AN ABSTRACT OF THE THESIS OF

Noha Elarief for the degree of Master of Science in Computer Science presented on October 16, 2008

Title: Diversity Combining ARQ over the $m$ ($> 2$)-ary Unidirectional Channel

Abstract approved:

________________________________________________________________________________________

Bella Bose

In diversity combining automatic repeat request (ARQ), erroneous packets are combined together forming a single, more reliable, packet. In this thesis, we give a diversity combining scheme for the $m$-ary unidirectional channel. A system using the given scheme with a $t$-unidirectional error detecting code is able to correct up to $E_{max} = \left\lfloor \frac{t}{2} \right\rfloor$ unidirectional errors. To use the given scheme, the decoder should be able to decide the error type (increasing or decreasing). Hence, we give simple techniques to make this decision for various unidirectional error detecting codes.
Diversity Combining ARQ over the $m \geq 2$-ary Unidirectional Channel

by

Noha Elarief

A THESIS

submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Master of Science

Presented October 16, 2008
Commencement June 2009
Master of Science thesis of Noha Elarief presented on October 16, 2008

APPROVED:

________________________
Major Professor, representing Computer Science

________________________
Director of the School of Electrical Engineering and Computer Science

________________________
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 request.

________________________
Noha Elarief, Author
ACKNOWLEDGMENTS

All praise is due to Allah, Most Gracious, Most Merciful. There is no ease except in that which He has made easy and only Him makes the difficulty - if He wishes - easy. I pray to Him that He would accept my humble achievement and would give me strength to seek more knowledge: “And say: O my Lord! Advance me in knowledge.” (Qur’an, Ta-Ha 20:114).

I would like to express my gratitude to all those who offered their valuable help to me throughout my studies. My deep and sincere thanks go to my supervisor, Dr. Bose, from whom I have learned alot, both on the professional and human levels. Inspite of his busy schedule, he has always been there to help. This work would not have been possible without his support and inspiration.

I would like to thank my parents, to whom I dedicate this work. They raised me, bore with me and loved me. I sincerely hope that they are be proud of me. I would like to convey my gratitude for my sisters and my wonderful nieces and nephew. The emotional environment they enveloped me with helped me overcome the hardest of times. I owe my most sincere gratitude to my teacher, Ms Heba Aly, who taught me what it means to have a goal in life. She has always been there for me, even while far away from home. Special thanks go to my friends for their love, care and support.

I would also like to convey my thanks to Dr. Mary Flahive, Dr. Thinh Nguyen and Dr. Jim Coakley for seving in my committee. Finally, I would like to acknowledge the National Science foundation as my work was supported by the NFS grants CCF-0801452 and CCF-0728810.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Automatic Repeat Request (ARQ) Protocols</td>
<td>3</td>
</tr>
<tr>
<td>3 Unidirectional errors’ correction using diversity combining</td>
<td>5</td>
</tr>
<tr>
<td>4 Performance Analysis</td>
<td>8</td>
</tr>
<tr>
<td>5 Determining error type for various $t$-UED codes</td>
<td>14</td>
</tr>
<tr>
<td>5.1 Optimal constant weight codes</td>
<td>14</td>
</tr>
<tr>
<td>5.2 Systematic AUED codes</td>
<td>14</td>
</tr>
<tr>
<td>5.3 Systematic $t$-UED codes</td>
<td>15</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>23</td>
</tr>
<tr>
<td>Bibliography</td>
<td>24</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>4.1 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 4)$-UED code of length $n = 30$</td>
<td>9</td>
</tr>
<tr>
<td>4.2 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 8)$-UED code of length $n = 30$</td>
<td>10</td>
</tr>
<tr>
<td>4.3 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using AUED code of length $n = 30$</td>
<td>10</td>
</tr>
<tr>
<td>4.4 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 4)$-UED code of length $n = 40$</td>
<td>11</td>
</tr>
<tr>
<td>4.5 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 8)$-UED code of length $n = 40$</td>
<td>11</td>
</tr>
<tr>
<td>4.6 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using AUED code of length $n = 40$</td>
<td>12</td>
</tr>
<tr>
<td>4.7 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 4)$-UED code of length $n = 50$</td>
<td>12</td>
</tr>
<tr>
<td>4.8 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 8)$-UED code of length $n = 50$</td>
<td>13</td>
</tr>
<tr>
<td>4.9 Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using AUED code of length $n = 50$</td>
<td>13</td>
</tr>
</tbody>
</table>
DIVERSITY COMBINING ARQ OVER THE $M(\geq 2)$-ARY UNIDIRECTIONAL CHANNEL

1. INTRODUCTION

During the process of data transmission, noise affecting the transmission channel often causes data to be distorted. Depending upon the transmission channel properties, errors can be classified as symmetric, asymmetric or unidirectional. Let $X = (x_{n-1}, x_{n-2}, \ldots, x_0)$ and $X' = (x'_{n-1}, x'_{n-2}, \ldots, x'_0)$ be two vectors over the digit set $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$, where $X$ is the transmitted word and $X'$ is the received word. We say that $X'$ is erroneous if $\exists i \in \{0, 1, \ldots, n-1\}$ such that $x_i \neq x'_i$; the error is of increasing type if $x_j \leq x'_j$ and is of decreasing type if $x_j \geq x'_j$, $\forall j \in \{0, 1, \ldots, n-1\}$. In channels where both increasing and decreasing errors are likely to occur simultaneously, the error is classified as symmetric.

On the other hand, when only one error type - known a priori - can occur, the error is said to be asymmetric. Unidirectional errors are similar to asymmetric errors, but the difference is that the type of error that can occur is not known a priori. The study of unidirectional errors is of particular importance when modeling common VLSI failures and connection faults.

In [1], common sources of unidirectional errors in VLSI devices are given: multiple faults in address decoders, open line and line shortening in word lines, failures in power supplies and stuck-at-faults in shift registers.

In order to achieve reliable data transmission through a noisy channel, two main error-control schemes are used: Forward Error Correcting (FEC) and Automatic Repeat Request (ARQ). FEC is a technique where the sender adds redundancy to the original message in such a way that it is possible for the receiver to detect the presence of errors and correct the erroneous digits (up to some tolerance level), recovering the original message whenever possible. This is particularly important when re-sending the message is not possible.

In ARQ systems, on the other hand, the sender encodes the message using error detecting codes. Then, the receiver checks the encoded message for errors, and requests retransmission of erroneous data. Sometimes FEC and ARQ are combined so that the receiver corrects
minor errors and requests retransmission only when the error overhead is too high; this is referred to as Hybrid ARQ (HARQ).

Two main approaches to ARQ are identified: plain ARQ and Diversity combining ARQ. In plain ARQ systems, the receiver completely discards erroneous packets and requests their retransmission. Nevertheless, these discarded packets may contain useful information. Thus, they can be combined together in order to recover the original packet. The idea of packet combining was first suggested by Sindhu [2] who introduced the ARQ-with-memory (MARQ) technique. Sindhu showed that MARQ is a significant improvement over plain ARQ in the sense that it increases the throughput. In [3], the authors proposed a diversity combining ARQ protocol for the $m \geq 2$-ary asymmetric channel and showed that the diversity combining scheme vastly improves the throughput. For example, in a system with error probability $p = 0.1$ using the all asymmetric error detecting (AAED) 50-out-of-100 code, the average number of transmissions required to successfully receive a packet is 2.45 when using diversity combining ARQ as opposed to 194.03 using plain ARQ. In this thesis, we extend that scheme to the $m \geq 2$-ary unidirectional channel. A system using this diversity combining ARQ protocol is able to correct up to $E_{\text{max}} = \left\lfloor \frac{t}{2} \right\rfloor$ errors using $t$-unidirectional error detecting ($t$-UED) codes. We also give techniques for determining the error type for various unidirectional error detecting codes since identifying the error type is a crucial step in the given diversity combining scheme. Furthermore, the given technique for deciding the error type in the systematic $m$-ary $t$-UED code [4, 5] is applicable to the binary Bose-Lin code [6] and is much simpler than the one given in [7].

This thesis is organized as follows: Chapter 2 describes the main ARQ protocols. Chapter 3 explains the process of diversity combining for unidirectional channels. In Chapter 4, we compare the performance of the given diversity combining ARQ with the plain ARQ protocol. Chapter 5 describes methods for determining the error type in the received word for various error detecting codes. Finally, the thesis is concluded in Chapter 6.
2. **Automatic Repeat Request (ARQ) Protocols**

As mentioned earlier, ARQ is an error-control protocol whose goal is to send packets reliably over noisy channels. The receiver checks the incoming packets for errors and automatically requests the retransmission of the erroneous packets. This is typically done using *acknowledgments* and *negative acknowledgments*. An acknowledgment (ACK) is a signal that the receiver sends to indicate that the packet has been successfully received. If, on the other hand, the receiver detects the presence of error in the received packet, it requests retransmission by sending a negative acknowledgment (NAK) signal to the sender [7].

There are three main ARQ protocols: Stop and Wait ARQ, Go-Back-N ARQ and Selective Repeat ARQ. We hereby give a brief explanation for each:

- **Stop and wait protocol**: In the simplest implementation of this protocol, the sender sends one packet at a time, and waits for an ACK from the receiver to send the next packet. If the sender receives a NAK instead, it re-transmits the codeword. The main advantage of the stop and wait protocol is its simplicity. On the other hand, when the response time is long, this technique is clearly inefficient since the sender remains idle, waiting for ACKs, which dramatically decreases the channel’s throughput.

- **Go-Back-N ARQ**: In this protocol, the sender continues to send a number of packets - specified by a window size - without waiting for acknowledgments from the receiver. The receiver sends an ACK for every packet it correctly receives. When the receiver detects an error in a received packet, it sends a NAK signal holding this packet’s id and ignores all subsequent packets until it is correctly received. Once the sender has finished sending all the packets in the current window, it will go back and send all packets following the last acknowledged one. Thus, some buffering is necessary at the sender. This protocol uses the channel more efficiently than the stop and wait protocol as the sender does not remain idle waiting for acknowledgments. However, packets are being sent multiple times. This problem is avoided in selective-repeat ARQ.
- Selective-repeat ARQ: In this protocol, likewise, the sender continues to send a number of packets - specified by a window size - without waiting for acknowledgments from the receiver. The receiver sends an ACK for every packet it correctly receives. However, when the receiver detects an error in a received packet, it will continue receiving and acknowledging all subsequent packets while keeping track of erroneous packets, requesting their retransmission through NAK signals. Once the sender has finished sending all the packets in the current window, it will re-send only the ones that were not acknowledged. Thus, buffering is needed at both the sender and the receiver.
3. Unidirectional errors’ correction using diversity combining

We extend the diversity combining ARQ described in [3, 7] for the \( m(\geq 2) \)-ary unidirectional channels as follows. Let \( X \) be the transmitted codeword and \( X_i' \) be the received word at time step \( i \), where \( i \in \{0, 1, \ldots\} \). The receiver stores two words, \( Z \) and \( Z' \), for the combining purpose. Based on the type of error that occurred in \( X_i' \), the receiver updates \( Z \) and \( Z' \) as follows:

- If no error is detected in \( X_i' \) then accept it.
- In case of decreasing error:
  
  \[
  Z_i = \max(Z_{i-1}, X_i') \\
  Z_i' = Z_{i-1}'
  \]

- In case of increasing error:
  
  \[
  Z_i' = \min(Z_{i-1}', X_i') \\
  Z_i = Z_{i-1}
  \]

with initial values, \( Z_0 = (0, \ldots, 0) \) and \( Z_0' = (m - 1, \ldots, m - 1) \). The functions \( \min \) and \( \max \) are the digit-by-digit minimum and maximum operations, respectively. The receiver sends an acknowledgment to the sender only if no error is detected in either \( X_i' \), \( Z_i \) or \( Z_i' \) and sends a negative acknowledgment otherwise. Note that, the receiver only stores the current values of \( Z \) and \( Z' \) since they are overwritten at each retransmission.

**Example 3.1.** Consider a system using constant weight 7 code, with length 5 and digits over \( \mathbb{Z}_4 \). In this code, the sum of the digits of the codeword is 7. Let \( X = (3 1 0 1 2) \) be the transmitted codeword. The sequence of transmissions is given in Table 1. After four transmissions, \( Z' \) is not in error and hence \( X \) is successfully received. Note that, the system needs to determine the error type at each iteration. As can be seen
from the example, the maximum and minimum operations guarantee that, once a symbol is received correctly, the corresponding symbol in the combined word will remain unchanged during retransmission.

<table>
<thead>
<tr>
<th>i</th>
<th>$X'_i$</th>
<th>$Z_i$</th>
<th>$Z'_i$</th>
<th>error type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0 0 0</td>
<td>3 3 3 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2 1 0 1 1</td>
<td>2 1 0 1 1</td>
<td>3 3 3 3</td>
<td>decreasing</td>
</tr>
<tr>
<td>2</td>
<td>3 1 3 2</td>
<td>2 1 0 1 1</td>
<td>3 1 3 2</td>
<td>increasing</td>
</tr>
<tr>
<td>3</td>
<td>0 1 0 2</td>
<td>2 1 0 1 2</td>
<td>3 1 3 2</td>
<td>decreasing</td>
</tr>
<tr>
<td>4</td>
<td>3 2 0 1 3</td>
<td>2 1 0 1 2</td>
<td>3 1 0 1 2</td>
<td>increasing</td>
</tr>
</tbody>
</table>

Table 3.1: Diversity combining example

Applying the above scheme, a system using a $t$-UED code is capable of correcting up to $E_{\text{max}} = \lfloor \frac{t}{2} \rfloor$ unidirectional errors. This is proved in Theorem 3.2. We first introduce some necessary notations. Let $X$ and $Y$ be n-tuples over $\mathbb{Z}_m$. We define $N(X, Y)$ as the number of positions in which the digits of $X$ have higher values than those of $Y$. We say that $X$ and $Y$ are unordered if $N(X, Y) \geq 1$ and $N(Y, X) \geq 1$. Furthermore, if $N(X, Y) = 0$, we say that $Y$ covers $X$. Finally, based on our definition of $N(X, Y)$, we can define the Hamming distance between $X$ and $Y$ as $D_H(X, Y) = N(X, Y) + N(Y, X)$.

**Theorem 3.1** [4, 5, 6, 8]. A code $C$ is capable of detecting $t$ unidirectional errors if and only if, $\forall X, Y \in C$, either $X$ and $Y$ are unordered or $D_H(X, Y) \geq t + 1$.

**Theorem 3.2.** A system using the given diversity combining scheme with a $t$-UED code, $C$, is capable of correcting up to $E_{\text{max}} = \lfloor \frac{t}{2} \rfloor$ unidirectional errors.

**Proof.** We need to show that, if the number of errors in the received word is less than or equal to $E_{\text{max}} = \lfloor \frac{t}{2} \rfloor$, the receiver will be able to decide the type of error that occurred and hence update the right word ($Z$ or $Z'$). Let $X$ be the transmitted codeword and $X'$ be
the received word. If the error is of decreasing type with the number of errors being
\[ \leq \left\lfloor \frac{t}{2} \right\rfloor, \] then \( X \) covers \( X' \) and \( D_H(X, X') \leq \left\lfloor \frac{t}{2} \right\rfloor \). Now, we need to prove that there is no \( Y \in \mathcal{C} \) such that \( X' \) covers \( Y \) with \( D_H(X', Y) \leq \left\lfloor \frac{t}{2} \right\rfloor \) (i.e., the error cannot be of increasing type). Assume, for the sake of contradiction, that such \( Y \) exists. Then, \( X \) would cover \( Y \) with \( D_H(X, Y) \leq t \). However, this would contradict the requirement specified by Theorem 3.1. Thus, the receiver can correctly decide the error type if up to \( E_{max} = \left\lfloor \frac{t}{2} \right\rfloor \) unidirectional errors occurred during transmission. \( \square \)
4. **Performance Analysis**

In communication systems, *throughput* is a parameter that is commonly used to evaluate the efficiency of the system. The throughput, $\Phi$, is related to the average number of transmissions, $\bar{R}$, as follows $\Phi = \frac{k}{(n\bar{R})}$, where $k$ is the number of information symbols and $n$ is the length of the code used. In [3], it is shown that an $m(\geq 2)$-ary asymmetric channel using the diversity combining ARQ scheme has a significantly lower expected number of transmissions than one using plain ARQ scheme. Moreover, it is shown that, using diversity combining, it is possible to operate very close to optimal. In this chapter, we show that the analytical findings in [3] also apply to the $m(\geq 2)$-ary unidirectional channel.

In order to evaluate the performance of the given ARQ scheme, we designed a process to simulate the expected number of transmissions for both plain ARQ and diversity combining ARQ. We hereby briefly describe the simulation process. We simulate a system that uses an arbitrary $t$–UED code of length $n$ (where $0 \leq t \leq n$). When the number of errors in the received vector exceeds $t$, we assume no error is detected. Therefore, let $q_i = \binom{n}{i} p^i (1 - p)^{n-i}$, where $p$ is the symbol error probability, then the probability that $i$ errors are detected, $q'_i$, is computed as follows

$$q'_i = \begin{cases} 
q_i, & 1 \leq i \leq t \\
q_0 + \sum_{i=t+1}^{n} q_i = 1 - \sum_{i=1}^{t} q_i, & i = 0 
\end{cases}$$

At each iteration (transmission), a random error vector is created based on the above probability distribution with increasing and decreasing errors being equally likely to occur. The error location is independent from one iteration to the next. With Plain ARQ, the word is successfully received if and only if the generated error vector is the zero vector (i.e. no errors in the received word). Hence, the expected number of transmissions for Plain ARQ protocol is the expected number of iterations to generate the zero error vector. On the other hand, with the Diversity Combining scheme, two additional vectors (for combining
purpose) are stored, one for each error type. Thus, as explained in the previous chapter, no retransmission is needed when one of the three vectors (the error vector or the two additional vectors) is the zero error vector.

The simulation results show that the given diversity combining scheme yields a significantly lower number of transmissions and is thus more efficient than the plain ARQ scheme. Figures 4.1 through 4.9 show the plots of the average number of transmissions for plain and diversity combining ARQ schemes for $t = 4, 8$ and $n$(AUED code) and $n = 30, 40$ and $50$. As $p$ increases, the number of errors in the received word and (accordingly) the expected number of transmissions increase. However, the rate of change in the case of plain ARQ is much higher than in the diversity combining case. For $t < n$, when $p$ reaches a certain threshold level, the number of errors exceeds the error detection capability of the code, errors are no longer being detected and the average number of transmissions starts decreasing. Hence, the given diversity combining scheme gives the best performance when used with an AUED code.

Figure 4.1: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 4)$-UED code of length $n = 30$. 
Figure 4.2: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(=8)$-UED code of length $n = 30$.

Figure 4.3: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using AUED code of length $n = 30$. 
Figure 4.4: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 4)$-UED code of length $n = 40$.

Figure 4.5: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 8)$-UED code of length $n = 40$. 

Figure 4.6: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using AUED code of length $n = 40$.

Figure 4.7: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using $t(= 4)$-UED code of length $n = 50$. 
Figure 4.8: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using \( t(= 8) \)-UED code of length \( n = 50 \).

Figure 4.9: Plot of the average number of transmissions for both plain and diversity combining ARQ schemes using AUED code of length \( n = 50 \).
5. Determining Error Type for Various $t$-UED Codes

As explained earlier, for the proposed scheme to be applicable, the receiver needs not only to detect errors but also to determine their types. In this section, we will show how to determine the error type for various $t$-UED codes over the digit set $\mathbb{Z}_m$.

5.1 Optimal constant weight codes [9]

In constant weight codes, all codewords have the same weight, where the weight of a vector $X = (x_{n-1}, x_{n-2}, \ldots, x_0)$ is defined as $w(X) = \sum_{i=0}^{n-1} x_i$. These codes are capable of detecting any number of unidirectional errors. In [9], it was shown that a code $C$, such that, $\forall X \in C$, $w(X) = \left\lfloor \left(\frac{m-1}{2}\right)n \right\rfloor$, has the maximum number of unordered codewords; thus, $C$ is an optimal AUED code. Given a received erroneous word $X'$, the type of error can be determined as follows:

- $w(X') > \left\lfloor \left(\frac{m-1}{2}\right)n \right\rfloor \Rightarrow$ increasing error
- $w(X') < \left\lfloor \left(\frac{m-1}{2}\right)n \right\rfloor \Rightarrow$ decreasing error

5.2 Systematic AUED codes [10]

In this code, a codeword is constructed as follows: Given the information word $(x_{k-1}, x_{k-2}, \ldots, x_0)$, compute the check value as $C = \sum_{i=0}^{k-1} (m - 1 - x_i)$; the codeword is then $X = (x_{k-1}, \ldots, x_0, c_{r-1}, \ldots, c_0)$, where $(c_{r-1}, c_{r-2}, \ldots, c_0)$ is the representation of $C$ in radix-$m$ system. Let $X' = (x'_{k-1}, \ldots, x'_0, c'_{r-1}, \ldots, c'_0)$ be the received word and $C'$ be the value of the received check digits. Compute $C'' = \sum_{i=0}^{k-1} (m - 1 - x'_i)$. Then, there is no error in the received word if and only if $C' = C''$. This code is capable of detecting any number of unidirectional errors. The type of error can be easily determined as follows.

For increasing error, we have the following cases:

1. Errors in the information part:

   $C' = C$ and $C'' < C$ thus $C' > C''$
(2) Errors in the check part:
\[ C'' = C \text{ and } C' > C \text{ thus } C' > C'' \]

(3) Errors in both information and check parts:
\[ C' > C \text{ and } C'' < C \text{ thus } C' > C'' \]

For decreasing error, we have the following cases:

(1) Errors in the information part:
\[ C' = C \text{ and } C'' > C \text{ thus } C' < C'' \]

(2) Errors in the check part:
\[ C'' = C \text{ and } C' < C \text{ thus } C' < C'' \]

(3) Errors in both information and check parts:
\[ C' < C \text{ and } C'' > C \text{ thus } C' < C'' \]

We can thus conclude that:

- \( C' > C'' \Rightarrow \text{increasing error} \)
- \( C' < C'' \Rightarrow \text{decreasing error} \)

### 5.3 Systematic \( t \)-UED codes [4, 5]

The error detecting capability of the codes proposed in [4, 5] depends on the number of check digits, \( r \), irrespective of the number of information digits. Two classes of codes are given; with \( r = 2 \) and \( r \geq 3 \). The codes with \( r = 2 \) check digits can detect up to 2 errors and with \( r \geq 3 \) can detect up to \( mr - 2 + r - 2 \) errors. The code construction is briefly given below.

1. **Code design with \( r = 2 \) check digits:** let 
   \[ b = \left( \sum_{i=0}^{k-1} (m - 1 - x_i) \right) \mod m^2. \]
   The check digits are then the representation of \( b \) in radix-\( m \) system.

2. **Code design with \( r \geq 3 \) check digits:** let 
   \[ b = \left( \sum_{i=0}^{k-1} (m - 1 - x_i) \right) \mod m^{r-1} \]
   and let \( (c_{r-2}, c_{r-3}, \ldots, c_0) \) be the representation of \( b \) in radix-\( m \) system. Then, the \( r \)-digit check is \( (m - c_{r-2}, c_{r-2}, c_{r-3}, \ldots, c_0) \).
Depending on the number of check digits, the following subsections describe how the type of error can be identified.

5.3.1 Error type determination with \( r = 2 \) check digits

Define the syndrome \( S = \left( \sum_{i=0}^{k-1} (m - 1 - x'_{i}) - v(C') \right) \mod m^2 \), where, as before, \( x'_{i} \) and \( v(C') \) are the received information digits and the check value respectively. Since this code detects double errors then, using the given diversity combining scheme, \( E_{\text{max}} = 1 \) error can be corrected. Thus we compute the syndrome value for single digit error. Let the \( j^{\text{th}} \) digit be in error, with error magnitude \( e \in \{1, 2, \ldots, m - 1\} \), then,

For increasing error:

(1) Error in the information part \( (x'_{j} = x_{j} + e) \):

\[
S = \left( \sum_{i=0}^{k-1} (m - 1 - x'_{i}) - v(C') \right) \mod m^2 \\
= \left( \left( \sum_{i=0}^{k-1} (m - 1 - x_{i}) - e \right) - v(C) \right) \mod m^2 \\
= -e \mod m^2 = m^2 - e
\]

(2) Error in the check part:

- \( c'_{0} \) is in error \( (c'_{0} = c_{0} + e) \):

\[
S = \left( \sum_{i=0}^{k-1} (m - 1 - x'_{i}) - v(C') \right) \mod m^2 \\
= \left( \left( \sum_{i=0}^{k-1} (m - 1 - x_{i}) \right) - (v(C) + e) \right) \mod m^2 \\
= -e \mod m^2 = m^2 - e
\]
• \( c_1' \) is in error (\( c_1' = c_1 + e \)):

\[
S = \left( \sum_{i=0}^{k-1} (m - 1 - x_i') - v(C') \right) \mod m^2
\]
\[
= \left( \sum_{i=0}^{k-1} (m - 1 - x_i) - (v(C) + m e) \right) \mod m^2
\]
\[
= -m e \mod m^2 = m^2 - m e
\]

For decreasing error:

(1) Error in the information part (\( x'_j = x_j - e \)):

\[
S = \left( \sum_{i=0}^{k-1} (m - 1 - x_i') - v(C') \right) \mod m^2
\]
\[
= \left( \sum_{i=0}^{k-1} (m - 1 - x_i) + e - v(C) \right) \mod m^2
\]
\[
= e \mod m^2 = e
\]

(2) Error in the check part:

• \( c_0' \) is in error (\( c_0' = c_0 - e \)):

\[
S = \left( \sum_{i=0}^{k-1} (m - 1 - x_i') - v(C') \right) \mod m^2
\]
\[
= \left( \sum_{i=0}^{k-1} (m - 1 - x_i) - (v(C) - e) \right) \mod m^2
\]
\[
= e \mod m^2 = e
\]
• $c'_1$ is in error ($c'_1 = c_1 + e$):

$$S = \left( \sum_{i=0}^{k-1} (m - 1 - x'_i) \right) - v(C') \mod m^2$$

$$= \left( \sum_{i=0}^{k-1} (m - 1 - x_i) \right) - (v(C) - m e) \mod m^2$$

$$= m e \mod m^2 = m e$$

Thus, when $r = 2$ :

• $S \in \{m^2 - m + 1, m^2 - m + 2, \ldots, m^2 - 1\} \Rightarrow$ increasing error

• $S \in \{1, 2, \ldots, m - 1\} \Rightarrow$ decreasing error

• $S \in \{m, 2m, \ldots, (m - 1)m\} \Rightarrow c_1$ is in error, re-compute the check value to recover the codeword.

### 5.3.2 Error type determination with $r \geq 3$ check digits

From the code construction we can see that, if one of the most two significant digits is in error then the error type can be determined as follows:

• $c'_{r-1} + c'_{r-2} > m - 1 \Rightarrow$ increasing error

• $c'_{r-1} + c'_{r-2} < m - 1 \Rightarrow$ decreasing error

In further analysis, we assume that the first two significant digits of the check part are not in error. Here again, we compute the syndrome $S$ as

$$S = \left( \sum_{i=0}^{k-1} (m - 1 - x'_i) \right) - v(C') \mod m^{r-1},$$

where $v(C')$ is the value of the received check. To decide the error type, we compute the range of possible values of the syndrome in case of increasing and decreasing errors. First we give an example illustrating the basic idea.

#### Example 5.1.

Let $m = 5$ and $r = 4$. Then, $E_{\text{max}} = \left\lfloor \frac{5^4 - 2 + 2}{2} \right\rfloor = 13$. The possible syndrome values for increasing errors are $S \in \{-1 \mod 125, -2 \mod 125, \ldots, -68 \mod 125\} = \{57, 58, \ldots, 124\}$. Similarly, for decreasing error, $S \in \{1 \mod 125, 2 \mod 125, \ldots, 68 \mod 125\} = \{1, 2, \ldots, 68\}$. Thus, if $1 \leq S \leq 56$ then the error must be of decreasing type and if
69 ≤ S ≤ 124 it must be of increasing type. However, if 57 ≤ S ≤ 68, then the error type cannot be determined by the syndrome value alone. In this case, both the syndrome and the check digits are used to find the error type as explained below.

Increasing error:

(1) Errors in the information part:

\[ S \in \left[ -((m - 1)E_{\text{max}}) \mod m^{r-1}, -1 \mod m^{r-1} \right] \]

\[ i.e. \; S \in \left[ m^{r-1} - (m - 1) \left\lfloor \frac{m^{r-2} + r - 2}{2} \right\rfloor, m^{r-1} - 1 \right] \]

(2) Errors in the check part:

\[ S \in \left[ -(m^{r-2} - 1) \mod m^{r-1}, -1 \mod m^{r-1} \right] \]

\[ i.e. \; S \in \left[ m^{r-1} - m^{r-2} + 1, m^{r-1} - 1 \right] \]

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

\[ S \in \left[ -((m - 1)(E_{\text{max}} - (r - 2)) + (m^{r-2} - 1)) \mod m^{r-1}, -1 \mod m^{r-1} \right] \]

\[ i.e. \; S \in \left[ m^{r-1} - ((m - 1) \left( \left\lfloor \frac{m^{r-2} + r - 2}{2} \right\rfloor - r + 2 \right) + (m^{r-2} - 1)) \right), (m^{r-1} - 1) \]

Decreasing error:

(1) Errors in the information part:

\[ S \in \left[ 1 \mod m^{r-1}, ((m - 1)E_{\text{max}}) \mod m^{r-1} \right] \]

\[ i.e. \; S \in \left[ 1, (m - 1) \left\lfloor \frac{m^{r-2} + r - 2}{2} \right\rfloor \right] \]

(2) Errors in the check part:

\[ S \in \left[ 1 \mod m^{r-1}, (m^{r-2} - 1) \mod m^{r-1} \right] \]

\[ i.e. \; S \in \left[ 1, m^{r-2} - 1 \right] \]
Errors in both the information and check parts:

\[ S \in \left[ 1 \mod m^{r-1}, \left( (m-1) (E_{max} - (r-2)) + (m^{r-2} - 1) \right) \mod m^{r-1} \right] \]
i.e. \[ S \in \left[ 1, (m-1) \left( \left\lfloor \frac{m^{r-2}+r-2}{2} \right\rfloor - r + 2 \right) + m^{r-2} - 1 \right] \]

Let \( S_{min} = m^{r-1} - \left( (m-1) \left( \left\lfloor \frac{m^{r-2}+r-2}{2} \right\rfloor - r + 2 \right) + m^{r-2} - 1 \right) \) and \( S_{max} = (m-1) \left( \left\lfloor \frac{m^{r-2}+r-2}{2} \right\rfloor - r + 2 \right) + m^{r-2} - 1 \), it can be seen that

- \( 1 \leq S < S_{min} \Rightarrow \) decreasing error
- \( S_{max} < S \leq m^{r-1} - 1 \Rightarrow \) increasing error

When the syndrome is in the range \( S_{min} \leq S \leq S_{max} \) the error type can’t be determined by the syndrome value alone. Thus, we give a method to make that decision depending upon both the syndrome and the value of the check digits; Let \( a = \#_0(c'_{r-3}, c'_{r-4}, \ldots, c'_0) \) and \( b = \#_{m-1}(c'_{r-3}, c'_{r-4}, \ldots, c'_0) \), where \( \#_i(V) \) is the number of elements, in a vector \( V\), equal to \( i \). Now, for the error to be increasing, the minimum possible value of the syndrome, given the received check digits \( c'_{r-3}, c'_{r-4}, \ldots, c'_0 \), is

\[ P_{min} = m^{r-1} - \left( (E_{max} - (r-2-a)) (m-1) + (c'_{r-3} m^{r-3} + c'_{r-4} m^{r-4} + \cdots + c'_0 m^0) \right) \text{ Maximum possible error in info} + \text{ Maximum possible error in check}, \quad (1) \]

For the error to be decreasing, the maximum possible value of the syndrome, given the received check digits \( c'_{r-3}, c'_{r-4}, \ldots, c'_0 \), is

\[ P_{max} = m^{r-1} - \left( (E_{max} - (r-2-b)) (m-1) + (m-1 - c'_{r-3}) m^{r-3} + (m-1 - c'_{r-4}) m^{r-4} + \cdots + (m-1 - c'_0) m^0 \right) \text{ Maximum possible error in info} + \text{ Maximum possible error in check} \]

\[ = \left( E_{max} - (r-2-b) \right) (m-1) + (m^{r-2} - 1) - \left( c'_{r-3} m^{r-3} + c'_{r-4} m^{r-4} + \cdots + c'_0 m^0 \right), \quad (2) \]

Thus, the error type can be determined as follows:

- \( S \geq P_{min} \Rightarrow \) increasing error
- \( S \leq P_{max} \Rightarrow \) decreasing error
For the previous method to be correct, it suffices to show that $P_{\text{min}} > P_{\text{max}}$. From (1) and (2) we can see that

$$P_{\text{min}} - P_{\text{max}} = m^{r-1} - (2E_{\text{max}} - 2(r - 2) + (a + b))(m - 1) - m^{r-2} + 1$$

But since $a + b \leq r - 2$, then

$$P_{\text{min}} - P_{\text{max}} \geq m^{r-1} - (2E_{\text{max}} - (r - 2))(m - 1) - m^{r-2} + 1$$
$$\geq m^{r-1} - (m^{r-2} + r - 2 - (r - 2))(m - 1) - m^{r-2} + 1$$
$$\geq 1$$

Note that this error-type determination technique is also applicable to the Bose-Lin code [6], and is more straightforward than the one given in [7].

To summarize the results:

- $1 < S < S_{\text{min}} \Rightarrow$ decreasing error
- $S > S_{\text{max}} \Rightarrow$ increasing error
- $S_{\text{min}} \leq S \leq S_{\text{max}}$, compute $P_{\text{min}}$ and $P_{\text{max}}$:
  - $S \leq P_{\text{max}} \Rightarrow$ decreasing error
  - $S \geq P_{\text{min}} \Rightarrow$ increasing error

**Example 5.2.** Consider the received word $X'$ over the radix $m = 5$ with $r = 5$ check digits. Suppose the value of the syndrome is $S = 310$, and that the check digits $c_2$ and $c_1$ are 2 and 1, respectively. Then, $E_{\text{max}} = \left\lfloor \frac{5^3 + 3}{2} \right\rfloor = 64$ is the maximum number of errors that can be corrected by the given scheme. Now, to determine the type of error we note that, in case the received word suffered an increasing error, the minimum possible value of the syndrome occurs when the two check digits $c_2$ and $c_1$ are in error and 62 other information digits are also in error (with maximum possible error magnitude), i.e.
\[ P_{\text{min}} = 5^4 - (62 \times 4 + 2 \times 25 + 1 \times 5) = 322. \]

On the other hand, if the received word suffered a decreasing error, the maximum possible value of the syndrome occurs when the two check digits \( c_2 \) and \( c_1 \) are in error and 62 other information digits are also in error (with maximum possible error magnitude), i.e.

\[ P_{\text{max}} = 62 \times 4 + (4 - 2) \times 25 + (4 - 1) \times 5) = 313. \]

Since the syndrome value is 310, we can conclude that the error that occurred is of decreasing type.
6. Conclusion

A diversity combining ARQ for the $m (\geq 2)$-ary unidirectional channel is presented. The main idea of the proposed scheme is to keep two combined words, one for each type of error (increasing/decreasing). The combining is done such that, once a symbol is received correctly it is stored in a combined word and is kept unchanged throughout the retransmission process. Thus, the receiver eventually receives the transmitted word correctly. This scheme is expected to reduce the number of retransmissions and hence to increase the channel’s throughput. Moreover, if applied with the hybrid ARQ protocol, we expect the system to be operating optimally as it is the case for asymmetric channels [3]. In addition, we presented techniques for determining the type of error that occurred during transmission. This is an essential step in the given diversity combining scheme.
BIBLIOGRAPHY