Analysis of Different Error Correction Codes

Mayank Wadhawan, Student, UFID - 59148122

Abstract— **Wireless links are more susceptible to noise/errors than wired links, which makes Error Detection Codes (EDC) and Error Correction Codes (ECC) crucial for reliable communication. EDC are used to determine if an error was introduced during transmission. If the data is corrupted, the sender retransmits the data. Whereas, ECC can recover corrupt data even when there is no back channel. Error Correction Codes have many applications, ranging from Internet, CDMA, and 3G to more advances wireless communication systems like Satellite Communication and Deep Space Communication. In this report, I have analyzed different error detecting codes and error correction codes. Additionally, I have plotted Bit Error Rate vs Signal-to-noise Ratio for various ECC. Furthermore, I have mentioned applications of different ECC and their shortcomings.**

Index Terms—**error correction codes, error detection codes. Bit error rate, signal-to-noise ratio, repetition code, checksum, parity bits, automatic repeat request, low density parity check, turbo code, reed Solomon code, convolutional code, additive white Gaussian noise.**

I. Introduction

When data is transmitted over a wired or wireless link, errors might be introduced because of channel noise. Error detection codes are used to check if errors occurred during transmission. In section II, I have briefly explained a few error detection codes like Parity Bits, Repetition Code, Checksum and Cyclic Redundancy Checks. Parity Bits is one of the most simple error detection codes. In this method, a parity bit is added to a 7 bit binary code, which results into 8 bit binary code. In Repetition code, we divide data streams into blocks. Each block is sent a certain number of times. In Checksum, we compute the checksum of data using a checksum function. This checksum is sent along with the data. The receiver separates the data and the checksum. Then it uses the same checksum function to determine the checksum. This new checksum is compared to the checksum received. If they are different, then the data received is corrupt. Cyclic Redundancy Checks is useful for finding burst errors [1].

Error correction codes help the receiver to recover the corrupt data even when there is no back channel. Sender sends redundant data along with the original message, the receiver can use this redundant data to recover the errors. In section IV, I have briefly explained a few error correction codes like Reed-Solomon code, Low Density Parity Check Code (LDPC Code), Turbo Code and BCH Codes. Reed-Solomon error correction code is capable of correcting burst errors. For LDPC codes, noise threshold is close to Shannon limit. In a channel, reliable communication is only possible at certain rates which are less than or equal to channel capacity. If they are more than channel capacity, then the communication will be unreliable. Turbo codes get their name from the feedback loop used during decoding. It is similar to exhaust feedback in engine turbocharge. The main advantage of BCH codes is that they are very easy to decode and can be decoded by low powered hardware.

II. Error Detection Codes

A few error detection codes are mentioned below.

A. Parity Bits

Error detection using Parity bits can be done using two ways. Even parity and Odd parity. Refer to Table 1.
1. In even parity, we determine total number of 1's in the binary code. If it's odd, we append 1 as a parity bit. If it's even, we append 0 as parity bit.
2. In odd parity, we determine total number of 1's in the binary code. If it's even, we append 1 as a parity bit. If it's odd, we append 0 as parity bit.

TABLE 1: Parity Bit Error Detection

7 bit code	Even Parity		Odd Parity	
	Parity Bit	8 bit code	Parity Bit	8 bit code
1011010	0	10110100	1	10110101
1010010	1	10100101	0	10100100
0100101	1	01001011	0	01001010
1111000	0	11110000	1	11110001
0011100	1	00111001	0	00111000

Parity Bits can detect errors in a noisy channel very quickly. One disadvantages of Parity Bits method is that it can only detect odd number of bit flips. That is, it can detect only when number of but flips are 1,3,5,7 … etc. Additionally, Parity Bits cannot determine all forms of errors.

B. Repetition Code

This is a simple error detection code. In (4,1) repetition code, the blocks are sent 4 times. This method therefore is very inefficient because data is sent multiple times.
For example in this case, 011 is sent like 011 011 011 011. If any of the block is different from the other, then there is an error. At the receiver, we can decode the signal by majority vote. In this method, if there is a bit flip in the same place for all the blocks, then this method would not be able to detect the error. Advantage of this method is that it is very simple.

C. Checksum

Two of the commonly used checksum algorithms are mentioned below.

1. Parity Word – In this method, first the data is divided into words with certain number of bits (n). Then we compute XOR of all these words. This is the checksum. This is then appended in the message and transmitted. Receiver extracts the data from the received message. Then it divides the message into words of n bits each. Then we compute the checksum of performing XOR on these words. This is then compared to the checksum received. If they are different, then the message received is corrupt. In this method, if there is an error which affects even number of bits in the same position of different words, then it will not be detected.

2. Modular Sum – This method is similar to the above method except that we add all the numbers in binary form.

D. Cyclic Redundancy Checks

This method can be implemented in hardware and therefore is used in storage devises. We use a generator polynomial to perform long division on data. In this method, the generator polynomial is the divisor, the data is dividend and remainder is used to find errors. This remainder is sent with the data. Receiver uses same method to find the remainder. This new remainder is compared with remainder received to determine if there is an error.

E. Using Cryptographic Hash Functions

We can use cryptographic hash functions to determine the integrity of the data. Cryptographic hash value of a data is unique. Probability of collision by using these functions is very less. We can use same cryptographic hash function at sender and receiver to find errors.

III. Error Correction using ARQ

A. Automatic Repeat Request (ARQ)

This is an error correction method which needs a back channel to be implemented. That is receiver can send a message back to the sender. This method makes use of error detection method, acknowledgement and negative acknowledgement. Receiver uses error detection method to determine if the received data is corrupt. If the data is corrupt, then it sends negative acknowledgement. Otherwise, it sends positive acknowledgement. Whenever there is a timeout, then the sender retransmits the data. ARQ is of 3 types. Stop and wait, Go back N and Selective Repeat. ARQ is most notably used in Internet because of internet's varying capacity. There is latency when we use this technique because of back channel communication [9].

1. Stop and Wait – In this method, sender sends a frame to receiver one at a time. When receiver receives a correct frame, it sends positive acknowledgement (ACK) to the sender. If the sender does not receive ACK, then timeout occurs and it retransmits the same data. This method is very inefficient because frames are sent one at a time.

2. Go Back N – In this method, the window size of sender is N and window size of receiver is 1. The sender will send all the frames in its window one by one. The receiver will check every frame for error and sends ACK if the frame is valid. If receiver receives incorrect frame or any frame out of order, then it send ACK with id of last correct received frame. The sender will then send all the frames from that id. This is why this method is called Go back N. This method is more efficient that Stop and Wait because more packets are being sent. However, frames will have to be sent multiple times when received data is corrupt or out of order.

3. Selective Repeat – In this method, the window size of sender and receiver is N. This allows receiver to buffer out of order frames. The receiver sends individual acknowledgement of received frames. The sender will only retransmit frames which have timed out. All the following frames don't have to be retransmitted. Therefore, it performs better than Go Back N.

IV. Error correction codes

Error Correction Codes are used in broadcasting and in storage devised like hard disk, CD ROM etc. Error Correction codes are of 2 types. Convolutional Codes and Block Codes.

A. Block Codes

1. Reed-Solomon

Error correction using these codes is done by adding redundant data to the data message. Then this is sent through the communication channel. The receiver then decodes the data by using Reed-Solomon decoder. Please refer figure 1.

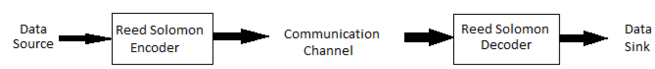

Figure 1: Reed-Solomon Encoding and Decoding

A Reed-Solomon code word of n bits contains k bit of data and 2t bits of parity data can detect 2t error symbols and correct t error symbols. Refer to figure 2.

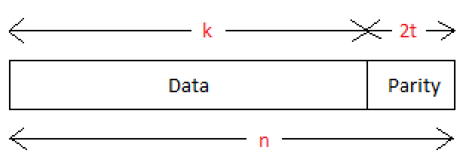

Figure 2: Reed-Solomon Code word

Reed-Solomon codes have the advantage that they are easy to code. They are extensively used in data storage disks like compact disk and DVD. They are also used in bar codes because we can get correct data even if a part of bar code is damaged. They are also used in space transmissions. However, they are being replaced by turbo codes [8].

The Matlab file ReedSolomon.m shows encoding and decoding data using Reed Solomon code. I have determined the BER for the SNR = 10 (Figure 3).

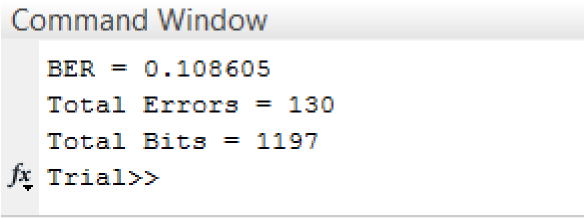

Figure 3: ReedSolomon.m output

Figure 4 shows performance of Reed Solomon code in Additive White Gaussian Noise channel. The Matlab file is ReedSolomonChart.m.

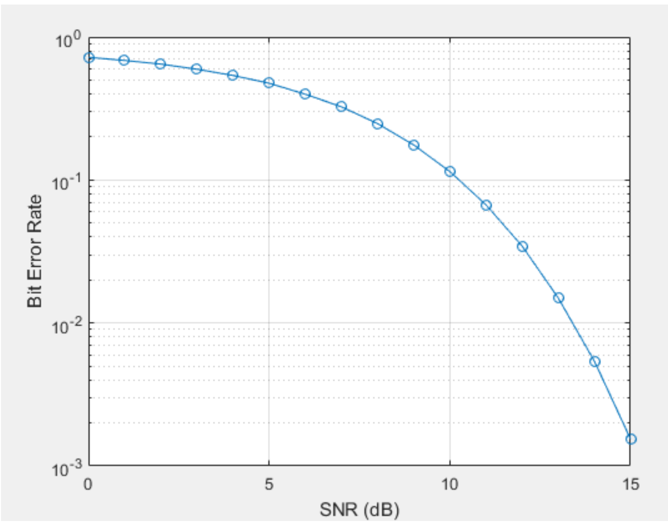

Figure 4: BER vs SNR for Reed Solomon code

2. Low Density Parity Check Code (LDPC Code)

LDPC codes are made using sparse bipartite graphs. Refer to Figure 5[3].

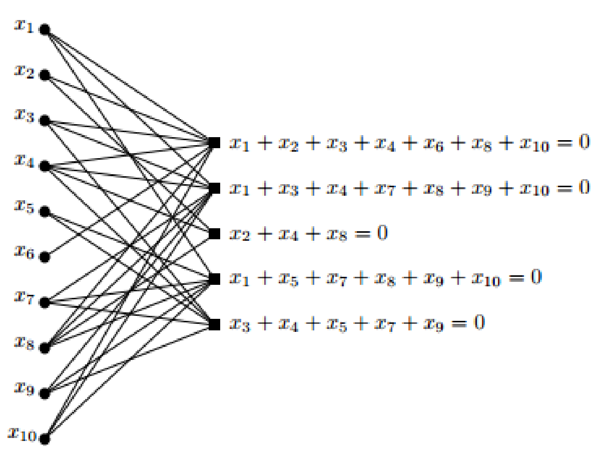

Figure 5: An example of LDPC Code

LDPC decoders have iterative algorithms [2]. That is, in each iteration message is sent from message nodes to check nodes and then from check nodes to message nodes. LDPC codes have advantage that their encoders and decoders are very fast. They have applications in space communication.
The file LDPC.m shows encoding and decoding using LDPC code. The LDPC BER for SNR = 0.81 is shown in Figure 6.

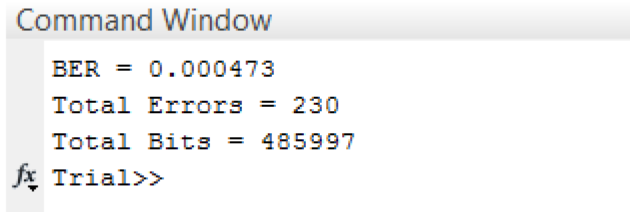

Figure 6: LDPC.m output

Figure 7 shows BER vs SNR for LDPC code in AWGN channel. The Matlab file is LDPCchart.m.

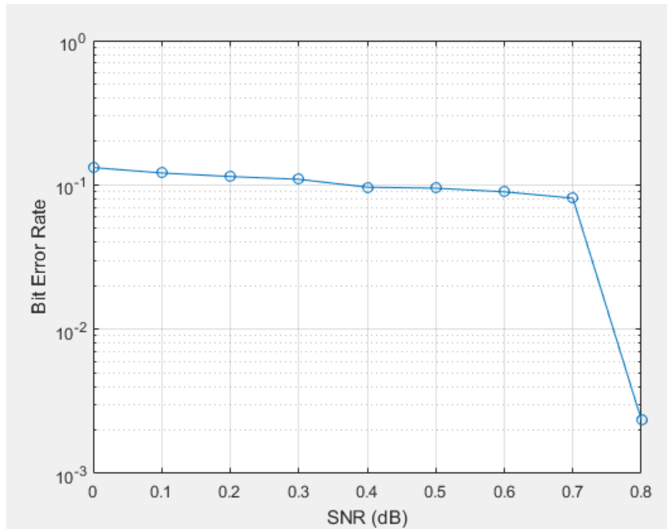

Figure 7: BER vs SNR for LDPC Code

3. Turbo Code

Turbo Codes have performance similar to LDPC codes (close to Shannon limit). These codes are currently competing with LDPC codes in high end applications. These codes have applications in 3G and 4G mobile [5]. They are also being used for satellite communications. Interleavers are a part of turbo codes and gives them good performance (Figure 8). However, this makes the decoding slow.

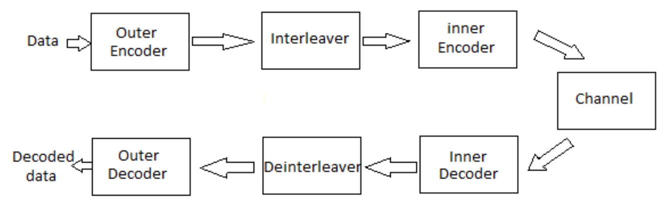

Figure 8: Turbo code encoding and decoding

TurboCode.m contains Matlab code for encoding and decoding using turbo code. The output is shown in Figure 9.

```
Command Window
BER = 0.005001
Total Errors = 96
Total Bits = 19197
fx Trial>>
```

Figure 9: TurboCode.m output

Figure 10 shows the turbo code BER performance in AWGN channel. The Matlab file is turboperf.m.

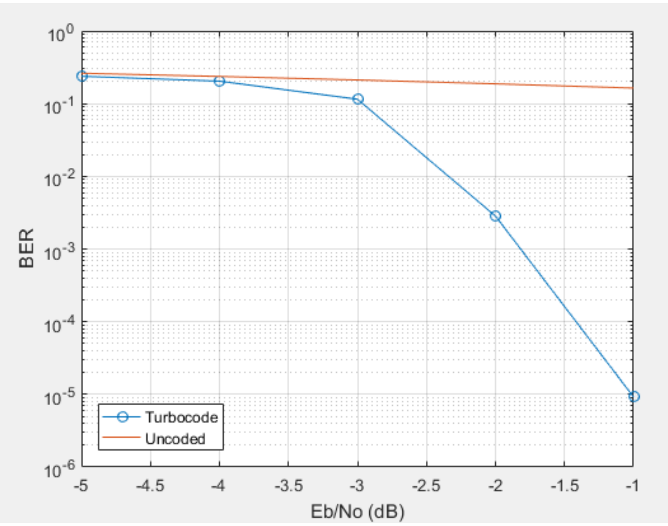

Figure 10: BER vs Eb/No for Turbo code and un-coded signal

4. BCH Codes

These codes have many applications in storage devices like CD, DVD and SSD. They are also used in satellite communication and 2 dimensional bar codes. They have very simple design. We can control how many symbols can be corrected initially during design.

The file named BCH.m shows encoding and decoding using BER code. The output shows BER for SNR = 10 for BCH code (Figure 11).

```
Command Window
BER = 0.010645
Total Errors = 17
Total Bits = 1597
fx Trial>>
```

Figure 11: BCH.m output

Figure 12 shows BER vs SNR for BCH code in AWGN channel.

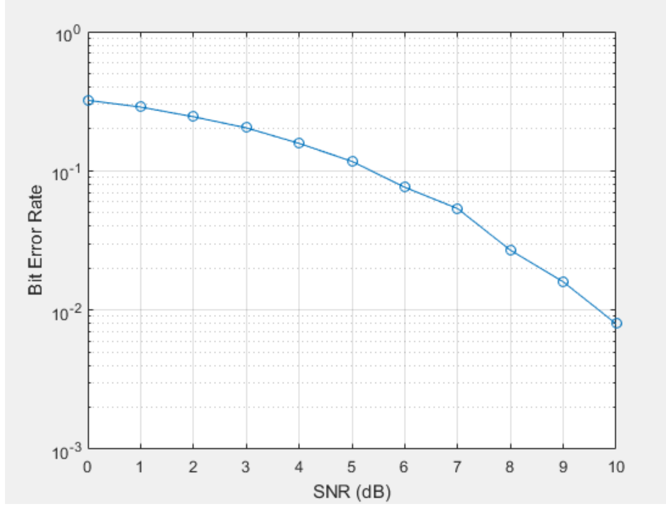

Figure 12: BER vs SNR for BCH code

B. Convolutional Codes

These codes can work on symbols of any length. It doesn't have to be fixed length like block codes. Data encoded using convolutional codes can be soft decoded using Viterbi Algorithm. This a Dynamic Programming algorithm which has applications in speech recognition, GSM, CDMA, satellite communication and bioinformatics. Convolutional codes can be implemented in hardware.

I have written a Matlab file (Viterbi.m) which shows encoding using Convolutional code and decoding using Viterbi decoder. The Figure 14 shows the output of that file.

```
Command Window
BER = 0.504188
Total Errors = 301
Total Bits = 597
fx Trial>>
```

Figure 14: Viterbi.m output

V. Related Work

There are research papers comparing a few ECC for specific applications.

[4] explains LDPC for hardware implementation

[8] describes a way to improve decoding of Reed-Solom codes

[9] shows how ARQ is used for wireless transmission

[15] explains use of Turbo codes and LDPC codes for deep space communications

These research papers explain ECC in specific situations. This makes it hard to compare these codes in more general form.

VI. Applications

There are many applications of Error Correction codes in the context of wireless communications. I have compiled a list of some wireless links and mentioned which error correction codes are suitable for them. Refer to Table 2.

TABLE 2: FEC codes for various wireless links

Applications	ARQ	Reed Solomon	LDPC	Turbo Code	BCH Code
Radio(2 way)	Yes	No	No	No	No
Satellite TV	No	Yes	Yes	Yes	Yes
Bluetooth	Yes	No	No	No	No
CDMA, GSM	Yes	Yes	Yes	Yes	Yes
GPRS, 3G, 4G	No	Yes	Yes	Yes	Yes
Satellite Communication	No	Yes	Yes	Yes	Yes
Deep space communication	No	Yes	Yes	Yes	Yes

VII. CONCLUSION

In the previous sections, I have described and analyzed many widely used error detecting codes and error correcting codes. It would be very unreasonable to conclude that a particular Error Correcting code would work optimally for all applications. The use of an ECC depends on factors like latency, channel noise, type of errors, simplicity, performance and hardware involved.

ARQ is widely used in network protocols for reliable communication. Reed-Solomon code and BCH Code are widely used for storage devices like CD ROM, DVD and SSD. BCH codes are used in 2-D bar codes. LDPC code, Turbo codes and Reed-Solomon code are used for satellite and deep space communication. Turbo codes are competing with LDPC codes for high end applications.

REFERENCES

[1] https://en.wikipedia.org/wiki/Error_detection_and_correction
[2] Low density parity check codes over GF(q). M. C. Davey ; Cavendish Lab., Cambridge Univ., UK ; D. J. C. MacKay
[3] https://www.ics.uci.edu/~welling/teaching/ICS279/LPCD.pdf
[4] LDPC code construction with flexible hardware implementation. D. E. Hocevar ; DSP Solutions R&D Center, Dallas, TX, USA
[5] Performance Comparison Between Turbo Code and Rate-Compatible LDPC Code for Evolved Utra Downlink OFDM Radio Access. Naoto Ohkubo; Nobuhiko Miki; Yoshihisa Kishiyama; Kenichi Higuchi; Mamoru Sawahashi
[6] Turbo-codes and high spectral efficiency modulation. S. Le Goff ; France Telecom Univ., Brest, France ; A. Glavieux ; C. Berrou
[7] Turbo Codes Extended with Outer BCH Code. Jakob Dahl Andersen
[8] Improved decoding of Reed-Solomon and algebraic-geometric codes. V. Guruswami ; Lab. for Comput. Sci., MIT, Cambridge, MA, USA ; M. Sudan
[9] A novel ARQ scheme applied to wireless communication. Peng Yu ; Institute of Zhengzhou Information Science and Technology, Zhengzhou, Henan Province, China, 450002 ; Xiaomei Wang ; Hongyi Yu
[10] Algebraic soft-decision decoding of Reed-Solomon codes. R. Koetter ; Coordinated Sci. Lab., Univ. of Illinois, Urbana, IL, USA ; A. Vardy
[11] Wireless image transmission using multiple-description-based concatenated codes. Daniel G. Sachs ; Anand Raghavan ; Kannan Ramchandran
[12] A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels. Shu Lin ; Texas A&M Univ., College Station, TX, USA ; P. Yu
[13] Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel. M. Bystrom ; Center for Telecommun. & Inf. Networking, Drexel Univ., Philadelphia, PA, USA ; J. W. Modestino
[14] Power control for the additive white Gaussian noise channel under channel estimation errors. T. E. Klein ; Lucent Technol., Murray Hill, NJ, USA ; R. G. Gallager
[15] The Development of Turbo and LDPC Codes for Deep-Space Applications. Kenneth S. Andrews ; California Inst. of Technol., Pasadena ; Dariush Divsalar ; Sam Dolinar ; Jon Hamkins more authors
[16] http://www.mathworks.com/