

# Novel Algorithm to Construct QC-LDPC Codes for High Data Rate Applications

Bhuvaneshwari P. Vairaperumal, Tharini Chandrapragasam\*

Department of Electronics and Communication Engineering B.S. Abdur Rahman Crescent Institute of Science & Technology, Vandalur, Chennai, India

E-mail: pvbunaperumal@gmail.com, tharini@crescent.education

\*Corresponding author

**Keywords:** low density Parity check (LDPC) codes, quasi-cyclic low density parity check (QC-LDPC) codes, cloud, storage overhead, software defined radio (SDR), AWGN channel

**Received:** June 9, 2023

*A novel algorithm to construct highly sparse, quasi-cyclic low-density parity check codes with large girth and high code rates that can be employed in high data rate applications is proposed. In this paper, a sparse girth six base matrix is designed, which is then substituted by a difference exponent matrix derived from a basic exponent matrix based on the powers of a primitive element in a finite field  $F_q$ , to build varying size high code rate QC-LDPC codes with girth six. The proposed exponent matrix generation is a one-time procedure and hence, a smaller number of computations is involved. Experimental evaluation of proposed QC-LDPC codes in S3 cloud storage system showed faster encoding-decoding speeds, better storage efficiency and reduced storage overhead compared to conventional LDPC codes, traditional RS codes and conventional QC-LDPC codes. Simulation results showed that the QC-LDPC codes constructed using the proposed algorithm performed very well over AWGN channel. Hardware implementation of the proposed high-rate QC-LDPC code ( $N = 1248$ ,  $R = 0.9$ ) in Software Defined Radio platform using the NI USRP 2920 hardware device displays very low bit error rates compared to conventional QC-LDPC codes and conventional LDPC codes of similar size and rate. Thus, from both the simulation and hardware implementation results, the proposed QC-LDPC codes with high code rate were found to be suitable for high data rate applications such as cloud data storage systems and 5G wireless communication systems.*

*Povzetek: Predstavljen je nov algoritem za ustvarjanje redkih QC-LDPC kod z veliko hitrostjo kodiranja, učinkovitih v aplikacijah z velikim pretokom podatkov.*

## 1 Introduction

Businesses employ cloud computing to gain access to premium IT solutions without the need to invest in expensive and space-consuming hardware. Such large businesses constantly deal with massive amounts of data, referred to as Big Data. With the integration of 5G technology, Cloud Service Providers (CSPs) have the potential to offer more efficient solutions for managing Big Data challenges. Employing a 5G network combined with cloud computing helps to deal with transferring such large quantities of data faster in real-time. Both cloud storage systems and 5G technology are high-data-rate applications that require an efficient coding mechanism capable of adapting to these high data rates while providing robust error correction capabilities and improved storage performance.

Traditional replication techniques [1-2] enhances data availability in cloud data storage systems by reducing latency and ensuring fast data recovery. However, it comes with the drawback of high storage overhead, which subsequently increases the overall storage costs.

Erasure coding techniques can achieve both high data reliability and low storage costs, as they do not generate multiple copies of data. Consequently, the storage overhead with erasure codes is much lower compared to traditional replication techniques. Erasure codes have therefore been employed to store data in cloud storage systems such as Google Cloud [3], Microsoft Azure [4], and Facebook [5]. Typical examples of erasure coding techniques employed in cloud and distributed storage systems are Reed Solomon codes (RS) [6-8], Cauchy Reed Solomon Codes (CRS) [10], Local reconstruction codes (LRC) [4,11] and Low-density parity check (LDPC) codes [12]. Despite their advantages, current erasure codes possess certain drawbacks in data storage systems, particularly in the cloud, which has led to ongoing research in search of improved coding schemes.

Reed Solomon (RS) codes are the most commonly employed erasure codes in Cloud data storage systems [8]. RS codes perform encoding in Galois Field  $GF(2^w)$ , which involves the usage of a complex Vandermonde matrix that acts as the generator matrix. In the encoding process, log and antilog tables are used to convert

multiplication into addition, and then addition into XOR operations over  $GF(2^w)$ . Since, the complexity of Vandermonde matrix inversion is very high, which is of the order of  $O(n^3)$ , the computational overhead is considerably very high. The high recovery and storage costs of RS-based cloud data storage systems motivate many researchers to propose new techniques. And still research is going on.

Cauchy Reed Solomon (CRS) codes are a variation of Reed-Solomon codes, which convert the Galois Field multiplications into simple XOR operations. With an  $(n,k)$  CRS code,  $k$  data blocks are encoded into  $m$  parity blocks and it can tolerate any  $m$  block lost without any data loss. However, the challenge arises when the requirement specifies that the  $k$  data blocks and  $m$  parity blocks must be stored on separate nodes within cloud storage servers. Otherwise, the failure of one node can result in multiple failures. Further, Cauchy Reed Solomon (CRS) Code [10], uses simple Cauchy matrix in place of the complex Vandermonde matrix. This helped in the reduction of computation complexity from  $O(n^3)$  to  $O(n^2)$ . The inverse matrix computation of the Cauchy matrix over Galois Field is of the order of  $O(n^2)$ , which incurs increased computational overhead.

The Minimal Storage Regenerating (MSR) code [13] falls within the category of regenerating codes designed to minimize repair bandwidth and enhance storage efficiency. While it has proven effective in reducing recovery overhead, it comes with a significantly higher computational cost compared to traditional RS-based erasure codes. To address these drawbacks, simpler XOR-based erasure codes with lower computational overhead, known as Low-density parity check (LDPC) codes, have been introduced. Some major applications of LDPC codes in the field of data storage systems are Wide Area Storage Network [14], Cloud running [15], P2P distributed storage systems [16], and distributed storage systems [17-19].

Conventional LDPC codes [20] designed by R.Gallager, are randomly constructed codes which show high error correction capabilities and are applied to many digital communication standards. Mackay constructed LDPC codes, whose error performances reached the Shannon's channel capacity limits for long codelengths [21]. Hence, LDPC codes were adopted in a wide range of applications such as satellite broadcasting, optical communications, wireless communications and high density storage systems. However, conventional LDPC codes are non-systematic unstructured codes due to which they have higher computational complexity and slower encoding-decoding speeds. Randomly constructed LDPC code structures increases the complexity of the decoder implementation [33]. It is because a random interconnection pattern between the variable nodes and the check nodes in the tanner graph of the code will introduce a more complex wire routing circuit on the hardware. Consequently, to mitigate the complexity associated with decoding and enhance the iterative

decoding performance, structured LDPC codes, referred to as Quasi Cyclic-LDPC codes, have been introduced.

Kim et al. [22] presented a novel design of Quasi-Cyclic LDPC codes employing cyclic shift matrices, which were faster in terms of encoding and decoding processes and required less memory for storage. Such QC-LDPC codes are being considered for cloud data storage systems.

Well-designed QC-LDPC codes with high level of sparsity and large girth (girth 6) combined together have the potential to address the primary challenges faced by cloud data storage systems. Furthermore, applications with high data rates like cloud computing and 5G technologies necessitate the use of high code rate codes to accommodate their speeds. Thus, a novel algorithm to construct high code rate Quasi Cyclic LDPC codes with a girth of atleast six from a low code rate and highly sparse base matrix is proposed. The girth six sparse base matrix along with an efficient exponent matrix that can eliminate the short cycles or girth four improves the decoding performance of QC-LDPC codes. Therefore, a difference exponent matrix derived from a basic exponent matrix built based on the powers of a primitive element in a finite field  $F_q$  is proposed in this paper, which can directly eliminate the girth 4 in parity check matrix. Simulation results show that the proposed high code rate QC-LDPC codes possess better error correction capabilities compared to conventional QC-LDPC codes and conventional LDPC codes using BPSK modulation scheme over AWGN channel. Table 1. presents the summary of the related works and their contributions. The proposed high code rate QC-LDPC code exhibits reduced storage overhead, increased storage efficiency, and increased speed of encoding and decoding operations, making it most suitable for application in cloud data storage systems and high data rate wireless communication systems as well.

The rest of the paper is organized as follows: An introduction to Quasi-cyclic codes is provided in Section 2. Section 3. presents a novel algorithm for constructing high rate QC-LDPC codes using a girth six base matrix of low code rate and a difference exponent matrix, constructed based on the powers of a primitive element in a finite field  $F_q$ . Section 4 focuses on the application of the proposed QC-LDPC codes in S3 Cloud storage system. Section 5. presents the simulation results which includes the speed of encoding-decoding operations of the proposed codes as well as key metrics related to storage performance, including storage overhead, storage efficiency, and decoding overhead. In Section 6, the application of proposed codes in wireless communication system and their comparative BER performance analysis with conventional LDPC and QC-LDPC codes over AWGN channel is presented. Section 6. Also presents the hardware implementation of proposed QC-LDPC codes in Software Defined Radio, employing the NI USRP 2920 hardware device, and includes comparisons of their BER performance. Section 7. provides the conclusion.

Table 1: Summary of related works and contributions

S/N	Author	Title	Methodology	Result
1	LAN et al.: 2007	Quasi-Cyclic LDPC Codes For AWGN And Binary Erasure Channels	Based on the multiplicative group of GF, an array of circulant permutation CPM and zero matrices are combined to construct QC-LDPC codes	Short QC-LDPC code with code rate of 0.75, a minimum distance of at least 31 and girth six property shows BER of $10^{-6}$ at about 3.8dB
2	SONG et al. (2009)	A Unified Approach To The Construction Of Binary And Non binary Quasi-Cyclic LDPC Codes Based On Finite Fields	To construct sparse parity check matrix (PCM) of QC-LDPC code with girth six.	Large QC-LDPC codes with R=0.86 having girth of atleast six shows good error performance in AWGN channel with BPSK modulation and SPA decoding with 50 iterations
3	S.Noor (2010)	Construction and Performance Evaluation of QC-LDPC Codes over Finite Fields	Short LDPC codes based on finite fields and array dispersion techniques. Most importantly, LDPC codes are constructed using multiplicative groups based on Finite fields.	LDPC codes are constructed using multiplicative groups based on finite fields shows a very good performance in the waterfall region.
4	G. Zhang	Type-II QC-LDPC Codes From Multiplicative Subgroup of Prime Field	A novel class of type-II QC-LDPC codes with girth six, and very small circulant sizes using Chinese Remainder Theorem (CRT) achieves theoretical lower bound.	Good decoding performance using the iterative decoding with sum-product algorithm (SPA) with over 50 iterations.
5	Sheng Huang 2016	Construction method of QC-LDPC codes based on multiplicative group of finite field in optical communication.	A novel method to construct QC-LDPC code by using the multiplicative group of finite field and with good distance property.	It provides higher code rate QC-LDPC codes and good error correction capability is achieved in 30 iterations using BPSK modulation over AWGN channel

## 2 Introduction to Quasi-Cyclic low density parity check codes

Quasi Cyclic LDPC codes [23] are a special class of structured LDPC codes composed of cyclic permutation sub-matrices or null matrices. The H matrix of QC-LDPC code as shown in eqn (1) is divided into a number of square ‘P’ sub-matrices, where each sub-matrix is either a null matrix of size (PxP) or a circular permutation matrix of size (PxP).

$$H = \begin{bmatrix} I(P_{1,1}) & I(P_{1,1}) & \dots & I(P_{1,n}) \\ I(P_{2,1}) & I(P_{2,1}) & \dots & I(P_{2,n}) \\ \dots & \dots & \dots & \dots \\ I(P_{m,1}) & I(P_{m,2}) & \dots & I(P_{m,n}) \end{bmatrix} \quad (1)$$

, where  $I(P_{i,j})$  represents the (PxP) cyclic permutation matrix (CPM) obtained after cyclically shifting each row of an identity matrix to the right and I (0) represent an all zero matrix of size (PxP).

The parameter P is known as the lift factor or the expansion factor. If P=3, then I (0), I (1) and I (2) is given by

$$I(0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$I(2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad (3)$$

The position of one’s in H matrix of QC-LDPC code is determined by an exponent matrix or a shift permutation matrix.

The exponent matrix E(H) or the shift permutation matrix is constructed based on new algorithms.

$$E(H) = \begin{bmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,n} \\ p_{2,1} & p_{2,2} & \dots & p_{2,n} \\ \dots & \dots & \dots & \dots \\ p_{m,1} & p_{m,2} & \dots & p_{m,n} \end{bmatrix} \quad (4)$$

The code rate of an (n,k) QC-LDPC code in general is computed by R

$$R = \frac{k}{n} \quad (5)$$

, where n represent the total number of blocks and k represent the number of data blocks.

Although high data rate applications demand high code rate codes, it is not necessary to construct high code rate QC-LDPC codes directly. Instead, following specific puncturing procedures, a base matrix with a low code rate can generate high code rate QC-LDPC codes.

### 2.1 Tanner graph

The parity check matrix, H of LDPC codes, is usually represented using a Tanner graph [25]. A tanner graph, also known as a bipartite graph, which is composed of variable nodes and check nodes connected via an edge. An edge appears if and only if there exists a 1 in the corresponding location in the H matrix. A cycle in a Tanner graph is defined as the sequence of check nodes and variable nodes connected in a cyclic manner, which starts and ends at the same node in the tanner graph. The girth of a Tanner graph, or an LDPC code, is defined as the length of the shortest cycle within it. In this context, the shortest cycle in a Tanner graph is referred to as a Cycle 4 or Girth 4, as illustrated in Fig .1.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

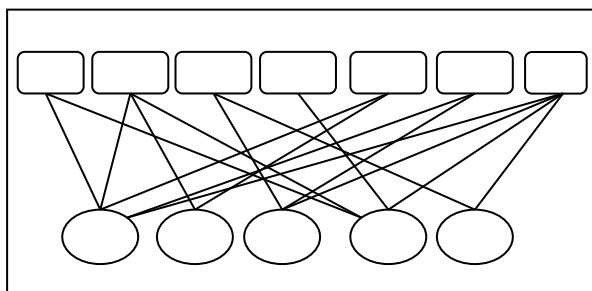


Figure 1: Tanner graph of H matrix given in (6)

It is well known that short cycles or girth four, affects the decoding performance of LDPC codes using the message passing- iterative decoding algorithm. Hence, QC-LDPC codes with girth six are required to be constructed for iterative decoding MP algorithm to perform well [25, 31, 33]. Furthermore, there exist both systematic and non-systematic QC-LDPC codes. In non-systematic codes, the parity blocks are integrated within

the codeword. In contrast, systematic QC-LDPC codes separate the data blocks and parity blocks, making it easier to extract data directly from the codeword. Systematic QC-LDPC codes are particularly advantageous in cloud storage systems because they enable data retrieval without the need for decoding. This leads to reduced computational complexity and enhances the speed of coding and decoding operations. In this paper, a novel algorithm to construct the parity check matrix ‘H’ of QC-LDPC code with high code rate {R ≥ 0.9} and a girth of at least six for cloud data storage systems is proposed. The upcoming section provides a comprehensive explanation of the proposed algorithm.

### 3 Proposed algorithm to construct high code rate QC-LDPC codes for cloud storage systems

This paper proposes a novel algorithm to construct the parity check matrix, H matrix of QC-LDPC codes with a high code rate and a girth of atleast six. The proposed algorithm comprises of two main steps: a) Construction of highly sparse low code rate base matrix H<sub>b</sub> with girth six. b) Construction of an improved difference exponent matrix P<sub>de</sub> that helps to build large sized QC-LDPC codes with high code rate, R ≥ 0.9 and girth six. The exponent matrices are constructed only once and can be stored in local machine unlike the conventional LDPC code, which requires to store the entire H matrix.

The proposed QC-LDPC code with size (M x N) is takes the following form:

$$H = [H_i \ H_p] \quad (7)$$

, where H<sub>i</sub> is the circular permutation matrix constructed using the proposed algorithm and H<sub>p</sub> is the dual diagonal identity parity matrix of (M x M) size, where M is the number of rows and N is the number of columns of the parity check matrix.

In order to construct H<sub>i</sub>, the information part of the proposed parity check matrix, H of QC-LDPC codes, a sparse base matrix, H<sub>b</sub> with girth six and an improved exponent matrix, called as difference exponent matrix P<sub>d</sub> derived from E(H) constructed based on the powers of a primitive element in a finite field F<sub>q</sub> are proposed.

#### 3.1 Proposed algorithm for constructing sparse base matrix H<sub>b</sub> with girth six

The core of QC-LDPC encoding lies in the generation of a sparse base matrix, H<sub>b</sub> with a girth of at least six. The elimination of girth 4 from the base matrix decreases the encoding and decoding complexity to a large extent and improves the BER performance. In this section, sparse base matrix H<sub>b</sub> with girth of atleast six is generated using the proposed algorithm and explained in the following steps:

**Step 1:** Define the code rate R<sub>b</sub> of the random binary matrix to be low, R<sub>b</sub>= 0.5.

**Step 2:** Specify the required number of columns,  $n$  such that  $n$  is always less than 100 (i.e  $n < 100$ ) and the required number of rows ‘ $m$ ’ of a small random binary matrix (B), such that

$$n > m \tag{8}$$

**Step 3:** Set the column weight ( $c_w$ ) of the B matrix as 3 or more.

$$c_w \geq 3 \tag{9}$$

The condition  $c_w \geq 3$  must be satisfied so that later puncturing can be performed to increase the code rate and sparsity level of the PCM.

**Step 4:** Perform puncturing on the rows of the random binary matrix (B) by searching for rows with large weights ( $r_w$ ) and remove the rows with row weight ( $r_w$ ) having an odd value starting from 9.....∞.

$$r_w = \begin{cases} \text{odd values} \geq 9, 11, \dots, \infty \\ \neq 1, 3, 5, 7 \end{cases} \tag{10}$$

This step increases the code rate and sparsity levels of the random binary matrix B

**Step 5:** Test for the presence of short cycles (cycle 4) in the B matrix obtained from step 4. If cycle 4 exists, flip the one’s which forms the cycle 4 in such a way that, the resultant matrix contains zero cycle 4 or otherwise known as girth six base matrix,  $H_b$ . Fig.2 shows how to flip one’s that forms the cycle 4.

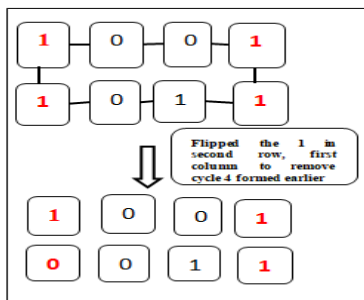


Figure 2: Removal of cycle 4 by flipping one’s that forms the short cycle

The base matrix with girth six constructed using steps 1 to 5 is known as the proposed base matrix,  $H_b$  with  $(n \times m)$  size, where  $n$  represent the number of columns of  $H_b$  matrix and  $m$  represent the number of rows of the  $H_b$  matrix. The generated  $H_b$  base matrix from section 3.1 is combined with the novel difference exponent matrix  $P_{de}$  with a lift factor,  $P$  given in section 3.2 to construct  $H_i$  matrix and the required  $H$  matrix.

### 3.2 Improved method of constructing the exponent matrix E(H) and the proposed difference exponent matrix $P_{de}$

This section provides a method of constructing an exponent matrix based on powers of an element of finite field  $F_q$  as mentioned in [26].The combination of the proposed base matrix and the exponent matrix generated produces a large number of cycle 4’s. To combat this negative effect, a difference exponent matrix is

proposed that can reduce the large number of cycle 4’s (in 1000’s) to single digits, that can be easily flipped to obtain girth six QC-LDPC codes and is explained as follows:

**Step 1.** Let  $F_q$  be a finite field with  $q$  as a prime power [26]. Assume ‘ $\alpha$ ’ as the primitive element of the finite field  $F_q$  and ‘ $r$ ’ is the largest prime factor of  $q-1$ , where  $q-1$  is cr. If  $\beta = \alpha^c$ , the order of  $\beta$  is  $r$  and this set forms a cyclic subgroup of the multiplicative group of  $F_q$ , subgroup,  $G_r = \{1, \beta, \beta^2, \dots, \beta^{r-1}\}$  [26].

**Step 2.** Construct the cyclic permutation matrices based on the cyclic subgroup  $G_r$  of the multiplicative group of  $F_q$ , where powers of  $\beta$  is taken in modulo  $r$  [26].

**Step 3.** The generalized exponent matrix E(H) matrix [26] is of the following form:

$$E(H) = \begin{bmatrix} 1 & 1 & 1 & \dots & \dots & 1 \\ 1 & \beta & \beta^2 & \dots & \dots & \beta^{r-1} \\ \dots & \beta^2 & (\beta^2)^2 & \dots & \dots & (\beta^2)^{r-1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & \beta^{r-1} & \beta^{(r-1)^2} & \dots & \dots & (\beta^{r-1})^{r-1} \end{bmatrix} \tag{11}$$

Example: Exponent matrix E(H) for the finite field  $F_{16}$ , with  $q=16$  is given in (12). The factor  $16-1=3 \times 5$  is obtained. Assuming  $r=5$  and  $c=3$ , the constituents of the multiplicative subgroup are  $\{\beta^0, \beta^1, \beta^2, \dots, \beta^4\}$ , where  $\beta$  is a prime number. The exponent matrix E(H) [26] is then calculated by

$$E(H) = \begin{bmatrix} 1 & 1 & 1 & \dots & \dots & 1 \\ 1 & \beta & \beta^2 & \dots & \dots & \beta^4 \\ 1 & \beta^2 & \beta^4 & \dots & \dots & \beta^3 \\ 1 & \beta^3 & \beta^1 & \dots & \beta^4 & \beta^2 \\ 1 & \beta^4 & \beta^3 & \beta^2 & \beta & \dots \end{bmatrix} \tag{12}$$

**Step 4.** Substitute the values of  $r$  and  $\beta$  in E(H) matrix to obtain the associated P matrix(exponent matrix) after performing modulo  $r$  of each entry  $\beta, \beta^2, \dots$  etc.

$$P = \begin{bmatrix} 1 & 1 & 1 & \dots & \dots & 1 \\ 1 & \beta & \beta^2 & \dots & \dots & \beta^4 \\ 1 & \beta^2 & \beta^4 & \dots & \dots & \beta^3 \\ 1 & \beta^3 & \beta^1 & \dots & \beta^4 & \beta^2 \\ 1 & \beta^4 & \beta^3 & \beta^2 & \beta & \dots \end{bmatrix} \text{mod } r \tag{13}$$

Example: exponent matrix after substituting  $r=5$  and  $\beta = 3$  in (11) we obtain the exponent matrix, P matrix as follows:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 6 & 9 & 12 \\ 0 & 6 & 12 & 3 & 9 \\ 0 & 9 & 3 & 12 & 6 \\ 0 & 12 & 9 & 6 & 3 \end{bmatrix} \tag{14}$$

, where  $\beta^0=1, \beta^1 = 1, \beta^2 = 2 \times 3 = 6, \beta^3 = 3 \times 3 = 9$  and  $\beta^4 = 4 \times 3 = 12$ .

For each of the given values in P matrix, the cyclic permuted copies of the base matrix ‘ $H_b$ ’ are substituted. And check for presence of girth 4 is conducted after substitution. The results showed a large number of cycle 4’s present. Hence, a new approach to get rid of all the cycle 4’s by computing the difference exponent matrix  $P_{de}$  is presented in this paper.

The proposed difference matrix  $P_{de}$ , is obtained by finding the difference between the 1st row and 2nd row of the exponent matrix P, excluding the first column and row values.

**Step 5.** Find the difference between the first and second rows of P matrix from step (4) in section 3.2 and exclude the first column and rows to get the difference exponent matrix,  $P_d$  which is of the form

$$P_d = \begin{bmatrix} p(2,1) - p(1,1) & \dots & p(2,n) - p(1,n) \\ \dots & \dots & \dots \\ p(m,1) - p(m-1,1) & \dots & p(m,n) - p(m-1,n-1) \end{bmatrix} \quad (15)$$

Example: Eqn. (13) after performing step 5. becomes

$$P_d = \begin{bmatrix} -3 & -6 & 6 & 3 \\ -3 & 9 & -9 & 3 \\ -3 & -6 & 6 & 3 \end{bmatrix} \quad (16)$$

**Step 6.** If the rows are identical, then any one of the rows is removed.

Example: The eqn (16) has two identical rows, hence one of them must be cancelled to get the new difference matrix as shown in (16)

$$P_d = \begin{bmatrix} -3 & -6 & 6 & 3 \\ -3 & 9 & -9 & 3 \end{bmatrix} \quad (17)$$

**Step 7.** Replace the negative integer values by zero. If no negative values are present, insert new zero matrices in between to increase the good distance property.

$$P_d = \begin{bmatrix} 0 & 0 & 6 & 3 \\ 0 & 9 & 0 & 3 \end{bmatrix} \quad (18)$$

**Step 8.** If two similar integer values are present in close locations, interchange the values such that they are separated by large distance to obtain the proposed exponent matrix,  $P_{de}$ .

Example: Interchanging the values in the first and the second rows of  $P_d$  matrix, as shown in Fig 3., gives the required difference exponent matrix  $P_{de}$ .

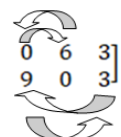
$$P_d = \begin{bmatrix} 0 & 6 & 0 & 3 \\ 0 & 9 & 0 & 3 \end{bmatrix}$$


Figure 3. Interchanging the values in  $P_d$  matrix

$$P_{de} = \begin{bmatrix} 0 & 6 & 0 & 3 \\ 0 & 3 & 0 & 9 \end{bmatrix} \quad (19)$$

The proposed exponent matrix,  $P_{de}$  reduces the formation of girth 4 to a large extent (in single digits).

Several exponent matrices can be built using the proposed algorithm which can be applied for various applications.

### 3.3 Construct the proposed QC-LDPC code using the proposed base matrix, CPM matrices and difference exponent matrix

**Step 8.** Substitute each integer value in the proposed exponent matrix,  $P_{de}$  with its corresponding  $(n \times m)$  CPM matrices (right shifted permutation base matrices) and substitute the zero value with  $(n \times m)$  zero matrix of the same size as that of the base matrix.

**Step 9.** Check for the presence of cycle 4’s after substituting the  $(m \times n)$  CPM matrices obtained from base matrix. Very few cycle 4’s will be present which can be eliminated by flipping the ones that forms the short cycle. The resultant matrix obtained is the required  $(K \times M)$   $H_i$  matrix (information part of the H matrix).

**Step 10.** Append a  $(M \times M)$  dual diagonal identity parity check matrix,  $H_p$  to the  $H_i$  matrix to obtain the proposed final PCM,  $(M \times N)$  H matrix of the proposed QC-LDPC code.

### 3.4 Structure of the proposed QC-LDPC code

The proposed QC-LDPC code has a defined structure with size  $(N, K, M)$ , which is shown in Fig.4.

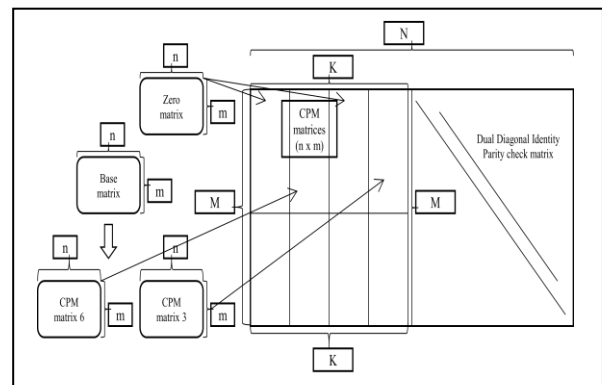


Figure 4: Structure of the proposed QC-LDPC code.

From Fig.4, it can be seen that the base matrix  $H_b$  of size  $(m \times n)$  is circularly right shifted to obtain the circulant permutation matrices (CPM) of same size and an all zero matrix of  $(m \times n)$  size is constructed. These sub-matrices are then substituted in the difference exponent matrix  $P_{de}$  to obtain the  $H_i$  matrix of size  $(M \times K)$ . A dual diagonal identity parity check matrix  $H_p$  is constructed with size  $(M \times M)$ , which is appended to  $H_i$  matrix to finally get the required QC-LDPC code H matrix of size  $(M \times N)$ .

### 3.5 XOR based Encoding

An illustration of a systematic LDPC code with 4 data nodes and 3 parity nodes is presented in Fig. 5.

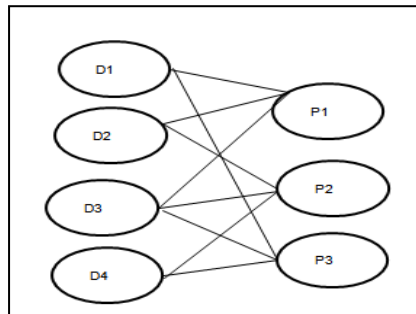


Figure 5: Systematic LDPC code with 4 data nodes and 3 parity nodes.

The systematic LDPC encoding process uses simple XOR operations [12], in which the data blocks (denoted by D on the left side of the graph) are computed by XOR'ing the parity blocks to which they are linked and the parity blocks (denoted by P on the right side of the graph) are computed by XOR'ing the data blocks to which they are linked. Both the data and parity blocks are then sent across the internet to be stored in different cloud servers.

### 3.6 Recovery based iterative message passing decoding

The QC-LDPC code decoding is performed using a recovery equation-based iterative decoding algorithm [27]. The recovery equations are derived using the same Tanner graph (H matrix) which was used for the encoding process. For example, if the (7, 4) LDPC code with H matrix Fig (5) is used for decoding then there are a total of 7 blocks with 3 parity blocks, and 4 data blocks ( $k=n-m=7-3=4$ ). All the operations performed in decoding to recover the lost data clocks are simple XOR operations.

Suppose the first data block is lost, then the decoder will find the recovery equations concerning the first data block. After several iterations, the decoder will find the correct recovery equation which has all the blocks including the parity blocks needed to retrieve the lost data block. The decoder contacts the cloud server in which the required parity blocks are stored and successfully reconstructs the original data block. Once the lost data block is retrieved, the decoder downloads the rest of the blocks and concatenates them to obtain the original user data. If the decoder has run the maximum number of iterations and still unable to retrieve the original data block, then the decoder returns the message as 'failed to retrieve the original data block'.

## 4. Application of proposed QC-LDPC code for cloud storage system

The proposed QC-LDPC coded cloud system model is shown in Fig.6. A user data file {image, text, and

video} is split into several parts and based on each data size, the corresponding QC-LDPC code is chosen to encode the data files. The QC-LDPC encoder is responsible for encoding each data segment, and the encoded segments are distributed across various cloud servers for storage. When a user requests the data, the decoder retrieves all the necessary files, including data blocks and the required number of parity blocks, to reconstruct the original user data. In cases of data loss, the QC-LDPC decoder employs a recovery-based decoding algorithm to recover the original user data.

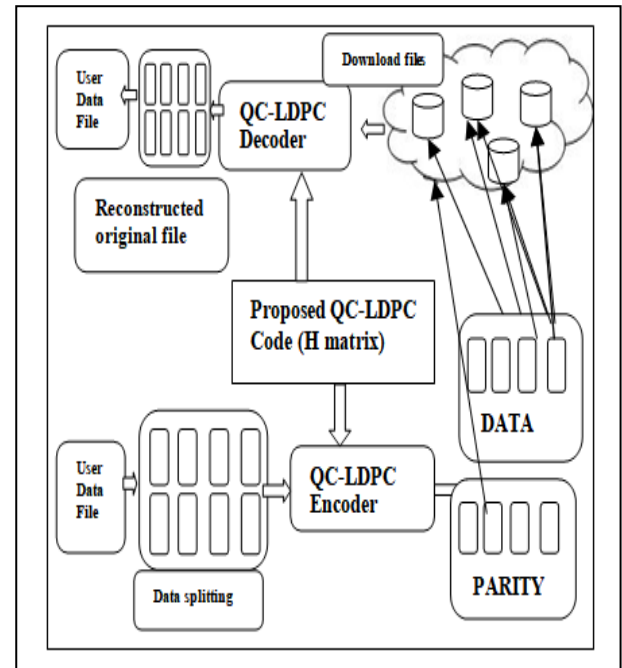


Figure 6: System model of QC-LDPC coded cloud storage system

In this paper, S3 cloud storage system has been used to store the proposed QC-LDPC encoded files and the performance in terms of encoding-decoding speeds and storage overhead has been analyzed. The subsequent section provides an explanation of the S3 cloud storage architecture.

### 4.1 Architecture of S3 cloud storage

The cloud storage architecture comprises a front-end system and a back-end system. The front-end includes the user device and the necessary applications for system access. Meanwhile, the back-end is composed of various components, including the LB (Load Balancer), API server, file servers, and partition server. Figures 7 and 8. depict the high-level architecture [28] and the back-end system of the S3 Cloud storage system, respectively.

The S3 Gateway plays a vital role in managing input and output communications, encompassing the transfer of input data to S3 buckets through the meta-data management module. Additionally, there is a dedicated management gateway that supports tasks like instance creation, setup, configuration, and more, in coordination with the S3 gateway. The metadata management layer comprises a set of distributed service engines that enable



users to upload data based on policies and permissions while efficiently managing the stored data.

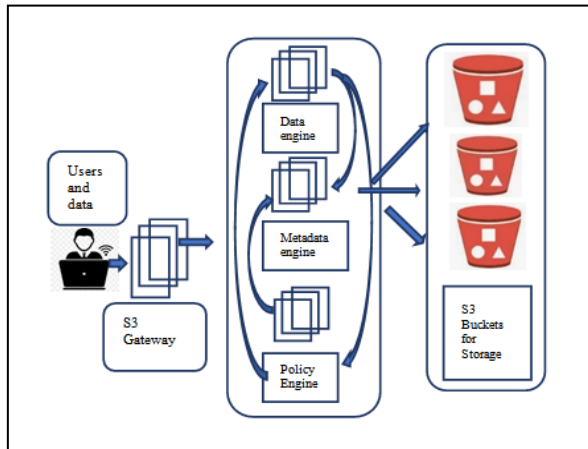


Figure 7. High level architecture of S3 cloud storage system.

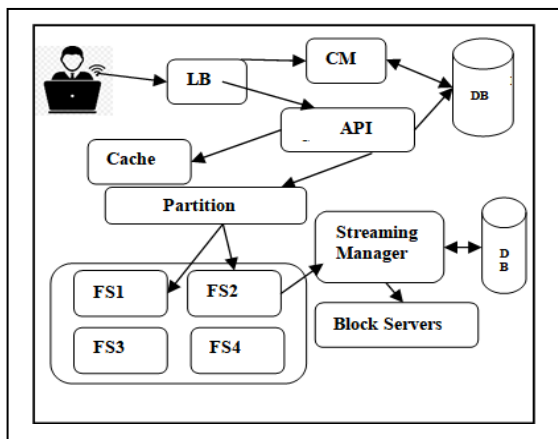


Figure 8: Backend system of the cloud data storage infrastructure.

The backend systems, including CM, LB, API server, DB, and SL, operate synchronously to deliver an effective data storage solution with high data reliability and availability for users. The key terms are described as follows: -

**Cluster Manager (CM)** - Manages all servers, handles account allocation, oversees account management, controls bucket creation, and grants or removes permissions.

**Load Balancer (LB)** - Performs various operations such as uploads, downloads, and archiving.

**API Server (API)** - Acts as the central application server responsible for handling all user requests.

**Database Server (DB)** - Stores user file information, encryption keys, and authentication data.

**Streaming Layer (SL)** - Performs replication and keeps another copy of the data in region 2 in other data centers.

Once a user creates an account with S3, CM allocates the user account by obtaining the necessary permissions from the Authentication Server or the DB. User creates new buckets through the S3 Console once the required permissions are granted. CM updates the bucket's

domain name in the DNS. User uploads the encoded files in the buckets and these operations are taken care of by the LB. When a file is uploaded, the API server assigns a unique UUID to facilitate easy file identification. The API server then transmits the file to the partition server, which communicates with the streaming layer. The Streaming Layer comprises numerous file servers, each with storage capacities ranging from 20 to 30 TB. The partition server writes the file into a designated stream file server, for example, "Stream Server 1." The Stream Layer retrieves the files, writes them to the hard disk, and creates replicas. All UUIDs are stored in the DB. Consequently, when a file is requested, CM retrieves the file from the HDD to the buckets, enabling users to promptly download the files.

In the event of a region experiencing downtime, the DNS will redirect to the LB of another region, and ensures that full data availability is maintained. CM monitors the available space in the file server and determines which resources can be utilized to store new files. This paper explores the utilization of the S3 cloud data storage system for storing proposed QC-LDPC encoded user data.

### 4.2 Advantages of the proposed algorithm

The proposed algorithm's main benefit is that it generates high code rate QC-LDPC code  $\{0.8, 0.9\}$  from a low code rate (0.5) base matrix  $H_b$ . The exponent matrix is derived as a difference of the rows of the original exponent matrix, and the base matrix is devoid of short cycles. Figure 9a. Shows the H matrix before girth 4 ( $64 \times 1216$ ) was eliminated i.e by applying CPM matrices of proposed  $H_b$  matrix in generalized exponent matrix  $E(H)$ . Figure 9b. Shows the proposed H matrix ( $1216 \times 32$ ) after girth 6 is eliminated by applying CPM matrices of the proposed  $H_b$  matrix in the proposed difference exponent matrix  $P_{de}$ .

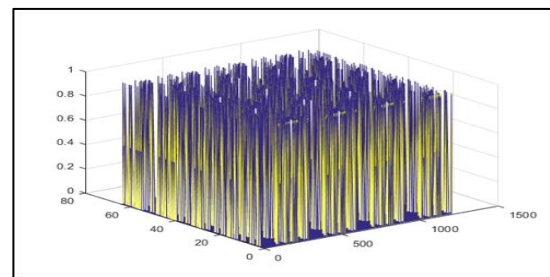


Figure 9: a) H matrix -Before removal of girth 4 ( $64 \times 1216$ ) by applying CPM matrices in generalized exponent matrix  $E(H)$

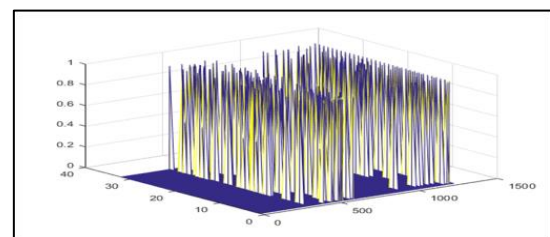


Figure 9: b) H matrix without girth 4 ( $32 \times 1216$ ) by applying CPM matrices in proposed difference exponent matrix  $P_{de}$



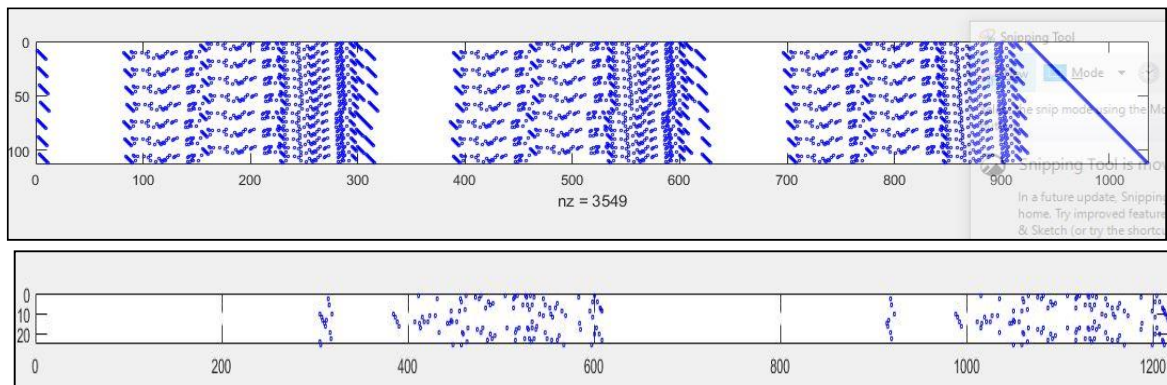


Figure 10: a) H matrix –Less sparse QC-LDPC code before applying algorithm b) H matrix with more sparsity after applying algorithm.

The proposed QC-LDPC code constructed using the proposed algorithm achieves highly sparse codes, which reduces the complexity of the codes and hence, increases the speed of operations. This is attributed to the fact that the complexity of the proposed code is determined by the number of one's present in the code. Figure 10a illustrates an example of a QC-LDPC code with low sparsity level, while Figure 10b. demonstrates an example of a highly sparse QC-LDPC code achieved through the proposed technique..

## 5 Results and discussions

In this section, simulation results that compare the encoding-decoding speeds of the proposed QC-LDPC codes with conventional LDPC, conventional QC-LDPC codes, and RS codes are presented. Here are a few examples of the H matrix constructed using the proposed algorithm: -

i) Proposed QC-LDPC codes: (266, 212, 54), (720, 640, 80), (1024, 860, 164), (2160, 1800, 180) with a code rate  $R=0.8$ .

ii) Proposed QC-LDPC codes: (1248, 1216, 36), (4352, 4096, 256) with a code rate  $R=0.9$ .

The proposed high-rate QC-LDPC code of size (1248,1216,  $R=0.9$ ) have been utilized to encode user data files of varying sizes {100 KB, 256KB, 500KB, 1MB, 10MB} to test the performance of the proposed algorithmic code in terms of encoding speeds.

In S3 cloud storage buckets, the encoded files composed of both the data and parity blocks are uploaded via the S3 console from a local machine. The local machine is equipped with both the encoder and decoder functionalities, implemented using MATLAB software.

The local machine is a Windows 11 Pro system with the following specifications: Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz 3.70 GHz and 64-bit operating system, x64-based processor, 64.0 GB RAM.

When the client needs the original data file, they simply download the necessary data blocks, and then concatenate them together to reconstruct the original file. However, in case of data loss, the decoder's role

comes into play. It decodes the available data blocks and attempts to reconstruct the lost data, ultimately leading to the recovery of the original data.

### 5.1 Comparison of encoding and decoding speeds of proposed QC-LDPC codes.

Table 2. shows the comparison between the encoding speeds of proposed QC-LDPC, conventional LDPC codes, conventional QC-LDPC codes and RS codes.

Table 2 reveals that the proposed QC-LDPC code with a high code rate of 0.9 and  $N=1248$  takes only 2.128 seconds to encode a 10MB file. In contrast, conventional QC-LDPC codes and conventional LDPC codes encode the same file in 8.42 seconds and 94.36 seconds, respectively. This demonstrates that the proposed QC-LDPC code exhibits significantly faster encoding speeds compared to its counterparts. This speed advantage is primarily attributed to the high level of sparsity and the girth six property of the proposed code. Furthermore, it's worth noting that the shorter length proposed QC-LDPC code takes less time for encoding compared to the longer codelength proposed code with high-rate codes.

Table 2: Encoding speeds of the proposed QC-LDPC code vs Conv. LDPC or QC-LDPC codes vs RS code

Encoding Speeds of Prop. QC-LDPC codes vs. Conv. QC-LDPC codes with code rate R=0.9					
Erasure codes (N, K, M)	100 KB	256 KB	500 KB	1 MB	10 MB
Prop. QC-LDPC code (1248,1216)	34.86 ms	63.44 ms	126.98 ms	226.45 ms	2.128 s
Conv. QC-LDPC code (1240,1116)	251.43 ms	554.26 ms	980 ms	1.921s	8.42 s
Conv. LDPC code (1024,940)	2.66 s	4.97 s	9.98s	23.82 s	94.36 s
Encoding Speeds of Proposed QC-LDPC codes vs., RS codes with code rate R=0.8					
Prop. QC-LDPC code (266,212)	2.12 ms	5.06 ms	11.24m s	36.9 ms	268.45 ms
Reed Solomon Code (256,224)	4.28 ms	9.64 ms	21.04m s	0.485s	3.682 s

Table 3: Decoding speeds of the proposed QC-LDPC code vs Conv. LDPC or QC-LDPC codes vs RS code

Decoding Speeds of Prop. QC-LDPC codes vs. Conv. QC-LDPC codes with code rate R=0.9					
Erasure codes (N, K, M)	100 KB	256 KB	500 KB	1 MB	10 MB
Prop. QC-LDPC code (1248,1216)	46.45 ms	94.33 ms	198.46 ms	380 ms	1.43 s
Conv. QC-LDPC code (1240,1116)	206.8 ms	422.46 ms	842.56 ms	1.76 s	6.85 s
Conv. LDPC code (1024,940)	1.99 s	3.92 s	8.1 s	19.86 s	85.6 s
Decoding Speeds of Proposed QC-LDPC codes vs., RS codes with code rate R=0.8					
Prop. QC-LDPC code (266,212)	1.04 ms	2.65 ms	9.92 ms	21.80 ms	192.45 ms
Reed Solomon Code (256,224)	3.41 ms	6.42 ms	16.84m s	1.07s	2.12s

Table 3 demonstrates that the proposed QC-LDPC code achieves faster decoding times, taking only 1.43 seconds to decode a 10 MB file. In contrast, conventional QC-LDPC codes and conventional LDPC codes decode the same file in 6.85 seconds and 85.6 seconds, respectively. Notably, the shorter length proposed QC-LDPC code outperforms RS codes of

similar size and rate, as it takes only 192.45 milliseconds to decode the file using the proposed code, which is significantly faster than the 2.12 seconds required by traditional RS codes.

The proposed codes exhibit high sparsity and possess the girth six property, resulting in higher decoding speeds compared to conventional LDPC, conventional QC-LDPC codes, and traditional RS codes.

## 5.2 Storage metrics

The key storage performance metrics [27] for an (N, K, M) QC-LDPC code, where N represents the code length, K denotes the number of information blocks, and M signifies the number of parity blocks, are elucidated as follows:

a) **Erasure correction capability** ‘ $\epsilon$ ’ of QC-LDPC code is computed by

$$\epsilon = (K + M) - fK \quad (20)$$

where ‘f’ is the **storage overhead factor**

It is well known that LDPC codes [29,27] requires  $fK$  blocks out of  $K + M$  encoded blocks to recover the original K data blocks.

b) **Storage overhead** ( $S_{ovh}$ ) is determined by the ratio of the number of parity blocks to the number of data blocks [32].

c) **Storage efficiency** ‘ $\eta$ ’ can be calculated by determining the percentage of the code rate of the QC-LDPC code.

$$\eta = \frac{K}{N} \times 100 \quad (21)$$

d) **Decoding Overhead**  $Dec_{ovr}$  [32]: It is defined as the number of blocks required for decoding –  $\epsilon$

e) **Decoding Overhead Ratio**  $Dec_{ovr}$  Ratio: It is defined as the number of blocks needed to decode by the actual number of data blocks minus 1. [32]

$$Dec_{ovr} \text{ Ratio} = \frac{\text{Number of blocks needed to decode}}{\text{Actual number of data blocks}} - 1 \quad (22)$$

Table. 4. shows the comparative analysis of the proposed QC-LDPC codes with conventional LDPC, QC-LDPC codes, and traditional RS codes based on storage performance metrics.

Table 4: Storage performance analysis

Erasure codes (N, K, M)	Code Rate R	$\eta$ ( %)	$\epsilon$	$f$	$Dec_{ovr}$	$Dec_{ovr}$ Ratio	$S_{ovh}$
Prop.QC-LDPC code (266,240,26)	0.9	90%	3	1.09	263	0.24%	0.108
Prop.QC-LDPC code (720,648,72)	0.9	90%	3	1.106	717	0.12%	0.106
Prop.QC-LDPC code (1024,940,84)	0.9	90%	3	1.108	1021	0.10%	0.108
Prop.QC-LDPC code (1248,1216,36)	0.9	90%	3	1.02	1245	0.02%	0.029
Prop.QC-LDPC code (2160,1800,180)	0.8	80%	3	1.19	2157	0.19%	0.11
Prop.QC-LDPC code (4352, 4096,256)	0.9	90%	3	1.06	4349	0.06%	0.062
Conv. LDPC code (1240,1116,124)	0.9	90%	2	1.109	1238	0.109%	0.1
Conv.QC-LDPC code (1260,1168,92)	0.9	90%	2	1.07	1258	0.07%	0.07
RS code(256,224,32)	0.8	80%	2	NA	NA	NA	0.14

From Table 4, it is evident that the proposed QC-LDPC codes. Further, proposed codes were found to have much better storage performance than the conventional LDPC codes and traditional RS codes codes, characterized by high code rates (0.9) and varying code lengths (1248, 1216), (4352, 4096), outperform others in terms of storage overhead and erasure correction capability. The storage overhead generated by the high code rate (0.8) proposed QC-LDPC code with a long code length (2160, 1800) is slightly higher (0.11) compared to other higher code rate (0.9) proposed QC-LDPC codes, which have a storage overhead of just 0.029. **Hence, High code rate proposed QC-LDPC code of 0.9 and above produces lesser storage overhead than medium-high code rate proposed QC-LDPC code of 0.8. Further, proposed QC-LDPC codes were found to have much better storage performance than the conventional LDPC codes and traditional RS codes.** Due to the highly sparse girth six property of proposed QC-LDPC codes, it can be seen that they have less decoding overhead compared to conventional codes and RS codes. Thus, we conclude that the proposed QC-LDPC codes are the best among the three candidates for cloud data storage systems based on their faster encoding decoding speeds, high storage efficiency and reduced storage overhead..

### 6 Application of proposed QC-LDPC codes in wireless communication system

A basic wireless communication system has been designed in LABVIEW software to analyze the error correction performance of the proposed high rate QC-LDPC codes, as depicted in Fig. 11.

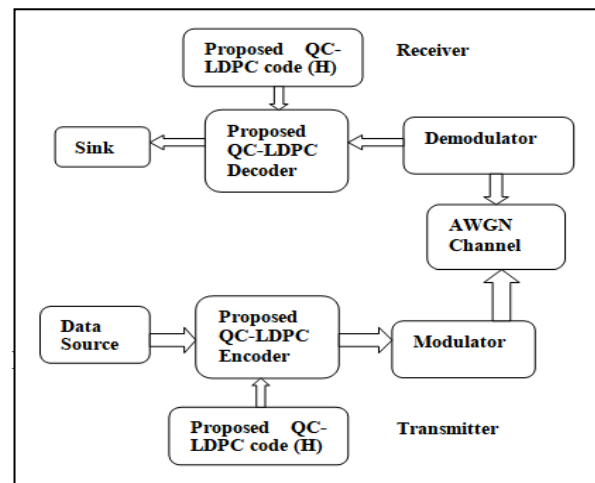


Figure 11: Basic digital wireless communication system.

The user data is transmitted to the proposed QC-LDPC encoder, where it is encoded using the proposed QC-LDPC code (H matrix) constructed using the proposed algorithm. The encoded data is sent to the modulator with binary phase shift keying BPSK modulation scheme and sent across the additive white Gaussian noise AWGN channel [25, 31]. On the receiver end, the received data is demodulated using BPSK demodulation scheme and decoded by the proposed QC-LDPC decoder, which is then finally received at the sink. The proposed parity check matrix H is constructed in MATLAB software and then inserted into the LABVIEW code in .IVM format to the encoder and decoder respectively.

## 6.1 BER performance analysis of the proposed QC-LDPC over AWGN channel.

The most basic communication channel, AWGN channel has been utilized for transmitting the encoded signal with noise. The received signal is given by

$$y = s_{0/1} + n \quad (23)$$

, where  $n$  denotes the noise added to the signal.

The capacity of the AWGN channel is given by

$$C = \frac{1}{2} \log_2 \left( 1 + \frac{1}{\sigma^2} \right) \quad (24)$$

, where  $\sigma^2$  is the noise variance which is the energy per transmitted bit  $E_s$ .

BER for BPSK modulated system [32] can be calculated

$$\text{by } BER = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_b}{N_0}} \quad (25)$$

$$\text{And } \frac{E_b}{N_0} \text{ can be calculated by } \frac{E_b}{N_0} = \frac{n}{k} \cdot \frac{1}{\sigma^2} \quad (26)$$

This section presents the BER performance of the proposed high code rate (0.9) QC-LDPC code using BPSK modulation scheme over AWGN channel. They are also compared with the results of conventional LDPC codes.

Figure 12. displays the performance of a high code rate proposed QC-LDPC code (1248,1216,R=0.9) over AWGN channel.

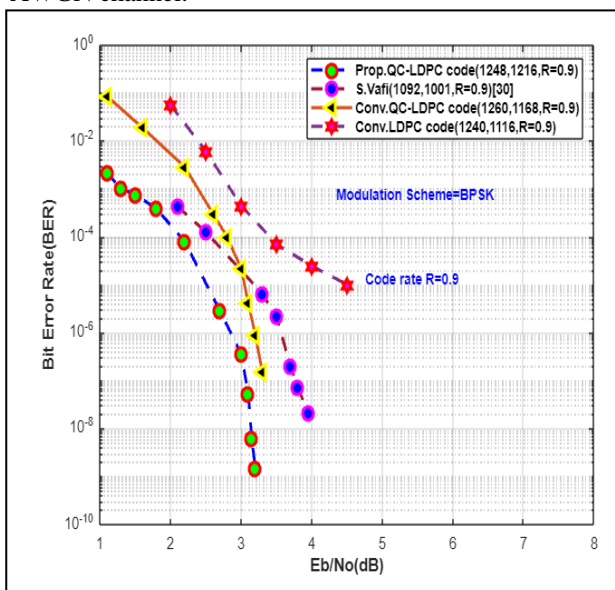


Fig. 12. BER performance of high code rate proposed QC-LDPC codes (1248,1216,R=0.9) on AWGN channel using BPSK modulation scheme.

In terms of BER, the simulation results show that the proposed QC-LDPC code outperforms the conventional QC-LDPC code and conventional LDPC code with BPSK modulation scheme on the AWGN channel. This is due to the fact that cycle 4 is absent from both the base matrix and the newly constructed proposed QC-LDPC code. When compared to a conventional LDPC code of the same size and rate, the newly constructed proposed QC-LDPC code with a high code rate of

$R=0.9$  performs quite well.

The proposed QC-LDPC code has a very low BER of  $10^{-9}$  at 3 dB  $E_b/N_0$ , whereas the conventional QC-LDPC code only has a BER of  $10^{-7}$  at 3.2 dB  $E_b/N_0$  and S. Vafi [30] LDPC code only has a BER of  $10^{-8}$  at 4 dB  $E_b/N_0$ . As a result, we find that the proposed high code rate QC-LDPC code with girth six outperforms the conventional QC-LDPC codes and other algorithmic codes in terms of error performance.

## 6.2 Hardware Implementation of the proposed QC-LDPC code (N=1248, R=0.9) in software defined radio using NI USRP 2920.

Software Defined Radio (SDR) is a radio device that can be reprogrammed to transmit and receive on every frequency within a certain range. SDR used in this work is National Instruments Universal Software Radio Peripheral (USRP) 2920. The USRP 2920 offers an integrated hardware and software solution, enabling rapid prototyping of high-performance wireless communication systems and facilitating the evaluation of newly designed algorithms. This SDR operates in the frequency range of 50 MHz to 2.2 GHz and connects to the host PC via Gigabit Ethernet. Gain range of USRP 2920 is 0 dB to 31 dB. By integrating the USRP 2920 with the LABVIEW communication suite, a complete wireless communication system is designed. This section presents the error performance evaluation of proposed QC-LDPC code (N=1248, K=1216, R=0.9) using USRP 2920 hardware device.

Figure 13. shows the experimental set-up used to conduct the tests and parameters for both the transmitter and receiver codes are set as given in Table 5.



Figure 13: Experimental setup

On the transmitter end, the parameters are set such as USRP's IP address, carrier frequency, IQ rate, active antenna and gain. The proposed QC-LDPC code is inserted into the program along with the source data bits and it is run. The user data bits are encoded and mapped into symbols, which are BPSK modulated and converted to analog signals.

Table 5: NI USRP 2920 Parameters.

S.No.	Parameters	Transmitter	Receiver
1	Device IP	192.168.10.1	
2	Antenna	Triband Antenna	
3	Modulation	BPSK	
4	Active Antenna	TX1	RX1
5	Gain of USRP	12dB	14dB
6	Carrier Frequency	915M	915M
7	Ethernet connection	1 Gigabyte	

These signals are transmitted via a triband antenna. Now the receiver, receives the signal and are converted into digital signals. The signals are now BPSK demodulated and decoded using proposed QC-LDPC decoder. BER is measured using BER block in the receiver code. One of the main issues faced during implementation was that of limited memory of SDR i.e. while dealing with large volumes of data, the transmitter and receiver blocks did not run properly. The programs kept freezing. So we split the complete data into smaller blocks and ran the programs. For a message block size of  $K=1216$  and higher code rate of  $R=0.9$ , there were no issues and the results are provided in Fig .13. The error correction performance of the proposed QC-LDPC code is analyzed using the Bit Error Rate (BER) vs  $E_b/N_0$  as given in Fig. 14.

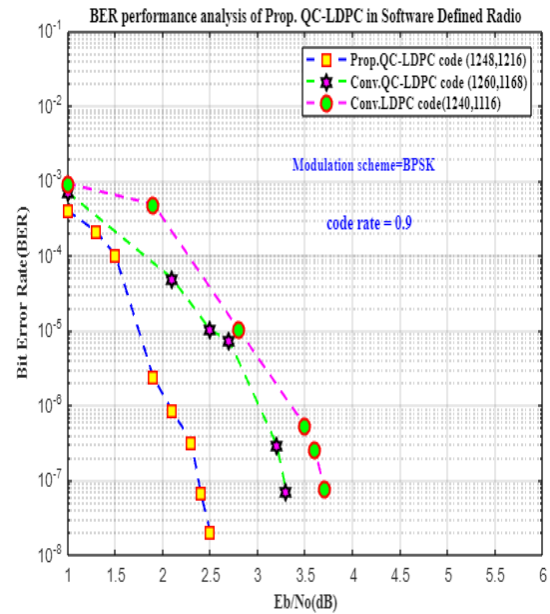


Fig 14. BER performance of proposed QC- LDPC vs Conv. LDPC and QC-LDPC codes

Figure.13. shows a very low BER of  $10^{-8}$  in 2.5 dB  $E_b/N_0$  compared to conventional LDPC and conventional QC-LDPC codes of similar rate and lengths. It is due to the high level of sparsity of the proposed base matrix with code rate 0.5, which is expanded to obtain a high code rate QC-LDPC code of 0.9 and with girth of atleast six.

Table 6: Comparison of the existing work with the proposed work.

S/N	Author	Goals	Result
1	G. Zhang et al. (2020)	To construct girth six QC-LDPC codes using Chinese Remainder Theorem (CRT) method and short codes that can achieve high error correction capability.	Girth-6 QC-LDPC codes constructed using this method show better decoding performance using the iterative decoding with sum-product algorithm (SPA) in about 50 iterations
2	S.Noor (2010)	To construct short length QC-LDPC codes with girth six based on multiplicative group in the finite field and to improve BER performance by applying masking technique .	Short length QC-LDPC codes achieved a BER of $10^{-6}$ at about 4 dB $E_b/N_0$ and no error floor is seen
3	LAN et al.(2007)	To construct RC-constrained arrays of circulant permutation matrices (CPM) based on the multiplicative groups of finite fields. The CPM matrices and zero matrices are combined to obtain QC-LDPC codes without cycle 4.	While decoding short QC-LDPC codes with medium code rate 0.75, and iterative decoding using SPA technique in AWGN channel with BPSK modulation shows good BER performance of $10^{-6}$ at about 3.8dB. The number of decoding iterations is set to either 50 or 100.
4	SONG et al. (2009)	To construct a base matrix over some finite field and which satisfies RC constraint. The elements in base matrix are replaced by binary circulants to form QC-LDPC codes with girth six over $GF(q)$	Large girth six QC-LDPC codes with high code rates ( $R=0.86$ ) an girth six shows better decoding performance using SPA decoding with 50 iterations over AWGN channel.
5	Sheng Huang(2016)	To construct high rate QC-LDPC codes with good distance property that can be applied to high speed applications	Achieved good error correction capability in about 30 iterations with SPA decoding algorithm
6.	Proposed work	To construct high rate QC-LDPC codes with girth six property by combining the proposed sparse base matrix of girth six and exponent matrix based on finite fields.	Achieved good error correction capability in about 15-20 iterations with SPA decoding algorithm. The proposed QC-LDPC codes are highly sparse code with high decoding performance due to their girth six property.



### 6.3 Discussion

Table .6 presents the comparison between the existing works and the proposed work in terms of BER performance. It is clear that, mainly due to the girth six property obtained by combining the proposed girth six base matrix, it's corresponding CPM matrices and the proposed difference exponent matrix ( $P_{de}$ ), the newly constructed proposed QC-LDPC codes show better decoding performance than the already existing codes given in literature as given in Table 6.. Furthermore, the high sparsity level of the proposed codes reduces the computations involved in encoding and decoding operations, which help in achieving high speeds.

### 7 Conclusion

In this paper, we have analyzed the performance of the highly sparse proposed QC- LDPC with high code rate (0.9) and girth six in terms of encoding decoding speeds and storage overhead produced. These codes have been constructed using the sparse girth six base matrix and a proposed difference exponent matrix which shows low storage overhead and high-speed encoding decoding computations with high storage efficiency. We evaluated the performance of our proposed QC-LDPC codes over the AWGN channel. Simulation results demonstrate superior performance compared to conventional LDPC codes, conventional QC-LDPC codes, and other algorithmic codes of similar size and rates. Software implementation of the proposed QC-LDPC code (N=1248) has been carried out using a hardware device NI USRP 2920(SDR) and a low BER of  $10^{-6}$  at 4.2dB  $E_b/N_0$  is achieved for a high code rate of  $R=0.9$ . Hence, we conclude that QC-LDPC codes with zero short cycles and high sparsity constructed using the proposed algorithm are suitable for both cloud data storage systems and wireless communication systems.

### Acknowledgement

Thanks to the reviewers for your valuable suggestions.

### References

- [1] W. K. Lin, D. M. Chiu, Y. B. Lee (2004). Erasure Code Replication Revisited, *4th International Conference on Peer-to-Peer Computing (P2P 2004)*, IEEE, Zurich, Switzerland, pp.90–97.  
<https://doi.org/10.1109/ptp.2004.1334935>
- [2] Mahfoud, Zohra & Nouali-Taboudjemmat, Nadia. (2019). Consistency in Cloud-based database systems. *Informatica*. 43.  
<https://doi.org/10.31449/inf.v43i3.2650>
- [3] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das,(2010). Towards characterizing cloud backend workloads: insights from google compute clusters, *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 4, pp. 34–41, 2010.  
<https://doi.org/10.1145/1773394.1773400>
- [4] C. Huang, H. Simitci, Y. Xu, A. Ogun, B. Calder, P. Gopalan, J. Li and S. Yekhanin, (2012). Erasure coding in Windows Azure storage, *Proc. of the 2012 USENIX Annual Technical Conference*, 2012, pp.15–26.  
<https://dl.acm.org/doi/10.5555/2342821.2342823>
- [5] Facebook's erasure coded hadoop distributed file system (hdfs-raid)  
<https://github.com/facebook/hadoop-20>.
- [6] S.Reed and G.Solomon.(1960). Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*,8(2):300–304,1960. <https://doi.org/10.1137/0108018>
- [7] Li and B. Li,(2013).Erasure coding for cloud storage systems: a survey, *Tsinghua Science and Technology*, vol. 18, no. 3, pp. 259–272,2013. <https://doi.org/10.1109/TST.2013.6522585>
- [8] Osama Khan, Randal Burns, James Plank, William Pierce, Cheng Huang. (2012). Rethinking erasure codes for cloud file systems: Minimizing I/O for recovery and degraded reads. *Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST'12)*,SanJose, CA, February14-17,2012.  
<https://dl.acm.org/doi/10.5555/2208461.2208481>
- [9] James Plank and Huang,(2013) Tutorial on Erasure Coding for Storage Applications, *11th USENIX Conference on File and Storage Technologies*, ,SanJose,CA,February12,2013.  
<http://web.eecs.utk.edu/~jplank/plank/papers/FAST-2013-Tutorial.html>
- [10] Plank, J. S. and Xu, L., (2006). Optimizing cauchy reed-solomon codes for fault-tolerant network storage applications. In *Proceedings of the IEEE International Symposium on Network Computing and Applications*. (IEEE NCA06), Cambridge, MA, July,2006.  
<http://www.cs.utk.edu/~plank/plank/papers/NCA-2006.html>
- [11] J. Blomer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, and D. Zuckerman.(1995) An XOR-based erasure-resilient coding scheme. *Technical Report TR-95-048*, *International Computer Science Institute*, August 1995.  
<http://www.icsi.berkeley.edu/ftp/global/pub/techreports/1995/tr-95-048.pdf>
- [12] Benjamin G., Birger K., Nuno S., (2007). Exploring high performance distributed file storage using LDPC codes, *Parallel Computing*, Vol. 33, Issue 4-5.  
<https://doi.org/10.1016/j.parco.2007.02.003>
- [13] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran,(2010).Network coding for distributed storage systems, *IEEE transactions on*



- information theory*, vol. 56, no. 9, pp. 4539–4551, 2010.  
<https://doi.org/10.48550/arXiv.0803.0632>
- [14] J. S. Plank and M. G. Thomason, (2004). A practical analysis of low-density parity-check erasure codes for wide-area storage applications, *2004 International Conference on Dependable Systems and Networks (DSN)*, June 2004, pp.115–124. <https://doi.org/10.1109/dsn.2004.1311882>
- S.I.Park,H.M.Kim,Y.Wu and J.Kim,(2013).A Newly Designed Quarter-Rate QC-LDPC Code for the Cloud Transmission System, *IEEE Transactions on Broadcasting*, vol. 59, no. 1, pp. 155-159, March 2013.<https://doi.org/10.1109/tbc.2012.2226673>
- [15] Park G.S., Song H.(2016). A novel hybrid P2P and cloud storage system for retrievability and privacy enhancement. *Peer-to-Peer Netw. Appl.* 9, 299–312 (2016).  
<https://doi.org/10.1007/s12083-015-0337-z>
- [16] Y.Weii, Y. W. Foo, K. C. Lim, and F. Chen, (2014). The auto-configurable ldpc codes for distributed storage,” *17th IEEE International Conference on Computational Science and Engineering*, Dec 2014, pp. 1332–1338.  
<https://doi.org/10.1109/cse.2014.254>
- [17] Y. Wei, F. Chen, and K. C. Lim, (2015). Large LDPC codes for big data storage, *Proceedings of the ASE Big Data & Social Informatics 2015*. ACM, p. 1-6, Oct 2015, Kaohsiung, Taiwan.  
<https://doi.org/10.1145/2818869.2818881>
- [18] Y. Wei and F. Chen, (2016), expanCodes: Tailored LDPC codes for big data storage,” *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing*, Aug 2016, pp. 620–625. Auckland, New Zealand.  
<https://doi.org/10.1109/dasc-picom-datacom-cyberscitech.2016.113>
- [19] R.Gallager,(1962).Low-density parity-check codes,*IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan 1962.  
<https://doi.org/10.1109/TIT.1962.1057683>
- [20] D. J. C. MacKay,(1999).Good error-correcting codes basedon very sparse matrices, *IEEE Trans. Inform. Theory*,vol.45, no.2, pp. 399-431, Mar. 1999. <https://doi.org/10.1109/isit.1997.613028>
- [21] S. Myung, K. Yang, and J. Kim,(2005). Quasi-cyclic LDPC codes for fast encoding, *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2894–2901, Aug. 2005.  
<https://doi.org/10.1109/tit.2005.851753>
- [22] M. P. Fossorier, (2004).Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.  
<https://doi.org/10.1109/tit.2004.831841>
- [23] Salah Abdulghani Alabady,(2018).Binary and Non-Binary Low Density Parity Check Codes: A Survey, *International Journal of Information Engineering and Applications* 2018; 1(3): 104-117.<https://api.semanticscholar.org/CorpusID:201651560>
- [24] H. Dehghani, M. Ahmadi, S. Alikhani and R. Hasni, (2012). Calculation of Girth of Tanner Graph in LDPC Codes, *Trends in Applied Sciences Research*, 7: 929-934.  
<https://scialert.net/abstract/?doi=tasr.2012.929.934>
- [25] Amirzade, F., Alishahi, M.,(Rafsanjani) Sadeghi, M. (2019). An algebraic construction of QC-LDPC codes based on powers of primitive elements in a finite field and free of small etss.*Algebraic structures and their applications*, 6(1), 129-140.  
<https://www.aims sciences.org/article/doi/10.3934/amc.2020062>
- [27] H. S. G. et al.,(2010).SpreadStore: A LDPC Erasure Code Scheme for Distributed Storage System, *2010 International Conference on Data Storage and Data Engineering*,Feb 2010, pp. 154-158. Bangalore, India.  
<https://doi.org/10.1109/dsde.2010.61>
- [28] Hitachi Vantara,(2021).Scale out S3 object storage platform, built on microservices architecture that can be deployed as software only or an appliance, *Architecture fundamentals*, November 2021.  
<https://www.hitachivantara.com/en-us/pdf/whitepaper/hcp-for-cloud-scale-whitepaper.pdf>
- [29] H. Park, D. Lee and J. Moon, (2018). LDPC Code Design for Distributed Storage: Balancing Repair Bandwidth, Reliability, and Storage Overhead, *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 507-520, Feb. 2018.  
<https://doi.org/10.1109/TCOMM.2017.2769116>
- [30] S. Vafi and N. R. Majid, (2018). Combinatorial design-based quasi-cyclic LDPC codes with girth eight , *Digit Commun Netw.* 4 (2018), 296– 300.  
<https://doi.org/10.1016/j.dcan.2018.01.001>
- [31] Ying Yu Tai, L. Lan, Lingqi Zeng, S. Lin and K. A. S. Abdel-Ghaffar, (2006).Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels, *IEEE Transactions on Communications*, vol. 54, no. 10, pp. 1765-1774, Oct. 2006.  
<https://doi.org/10.1109/TCOMM.2006.881361>
- [32] V. Bhuvaneshwari, C. Tharini,(2022). Novel construction of quasi-cyclic low-density parity-check codes with variable code rates for cloud data storage systems, *ETRI Journal, Wiley Publications*,08 November 2022,  
<https://doi.org/10.4218/etrij.2021-0449>
- [33] Swapnil Mhaske,(2015).High-Throughput FPGA QC-LDPC Decoder Architecture for 5G Wireless, 2015. <https://doi.org/doi:10.7282/T3JH3P5W>

