (-1)^{n} + (-1)^{n} \sum_{\tau=1}^{\infty} \tau \left[\sum_{k=1}^{n} \left(\binom{n}{k} \left(\frac{p^{\tau-1}}{2}\right)^{k} (-1)^{n-k}\right)\right]$$

$$= \sum_{\tau=1}^{\infty} \tau + (-1)^{n} \sum_{k=1}^{n} \left[\binom{n}{k} (-1)^{n-k} \sum_{\tau=1}^{\infty} \tau \left(\frac{p^{\tau-1}}{2}\right)^{k}\right]$$

$$= \sum_{\tau=1}^{\infty} \tau + (-1)^{n} \sum_{k=1}^{n} \binom{n}{k} (-1)^{n-k} \frac{1}{p^{k}} \sum_{\tau=1}^{\infty} \tau \left[\frac{p^{\tau}}{2}\right]^{k}$$

$$= \sum_{\tau=1}^{\infty} \tau + (-1)^n \sum_{k=1}^{n} \binom{n}{k} (-1)^{n-k} \left[\frac{1}{2^k (1-p^k)^2} \right] .$$

Substituting the expression of S_1 and S_2 in Equation (5.3), it follows:

$$\begin{aligned}
\mathbb{E}[R(X)] &= \frac{1}{2^n} + \sum_{\tau=1}^{\infty} \tau + (-1)^n \sum_{k=1}^{n} \binom{n}{k} (-1)^{n-k} \left[\frac{p^k}{2^k (1-p^k)^2} \right] \\
&\quad - \sum_{\tau=1}^{\infty} \tau - (-1)^n \sum_{k=1}^{n} \binom{n}{k} (-1)^{n-k} \left[\frac{1}{2^k (1-p^k)^2} \right] \\
&= \frac{1}{2^n} + (-1)^n \sum_{k=1}^{n} \binom{n}{k} (-1)^{n-k} \left[\frac{p^k - 1}{2^k (1-p^k)^2} \right] \\
&= \frac{1}{2^n} + (-1)^n \sum_{k=1}^{n} \binom{n}{k} (-1)^{n-k+1} \frac{1}{2^k (1-p^k)} \\
&= \frac{1}{2^n} - \sum_{k=1}^{n} \binom{n}{k} \frac{(-1)^k}{2^k (1-p^k)} .
\end{aligned}$$

Assume that the codewords are equally likely transmitted, the average number of retransmissions for the code $\mathcal{C}$ is:

$$\begin{aligned}
\mathbb{E}[R(\mathcal{C})] &= \frac{1}{|\mathcal{C}|} \sum_{X \in \mathcal{C}} \mathbb{E}[R(X)] \\
&= \frac{1}{|\mathcal{C}|} \sum_{X \in \mathcal{C}} \left[\frac{1}{2^n} - \sum_{k=1}^{n} \binom{n}{k} \frac{(-1)^k}{2^k (1-p^k)} \right] .
\end{aligned}$$

5.3. Analysis of Type-I Hybrid ARQ Protocols using *t-AEC/AAED* Codes over the *m*-ary *Z*-Channel, $m \geq 2$.

Assume that the type-I hybrid ARQ communication system shown in Figure 5.3 is used. Also, assume that the used code, $\mathcal{C}$, is a t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$AEC/AAED$) codes.

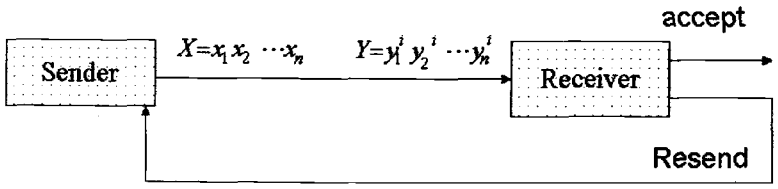

FIGURE 5.3. ARQ system.

After transmitting the codeword $X \in \mathcal{C} \subseteq \mathbb{Z}_m^n$, if the receiver declares no error in the received word Y, then the receiver sends the transmitter ACK requesting it to transmit the next codeword. If the receiver detects an error in Y and the number of errors, e, is within the error correcting capability of the designed code, i.e $e \leq t$, then the errors will be corrected and the receiver sends the transmitter ACK requesting it to send the next codeword. On the other hand, if $e > t$, then the receiver discards Y, and sends NAK to the transmitter requesting it to resend X. This process is continued until the received word is successfully accepted. In this section, we compute the average number of retransmissions needed to receive a given codeword X correctly. The transmission and retransmission procedure for the type-I hybrid ARQ scheme is given in Figure 5.4.

Let $P_a(X) \in [0,1]$ and $P_r(X) \in [0,1]$ be respectively the probabilities of accepting and rejecting the received word Y, where $P_a(X) + P_r(X) = 1$. Also, let $R^{(t)}(X) : \Omega \longrightarrow \mathbb{N} - \{0\}$ be the random variable:

$$R^{(t)}(X) \stackrel{\text{def}}{=} \text{number of retransmission needed by the receiver to accept the received word } Y.$$

In this simple case, the average number of retransmissions is

$$\mathbb{E}\left[R^{(t)}(X)\right] = \sum_{i=1}^{\infty} i[P_r(X)]^{i-1} P_a(X) = P_a(X) \sum_{i=1}^{\infty} i[P_r(X)]^{i-1}$$
$$= \frac{P_a(X)}{[1 - P_r(X)]^2} = \frac{1}{P_a(X)}. \tag{5.4}$$

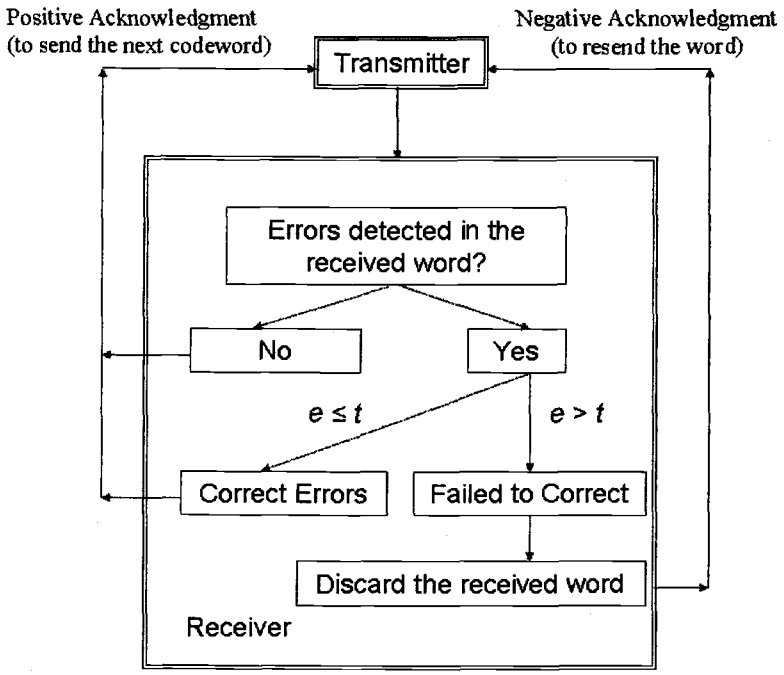

FIGURE 5.4. The proposed transmission and retransmission procedure for type-I hybrid ARQ scheme.

This is because,

$$f(x) \stackrel{\text{def}}{=} \sum_{i=1}^{\infty} x^i = \frac{1}{1-x} - 1 \quad \Longrightarrow \quad f'(x) = \sum_{i=1}^{\infty} i x^{i-1} = \frac{1}{(1-x)^2}.$$

Hence, in order to calculate the average number of retransmissions, we need to find an expression for $P_a(X)$. Since a t-$AEC/AAED$ codes is used, a transmitted word X is accepted iff the number of errors occurred during the transmission is less than or equal to t. Hence,

$$P_a(X) = \sum_{\tau=0}^{t} P(Y \ contains \ exactly \ \tau \ errors | X). \tag{5.5}$$

Also, in this case, we assume that the channel of the system shown in Figure 5.3 is a DMC. Let

$$\delta(x) \stackrel{\text{def}}{=} P(y \neq x|x) = \sum_{a \in \mathbb{Z}_m - \{x\}} P(a|x), \quad \text{for all } x \in \mathbb{Z}_m, \tag{5.6}$$

be the probability that the symbol x is received in error, and hence

$$1 - \delta(x) = P(y = x|x) = P(x|x), \quad \text{for all } x \in \mathbb{Z}_m,$$

is the probability that the symbol x is received correctly. Since the channel is DMC we have

$$P(y_1 y_2 \dots y_n | x_1 x_2 \dots x_n) = \prod_{i=1}^{n} P(y_i|x_i), \quad \text{for all } X, Y \in \mathbb{Z}_m^n.$$

Note that, the above formula implies that for all $X, Y \in \mathbb{Z}_m^n$,

$$P(\ y_1 y_2 \dots y_{i-1} = x_1 x_2 \dots x_{i-1}\ ,\ y_i \neq x_i \text{ and } y_{i+1} y_{i+2} \dots y_n = x_{i+1} x_{i+2} \dots x_n\ |X)$$

$$= P(y_1 \dots y_{i-1} = x_1 \dots x_{i-1} | x_1 \dots x_{i-1}) P(y_i \neq x_i | x_i)$$

$$\times\ P(y_{i+1} \dots y_n = x_{i+1} \dots x_n | x_{i+1} \dots x_n).$$

Further, given $X \in \mathbb{Z}_m^n$, let

$$\mathbf{w}(X) = (w_0(X), w_1(X), \dots, w_{m-1}(X))$$

be the weight of X, where

$$w_a(X) = |\{j = 1, 2, \dots, n :\ x_j = a\}|$$

indicates the number of occurrences of the symbol $a \in \mathbb{Z}_m$ in the word X. For example, if $m = 5$ then

$$\mathbf{w}(X = 1302043014) = (3, 2, 1, 2, 2).$$

Let us find an expression for $P(Y \text{ contains exactly } \tau \text{ errors}|X)$. Note that, since the channel is a DMC, we have

$P(Y \text{ contains exactly } \tau \text{ errors}|X)$

$$= \sum_{\substack{\tau_0,\tau_1,\ldots,\tau_{m-1}: \\ \tau_0+\tau_1+\ldots+\tau_{m-1}=\tau}} \prod_{a=0}^{m-1} P(Y \text{ contains exactly } \tau_a \text{ errors in the } a\text{'s of } X|X),$$

where the sum is over the $\binom{\tau+m-1}{m-1}$ different compositions of the number τ. Again, since the channel is a DMC, for all $a \in \mathbb{Z}_m$,

$$P(Y \text{ contains exactly } \tau_a \text{ errors in the } a\text{'s of } X|X)$$

$$= \binom{w_a(X)}{\tau_a} [\delta(a)]^{\tau_a} [1-\delta(a)]^{w_a(X)-\tau_a}$$

$$= \binom{w_a(X)}{\tau_a} [1-\delta(a)]^{w_a(X)} \left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a}.$$

Hence,

$$\prod_{a=0}^{m-1} P(Y \text{ contains exactly } \tau_a \text{ errors in the } a\text{'s of } X|X)$$

$$= \prod_{a=0}^{m-1} \binom{w_a(X)}{\tau_a} [1-\delta(a)]^{w_a(X)} \left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a}$$

$$= \prod_{a=0}^{m-1} [1-\delta(a)]^{w_a(X)} \prod_{a=0}^{m-1} \binom{w_a(X)}{\tau_a} \left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a},$$

and

$$P(Y \text{ contains exactly } \tau \text{ errors}|X)$$

$$= \prod_{a=0}^{m-1} [1-\delta(a)]^{w_a(X)} \sum_{\substack{\tau_0,\tau_1,\ldots,\tau_{m-1}: \\ \tau_0+\tau_1+\ldots+\tau_{m-1}=\tau}} \prod_{a=0}^{m-1} \binom{w_a(X)}{\tau_a} \left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a}.$$

So, from the above relation and (5.5), if X is a transmitted word of weight

$$\mathbf{w}(X) = (w_0(X), w_1(X), \ldots, w_{m-1}(X)),$$

then

$$P_a(X) = \prod_{a=0}^{m-1} [1-\delta(a)]^{w_a(X)} \left\{ \sum_{\tau=0}^{t} \sum_{\substack{\tau_0,\tau_1,\ldots,\tau_{m-1}: \\ \tau_0+\tau_1+\ldots+\tau_{m-1}=\tau}} \prod_{a=0}^{m-1} \binom{w_a(X)}{\tau_a} \left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a} \right\} \tag{5.7}$$

Relation (5.7) holds true for any t-Asymmetric Error Correcting/All Asymmetric Error Detecting ARQ system used with any discrete memoryless m-ary asymmetric Z-channel.

Note 2 If $\delta(0) = 0$ as in the case of the m-ary Z-channel ($\delta(0) = 0 \Leftrightarrow 0$ is always received error-free) then the symbol 0 "disappears" from relation (5.7). And so, for any m-ary Z-channel,

$$P_a(X) = \prod_{a=1}^{m-1} [1-\delta(a)]^{w_a(X)} \left\{ \sum_{\tau=0}^{t} \sum_{\substack{\tau_1,\tau_2,\ldots,\tau_{m-1}: \\ \tau_1+\tau_2+\ldots+\tau_{m-1}=\tau}} \prod_{a=1}^{m-1} \binom{w_a(X)}{\tau_a} \left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a} \right\}. \tag{5.8}$$

Similarly, the same thing holds true if $a_1, a_2, \ldots, a_l \in \mathbb{Z}_m$ are received error-free. In the following we assume $\delta(0) = 0$.

Note 3 The function

$$f(x) \stackrel{\text{def}}{=} \frac{x}{1-x} = x(1+x+x^2+\ldots) = x\sum_{i=0}^{\infty} x^i,$$

is increasing with $x \in [0,1)$ and such that $f(0) = 0$ and $f(1) = +\infty$. Now, if $c_1, c_2 \in \mathbb{R}^+$ are two positive constants such that

$$\text{for all } a \in \mathbb{Z}_m, \qquad \delta(a) \neq 0 \quad \Longrightarrow \quad c_1 \leq \delta(a) \leq c_2,$$

then from (5.8),

$$P_a(X) \leq \prod_{a=1}^{m-1}(1-\delta(a))^{w_a(X)} \left\{ \sum_{\tau=0}^{t} \sum_{\substack{\tau_1,\tau_2,\ldots,\tau_{m-1}: \\ \tau_1+\tau_2+\ldots+\tau_{m-1}=\tau}} \prod_{a=1}^{m-1} \binom{w_a(X)}{\tau_a} \left(\frac{c_2}{1-c_2}\right)^{\tau_a} \right\}$$

$$= \prod_{a=1}^{m-1}(1-\delta(a))^{w_a(X)} \left\{ \sum_{\tau=0}^{t} \left(\frac{c_2}{1-c_2}\right)^{\tau_a} \sum_{\substack{\tau_1,\tau_2,\ldots,\tau_{m-1}: \\ \tau_1+\tau_2+\ldots+\tau_{m-1}=\tau}} \prod_{a=1}^{m-1} \binom{w_a(X)}{\tau_a} \right\}$$

$$= \prod_{a=1}^{m-1}(1-\delta(a))^{w_a(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{c_2}{1-c_2}\right)^{\tau_a},$$

where $w(X) = w_H(X) = w_1(X) + w_2(X) + \ldots + w_{m-1}(X)$ is the Hamming weight of X. The above relations follows because obviously

$$\binom{w(X)}{\tau} = \binom{w_1(X) + w_2(X) + \ldots + w_{m-1}(X)}{\tau_1 + \tau_2 + \ldots + \tau_{m-1}}$$

$$= \sum_{\substack{\tau_1,\tau_2,\ldots,\tau_{m-1}: \\ \tau_1+\tau_2+\ldots+\tau_{m-1}=\tau}} \prod_{a=1}^{m-1} \binom{w_a(X)}{\tau_a}.$$

The analogous lower bound can be obtained similarly, and so

$$(1-c_2)^{w(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{c_1}{1-c_1}\right)^{\tau} \leq \prod_{a=1}^{m-1}(1-\delta(a))^{w_a(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{c_1}{1-c_1}\right)^{\tau}$$

$$\leq P_a(X) \prod_{a=1}^{m-1}(1-\delta(a))^{w_a(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{c_2}{1-c_2}\right)^{\tau}$$

$$\leq (1-c_1)^{w(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{c_2}{1-c_2}\right)^{\tau}. \tag{5.9}$$

Further, should $c_1 = \delta(a) = c_2 = \delta$, for all $a \in \mathbb{Z}_m - \{0\}$ then

$$P_a(X) = (1-\delta)^{w(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{\delta}{1-\delta}\right)^{\tau}. \tag{5.10}$$

The above relation can be also written as

$$P_a(X) = \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \delta^{\tau}(1-\delta)^{w(X)-\tau}. \tag{5.11}$$

In order to evaluate the average number of retransmissions, it might be useful to approximate $P_a(X)$ as follows. Let

$$f(\delta, w, t) \stackrel{\text{def}}{=} \sum_{\tau=0}^{t} \binom{w}{\tau} \delta^{\tau}(1-\delta)^{w-\tau}, \tag{5.12}$$

and

$$f_1(\delta, w, t) \stackrel{\text{def}}{=} \sum_{\tau=t+1}^{w} \binom{w}{\tau} \delta^{\tau}(1-\delta)^{w-\tau} \tag{5.13}$$

$$= \binom{w}{t+1} \delta^{t+1}(1-\delta)^{w-t-1} + \binom{w}{t+2} \delta^{t+2}(1-\delta)^{w-t-2} + \cdots + \binom{w}{w} \delta^{w}.$$

Note that, from the definitions of $f(\delta, w, t)$ and $f_1(\delta, w, t)$, $P_a(X) = f(\delta, w, t) = 1 - f_1(\delta, w, t)$ with $w = w(X)$. It is clear that

$$\binom{w}{t+1} \delta^{t+1}(1-\delta)^{w-(t-1)} \leq f_1(\delta, w, t).$$

On the other hand,

$$f_1(\delta, w, t) = \delta^{t+1} \left[\frac{\binom{w}{t+1}}{\binom{w-t-1}{0}} \right] \binom{w-t-1}{0} (1-\delta)^{w-t-1}$$

$$+ \delta^{t+1} \left[\frac{\binom{w}{t+2}}{\binom{w-t-1}{1}} \right] \binom{w-(t+1)}{1} \delta^{1}(1-\delta)^{w-t-2}$$

$$+ \cdots + \delta^{t+1} \left[\frac{\binom{w}{w}}{\binom{w-t-1}{w-t-1}} \right] \binom{w-t-1}{w-(t+1)} \delta^{w-t-1}$$

$$= \delta^{t+1} \sum_{\tau=0}^{w-t-1} \left[\frac{\binom{w}{t+1+\tau}}{\binom{w-t-1}{\tau}} \right] \binom{w-t-1}{\tau} \delta^{\tau}(1-\delta)^{w-(t-1-\tau)}.$$

But

$$\frac{\binom{w}{t+1+\tau}}{\binom{w-t-1}{\tau}} = \frac{\binom{w}{t+1}}{\binom{t+1+\tau}{t+1}},$$

and so,

$$f_1(\delta,w,t)=\delta^{t+1}\sum_{\tau=0}^{w-t-1}\left[\frac{\binom{w}{t+1}}{\binom{w+1+\tau}{t+1}}\right]\binom{w-t-1}{\tau}\delta^{\tau}(1-\delta)^{w-t-1-\tau}$$

$$=\binom{w}{t+1}\delta^{t+1}\sum_{\tau=0}^{w-t-1}\left[\frac{\binom{w-t-1}{\tau}}{\binom{t+1+\tau}{t+1}}\right]\delta^{\tau}(1-\delta)^{w-t-1-\tau}$$

$$\leq\binom{w}{t+1}\delta^{t+1}\sum_{\tau=0}^{w-t-1}\binom{w-t-1}{\tau}\delta^{\tau}(1-\delta)^{w-t-1-\tau}$$

$$=\binom{w}{t+1}\delta^{t+1}[\delta+(1-\delta)]^{w-t-1}=\binom{w}{t+1}\delta^{t+1}.$$

Hence,

$$\binom{w}{t+1}\delta^{t+1}(1-\delta)^{w-(t+1)}\leq f_1(\delta,w,t)=1-f(\delta,w,t)\leq\binom{w}{t+1}\delta^{t+1}. \quad (5.14)$$

From (5.14) and (5.4), it follows:

$$\frac{1}{1-\binom{w}{t+1}\delta^{t+1}(1-\delta)^{w-(t+1)}}\leq\mathbb{E}\left[R^{(t)}(X)\right]=\frac{1}{1-f_1(\delta,w,t)}\leq\frac{1}{1-\binom{w}{t+1}\delta^{t+1}}, \quad (5.15)$$

and if $(1-\delta)^{w-(t+1)}\approx 1$ then

$$\mathbb{E}\left[R^{(t)}(X)\right]=\frac{1}{1-\binom{w}{t+1}\delta^{t+1}}.$$

In the **general case**, relation (5.4) and (5.8) imply that the average number of retransmissions of a transmitted codeword depends only on its vector weight, $w=(w_0,w_1,\ldots,w_{m-1})$. Hence,

$$\mathbb{E}\left[R^{(t)}(X)\right]=\overline{R}^{(t)}_{Hyb}(w)$$

$$=1/\left\{\prod_{a=1}^{m-1}[1-\delta(a)]^{w_a}\left\{\sum_{\tau=0}^{t}\sum_{\substack{\tau_1,\tau_2,\ldots,\tau_{m-1}:\\ \tau_1+\tau_2+\ldots+\tau_{m-1}=\tau}}\prod_{a=1}^{m-1}\binom{w_a}{\tau_a}\left[\frac{\delta(a)}{1-\delta(a)}\right]^{\tau_a}\right\}\right\},$$

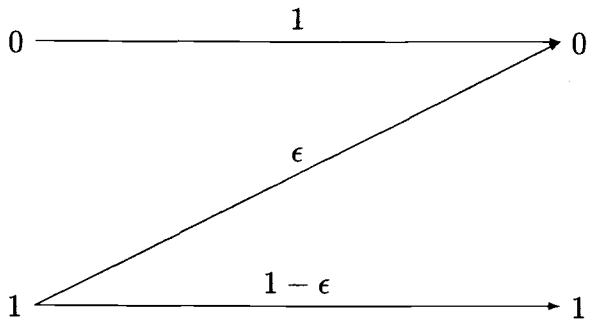

FIGURE 5.5. The binary Z-channel.

where $\delta(a)$ is the probability that symbol $a \in \mathbb{Z}_m - \{0\}$ is received in error. Now, the average number of retransmissions for the given code, $\mathcal{C}$, of length n can be obtained by taking the average of $\mathbb{E}\left[R^{(t)}(X)\right]$ over all codewords. Assume that the codewords are equally likely transmitted, the average number of retransmissions for the code $\mathcal{C}$ is

$$\mathbb{E}\left[R^{(t)}(\mathcal{C})\right] = \frac{1}{|\mathcal{C}|}\sum_{X\in\mathcal{C}} \mathbb{E}\left[R^{(t)}(X)\right] = \frac{1}{|\mathcal{C}|}\sum_{w\in\mathbb{Z}_n^m} A_w \overline{R}_{Hyb}^{(t)}(w),$$

where $A_w = |\{X \in \mathcal{C} : \ w(X) = w\}|$ defines the weight distribution of the code.

In the rest of this section, we calculate the expected number of the retransmissions, $\mathbb{E}\left[R^{(t)}(X)\right]$, for different m-ary Z-channels, $m \geq 2$.

5.3.1. Analysis of the throughput of Type-I Hybrid ARQ protocol over the binary asymmetric channel (Z-channel)

In this case, the channel model is shown in Figure 5.5. From (5.6), $\delta(0) = 0$ and $\delta(1) = \epsilon \in [0, 1]$. Hence from (5.8),

$$P_a(X) = (1-\epsilon)^{w(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{\epsilon}{1-\epsilon}\right)^{\tau} = \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \epsilon^{\tau}(1-\epsilon)^{w(X)-\tau},$$

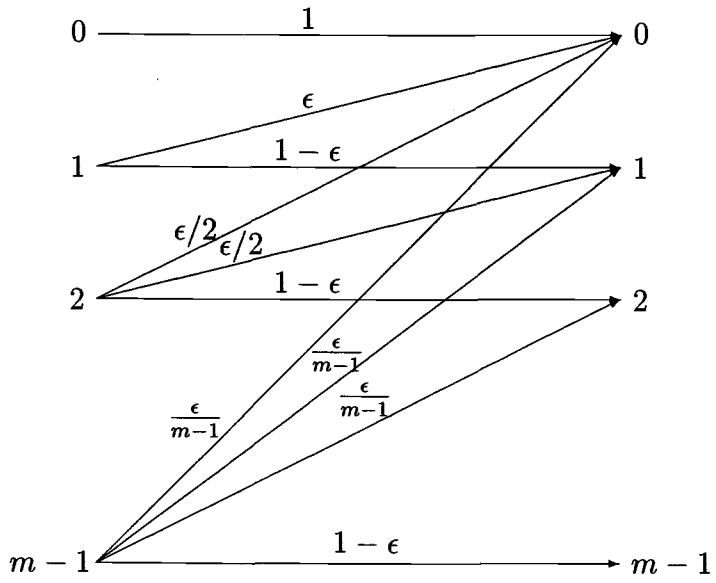

FIGURE 5.6. The m-ary Z-channel where every symbol error is equally likely.

and from (5.4)

$$\mathbb{E}\left[R^{(t)}(X)\right] = \frac{1}{\sum_{\tau=0}^{t}\binom{w(X)}{\tau}\epsilon^{\tau}(1-\epsilon)^{w(X)-\tau}}.$$

Example 5 *If* $\epsilon = 0.01$, $t = 2$ *and* $w = w(X) = 100$, *then* $\mathbb{E}\left[R^{(2)}(X)\right] = 1.086216480$. *When we use the bound in (5.15),* $1.064961780 \leq \mathbb{E}\left[R^{(2)}(X)\right] = 1.086216480 \leq 1.192890373$. *On the other hand, if* $\epsilon = 0.001$, *then* $1.000146766 \leq \mathbb{E}\left[R^{(2)}(X)\right] = 1.000150399 \leq 1.000161726$.

5.3.2. Analysis of the throughput of Type-I Hybrid ARQ protocol over the m-ary asymmetric Z-channel where every symbol error is equally likely

Here the channel model is as depicted in Figure 5.6. For all $\epsilon \in [0,1]$ the transition probabilities of such a channel satisfy:

$$P(y|x) = \begin{cases} 0 & \text{if } x < y, \\ 1 & \text{if } x = y = 0, \\ 1-\epsilon & \text{if } x = y \in \mathbb{Z}_m - \{0\}, \\ \epsilon/x & \text{if } x > y, \end{cases} \qquad \text{for all } x, y \in \mathbb{Z}_m.$$

So, in this case, from (5.6), $\delta(x) = \epsilon$ for all $x \in \mathbb{Z}_m - \{0\}$ and $\delta(0) = 0$. Again, from (5.11),

$$P_a(X) = (1-\epsilon)^{w(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{\epsilon}{1-\epsilon}\right)^{\tau} = \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \epsilon^{\tau} (1-\epsilon)^{w(X)-\tau},$$

and from (5.4)

$$\mathbb{E}\left[R^{(t)}(X)\right] = \frac{1}{\sum_{\tau=0}^{t} \binom{w(X)}{\tau} \epsilon^{\tau} (1-\epsilon)^{w(X)-\tau}}.$$

This is similar to the previous binary case.

Hence,

$$\mathbb{E}\left[R^{(t)}(\mathcal{C})\right] = \frac{1}{|\mathcal{C}|} \sum_{w=0}^{n} A_w \overline{R}_{Hyb}^{(t)}(w) = \frac{1}{|\mathcal{C}|} \sum_{w=0}^{n} \frac{A_w}{\sum_{\tau=0}^{t} \binom{w}{\tau} \epsilon^{\tau} (1-\epsilon)^{w-\tau}},$$

with $A_w = |\{X \in \mathcal{C} : \ w(X) = w\}|$, for all $w = 0, 1, 2, \ldots, n$.

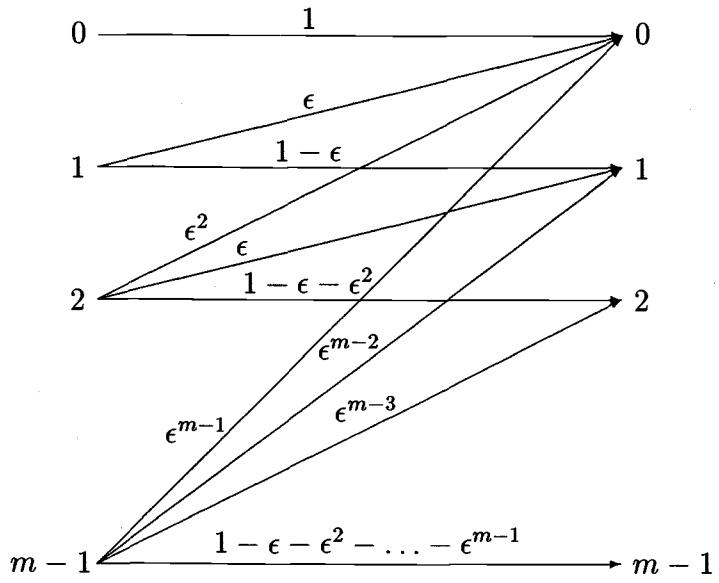

FIGURE 5.7. The m-ary Z-channel which takes into account the error magnitude.

5.3.3. Analysis of the throughput of Type-I Hybrid ARQ protocol over the m-ary Z-channel which takes into account the error magnitude

In this case, the channel model is as depicted in Figure 5.7. For all $\epsilon \in [0,1]$ the transition probabilities of such a channel satisfy:

$$P(y|x) = \begin{cases} 0 & \text{if } x < y, \\ 1 & \text{if } x = y = 0\,, \\ 1 - (\epsilon + \epsilon^2 + \ldots + \epsilon^x) & \text{if } x = y \in \mathbb{Z}_m - \{0\}, \\ \epsilon^{x-y} & \text{if } x > y, \end{cases} \qquad \text{for all } x, y \in \mathbb{Z}_m,$$

where $\epsilon \in [0,1]$. Hence, from (5.6),

$$\delta(x) = \epsilon \frac{1-\epsilon^x}{1-\epsilon} \approx \frac{\epsilon}{1-\epsilon}, \text{for all } x \in \mathbb{Z}_m - \{0\}.$$

From (5.11), and the above approximation,the following approximation holds:

$$P_a(X) \approx \left(1 - \frac{\epsilon}{1-\epsilon}\right)^{w(X)} \sum_{\tau=0}^{t} \binom{w(X)}{\tau} \left(\frac{\epsilon}{1-2\epsilon}\right)^{\tau}.$$

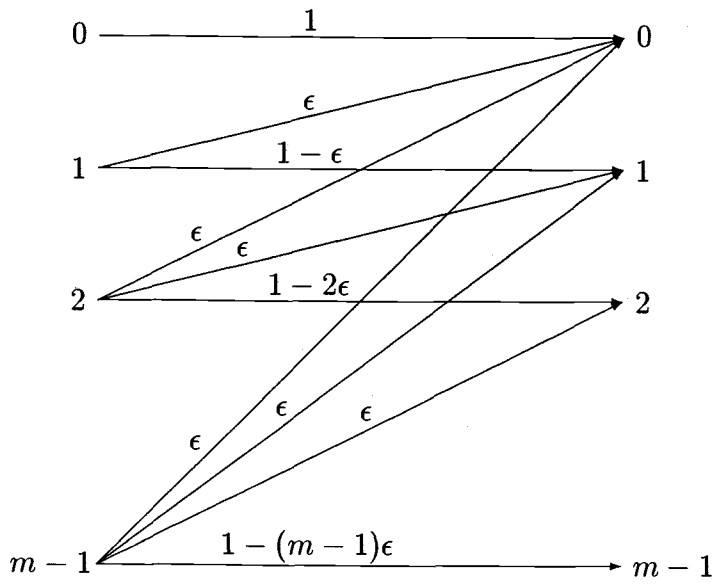

FIGURE 5.8. The m-ary Z-channel where every error type is equally likely=ϵ.

Hence, this case is approximately equivalent to the previous case. Note that analogous simple upper and lower bounds can be obtained from $\epsilon(1-\epsilon^m)/(1-\epsilon) \leq \delta(x) \leq \epsilon/(1-\epsilon)$, for all $x \in \mathbb{Z}_m - \{0\}$, and relation (5.9).

5.3.4. Analysis of the throughput of Type-I Hybrid ARQ protocol over the m-ary asymmetric Z-channel where every error type is equally likely

In this case, the transition probabilities of the channel shown in Figure 5.8 are:

$$P(y|x) = \begin{cases} 0 & \text{if } x < y, \\ 1 & \text{if } x = y = 0, \\ 1 - x\epsilon & \text{if } x = y \in \mathbb{Z}_m - \{0\}, \\ \epsilon & \text{if } x > y, \end{cases} \qquad \text{for all } x, y \in \mathbb{Z}_m,$$

where $\epsilon \in [0,1]$. And so, from (5.6), $\delta(x) = x\epsilon$, for all $x \in \mathbb{Z}_m - \{0\}$. From (5.9) and since $c_1 = \epsilon \leq \delta(x) \leq (m-1)\epsilon = c_2$, it is possible to obtain the following

bounds:

$$(1-(m-1)\epsilon)^{w(X)}\sum_{\tau=0}^{t}\binom{w(X)}{\tau}\left(\frac{\epsilon}{1-\epsilon}\right)^{\tau}$$

$$\leq \prod_{a=1}^{m-1}(1-a\epsilon)^{w_a(X)}\sum_{\tau=0}^{t}\binom{w(X)}{\tau}\left(\frac{\epsilon}{1-\epsilon}\right)^{\tau} \leq P_a(X)$$

$$\leq \prod_{a=1}^{m-1}(1-a\epsilon)^{w_a(X)}\sum_{\tau=0}^{t}\binom{w(X)}{\tau}\left(\frac{(m-1)\epsilon}{1-(m-1)\epsilon}\right)^{\tau}$$

$$\leq (1-\epsilon)^{w(X)}\sum_{\tau=0}^{t}\binom{w(X)}{\tau}\left(\frac{(m-1)\epsilon}{1-(m-1)\epsilon}\right)^{\tau}.$$

5.4. Analysis of ARQ Protocols with Diversity Combining using t-$AEC/AAED$ Codes over m-ary asymmetric Z-channel, $m \geq 2$

Assume that $\mathcal{C}$ is a t-Asymmetric Error Correcting and All Asymmetric Error Detecting (t-$AEC/AAED$) codes. Assume that the codeword $X \in \mathcal{C} \subseteq \mathbb{Z}_m^n$ is transmitted over the m-ary asymmetric Z-channel, $m \geq 2$. In the case of diversity combining, at each step τ, the receiver combines the received word, Y_τ, with the previous saved word, $Z_{\tau-1}$, to form a new word, Z_τ, as follows:

$$Z_\tau = \bigcup_{i=1}^{\tau} Y_i, \qquad \tau \in \mathbb{N} \text{ and } \tau \geq 1.$$

At the receiver, if there is no error in the combined word Z_τ, then Z_τ will be accepted and ACK will be sent to the transmitter requesting it to send the next codeword. If the receiver detects an error in the combined word Z_τ and the number of errors, e is within the error correcting capability of the designed code, i.e $e \leq t$, then the error(s) will be corrected and the receiver sends the transmitter ACK requesting it to send the next codeword. On the other hand, if $e > t$ then the receiver saves the erroneous combined word, Z_τ, and sends NAK to

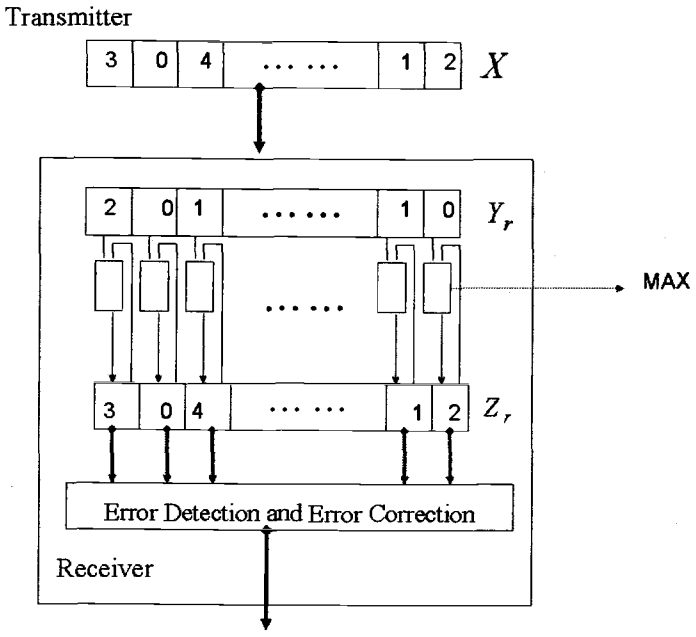

FIGURE 5.9. Packet reusing scheme

the transmitter requesting it to resend X. In this case, the word Z_τ is saved to combine later with the following received word, $Y_{\tau+1}$, to form a new word $Z_{\tau+1}$. This process continues until the combined word is successfully accepted. This scheme is shown in Figure 5.9 where the combining consists of a digit-by-digit MAX operation. In other words, the MAX operation is done on the digit-by-digit of the saved one $Z_{\tau-1}$, and the corresponding received word Y_τ, for all $\tau \geq 1$. The transmission and retransmission procedure for this diversity combining scheme is illustrated in Figure 5.10.

In this section, again we compute the number of retransmissions required to accept the code C correctly using this diversity combining scheme.

Let

$$\delta(x) = P(y \neq x|x) = \sum_{a \in \mathbf{Z}_m - \{x\}} P(a|x), \quad \forall x \in \mathbf{Z}_m$$

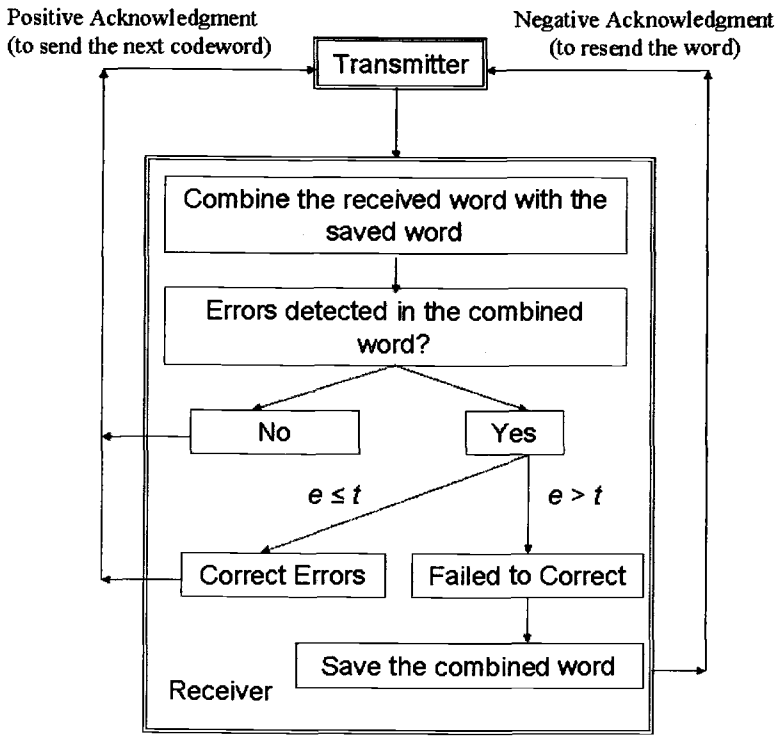

FIGURE 5.10. The proposed transmission and retransmission procedure for the diversity combining scheme.

be the probability that the symbol x is received in error, and

$$1 - \delta(x) = P(y = x|x) = P(x|x), \quad \forall x \in \mathbb{Z}_m$$

be the probability that the symbol x is received correctly. Also, assume that the channel model shown in Figure 5.6 is used, so that

$$\delta(x) = \text{constant} = \delta \in \mathbb{Z}_m - \{0\} \text{ and } \delta(0) = 0 \; . \tag{5.16}$$

Under this assumption, the 0's of the transmitted codeword $X \in \mathcal{C}$ are always received correctly and hence only the x_i's $\neq 0$ influence the average retransmission time of a word $X \in \mathcal{C}$. Now, let $R_i(X) : \Omega \to \mathcal{N} - \{0\}$ be the random variable defined as:

$$R_i = R_i(X) = \text{number of retransmissions needed to receive}$$

the i-th non-zero component of X correctly,

for all $i = 1, 2, \cdots, w$ where $w = w_H(X)$ is the Hamming weight of X.

From the above definition, it follows that the number of retransmissions needed to receive X correctly (i.e. the number of errors in Z_τ is $\leq t$) with a t-$AEC/AAED$ system is the random variable

$$R^{(t)}(X) = R^{(t)} = (t+1)\text{-th largest element in the set } \{R_1, R_2, \cdots, R_{w_H(X)}\}.$$

The following example shows how this scheme works.

Example 6 *Let* $X = (0122104130) \in \mathcal{C} \subseteq \mathbb{Z}_4^{10}$ *be the transmitted word over an m-ary Z-channel and assume the code can detect all errors. Suppose that* $Y_1 = (0012102110)$ *is the first received word. Table 5.1 shows how the original word can be recovered using diversity combining scheme. However, if t-AEC/AAED codes is used, then the original word can be recovered after* $R^{(t)}(X)$ *retransmissions where* $R^{(t)}(X) = t+1$*-th largest element in the set* $\{R_1, R_2, \cdots, R_7\}$*, i.e. the number of retransmissions required to recover the word is* $t+1$*th largest element in the set* $(2, 3, 1, 1, 4, 1, 2)$*. For example, if* 1*-AEC/AAED codes is used, then the error will be recovered after* $R^{(1)}(X) = 3$*nd largest element in the set* $(2, 3, 1, 1, 4, 1, 2)$*, i.e. after 3 retransmissions. Also, when* 2*-AEC/AAED codes is used, the errors will be recovered after* $R^{(2)}(X) = 2$ *retransmissions. Also,* $R^{(0)}(X) = 4$*,* $R^{(3)}(X) = 2$*, etc.*

In the rest of this section, we find an analytical expression for the average number of retransmissions needed to receive a transmitted word X correctly, $\mathbb{E}\left[R^{(t)}(X)\right]$.

Note that, since the channel is assumed to be a DMC, for the given X, the R_i's are independent. Also, since $\delta(x) = \delta = \text{constant}$, for all $x \in \mathbb{Z}_m$ –

τ	Y_τ	Z_τ
0	–	0000000000
1	0012102110	0012102110
2	0100003030	0112103130
3	0120102210	0122103130
4	0011104020	0122104130

TABLE 5.1. A sequence of transmissions example.

$\{0\}$, the R_i's are equally distributed as it will be shown below. So, the R_i's are independent and equally distributed. Hence, from the theory of order statistics [16], the cumulative distribution function (cdf) of $R^{(t)}$ in this case is

$$F_{R^{(t)}}(\tau) = P(R^{(t)} \leq \tau | X) = \sum_{h=0}^{t} \binom{w_H}{h} [F(\tau)]^{w_H - h} [1 - F(\tau)]^h , \qquad (5.17)$$

where

$$P(t+1\text{th largest element} \leq \tau) = P(\text{ all but } 1 \leq \tau) + \cdots + P(\text{ all but } t \leq \tau)$$

and $F(\tau) = F_{R_i}(\tau) = P(R_i \leq \tau | X)$ is the cdf of the R_i's. Let us now find out an expression for $F(\tau) = \sum_{j=1}^{\tau} P(R_i = j|X)$. From Equation (5.16) we have

$$P(R_i = j|X) = P(R_i = j| \text{ the } i\text{-th non zero component of } X) = \delta^{j-1}(1-\delta).$$

Hence,

$$F(\tau) = \sum_{j=1}^{\tau} \delta^{j-1}(1-\delta) = (1-\delta)\sum_{j=1}^{\tau} \delta^{j-1} = (1-\delta)\frac{1-\delta^\tau}{(1-\delta)} = 1 - \delta^\tau. \qquad (5.18)$$

Now, we compute the average number of retransmissions ($\mathbb{E}\left[R^{(t)}(X)\right]$), where

$$\mathbb{E}\left[R^{(t)}(X)\right] = \sum_{\tau=1}^{\infty} \tau \, P(R^{(\tau)} = \tau | X).$$

From (5.17) and (5.18), we have

$$P(R^{(t)} = \tau|X) = P(R^{(t)} \leq \tau|X) - P(R^{(t)} \leq \tau - 1|X)$$

$$= \sum_{h=0}^{t} \binom{w}{h} \left\{[F(\tau)]^{w-h}[1 - F(\tau)]^h - [F(\tau - 1)]^{w-h}[1 - F(\tau - 1)]^h\right\}$$

$$= \sum_{h=0}^{t} \binom{w}{h} \left\{(1 - \delta^\tau)^{w-h}\, \delta^{\tau h} - (1 - \delta^{\tau-1})^{w-h}\, \delta^{(\tau-1)h}\right\}$$

$$= \sum_{h=0}^{t} \binom{w}{h} \left\{\sum_{k=0}^{w-h} \binom{w-h}{k} (-1)^k\, \delta^{\tau k}\, \delta^{\tau h}\right.$$

$$\left. - \sum_{k=0}^{w-h} \binom{w-h}{k} (-1)^k\, \delta^{(\tau-1)k}\, \delta^{(\tau-1)h}\right\}$$

$$= \sum_{h=0}^{t} \binom{w}{h} \sum_{k=0}^{w-h} \binom{w-h}{k} (-1)^k \left\{\delta^{(h+k)\tau} - \delta^{(h+k)(\tau-1)}\right\}$$

$$= \binom{w}{0}\binom{w}{0}(-1)^1[1-1] + \sum_{k=1}^{w} \binom{w}{0}\binom{w}{k}(-1)^{k+1} \left\{\delta^{k(\tau-1)} - \delta^{k(\tau)}\right\}$$

$$+ \sum_{h=1}^{t} \sum_{k=0}^{w-h} \binom{w}{h}\binom{w-h}{k}(-1)^{k+1} \left\{\delta^{(h+k)(\tau-1)} - \delta^{(h+k)(\tau)}\right\}$$

$$= \sum_{k=1}^{w} (-1)^{k+1} \binom{w}{k} \left\{\delta^{k(\tau-1)} - \delta^{k\tau}\right\}$$

$$+ \sum_{h=1}^{t} \sum_{k=0}^{w-h} (-1)^{k+1} \binom{w}{h}\binom{w-h}{k} \left\{\delta^{(h+k)(\tau-1)} - \delta^{(h+k)\tau}\right\}.$$

Hence,

$$\mathbf{IE}\left[R^{(t)}(X)\right] = \sum_{\tau=1}^{\infty} \tau\ P(R^{(t)} = \tau|X) = \sum_{k=1}^{w} (-1)^{k+1} \binom{w}{k} \sum_{\tau=1}^{\infty} \tau\ \left\{\delta^{k(\tau-1)} - \delta^{k\tau}\right\}$$

$$+\sum_{h=1}^{t}\sum_{k=0}^{w-h}(-1)^{k+1}\binom{w}{h}\binom{w-h}{k}\sum_{\tau=1}^{\infty}\tau\ \left\{\delta^{(h+k)(\tau-1)}-\delta^{(h+k)\tau}\right\}$$

$$=S_1+S_2 \tag{5.19}$$

where

$$S_1=\sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}S(\delta,k), \tag{5.20}$$

$$S_2=\sum_{h=1}^{t}\sum_{k=0}^{w-h}(-1)^{k+1}\binom{w}{h}\binom{w-h}{k}S(\delta,h+k), \tag{5.21}$$

and

$$S(\delta,k)=\sum_{\tau=1}^{\infty}\tau\ \left(\delta^{k(\tau-1)}-\delta^{k\tau}\right),\quad \forall\ k=1,2,\cdots$$

$$=\sum_{\tau=0}^{\infty}\left(\delta^{k}\right)^{\tau}(\tau+1)-\sum_{\tau=1}^{\infty}\tau\ \left(\delta^{k}\right)^{\tau}$$

$$=1+\sum_{\tau=1}^{\infty}\left(\delta^{k}\right)^{\tau}$$

$$=1+\delta^{k}\sum_{\tau=0}^{\infty}\left(\delta^{k}\right)^{\tau}=1+\frac{\delta^{k}}{1-\delta^{k}} \tag{5.22}$$

Hence, from (5.20), S_1 can be derived as follows:

$$S_1=\sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}\left(1+\frac{\delta^{k}}{1-\delta^{k}}\right)$$

$$=\sum_{k=1}^{w}\binom{w}{k}(-1)^{k+1}\times 1+\sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}\left(\frac{\delta^{k}}{1-\delta^{k}}\right)$$

$$=\sum_{k=0}^{w}\binom{w}{k}(-1)^{k+1}-\binom{w}{0}(-1)+\sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}\left(\frac{\delta^{k}}{1-\delta^{k}}\right)$$

$$=-(1-1)^{w}+1+\sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}\left\{\frac{\delta^{k}}{1-\delta^{k}}\right\}.$$

Hence,

$$S_1 = 1 + \sum_{k=1}^{w} (-1)^{k+1} \binom{w}{k} \left(\frac{\delta^k}{1-\delta^k} \right) \tag{5.23}$$

An approximation for the above expression can be obtained if w is a constant as follows:

$$S_1 = 1 + \binom{w}{1} \frac{\delta}{1-\delta} - \binom{w}{2} \frac{\delta^2}{1-\delta^2} + \cdots + (-1)^{w+1} \binom{w}{w} \frac{\delta^w}{1-\delta^w}$$

$$\approx 1 + w \frac{\delta}{1-\delta} + O(\delta^2).$$

Now, consider S_2.

$$S_2 = \sum_{h=1}^{t} \sum_{k=0}^{w-h} (-1)^{k+1} \binom{w}{h} \binom{w-h}{k} \left(1 + \frac{\delta^{h+k}}{1-\delta^{h+k}} \right)$$

$$= \sum_{h=1}^{t} \sum_{k=0}^{w-h} (-1)^{k+1} \binom{w}{h} \binom{w-h}{k} + \sum_{h=1}^{t} \sum_{k=0}^{w-h} (-1)^{k+1} \binom{w}{h} \binom{w-h}{k} \left(\frac{\delta^{h+k}}{1-\delta^{h+k}} \right).$$

Assume that $t \leq w-1$, it follows:

$$\sum_{h=1}^{t} \sum_{k=0}^{w-h} \binom{w}{h} \binom{w-h}{k} (-1)^{k+1} = -\sum_{h=1}^{t} \binom{w}{h} \sum_{k=0}^{w-h} \binom{w-h}{k} (-1)^k$$

$$= \sum_{h=1}^{t} \binom{w}{h} (-1+1)^{w-h} = 0.$$

And so,

$$S_2 = \sum_{h=1}^{t} \sum_{k=0}^{w-h} (-1)^{k+1} \binom{w}{h} \binom{w-h}{k} \left(\frac{\delta^{h+k}}{1-\delta^{h+k}} \right)$$

$$= -\sum_{h=1}^{t} \sum_{k=0}^{w-h} (-1)^{k} \binom{w}{h} \binom{w-h}{k} \left(\frac{\delta^{h+k}}{1-\delta^{h+k}} \right) \tag{5.24}$$

Again, an approximation for the above expression can be obtained if w is a constant as follows:

$$S_2 = -w\frac{\delta}{1-\delta} + w(w-1)\frac{\delta^2}{1-\delta^2} + \cdots + (-1)^{w-t}\binom{w}{w-t}\binom{w-t}{w-t}\frac{\delta^w}{1-\delta^w}$$
$$= \approx -w\frac{\delta}{1-\delta} + O(\delta^2).$$

Substituting the expression of (5.23) and (5.24) in (5.19), the average retransmission time for a given word X to be received correctly is given by

$$\mathbb{E}\left[R^{(t)}(X)\right] = 1 + \sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}\frac{\delta^k}{1-\delta^k}$$
$$-\sum_{h=1}^{t}\sum_{k=0}^{w-h}(-1)^k\binom{w}{h}\binom{w-h}{k}\left\{\frac{\delta^{h+k}}{1-\delta^{h+k}}\right\}. \tag{5.25}$$

Note 4 From (5.22), the sum $S(\delta, k)$ can also be expressed as

$$S(\delta, k) = \sum_{r=0}^{+\infty}\left(\delta^k\right)^r, \tag{5.26}$$

and so, it is possible to derive another analytic expression for $\mathbb{E}\left[R^{(t)}(X)\right]$ as follows:

From (5.20) and (5.26),

$$S_1 = \sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}S(\delta, k) = \sum_{k=1}^{w}(-1)^{k+1}\binom{w}{k}\sum_{r=0}^{+\infty}\left(\delta^k\right)^r$$
$$= -\sum_{r=0}^{+\infty}\sum_{k=1}^{w}\binom{w}{k}\left(-\delta^r\right)^k = -\sum_{r=0}^{+\infty}\left[\sum_{k=0}^{w}\binom{w}{k}\left(-\delta^r\right)^k - \binom{w}{0}\left(-\delta^r\right)^0\right]$$
$$= \sum_{r=0}^{+\infty}\left[1-(1-\delta^r)^w\right] = 1 + \sum_{r=1}^{+\infty}\left[1-(1-\delta^r)^w\right].$$

Whereas, from (5.21), (5.26) and the non-restrictive assumption $t < w$,

$$S_2 = \sum_{\tau=1}^{t}\sum_{k=0}^{w-\tau}(-1)^{k+1}\binom{w}{\tau}\binom{w-\tau}{k}S(\delta, \tau+k)$$

$$= \sum_{\tau=1}^{t} \sum_{k=0}^{w-\tau} (-1)^{k+1} \binom{w}{\tau} \binom{w-\tau}{k} \sum_{r=0}^{+\infty} \left(\delta^{\tau+k}\right)^r$$

$$= -\sum_{r=0}^{+\infty} \sum_{\tau=1}^{t} \binom{w}{\tau} (\delta^r)^\tau \sum_{k=0}^{w-\tau} \binom{w-\tau}{k} (-\delta^r)^k$$

$$= -\sum_{r=0}^{+\infty} \sum_{\tau=1}^{t} \binom{w}{\tau} (\delta^r)^\tau (1-\delta^r)^{w-\tau}$$

$$= -\sum_{r=1}^{+\infty} \sum_{\tau=1}^{t} \binom{w}{\tau} (\delta^r)^\tau (1-\delta^r)^{w-\tau}.$$

Hence,

$$\mathbf{IE}\left[R^{(t)}(X)\right] = S_1 + S_2 = 1 + \sum_{r=1}^{+\infty} \left[1 - (1-\delta^r)^w - \sum_{\tau=1}^{t} \binom{w}{\tau} (\delta^r)^\tau (1-\delta^r)^{w-\tau}\right]$$

$$= 1 + \sum_{r=1}^{+\infty} \left[1 - \sum_{\tau=0}^{t} \binom{w}{\tau} (\delta^r)^\tau (1-\delta^r)^{w-\tau}\right]$$

$$= 1 + [1 - f(\delta, w, t)] + \left[1 - f\left(\delta^2, w, t\right)\right] + \left[1 - f\left(\delta^3, w, t\right)\right] + \ldots. \qquad (5.27)$$

where $f(\delta, w, t)$ is the function given in (5.12). Further from (5.27) and the upper bound in (5.14) it follows that:

$$\mathbf{IE}\left[R^{(t)}(X)\right] = 1 + \sum_{r=1}^{+\infty} \left[1 - f\left(\delta^r, w, t\right)\right] \leq 1 + \binom{w}{t+1} \sum_{r=1}^{+\infty} \left(\delta^{t+1}\right)^r$$

$$= 1 + \binom{w}{t+1} \frac{\delta^{t+1}}{1-\delta^{t+1}}.$$

From (5.27) and the lower bound in (5.14) it follows:

$$\mathbf{IE}\left[R^{(t)}(X)\right] = 1 + \sum_{r=1}^{+\infty} \left[1 - f\left(\delta^r, w, t\right)\right]$$

$$\geq 1 + \sum_{r=1}^{+\infty} \binom{w}{t+1} (\delta^r)^{t+1} (1-\delta^r)^{w-(t+1)}$$

$$= 1 + \binom{w}{t+1}(1-\delta)^{w-t-1}\sum_{r=1}^{+\infty}(\delta^r)^{t+1}$$

$$= 1 + \binom{w}{t+1}(1-\delta)^{w-t-1}\frac{\delta^{t+1}}{1-\delta^{t+1}}.$$

From the previous two inequalities, we obtain

$$1 + \binom{w}{t+1}\frac{\delta^{t+1}}{1-\delta^{t+1}}(1-\delta)^{w-t-1} \leq \mathbb{E}\left[R^{(t)}(X)\right] \leq 1 + \binom{w}{t+1}\frac{\delta^{t+1}}{1-\delta^{t+1}}. \quad (5.28)$$

Hence, when $(1-\delta)^{w-(t+1)} \simeq 1$,

$$\mathbb{E}\left[R^{(t)}(X)\right] \simeq 1 + \binom{w}{t+1}\frac{\delta^{t+1}}{1-\delta^{t+1}}. \quad (5.29)$$

Note that expression (5.27) can be used to compute the exact value of $\mathbb{E}\left[R^{(t)}(X)\right]$ in a very efficient manner, especially when w is very large. In this way we calculated the values in Table 5.2. Further, expression (5.27) truncated to a certain value of r gives generally a better approximation for $\mathbb{E}\left[R^{(t)}(X)\right]$ than the one given in (5.29), as shown Example 7.

The expression in (5.27) implies that the average number of retransmissions required to receive the word X correctly depends only on its *Hamming weight* $w = w_H(X)$. For a code $\mathcal{C}$ used in the system, the average number of retransmissions can be obtained by taking the average over all codewords. We also assume that all codewords are equally likely transmitted. Hence, the average number of retransmissions for the code $\mathcal{C}$ is

$$\mathbb{E}\left[R^{(t)}(\mathcal{C})\right] = \frac{1}{|\mathcal{C}|}\sum_{X\in\mathcal{C}}\mathbb{E}\left[R^{(t)}(X)\right]$$

$$= \frac{1}{|\mathcal{C}|}\sum_{w\in\mathbb{Z}_m^n} A_w \times \bar{R}^{(t)}_{DC\ ARQ}(\delta, w_H),$$

with $A_w = |\{X \in \mathcal{C} : \ w(X) = w\}|$, for all $w \in \mathbb{Z}_n^m$, being the weight distribution of the code $\mathcal{C}$.

The above equation is valid under the same assumption, $\delta(0) = 0$ and $\delta(a)$ = constant = δ, for all $a \in \mathbb{Z}_m - \{0\}$.

Table 5.2 shows the average number of retransmissions using a type-I hybrid ARQ and diversity combining using *t-AEC/AAED* codes over the m-ary asymmetric Z-channel where every symbol error is equally likely.

Example 7 *Assume that every symbol error is equally likely, then* $\delta(0) = 0$ *and* $\delta(x) = \epsilon$ *for all* $x \in \mathbb{Z}_m - \{0\}$. *So, from (5.25) and (5.27), the average number of retransmissions needed to receive the codeword* X *correctly is*

$$\mathbb{E}\left[R^{(t)}(X)\right] = \overline{R}^{(t)}_{DCARQ}(w) = 1 + \sum_{r=1}^{+\infty}\left[1 - \sum_{\tau=0}^{t}\binom{w}{\tau}(\epsilon^r)^\tau(1-\epsilon^r)^{w-\tau}\right]$$

$$= 1 + [1 - f(\epsilon, w, t)] + \left[1 - f\left(\epsilon^2, w, t\right)\right] + \left[1 - f\left(\epsilon^3, w, t\right)\right] + \ldots .$$

If $\epsilon = 0.01$, $w = w(X) = 100$ *and 2-AEC/AAED codes is used, i.e.* $t = 2$, *then the average number of retransmissions needed to receive the transmitted word* X *correctly is:*

$$\begin{aligned}\mathbb{E}\left[R^{(t)}(X)\right] &= \bar{R}^{(2)}_{DC}(100)\\ &= 1 + f_1(\epsilon = 0.01, w = 100, t = 2) + f_1(\epsilon = 0.01^2, w = 100, t = 2)\\ &+ f_1(\epsilon = 0.01^3, w = 100, t = 2) + \cdots = 1.079373362.\end{aligned}$$

On the other hand, if we use the bound given in (5.28), we obtain

$$1.060999226 \leq \mathbb{E}\left[R^{(t=2)}(X)\right] \leq 1.1617001.$$

	$\overline{R}^{(t)}_{Hyb}(w)$	$\overline{R}^{(t)}_{DC}(w)$	$\overline{R}^{(t)}_{Hyb}(w)$	$\overline{R}^{(t)}_{DC}(w)$	$\overline{R}^{(t)}_{Hyb}(w)$	$\overline{R}^{(t)}_{DC}(w)$
$t\backslash\epsilon$	0.1		0.01		0.001	
0	719380.7160	2.858090371	3.619887649	1.736596677	1.136625792	1.120331087
1	47258.58719	2.374110939	1.578717521	1.366655067	1.007532535	1.007476228
2	6224.745005	2.137628681	1.159390832	1.137478423	1.000311003	1.000310906
3	1232.137544	2.039483749	1.041976746	1.040285685	1.000009662	1.000009662
4	325.5382409	2.006534669	1.009699440	1.009606265	1.000000238	1.000000238
5	107.5263361	1.992620862	1.001924612	1.001920915	1.000000004	1.000000004
6	42.57759872	1.976842982	1.000329682	1.000329505	1	1
7	19.62371020	1.949090571	1.000049345	1.000049343	1	1
8	10.29575810	1.902879204	1.000006540	1.000006540	1	1
9	6.041371480	1.834475419	1.000000776	1.000000776	1	1
10	3.906820493	1.744037443	1.000000083	1.000000083	1	1
11	2.749091331	1.636243484	1.000000008	1.000000008	1	1
12	2.081021117	1.519466685	1.000000001	1.000000001	1	1
13	1.676974460	1.403688044	1.000000000	1.000000000	1	1
14	1.424535886	1.298016991	1.000000000	1.000000000	1	1
15	1.263876864	1.208783691	1	1	1	1
16	1.161116970	1.138760346	1	1	1	1
$t\backslash\epsilon$	10^{-4}		10^{-5}		10^{-6}	
0	1.012882918	1.012720340	1.001280825	1.001279200	1.000128008	1.000127992
1	1.000080606	1.000080600	1.000000812	1.000000812	1.000000008	1.000000008
2	1.000000338	1.000000338	1.000000000	1.000000000	1	1
3	1.000000001	1.000000001	1	1	1	1
4	1	1	1	1	1	1
$t\backslash\epsilon$	10^{-7}		10^{-8}		10^{-9}	
0	1.000012800	1.000012799	1.000001280	1.000001280	1.000000128	1.000000128
1	1.000000000	1.0000000	1	1	1	1
2	1	1	1	1	1	1
$t\backslash\epsilon$	10^{-10}		10^{-11}		10^{-12}	
0	1.000000012	1.000000012	1.000000001	1.000000001	1	1
1	1	1	1	1	1	1

TABLE 5.2. Average number of retransmissions for a word X with weight $w = 128$ using type-I hybrid ARQ, $\overline{R}^{(t)}_{Hyb}(w)$, and diversity combining scheme, $\overline{R}^{(t)}_{DC}(w)$.

6. CONCLUSION AND FUTURE WORK

In this thesis, some new results on error control techniques for the asymmetric channel are given. Specifically, new results are given on:

* The capacity of the asymmetric channel.

* Analysis of the extended error detecting capabilities of Bose-Lin codes.

* t-unidirectional error detecting codes over Z_m, $m \geq 2$.

* Type-I hybrid ARQ using $t\text{-}AEC/AAED$ codes over the m-ary Z-channel, $m \geq 2$.

* Diversity combining scheme using $t\text{-}AEC/AAED$ codes over the m-ary Z-channel, $m \geq 2$.

More specifically, in Chapter 2, an expression for the capacity of the binary asymmetric channel is derived. Using this expression, the capacities of the binary asymmetric channel (BSC) and the Z channel can be obtained as special cases.

In Chapter 3, some analysis of Bose-Lin codes, for error detecting capabilities beyond the maximum designed error detection, are given. It is shown that the codes can detect errors beyond the designed maximum error detection capabilities of the codes. When the error characteristic of a channel is asymmetric or unidirectional, these codes can be successfully applied. Such conditions can occur in applications where Bose-Lin codes are applied to relatively large blocks of data or circuitry where the number of errors in the code word can occasionally exceed the maximum that the code is designed to detect.

A new class of a systematic t-unidirectional error detecting codes over Z_m is designed in Chapter 4. It is shown that the constructed codes can detect up to

2 errors when using 2 check bits. Also, it is shown that the constructed codes can detect up to $m^{r-2} + r - 2$ using $r \geq 3$ check bits. When $m = 2$, these codes are equivalent to the Method 1 of Bose-Lin codes [11]. By using r check digits, an upper bound on the maximum number of detectable errors is given. This bound is a generalization of the bound given in [23], which is for $m = 2$.

In chapter 5, the throughput of the pure ARQ protocols using different codes over the m-ary Z channel, $m \geq 2$, is derived. We derive the throughput of the system by computing the number (or the expected number) of retransmissions needed to receive all codewords correctly. We analyze the throughput first for ARQ protocols. Then we consider type-I hybrid ARQ protocols, which use t-Asymmetric Error Detecting (t-AED) codes and also using All Asymmetric Error Detecting ($AAED$) codes. We also explain a simple diversity combining scheme for general m-ary Z-channel, and again derive the throughput efficiency of these schemes. From these results, it can be seen that the type-I hybrid ARQ protocol is inferior to the diversity combining scheme, especially when ϵ is large and/or t is small. On the other hand, when ϵ is small and/or t is large, their performance is essentially the same.

6.1. Further Research

The capacity of the binary asymmetric channel is derived in this thesis. One of the future research problems is to design asymmetric codes with rate close to the capacity of the asymmetric channel. Another problem in this area is to derive the capacity of the various m-ary Z asymmetric channels described in Chapter 5.

It is known that the binary Bose-Lin codes are optimal when the number of check bits, r, is 2,3, and 4. It is not clear whether the codes are optimal for $r \geq 5$ and this is an open problem. In addition, the optimality of the proposed t-asymmetric error detecting with $m \geq 2$ needs further investigation. We have not investigated the error detecting capabilities of the proposed t-ary error detecting codes for the number of errors beyond the designed maximum value. This problem is worth studying.

Most of the ARQ protocols designed here are for asymmetric channels. In future, we would like to investigate similar techniques for unidirectional errors. In fact, a simple diversity combining scheme similar to the one proposed here for asymmetric errors can be also designed for unidirectional errors. This scheme can correct up to $\lfloor \frac{t}{2} \rfloor$-unidirectional errors using t-unidirectional error detecting (t-UED) codes as we briefly mentioned.

The diversity schemes now consists of bit-by-bit OR operation for the $1 \rightarrow 0$ errors and bit-by-bit AND operation for $0 \rightarrow 1$ errors. When t-UED code is used and the number of errors is less than or equal to $\lfloor \frac{t}{2} \rfloor$, the receiver can find out whether $1 \rightarrow 0$ errors or $0 \rightarrow 1$ errors have occurred in the received word. For example, assume that a system uses Borden's-4-error detecting code with length $n = 20$, i.e. the codeword weights are 0, 5, 10, 15, and 20. Suppose, the receiver receives a word with weight 12. Then, it is clear that the original word must have weight 10 and the errors are of $0 \rightarrow 1$ type. On the other hand, if the receiver receives a word with weight 8 then again the original word must have weight 10; however, in this case, the errors must be of $1 \rightarrow 0$ type. This is because it is assumed at most 2 unidirectional errors can occur in the codewords. Thus, performing bit-by-bit OR operation for $1 \rightarrow 0$ errors and bit-by-bit AND operation for $0 \rightarrow 1$ errors, the receiver can obtain the original word. For

general m-ary codes, these operations must be digit-by-digit MAX and MIN. The throughput analysis for this method for different m-ary ($m \geq 2$) codes requires further investigation.

6.2. REFERENCES

[1] S. Al-Bassam and B. Bose. Asymmetric/unidirectional error correcting and detecting codes. *IEEE Trans. Comput.*, C-43:590–597, May 1994.

[2] R.J. Benice. An analysis of retransmission systems. *IEEE Transactions on Communications*, pages 135–154, December 1964.

[3] J.M. Berger. A note on error detecting codes for asymmetric channels. *Information and Control*, 4:68–73, March 1961.

[4] E.R. Berlekamp. *Algebraic Coding Theory.* Aegean Park Press, 1984.

[5] M. Blaum. *Codes for Detecting and Correcting Unidirectional Errors.* IEEE Computer Society Press, Los Alamitos, CA, 1993.

[6] M. Blaum and H. van Tilborg. On t-error correcting/all unidirectional error detecting codes. *IEEE Trans. Comput.*, 38:1493–1501, November 1989.

[7] J.M. Borden. Optimal asymmetric error detecting codes. *Information and Control*, 53:66–73, April 1982.

[8] B. Bose and S. Al-Bassam. On systematic single asymmetric error correcting codes. *IEEE Trans. Inform. Theory*, pages 669–672, March 2000.

[9] B. Bose and S. Cunningham. Asymmetric error correcting codes. In *Methods in Communication, Security and Computer Science.* Springer-Verlag, New York, 1993.

[10] B. Bose, Samir Elmougy, and L.G. Tallini. Systematic t-unidirectional error detecting codes in z_m. *To be submitted to IEEE Trans. on Computers.*

[11] B. Bose and D. Lin. Systematic unidirectional error-detecting codes. *IEEE Trans. Comput.*, 34:63–69, November 1985.

[12] B. Bose and D.K. Pradhan. Optimal unidirectional error detecting/correcting codes. *IEEE Trans. Comput.*, 31:564–568, June 1982.

[13] R.C. Bose and D.K. Ray-Chaudhuri. Further results on error correcting binary group codes. *Information and Control*, pages 279–290, September 1960.

[14] R.C. Bose and D.K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, page 68, March 1960.

[15] M. Covers and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, 1991.

[16] H.A. David. Order Statistics. John Wiley & Sons, New York, 1970.

[17] N.G. de Bruijin, C. Van Ebbenhorst Tengbergen, and D. Kruswijk. On the set of divisors of a number. *Nieuw Archief Voor Wiskunde*, 23:191–193, 1951.

[18] P. Delsarte and P. Piret. Bounds and constructions for binary asymmetric error-correcting codes. *IEEE Trans. Inform. Theory*, pages 125–131, January 1981.

[19] P. Elias. Coding for noisy channel. *IRE Conv. record*, pages 47–57, 1955.

[20] Samir Elmougy and Steven S. Gorshe. Some error detecting properties of bose-lin codes. *Submitted to the Journal of IEE proceedings on Computers and Digital Techniques.*

[21] C.V. Freiman. Optimal error detecting codes for completely asymmetric binary channels. *Information and Control*, 5:66–71, March 1962.

[22] S. Gorshe and B. Bose. A self-checking alu design with efficient codes,. *Proceedings of 14th IEEE VSLI Test Symposium,*, pages pp. 157–161, 1996.

[23] Luisa Gargano Grard D. Cohen and Ugo Vaccaro. Unidirectional error-detecting codes. *EUROCODE 90, Lectures Notes in Computer Science.*, 514(5):94–105, 1991.

[24] R.W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 1950.

[25] N.K. Jha. A new class of symmetric error correcting/unidirectional error detecting codes. *Computers and Mathematics with Applications*, pages 95–104, May 1990.

[26] N.K. Jha and M.B. Vora. A t-unidirectional error detecting systematic code. *Computers and Mathematics with Applications*, pages 705–714, 1988.

[27] S. Jiang and E. Fujiwara. A class of unidirectional byte error locating codes with single symmetric bit error correction capability. *IEICE Trans. Comput*, November 1994.

[28] R.S. Katti and M. Blaum. An improvement on constructions of t-ec/aued codes. *IEICE Trans. Comput*, May 1996.

[29] T. Kløve and V. Korzhik. *Error Detecting Codes. General Theory and their Application in Feedback Communication Systems.* Kluwer Academics, Boston/London/Dordrecht, 1995.

[30] T. Klove, P. Oprisan, and B. Bose. Diversity combining for the z-channel. *IEEE Trans. on Information Theory*, pages 1174–1178, 2005.

[31] S. Kundu and S. Reddy. On systematic t-error correcting/all unidirectional error detecting codes. *IEEE Trans. Comput.*, 39:752–761, June 1990.

[32] C.S. Laih and C.N. Yang. On the analysis and design of group theoretical t-syec/auec codes. *IEICE Trans. Comput*, January 1996.

[33] S. Lin and D.J. Costello. *Error Control Coding: Fundamentals and Applications.* Prentice Hall, 1983.

[34] Hang Liu, Hairuo Ma, Magda El Zarki, and Sanjay Gupta. Error control schemes for networks: an overview. *Mob. Netw. Appl.*, 2(2):167–182, 1997.

[35] R.J. McEliece. Comment on 'a class of codes for asymmetric channels and a problem from the additive theory of numbers. *IEEE Trans. Inform. Theory*, page 137, January 1973.

[36] R.J. McEliece and E.R. Rodemich. The Constantin-Rao construction for binary asymmetric error correcting codes. *Inform. and Control*, pages 187–196, January 1980.

[37] B.L. Montgomery and B.V.K.V. Kumar. Systematic random error correcting and all unidirectional error detecting codes. *IEEE Trans. Comput.*, pages 836–840, June 1990.

[38] K. Naemura. Semidistance codes and t-symmetric error correcting/all unidirectional error detecting codes. *IEEE Trans. Inform. and Systems*, pages 873–883, November 1992.

[39] P. Oprisan. *Error Control Techniques for the Z-Channel.* Ph.D. Thesis, Oregon State University, USA, 2005.

[40] Vera Pless. *Introduction to the Theory of Error-Correcting Codes.* John Wiley & Sons, New York, 3^{rd} edition, 1998.

[41] E. Prange. Cyclic error-correcting codes in two symbols. *Air Force Cambridge Research Center*, pages 57–103, 1957.

[42] E. Prange. Some cyclic error-correcting codes with simple decoding algorithms. *Air Force Cambridge Research Center*, pages 59–164, 1958.

[43] E. Prange. The use of coset equivalence in the analysis and decoding of group codes. *Air Force Cambridge Research Center*, pages 59–164, 1959.

[44] I.S. Reed and G. Solomon. Polynomial codes over certain finite fields. *SIAM Journal on Applied Mathematics*, page 300, 1960.

[45] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, pages 379–423, 1948. Part I.

[46] A. Shiozaki. Construction for binary asymmetric error-correcting codes. *IEEE Trans. Inform. Theory*, pages 787–789, September 1982.

[47] A. Shiozaki. Single asymmetric error-correcting cyclic AN codes. *IEEE Trans. Comput.*, pages 554–555, June 1982.

[48] V. Skacheck, T. Etzion, and R.M. Roth. Efficient encoding algorithm for third-order spectral-null codes. *IEEE Trans. Inform. Theory*, March 1998.

[49] L.G. Tallini, S. Al-Bassam, and B. Bose. Capacity and codes for the Z-channel. *Proceedings, Int. Symp. of Information Theory*, June 2002.

[50] L.G. Tallini, Samir Elmougy, and B. Bose. Type-i hybrid arq protocols and diversity combining arq protocols over the m-ary asymmetric channels, $m \geq 2$. *Submitted to IEEE Trans. on Information Theory.*

[51] N.H. Vaidya. Unidirectional bit/byte error control. *IEEE Trans. Comput*, May 1995.

[52] S.B. Wicker. *Error Control Systems for Digital Communication and Storage.* Prentice Hall, 1995.

[53] J.M. Wozencraft and B. Reiffen. *Sequential Decoding.* Cambridge, MA: MIT Press, 1961.