Analysis of Error Detection and Correction Codes

Mayank Wadhawan, Student, UFID - 59148122

***Abstract*— Error detection and correction can be performed by adding extra data to the message. This extra data can be used to detect if there is an error in the message. In this project, I will analyze and discuss various error detection techniques like Checksums, Parity Bits, and Repetition Codes. I will also analyze and discuss various error correction strategies like Automatic repeat request (ARQ) and Hybrid schemes. Additionally, I will conduct performance analysis based on power consumption, transmitted power and information throughput of different error correction codes like low density parity check code (LDPC) and turbo code for wireless networks. LDPC codes can be used for transmitting message over a noisy channel. They are capacity approaching codes i.e. noise threshold can be set close to theoretical maximum. Turbo codes are forward error correction codes with performance close to Shannon theoretical limit.**

I. Introduction

Error detection and correction techniques are required to transmit data reliably over unreliable communication channel. Errors are introduced because of channel noise during transmission from sender to receiver. These errors can be detected using various error detection techniques. In Checksum method, checksum of original data is calculated and is transmitted along with the data. Receiver calculates the new checksum and compares it with existing checksum. If they are different, then there is an error. In parity bits method, a parity bit is added to a group of bits to ensure that number of bits with value 1 is even or odd. This method detects odd number of errors. Even number of changed bits will make the data appear correct even when it's not. In repetition codes, each block of code is transmitted certain number of times. If the first block of code is different compared to other block of code, then there is an error. Error detection and correction can be systematic or non-systematic. In systematic error detection, transmitter sends the original data along with parity bits. Parity bits are determined using an algorithm. Receiver applies same algorithm to received data bits and then compares it with parity bits. If they are not same, then there is error. In non-systemic scheme, the original message is transformed into encoded message. Error correction can be performed in two ways, Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). In ARQ, error detection techniques are used to determine if there is an error. If error is detected in data, then erroneous data is retransmitted. ARQ uses acknowledgements (or negative acknowledgements) and timeouts to transmit data reliably. There are three types of ARQ protocols. They are stop-and-wait ARQ, go-back-N ARQ and selective repeat ARQ. In FEC, parity data is added to the message. Receiver can correct the errors without asking the sender to re-transmit the data. This is useful in broadcasting data from one source to multiple destinations. Wired networks are more reliable than wireless networks and are not susceptible to these many errors.

II. Project plan

A. Collecting relevant information

I will read reputed journals, research papers and textbooks for information pertaining to this project. I will read information related to various error detection codes like Checksums, Parity Bits, and Repetition Codes and different error correction codes like automatic repeat request, forward error correction and hybrid schemes. I will also gain deeper understanding of error correction codes such as LDPC, turbo code and Reed Solomon codes.
I will finish this task by 20th March.

B. Implementation Stage

(1) I will complete the implementation for error detection codes like checksum, parity bits and repetition codes.
I will finish this by 25th March

(2) I will implement different decoders and compare their performance till 5th April.

(3) Determine best error correcting code depending on requirements of the wireless network.
This will be done by 10th April.

C. Result Analysis

This stage involves analyses of results determined from implementation stage. I will determine why different codes work better for certain situations.
This must be completed by 15th April.

D. Final Changes

I will determine if any final changes are required in the project. If there are contradiction with known concepts, then I will analyze the results determine the reason behind it.
I will finish this by 19th April.

E. Final Report and Demo video creation

In this stage, I will create the final report, powerpoint presentation and create a video explaining the approach used and final results.
I will complete this stage by 25th April.

ACKNOWLEDGMENT

I would like to thank Dr. Dapeng Oliver Wu for his help in selecting project topic and for information provided on his website.

REFERENCES

[1] https://en.wikipedia.org/wiki/Error_detection_and_correction.
[2] Hoang Anh Ngo. Hybrid Automatic-Repeat-reQuest Systems for Cooperative Wireless Communications.
[3] R. Puri ; Dept. of Electr. Eng. & Comput. Sci., California Univ., Berkeley, CA, USA ; K. Ramchandran. Multiple description source coding using forward error correction codes.
[4] Jinghu Chen ; Dept. of Electr. Eng., Univ. of Hawaii, Honolulu, HI, USA ; A. Dholakia ; E. Eleftheriou ; M. P. C. Fossorier more authors. Reduced-Complexity Decoding of LDPC Codes.
[5] Sheryl L. Howard, Christian Schlegel, and Kris Iniewski. Error control coding in low-power wireless sensor networks: when is ECC energy-efficient?
[6] L. Li ; Sch. of ECS, Univ. of Southampton, Southampton, UK ; R. G. Maunder ; B. M. Al-Hashimi ; L. Hanzo. A Low-Complexity Turbo Decoder Architecture for Energy-Efficient Wireless Sensor Networks.
[7] Ravi Motwani, Zion Kwok, Scott Nelson. Low Density Parity Check (LDPC) Codes and the Need for Stronger ECC.