Error Detection and Correction over Two-Dimensional and Two-Diagonal Model and Five-Dimensional Model

Danial Aflakian 1 , Dr.Tamanna Siddiqui 2, Najeeb Ahmad Khan 3 ,Davoud Aflakian 4
Department of Computer Science
E-mails: a_danial85@yahoo.com, ja_zu_siddiqui@hotmail.com, nakhan@jamiahamdard.ac.in , davoud2007@yahoo.com

***Abstract*— In this research paper we discover two different schemes of error detection and correction which are based on parity-check code for optimal error detection and correction utilizing check bits without degrading the data rate too much.**
The scheme of the first method is obtained by arranging data in rows, to form a square, and then parity bits are added. The second scheme builds upon the first, by forming a square cube and the parity scheme is extended to this shape.

Keywords-component; Data Communication; Error detection and correction; Linear Block Coding; Parity check code.

I. Introduction

For transmission purposes, all signals consist of digital packets. To protect against data corruption, extra bits are added for error detection and correction. Corruption of data can occur due to the degradation of the signal due to noise or other impairments during transmission of signal from transmitter to receiver. Error detection [1] involves additional bits added to data at the transmitter end and sent as part of the packet. On reception, if the error scheme detects the data received is incorrect, then the receiver asks the transmitter to retransmit the data. In error correction, not only does the receiver detect the error, but is also able to recover correct data depending on the level of error correction implemented, i.e. the number of bits incorrect from which the error correction scheme is able to recover the correct data. The greater the level of error correction, the more bits required to be added to the data. There are two types of codes, the non-systematic and the systematic. A non-systematic code is one which accomplishes multiple error detection and correction through construction of a directory or catalog of word entries.

The first systematic error correction code was invented by W. Hamming [2]. There are many error correction schemes. One of the schemes involves dividing the data into blocks of k bits, called datawords, and adding r redundant bits. The resultant n bit blocks (consisting of k + r bits each) are called codewords [3]. The method considered here consists of adding a single bit, known as a parity bit, to make either the number of 0's even or the number of 1's even (we consider making the number of 1's even in this article).

This scheme detects only an odd number of error bits. On reception, this scheme counts the number of 1's in the block. If it is even, it considers the signal to be correct. If an even number of bits have been flipped during transmission, i.e. an even number of bits are incorrect on reception, this scheme does not detect an error.

The above method is called Simple parity check code, there are some methods which are based on simple parity check code known as Two-dimensional parity-check code, N-dimensional parity check code, three-dimensional parity-check code(S-codes families)[4] ,Multi-dimensional parity-check code theory[5], and Three and Four-dimensional Parity-check codes[6]. In Two-dimensional parity-check code, codewords are arranged separate rows to form a rectangle. Then, parity checking is also done for each column of the rectangle.

Four-dimensional parity-check code improves on Two-dimensional parity-check code by adding parity bits for the diagonal lines, as shown in figure 1, of the square or rectangle. For more information please refer to [6].

We studied four-dimensional and S-code and found that both are the same in the number of errors detected and corrected, but S-code's parity bits are more reliable and it has a better code transmission rate due to S-bits. We also discovered that S-code is more suited to a square shape, as this gives this scheme more reliability in the parity bits and makes it less complex.

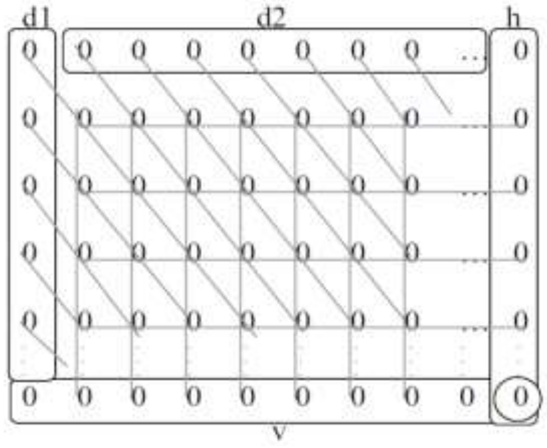

Figure 1, Four dimensional parity check code[6].

II. Proposed Work

In this publication we improve on 4-dimensional parity-check code, which was published on 2004 which is final related work . We discovered two methods of improvements:

Case 1: Two-dimensional and Two-diagonal parity-check codes.

Case 2: Five-dimensional parity-check code.

A. *Case 1: Two-dimensional and Two-diagonal parity-check code:*

In this scheme, we add parity bits for the diagonals of the square to have the same bit rate of 4-dimensional parity-check code. This method involves five sequential steps, as follows:

- First Step: The top parity row is constructed as follows (see figure 2). Each parity bit in the top row is a parity bit for a fixed number of bits, the count of which is decided by the main diagonal. The diagonals being considered to construct the top parity row are those oriented from the top left to the bottom right. The main diagonal is taken from the top left corner to the bottom right corner (shown as a line in black in figure 2, with its parity bit also shown as black). Subsequent parity bits consist of two portions. The first portion consists of the diagonal below the parity bit, and the second portion consists of furthest diagonal below the main diagonal not used yet, e.g. the second parity bit is shown in yellow and its two portions are also shown as yellow lines in figure 2. Similarly each parity bit is shown in a different colour and its two portions are shown in the same colour in figure 2.

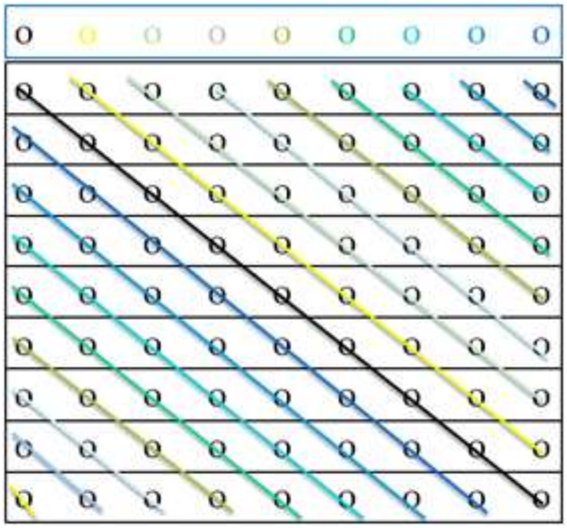

Figure 2, Top parity row.

- Second Step: The parity bits for the left column of parity bits are formed using the same method as step one, but consist of diagonals formed from bottom left to top right (as shown in figure 3).
- Third Step: The parity bits for the right column of parity bits are formed by taking each bit as the parity bit for its row of bits.
- Fourth Step: The parity bits for the bottom row of parity bits are formed by taking each bit as the parity bit for its column of bits.
- Fifth Step: Four bits are coloured red and inside red squares in figure 3. We use these bits as parity bits for others parity bits formed in steps one to four. The top right corner bit is the parity bit for the top row of parity bits. The bottom left corner bit is the parity bit for the left column of parity bits .The bottom right corner bit is the parity bit for the bottom row of bits, including the corner bit obtained in the previous calculation .The top left corner bit is the parity bit for the right column of parity bits, including the top right corner and bottom right corner bits. This scheme enables detection and correction of all errors in top parity row and left parity column. It detects odd number of errors in bottom parity row and right parity column. It also detects errors in the first three parity bit corners in red. The last calculated corner red bit (top left corner) has no parity bit associated with it and is critical in our error detection and correction scheme. One solution involves transmitting this top left corner parity bit thrice to ensure that there is no error in receiving it during transmission.

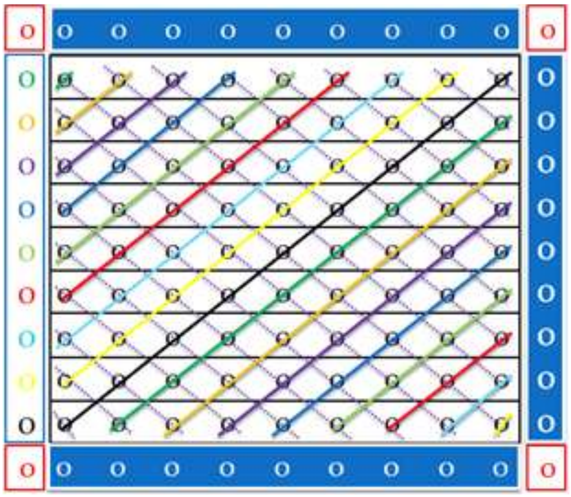

Figure.3: Two-dimensional and Two-diagonal parity-check code.

This scheme can detect 7 errors with 100% insurance (i.e. even if there is an error in parity bits). If there are 8 errors on vertices of a regular octagon, then the errors go undetected (as shown in figure 4). This scheme also corrects up to 4 errors. This error correction can also be improved upon by using the trial and error method of [7].

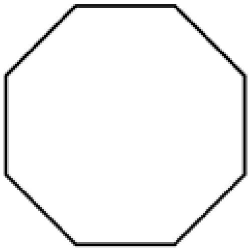

Figure 4, Undetectable 8-bit error pattern.

B. Case 2: Five-dimensional parit- check code

In this scheme a cube of data bits are formed. Then parity bits are present on five sides of the cube utilizing the following steps:

- First Step: The cube of data bits is divided into planes of squares. For each square, parity bits are formed using the scheme of Two-dimensional and Two-diagonal parity-check code.
- Second Step: In first step we obtained parity bit planes for four sides of the data cube. The fifth plane of parity bits is obtained as follows. We consider the forward facing face of cube (the only other remaining face is the back face), as seen in Figure 5. Each bit in this parity plane is calculated as the parity bit for the depth bits, except for the left column and top row. The left column parity bits are obtained by considering each bit the parity bit for its row in the parity plane. Similarly, each parity bit in the top row is obtained by considering it the parity bit of its column in the parity plane.
- Third Step: The common bit for the left column and top row is named as the share bit here. The share bit is got by taking the parity of its row, column and depth.

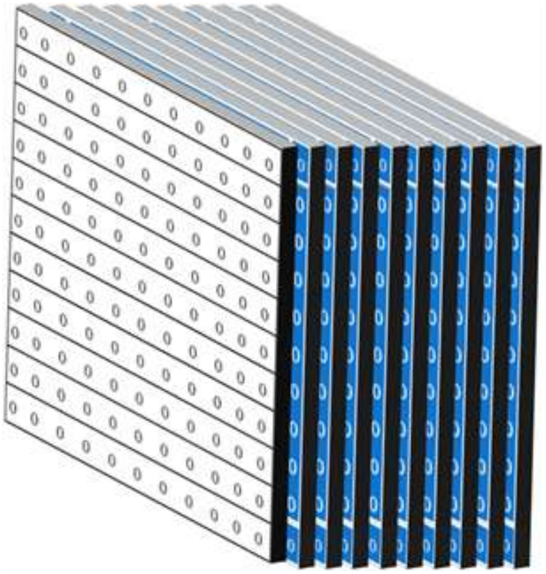

Figure 5, Five-dimensional parity-check code.

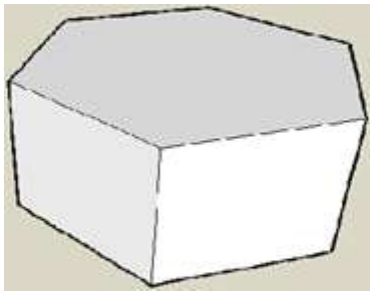

Figure 6,Undetecteble 16-bit error pattern.

In this scheme, errors will go undetected if the errors form the shape of a regular octagonal prism (shown in figure 6). The probability of this happening is quiet low.

III. Analysis

The advantage of proposed schemes can be demonstrated through overhead and code rate calculation.

A. Case1 overhead and code rate:

In case 1, the scheme uses data formed into a square, say of side of length L. Then:

Number of information bits= L^2

Number of parity check codes=L+L+L+L+4=4L+4

Total number of bits (information bits +parity bits)
$=L^2+4L+4$

$$\text{Overhead} = \frac{\text{No. of parity bits}}{\text{No. of information bits}} = \frac{4L+4}{L^2}$$

$$\text{Code Rate} = \frac{\text{No. of information bits}}{\text{Total number of bits}} = \frac{L^2}{L^2+4L+4}$$

If we compare four-dimensional parity-check code of [6] with case 1, we find that both have the same overhead and code rate, but our scheme makes a notable improvement in error detection and correction on information bits as well as parity bits.

B. Case2 overhead and code rate

This scheme uses a cube of data bits, say of side L. Then:

Number of information bits= $L*L*L=L^3$

Number of parity bits=(L+L+L+L)*L+4L+(L+1)*(L+1)
. $=4L^2+4L+L^2+1+2L=5L^2+6L+1$

Total number of bits (information bits +parity bits)
$= L^3+5L^2+6L+1$

$$\text{Overhead} = \frac{\text{No. of parity bits}}{\text{No. of information bits}} = \frac{5L^2+6L+1}{L^3}$$

$$\text{Code Rate} = \frac{\text{No. of information bits}}{\text{Total number of bits}} = \frac{L^3}{L^3+5L^2+6L+1}$$

Overhead and code rate is worse than previous methods, but error detection and correction is better.

C. Diagram of both cases

. In Figure 7, the vertical axis shows code rate and horizontal axis shows number of information bits. we replaced

L with some numbers to give comparisons. In figure 7, red indicates Two-dimensional parity-check code, blue indicates Four-dimensional parity-check code, green shows case 1, and violet shows case 2. We have shown case 2 separately, since it is not possible to compare its bit rates with others.

IV. Conclusion

In the Two-dimensional and Two-diagonal parity-check scheme we increase the number of errors detected and corrected by four-dimensional parity-check [6] by two, while maintaining the same overhead and code rates. The errors detected and corrected can be improved further by the trial and error method of [7]. Also it has a better detection on right column of parity bits and bottom row of parity bits. Errors on left column of parity bits and top row of parity bits are possible to correct.

For Five-dimensional parity-check code, up to 15 errors can be detected with the same level of error correction as the previous method. It is possible to improve on the error detection and correction by using methods similar to trial and error method of [7].

References

[1] Shu Lin Daniel J. Costello Jr, “Error Control Coding: Fundamentals and Applications”. Prentice Hall. ISBN 0-13-283796-X. , 1983.

[2] R. W. Hamming, “Error detecting and error correcting codes”, Bell System Tech. J. 29 , 1950 ,pp. 147-160.

[3] Behrouz Forouzan, “Data communications and networking” ISBN-13:978-0-07-063414-5,2006.

[4] Morris Rubinoff, “N-Dimensional Codes for Detecting and Correcting Multiple Errors”,Communications of the ACM , Dec. 1961 .

[5] Tan F.Wong and John M.Shea, “Multidimensional parity- check codes for bursty channels”, Proceedings of the IEEE international Washington, DC , USA, June. 2001.

[6] Naveen Babu Anne, Utthaman Thirunavukkarasu, Dr. Shahram Latifi,“Three and Four-dimensional Parity-check Codes” Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’04) 0-7695-2108-8/04, pp. 267-282, 2004.

[7] Ashby, W. R,“Design for a Brain”. Chapman & Hall (1960: Second Edition

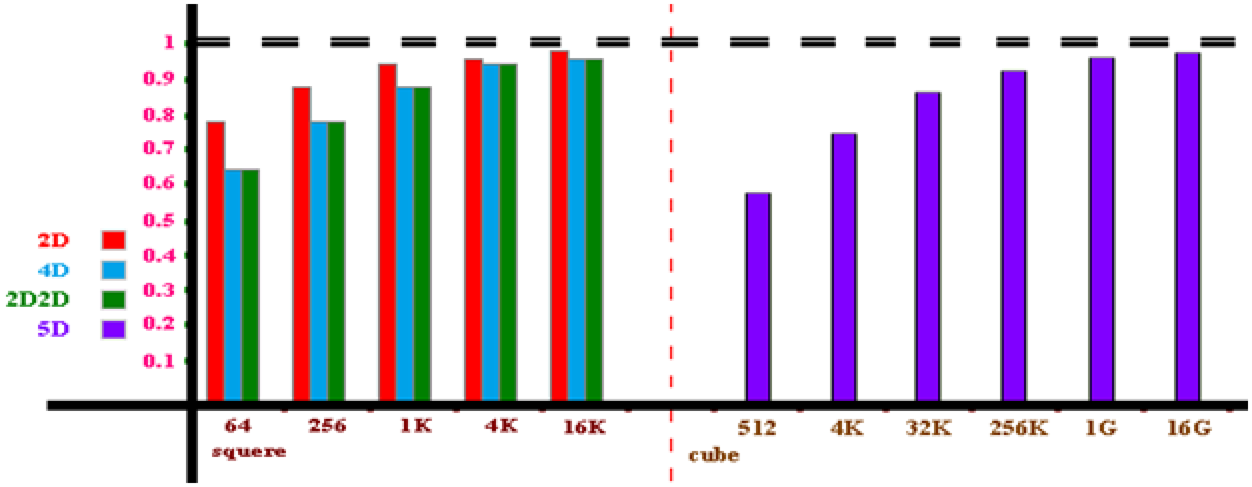

Figure 7, Code rate diagram for two proposed work schemes as well as Two and Four-dimensional parity-check codes.