Analysing the Efficiency and Performance of Hamming Code in High-Speed Data Networks

# 1. Introduction

## 1.1 Background

Hamming Code is a widely used error detection and correction technique in digital communication and data storage systems. Developed by Richard Hamming in 1950, this code provides a reliable method for detecting and correcting errors that may occur during data transmission or storage. This report aims to provide an overview of the Hamming Code, its principles, applications, and advantages.

## 1.2 Objectives

The primary objectives of this study are:

1. Evaluate the efficiency and performance of Hamming code in terms of error detection.

2. Analyse the computational overhead introduced by Hamming code in high-speed data networks.

3. Discussing an improved version of Hamming code for burst error detection and correction.

## 2. Hamming Code Fundamentals

Hamming Code, a widely used error correction technique, employs a systematic approach for both encoding and decoding processes. During encoding, redundant parity bits are strategically inserted into the data stream to create a codeword, expanding the original information bits. These parity bits are carefully calculated to facilitate error detection and correction during the decoding phase. The position of each parity bit is determined by its binary representation, and each parity bit is responsible for checking specific combinations of information bits. This redundancy enables the detection and correction of single-bit errors.

In the decoding process, the received codeword is compared with the expected codeword, and a syndrome is computed by XORing the two. A non-zero syndrome indicates the presence of an error, and the position of the error is identified based on the syndrome's value. The erroneous bit is then flipped to correct the error. Hamming Code's ability to both detect and correct errors provides a robust mechanism for ensuring data integrity in applications such as telecommunications and computer memory systems.

Top of Form

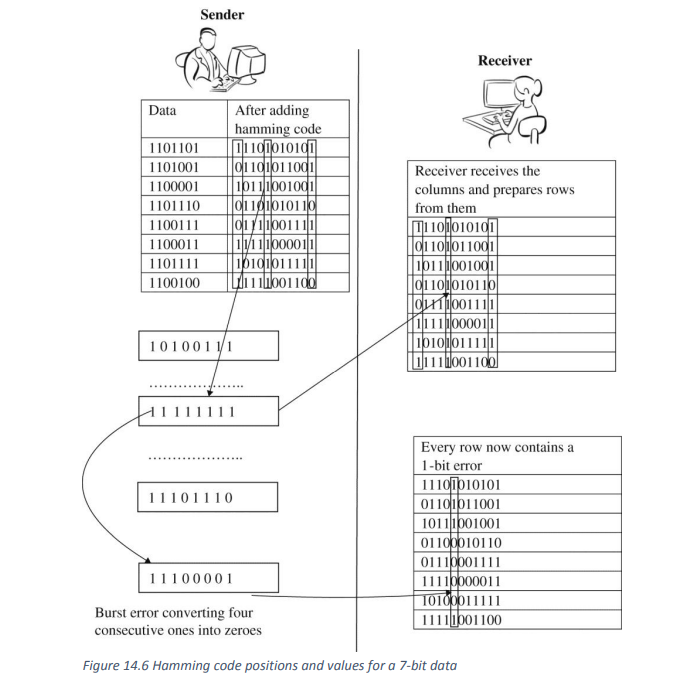
# 3. Implementing Hamming Code in High-Speed Data Networks

High-speed networks demand robust error detection and correction mechanisms to ensure the reliability of data transmission. Hamming Code, with its capability for both error detection and correction, has found extensive application in high-speed networks. This report delves into the implementation of Hamming Code in such networks, focusing on its significance, benefits, and challenges.

In high-speed networks, data is transmitted at rapid rates, increasing the likelihood of errors due to noise, interference, or signal degradation. Reliable error detection and correction mechanisms are essential to maintain data integrity, prevent data loss, and ensure the accurate delivery of information. Hamming Code's ability to identify and correct errors in real-time makes it a crucial tool for enhancing the performance of high-speed networks.

# 4. Improvement in Hamming Code

We have seen that data traffic is more likely to encounter the burst errors and thus we need a solution to handle burst errors. With a simple trick, the very Hamming code can be used to catch and correct burst errors. Data bytes are converted to hamming code and thus every 7-bit entity is converted to 11 bit and placed one below another. We accumulate 8 such rows and pick up the first column (of length 8 bits) and send it across. We send 11 such bytes across. A burst error might corrupt major portion of one of the bytes. The receiver rearranges the received bytes in columnar fashion and thus generate the same 11- bit entities which the sender has generated. As we have a corrupted column, most of the rows now have a one-bit error. Hamming code corrects that error and the problem are solved! Only if the burst error is longer than one byte or occurs in multiple bytes, the Hamming code fails to correct. Many researchers worked and many additional solutions are proposed.



# Comparing the error detection and correction performance with different types of errors and data length

A graph of data size and numbers

Description automatically generated with medium confidence

A graph with numbers and a line

Description automatically generated with medium confidence

# Comparing Hamming code and Improved hamming code correction efficiency in different types of error

A graph with a line and text

Description automatically generated with medium confidence