Encoding Schemes for Memory Efficient Quasi Cyclic Low Density Parity Check Codes

MOHAMMAD RAKIBUL ISLAM[1], SYED IFTEKHAR ALI[2]
EEE Department
Islamic University of Technology
Boardbazar, Gazipur-1704, Dhaka
BANGLADESH
[1]rakibultowhid@ yahoo.com and [2]s_ali@iut-dhaka.edu

Abstract: - Low Density Parity Check (LDPC) codes have attracted current researchers due to their excellent performance and capability of parallel decoding. One major criticism concerning LDPC codes has been their apparent high encoding complexity and memory inefficient nature due to large parity check matrix. This led the researchers to move into a memory efficient LDPC called Quasi-Cyclic (QC)-LDPC which shows the similar performance as LDPC does. This review gives a theoretical and analytical survey on different encoding schemes for QC-LDPC codes. The encoding schemes are categorized in three broad categories. These schemes are then analyzed under those three broad fields. Also different encoding schemes are compared.

Key-Words: - LDPC, QC-LDPC, efficient encoding, Richardson, Low complexity

1 Introduction

Low-Density Parity-Check (LDPC) codes have been the subject of intense research lately because of their capacity-achieving performance and linear decoding complexity. They were invented and proposed in 1962 by Robert Gallager [1, 2]. In the late 90's LDPC codes were rediscovered by Mackay and Neal [3, 4] and also by Wiberg [5]. Current hardware speeds make them a very attractive option for wired and wireless systems. Gallager considered only regular LDPC, i.e., codes that are represented by a sparse parity-check matrix with a constant number of 'ones' (weight) in each column and in each row. Later it was shown that the performance of LDPC codes can be improved by using irregular LDPC codes, i.e., both non uniform weight per column and non uniform weight per row [6, 7]. The parity-check matrix of a code can be viewed as defining a bipartite graph [8] with "variable" vertices corresponding to the columns and "check" vertices corresponding to the rows. Each non-zero entry in the matrix corresponds to an edge connecting a variable to a check.

One major criticism concerning LDPC codes has been their apparent high encoding complexity. Some low complexity LDPC encoding methods having near-linear complexity were introduced by Richardson et. al. [9] to lower encoding complexity, and its encoding method can be further simplified by employing LDPC codes whose binary base parity-check matrices have dual diagonal structure, suggested in standards such as IEEE 802.11n and IEEE 802.16e. Quasi-Cyclic (QC)-LDPC has been proposed to reduce the complexity of the LDPC while obtaining the similar performance [10, 11]. A modified scheme based on adaptive message length (AML) is proposed in [12]. Further modifications have been done over Richardson algorithm in [32], [33] and [34]. An algebraic construction for the regular and irregular QC-LDPC codes is shown in [10]. The modified algebraic construction is presented in [13]. An arbitrary bit generation and correction technique for encoding QC-LDPC codes with dual-diagonal parity structure is shown in [14], [15]. QC-LDPC code under fading channel was proposed in [16]. A construction of QC-LDPC codes for Additive White Gaussian Noise (AWGN) and Binary Erasure Channels (BEC) channels has been proposed by L. Lan [17]. Hardware implementations of decoders for QC-LDPC codes are being analyzed in some current research works [18, 19]. Some researchers are working on Quantum QC-LDPC codes in which, error detection and correction can be performed efficiently in quantum memory [20-22]. Girth of QC-LDPC codes is an important issue and several current researches are working on this topic [23-26]. It has been shown that increasing the girth or average girth of a code increases its decoding performance. The girth also determines the number of iterations before a message propagates back to its original node. Performance of structured codes could therefore be improved by increasing their girths. QC-LDPC based encoding is already suggested in some standards. Further applications and modifications of these proposals are proposed in [14], [27].

The encoding schemes used in QC-LDC codes are categorized in three types. Type I explains the encoding schemes related to approximate lower triangulation. In these schemes, H matrix is transformed into its approximate lower triangulation form. These schemes are applicable to both the LDPC and QC-LDPC codes. Type II explains the encoding schemes related to the algebraic construction of QC-LDPC codes and category III explains the encoding schemes which don't fall in the earlier two types. These different types of encoding are summarized in Table 1.

The rest of this paper is organized as follows. In section 2, the encoding schemes related to approximate lower triangulation schemes are introduced. Section 3 introduces the encoding schemes related to the algebraic construction of QC-LDPC codes. The other encoding schemes are introduced in section 4. Then section 5 concludes the paper.

Table 1
Different Encoding Schemes

Encoding Scheme type	**Encoding Schemes**
Type-I: Approximate Lower Triangulation Schemes	*1) Richardson Encoding Scheme* *2) Adaptive Message Length Encoding Scheme* *3) Arbitrary Bit-Generation and Correction Encoding Scheme* *4) Encoding with a systematic approximate lower triangular form* *5) Encoding for GLDPC codes* *6) Two stage encoding with Triangular Factorization*
Type-II: Families of Algebraic Construction of QC-LDPC codes	*1) Algebraic Construction of QC-LDPC codes: Bresnan Code* *2) Algebraic Construction of QC-LDPC codes by Dispersion* *3) Algebraic Construction of QC-LDPC codes: Rakibul Code*
Type-III: Other Encoding Schemes	*1) Encoding of QC-LDPC Codes Related to Cyclic MDS Codes* *2) Efficient Encoding of IEEE 802.11n LDPC Codes* *3) Encoding of Array LDPC Codes*

2 Approximate lower triangulation Schemes

2.1 Richardson Encoding Scheme

LDPC codes are linear codes. Hence, they can be expressed as the null space of a parity-check matrix H, i.e., x is a codeword if and only if

$$Hx^T = 0^T$$

The modifier "low-density" applies to H; the matrix H should be sparse and chosen at a random fashion. The sparseness of H enables efficient (suboptimal) decoding, while the randomness ensures (in the probabilistic sense) a good code. By means of Gaussian elimination, matrix H can be brought into an equivalent lower triangular form as shown in Fig. 1. Split the vector x into a systematic part s, and a parity part p, such that $x = (s, p)$. Construct a systematic encoder filling with the desired information symbols and determining the parity-check symbols using back-substitution.

But the actual encoding requires $O(n^2)$operations since, in general, after the preprocessing the matrix will no longer be sparse. Given that the original parity-check matrix is sparse, encoding can be accomplished in $O(n)$. Richardson proposed their encoding scheme with near $O(n)$ complexity and is shown in Fig. 2.

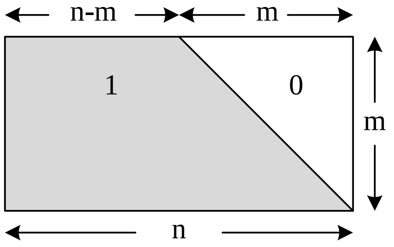

Fig.1. Equivalent lower triangular matrix using Gaussian elimination

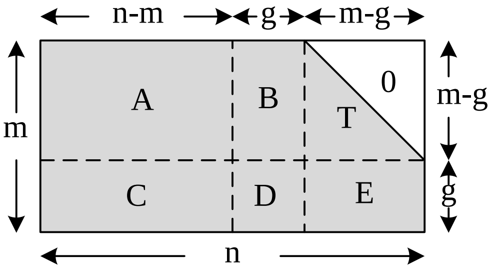

Fig. 2. Richardson's proposal for efficient encoding

They have taken the codeword $x = (s, p_1, p_2)$ and calculated p_1 and p_2 using the following equation

$$p_1^T = -\varphi^{-1}(-ET^{-1}A + C)s^T$$

$$p_2^T = -T^{-1}(As^T + Bp_1^T)$$

The overall complexity of determining p_1^T is $O(n + g^2)$ and that of p_2^T is $O(n)$. If the width of g can be kept as minimum as possible, the complexity can be kept close to $O(n)$. Bringing the randomness in the code makes the LDPC code memory inefficient. To make it memory efficient, the QC-LDPC code was evolved.

2.2 Adaptive Message Length Encoding Scheme

By means of Gaussian elimination, a matrix H can be transformed into an equivalent lower triangular form. However this approach requires $O(n^2)$ complexity encoding step. Richardson proposed $O(n - g^2)$ complexity encoding algorithm. Two different types of equations are required to compute parity bits using this algorithm. In order to reduce the complexity to *O(n)*, the author utilized approximate lower triangulation with post processing step where the parity bits can be calculated using a single equation. They assumed that the rows of a parity check matrix H have full rank. The proposed AML encoding scheme consists of the following three steps.

i) Preprocessing step: In the preprocessing step, row and column permutations of a nonsingular parity check matrix H is performed to transform the parity-check matrix into approximate lower triangular form as shown in Fig. 3.

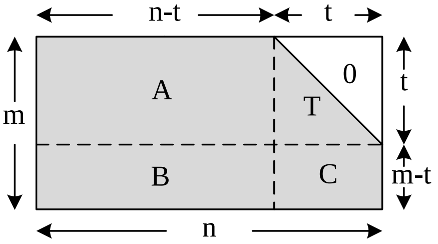

Fig 3. AML scheme before post processing step

Since this transformation is accomplished solely by permutations, the transformed matrix $H = \begin{bmatrix} A & T \\ B & C \end{bmatrix}$ is still sparse. All the submatrices are sparse and T is lower triangular with ones along the diagonal. The dimension of the matrix A is variable since the dimension of matrix T can vary according to the randomly generated H matrix.

ii) Encoding step: Let $x = (s, p)$ where s, the systematic part has length of $(n - t)$ and p, the parity part has length of t. Constraint $Hx^T = 0^T$ results in two equations, namely

$$As^T + Tp^T = 0 \quad (1)$$

$$Bs^T + Cp^T = 0 \quad (2)$$

From the above equations we get

$$p^T = T^{-1}As^T \quad (3)$$

$$B = CT^{-1}A \quad (4)$$

Once the $t \times (n - t)$ matrix, $T^{-1}A$ has been precomputed, the calculation of p can be done with complexity $O\big(t\,(n - t)\big)$. Rather than precomputing $T^{-1}A$ and then multiplying by s^T, we can get p by breaking the computation into two smaller steps, each of which is efficiently computable. Since A is sparse, As^T is computed with complexity O(n) and then As^T is multiplied by T^{-1}. $T^{-1}[As^T] = y^T$ is equivalent to the system $As^T = Ty^T$. This computation can be performed with $O(n)$ by back-substitution, since T is lower triangular and sparse. Therefore, the overall complexity for computing p is $O(n)$.

iii) Post processing step: Parity bits $\mathrm{p}_1, \ldots, \mathrm{p}_\mathrm{t}$ can be computed by using A and T submatrices since the row rank of the matrix [A T] is t.Therefore, we can ignore $\mathtt{B}$ and $\mathtt{C}$ submatrices in the transformed $\mathtt{H}$ matrix to reduce complexity during the encoding. After this post processing step, the resultant parity check matrix becomes $[\mathrm{A} \quad \mathrm{T}]$ as shown in Fig. 4. The comparison between Richardson and the AML encoding schemes is summarized in Table 2.

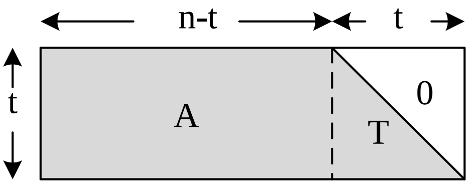

Fig. 4. AML scheme after post processing step

Table 2
Comparison in encoding steps between Richardson scheme and AML scheme

Richardson scheme	AML scheme
Preprocessing step: Input: Non singular H matrix. Output: matrix $\begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix}$	*Preprocessing step:* Input: Non singular H matrix. Output: matrix $\begin{bmatrix} A & T \\ B & C \end{bmatrix}$
Check rank step: Ensure that $-ET^{-1}B + D$ is nonsingular	*Encoding step*: Determine p to construct the codeword $x = (s, p)$ Complexity in encoding step = *O(n)*
Encoding step: Determine p_1 and $p_2$to construct the codeword $x = (s, p_1, p_2)$ Complexity in encoding step = *O(n+g²)*	*Post processing step:* Delete the sub matrix B and C to reduce encoding complexity

2.3 Arbitrary Bit-Generation and Correction Encoding

An arbitrary bit generation and correction (ABC) technique for encoding QC-LDPC codes with dual-diagonal parity structure is shown in [14], [15]. The QC-LDPC codes based on circulant sub-matrices is analyzed. Parity check matrix is modified and the encoder of quasi-cyclic LDPC codes is implemented using shift registers where the complexity of encoding is linearly proportional to the code length. Although modifying the parity matrix could result a slight performance loss due to short cycles, they reported that BLER performance of their proposed scheme is almost similar to the unaltered standard $\mathtt{H}$ matrix while complexity of encoding is slightly reduced. There are three main phases of encoding: first is the arbitrary parity-bit generation, second is sequential process to find remaining parity-bits exploiting dual-diagonal structure, and third is correction process for parity-bits. As it is true for all type to LPDC codes, the parity-check result of output code word vector c should meet $H.c = 0$. After modification of rate $R = 1/2$ mother matrix H, it can be sectorized into three sub matrices as shown in Fig.10. The information bit region $\mathtt{A}$, parity bit region $\mathtt{Q}$ for bit-flipping operation and parity bit region $\mathtt{U}$ for non bit-flipping,

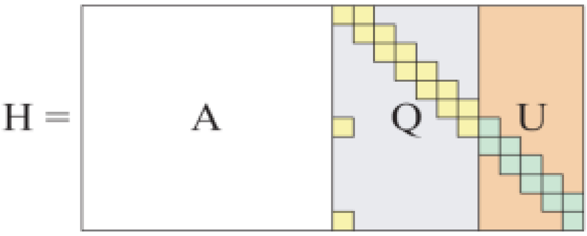

Fig. 5. Sectorized H matrix for codeword length 1944

$$\mathrm{H} = [\mathrm{A\,Q\,U}] \tag{5}$$

$$\mathrm{x} = \mathrm{As} \tag{6}$$

Parity part of matrix **H** is partitioned into two parts as **Q** and **U**. The boundary line is placed between second and third sub-block where three identity matrices are placed in a row. For example, three sub-blocks with zero cyclic shifts is located at $m/2$-th row in Fig. 5. Thus, boundary line between **Q** and **U** is set at $m/2$ -th and $(m/2+1)$ -th column. The vector x is formed by multiplying information bit vector s to sub-matrix A, as defined in Eq. (6). The proposed LDPC encoder starts encoding by generating Z arbitrary parity bits $p_0, p_1, \cdots p_{Z-1}$ for first column subblock in region **Q**. For example, all zeros can be set for $p_0, p_1, \cdots p_{Z-1}$. Assuming all zero is correct, parity bit values for $p_Z, p_{Z+1}, \cdots p_{2Z-1}$ are determined since $(x_0, x_1, \ldots, x_{Z-1})^T + \mathbf{Q}_{0\,\cdots 2Z-1} \cdot (p_0, p_1, \cdots p_{2Z-1})^T = 0$ is true for first sub-block row. Next, $p_{2Z}, p_{2Z+1}, \cdots p_{3Z-1}$ are determined sequentially since $(x_Z, x_{Z+1}, \ldots, x_{2Z-1})^T + \mathbf{Q}_{Z\,\cdots 3Z-1} \cdot (p_Z, p_{Z+1}, \cdots p_{3Z-1})^T = 0$. Note that $p_Z, p_{Z+1}, \cdots p_{2Z-1}$ as well as $x_Z, x_{Z+1}, \ldots, x_{2Z-1}$ are previously found. Exploiting the dual-diagonal parity structure, this recursive procedure is done until all parity bits (i.e. $p_Z, \ldots, p_{(mZ-1)}$) are determined. After recursion procedure, validity of last sub-block parity bits located at $(m-1)$ −th row, $p_{(m-1)Z-1}, p_{(m-1)Z}, \ldots, p_{mZ-1}$, is checked. It must hold true that last Z parity bits must check by satisfying $(x_{(m-1)Z}, x_{(m-1)Z+1}, \ldots, x_{mZ-1})^T + (p_0, p_1, \cdots\cdots\cdots p_{Z-1})^T + (p_{(m-1)Z}, p_{(m-1)Z+1}, \ldots, p_{mZ-1})^T = 0.$ If some parity bits are not correctly generated, their check results are not zero, and check is failed for specific bits. The final check results are stored in a vector **f**.

$$\mathbf{f} = (x_{(m-1)Z}, \ldots, x_{mZ-1})^T + (p_0, p_1, \ldots, p_{Z-1})^T + (p_{(m-1)Z}, p_{(m-1)Z+1}, \ldots, p_{mZ+1})^T \quad (7)$$

Thus, vector **f** is defined as

$\mathbf{f} = (x_{891}, \ldots, x_{971})^T + (p_0, p_1, \ldots, p_{81})^T + (p_{891}, p_{892} \quad , \ldots, p_{971})^T$in case of $R = 1/2$, codeword length $n = 1944$, sub-block size $Z = 81$. The LDPC encoding is summarized as following steps.

Step 1: Form accumulated information-bit vector x by doing matrix operation $x = As$.

Step 2: Set parity bits $p_0, p_1, \cdots p_{Z-1}$ as arbitrary binary values. Exploiting the dual-diagonal parity structure, solve unknown parity bits, $H \cdot (s^T, p_0, \ldots, p_{mZ-1})^T = 0$, by recursion.

Step 3: Store final check result vector

$(f_0, \ldots, f_{Z-1})^T = (x_{(m-1)Z}, \ldots, x_{mZ-1})^T + Q(p_0, p_1, \cdots\cdots p_{Z-1}) + (p_{(m-1)Z}, p_{(m-1)Z+1} \;, \ldots, p_{mZ-1})^T$ for correction of initially calculated parity bits, and create an vector v which is an augmented version of vector f with the column length of block $\boldsymbol{Q}$; $\boldsymbol{v} = (\mathbf{f}^{\mathrm{T}}, \mathbf{f}^{\mathrm{T}}, \mathbf{f}^{\mathrm{T}}, \mathbf{f}^{\mathrm{T}}, \mathbf{f}^{\mathrm{T}}, \mathbf{f}^{\mathrm{T}}, \mathbf{f}^{\mathrm{T}})^{\mathrm{T}}$. The number of final check result vector **f** to be augmented is $\frac{m}{2} + 1$ in case of *802.11n* draft standard.

Step 4: Add vector **v** to parity bits $p_0, p_1, \cdots p_{(\frac{m}{2}+1)Z-1}$ in region Q to correct them; $(\acute{p}_0, \acute{p}_1, \cdots\cdots\cdots \acute{p}_{(\frac{m}{2}+1)Z-1})^T = (p_0, p_1, \cdots p_{(\frac{m}{2}+1)Z-1})^T + \mathbf{v}$. Parity bits in block **U** are not changed.

The complexity comparison for Richardson, Rakibul and ABC technique is shown in Table 3. The complexity calculations are performed by using 802.11n based H matrix.

Table 3
Computational complexity in different encoding schemes

Parameter	Richardson Scheme	Rakibul Scheme	ABC Scheme
$\mathrm{p}_1^{\mathrm{T}}$	4941	–	–
As^{T}	4455	4617	4941
$\mathrm{Bp}_1^{\mathrm{T}}$	–	–	–
$\mathrm{Tp}_2^{\mathrm{T}}$	972	–	972
Tp^{T}	–	–	–
f	–	–	162
v	–	–	486
Total	10368	4617	6561

2.4 Encoding with a systematic approximate lower triangular (SALT) form

The first step in this encoding [32] is to transform the H matrix with as small gap g as possible, into an equivalent almost lower triangular form H_1, as illustrated by Fig. 2. As the ALT form, H_1, of the H matrix, is obtained by row and column permutations only, the submatrices A, B, C, D, T and E are all sparse matrices. In a second step the matrices A, B and T are kept, and the matrix E is transformed into an all-zero matrix and the matrix D into an identity matrix, both by Gaussian elimination. The resulting equivalent H matrix has systematic approximate lower triangular (SALT) form and full rank, and this parity check matrix is denoted by HH; it is illustrated by Fig. 6. It is assumed that during the process of transformation of the original H matrix into the equivalent form, HH, any linear dependent rows (which frequently but not necessarily occur in LDPC code constructions) are removed, so that the equivalent SALT form HH of the H matrix has full rank and the number of rows equals the number m of parity bits. To obtain the diagonal structure for the matrix T permutation of columns may be necessary, which means that bit-positions within the code word are relocated. Although this means that the matrices H and HH will not describe exactly the same code, the codewords will only differ in the ordering of the bits. This trivial type of change is assumed to be contained in our notion of 'equivalence' of the parity check matrices.

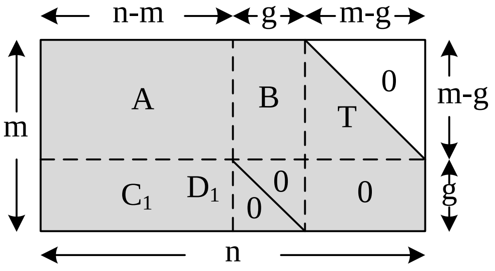

Fig. 6. Parity Check Matrix, HH, in systematic approximate lower triangular (SALT) form

Due to the structure of the SALT form Fig. 6 we can conveniently pick the first $n-m$ bit positions (from the left) in the codeword to be the data bit positions, i.e., the columns corresponding to the matrices A and C$_1$ are those of the data bits. Hence, the codewords have the following structure: $v = (\mathbf{u}, \mathbf{p_1}, \mathbf{p_2})$ with u the $n-m$ data bits, p_1 the first g parity bits and $\mathbf{p_2}$ the remaining $(m-g)$ parity bits.

The first g parity bits $\mathbf{p_1}$ can be directly determined from the sub-matrices C$_1$ and D$_1$ according to $\mathbf{p_1} = \mathbf{u}\,.C_1^T$. Further, from the parity-check condition $\mathbf{H}\,.\mathbf{v}^T = \mathbf{0}_{n\times 1}$ for any codeword $\mathbf{v}$, we obtain $A.\mathbf{u^T} + B.\mathbf{p_1^T} + T.\mathbf{p_2^T} = \mathbf{0}_{m\times 1}$. As the matrix T has lower triangular form, we obtain the second set $\mathbf{p_2} = \{p_2(1), p_2(2), \cdots, p_2(m-g)\}$ of parity bits by back-substitution.

2.5 Efficient encoding approach for generalized low density parity check codes

Inspired by the work in [9], the authors in [33] investigated a similar efficient encoding scheme for $(N, 2, n)$ generalized low-density (GLD) parity check codes. In [9] the greedy algorithms are used to construct approximate upper/lower triangular LDPC parity check matrix. Different with that approach, based on the structure of GLD parity check matrix, the authors proposed a systematic approach to construct approximate upper triangular $(N, 2, n)$ GLD parity check matrix H under the condition that no two constituent submatrices have more than one overlapping nonzero column.

Construction of H

Let the constituent code C_0 be an (n, k) code and its parity check matrix H_0 have systematic form $[I, P]$, where I is an $(n-k)$ by $(n-k)$ identity matrix. They defined N/n as s and $s\cdot(n-k)$ as L, respectively. The systematic construction approach of H can be shown in two steps:

1. Construct a matrix $\hat{H} = \left[\hat{H}^{1^T}, \hat{H}^{2^T}\right]^T$ where both $\hat{H}^1$ and $\hat{H}^2$ are L by N dimensional and contain s constituent submatrices.

2. Obtain H by reordering certain columns of $\hat{H}$.

First, $\hat{H}^1$ is constructed as $[I, SP]$, where I is an L by L identity matrix and SP is a block diagonal matrix containing s copies of submatrix P as shown in Fig. 7. It is noted that $\hat{H}^1$ can be seen as the parity check matrix of a super-code which consists of s constituent codes.

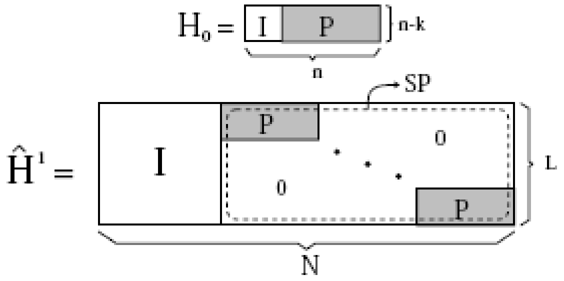

Fig. 7. Structure of matrix $\hat{H}^1$

$\hat{H}^2$ is constructed by permuting columns of matrix Q as shown in Fig. 8. They wrote matrix Q in block matrix form as $[Q_1, Q_2]$, where Q_1 and Q_2 are $L \times (N-L)$ and $L \times L$, respectively. By introducing two column permutations, π_1 and π_2, we construct $\hat{H}^2$ as $[\pi_1(Q_2), \pi_2(Q_1)]$. $\hat{H}^2$ also defines a super-code consisting of s constituent codes. Combining $\hat{H}^1$ and $\hat{H}^2$ together, a $(N;\ 2;\ n)$ GLD parity check matrix $\hat{H} = \left[\hat{H}^{1^T}, \hat{H}^{2^T}\right]^T$ is developed. Here π_1 and π_2 are chosen at random with the condition that no two constituent submatrices in $\hat{H}$ have more than one overlapping nonzero column. Based on the prerequisite that $N/n \geq n$ and the structure of $\hat{H}^1$ and Q, it can be proved that such two permutations always exist.

Since $\hat{H}_2^2 = \pi_2(\mathrm{Q}_1)$ and Q$_1$ contains $\left\lfloor\frac{N-L}{n}\right\rfloor = \left\lfloor\frac{s\cdot n - s\cdot(n-k)}{n}\right\rfloor = \left\lfloor\frac{s\cdot k}{n}\right\rfloor$ complete copies of systematic parity check matrix H_0, A column permutation π_3 can always be found which makes $H = \pi_3(\hat{H})$. The matrix has the approximate upper triangular form as shown in Fig. 9, in which each $P_i, i = 1, \cdots, s$, is obtained by removing some columns from matrix P. As shown in Fig. 9, H can be written as

$$H = \begin{bmatrix} T & B & D \\ A & C & E \end{bmatrix}$$

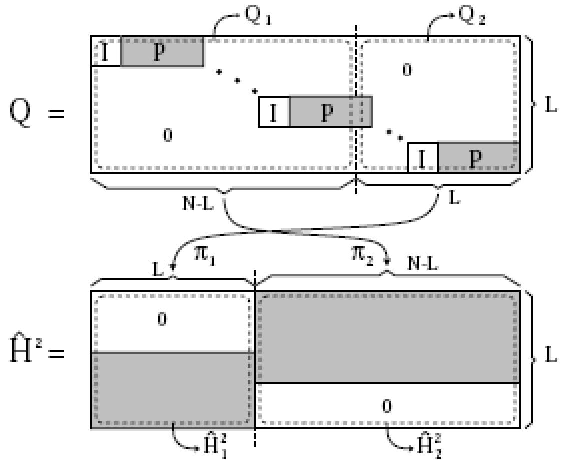

Fig. 8. Structure of matrix $\widehat{H}^2$

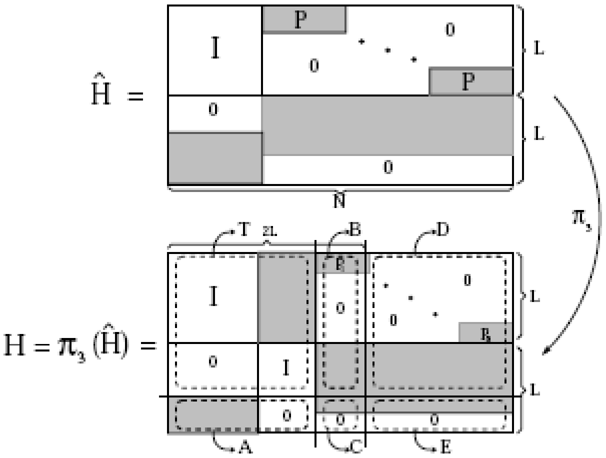

Fig. 9. Structure of matrix H

Encoding Process

i. Compute $y_C = D.x_c$ and $z_C = E.x_c$ which is efficient because both D and E are sparse;

ii. Solve $T.\hat{x}_a = y_C$. Since T has the form as shown in Fig 8, it can be proved that $T^{-1} = T$. Therefore, it can be written $\hat{x}_a = T.y_C$ which can be easily computed since T T is sparse;

iii. Evaluate $\hat{s} = A.\hat{x}_a + z_C$. which is also efficient since A A is sparse;

iv. Compute $x_b = \varphi \cdot \hat{s}$, where $\varphi = (A \cdot T \cdot B + C)^{-1}$. In this step, the complexity is scaled by $[(L - k \cdot Floor\left\{\frac{s-k}{n}\right\})]^2$.

v. Finally x_a can be obtained by solving $T \cdot x_a = B \cdot x_b + y_C$. Since $T^{-1} = T, x_a = T \cdot (B \cdot x_b + y_C)$. This is efficient since both T and B are sparse.

2.6 Two stage encoding with Triangular Factorization

Two stage encoding with Triangular Factorization (TSTF) algorithm shown in [34] explains the encoding in two steps.

1. Pre-computation step: Permute row vectors and column vectors of the parity check matrix H so that the H_2 part of H satisfies the LP condition, and the triangular matrices L and U with $H_2 = LU$ mod 2 are sparse.
2. Encoding step: Given an information vector $\boldsymbol{s}$ and parity check vector $\boldsymbol{p}$**,** the encoding stages are

   i. Compute $\boldsymbol{u}^T = H_1$.

   ii. Solve $H_2\boldsymbol{p}^T = \boldsymbol{u}^T$ after computing $\boldsymbol{v}^T = L^{-1}\boldsymbol{u}^T$ by back substitution for L and computing $\boldsymbol{p}^T = U^{-1}\boldsymbol{v}^T$ by back substitution for U.

3 Families of Algebraic Construction of QC-LDPC codes

3.1 Algebraic Construction of QC-LDPC codes: Bresnan Code

The paper in [10] discusses an algebraic construction for the regular and irregular QC-LDPC codes. The regular LDPC codes have the same number of ones in every row and column. The irregular LDPC codes have a different number of ones in columns and rows. The QC-LDPC codes consist of horizontally concatenated circulant sub-matrices. Each circulant sub-matrix is a square matrix for which every row is the cyclic shift of the previous row, and the first row is obtained by the cyclic shift of the last row. In this way, every column of each circulant sub-matrix is automatically the cyclic shift of the previous column, and the first column is obtained by the cyclic shift of the last column. The H matrix of dimension $(m \times L_m)$ for the QC-LDPC can be written as

$$H = [H_1 \; H_2 \; H_3 \; \cdots \; H_L] \tag{8}$$

where H_i is the i-th circulant sub-matrix of dimension $(m \times m)$, $i = 1, \cdots, L$. For the circulant matrices, the row weight and column weight are the same and fixed. Once the parity check matrix H is defined, the generator matrix is obtained. The matrices are created such that they should satisfy the constraint $GH^T = 0$. All the bits to be encoded are run through the generator matrix, and, therefore, all valid code words obey the property $CH^T = 0$ where C is the codeword.

The Quasi-Cyclic Generator matrix of rate $R = (L-1)/L$ has the following structure:

$$G = \begin{bmatrix} P_2^T & I_m & 0 & 0 & \cdots & 0 \\ P_3^T & 0 & I_m & 0 & \cdots & 0 \\ P_4^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & 0 & 0 & 0 & 0 & I_m \end{bmatrix}$$

As one of the requirements is $GH^T = 0$, we can write

$$GH^T = \begin{bmatrix} P_2^T & I_m & 0 & 0 & \cdots & 0 \\ P_3^T & 0 & I_m & 0 & \cdots & 0 \\ P_4^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & 0 & 0 & 0 & 0 & I_m \end{bmatrix} \times \begin{bmatrix} H_1^T \\ H_2^T \\ H_3^T \\ \vdots \\ H_L^T \end{bmatrix} = 0 \tag{9}$$

From the above relation, we can get $P_i = H_1^{-1} H_i$, where $i = 1 \cdots L$. The inverse of a circulant matrix is a circulant, and the product of two circulant matrices is also a circulant matrix.

Therefore, the QC-LDPC of different rates $(L-1)/L$ can be produced from the above-defined generator matrix G. By using this construction, the quasi-cyclic nature of generator matrix is preserved. Since the generator matrix is quasi-cyclic, the first row of each circulant sub-matrix is stored, and successive rows can be generated by a shift register generator. This greatly simplifies the encoder design. It is crucial that the circulant sub-matrix H_1 must be a nonsingular matrix. In order to maintain the non singularity of the circulant sub matrix H_1, polynomial representation of its first row should not be a factor of $x^m - 1$.

3.2 Algebraic Construction of QC-LDPC codes by Dispersion

In this section, a dispersion method for constructing QC-LDPC codes is presented for correcting erasure bursts [31]. The codes constructed by this method also perform well over the AWGN and binary random erasure channels. Consider a $4 \times 4k$ $\mathbf{H}_{\mathrm{EG}}(4,4k)$ subarray of an array $\mathbf{H}_{\mathrm{EG}}$ of circulant permutation matrices given by

$$\mathbf{H}_{\mathrm{EG}} = \mathbf{M}_{\mathrm{EG}}^{\mathrm{T}}$$

Where $\mathbf{H}_{\mathrm{EG}}$ is the transpose of $\mathbf{M}_{\mathrm{EG}}$ and $\mathbf{H}_{\mathrm{EG}}$ is a $q \times K$ array of $(q^{m-1}-1) \times (q^{m-1}-1)$ circulant permutation matrices and is a $q(q^{m-1}-1) \times K(q^{m-1}-1)$ matrix over $GF(2)$ with column and row weights q and K, respectively. Here $4 \leq q$ and $1 \leq k \leq \left\lfloor \frac{K}{4} \right\rfloor$. Dividing $\mathbf{H}_{\mathrm{EG}}(4,4k)$ into k 4×4 sub arrays as follows: $\mathbf{H}_{\mathrm{EG}}(4,4k) = [\mathbf{M}_0 \; \mathbf{M}_1 \cdots \; \mathbf{M}_{k-1}]$, where for $0 \leq j < k$

$$\mathbf{M}_j = \begin{bmatrix} \mathbf{A}_{0,4j} & \mathbf{A}_{0,4j+1} & \mathbf{A}_{0,4j+2} & \mathbf{A}_{0,4j+3} \\ \mathbf{A}_{1,4j} & \mathbf{A}_{1,4j+1} & \mathbf{A}_{1,4j+2} & \mathbf{A}_{1,4j+3} \\ \mathbf{A}_{2,4j} & \mathbf{A}_{2,4j+1} & \mathbf{A}_{2,4j+2} & \mathbf{A}_{2,4j+3} \\ \mathbf{A}_{3,4j} & \mathbf{A}_{3,4j+1} & \mathbf{A}_{3,4j+2} & \mathbf{A}_{3,4j+3} \end{bmatrix}$$

Since $\mathbf{H}_{\mathrm{EG}}(4,4k)$ satisfies the RC constraint, each subarray $\mathbf{M}_j$ also satisfies the RC constraint. From $\mathbf{M}_j$, we form an 8×8 array of circulant permutation and zero matrices, as shown below

$$\mathbf{D}_j = \begin{bmatrix} \mathbf{A}_{0,4j} & 0 & 0 & 0 & 0 & \mathbf{A}_{0,4j+1} & \mathbf{A}_{0,4j+2} & \mathbf{A}_{0,4j+3} \\ \mathbf{A}_{1,4j} & \mathbf{A}_{1,4j+1} & 0 & 0 & 0 & 0 & \mathbf{A}_{1,4j+2} & \mathbf{A}_{1,4j+3} \\ \mathbf{A}_{2,4j} & \mathbf{A}_{2,4j+1} & \mathbf{A}_{2,4j+2} & 0 & 0 & 0 & 0 & \mathbf{A}_{2,4j+3} \\ \mathbf{A}_{3,4j} & \mathbf{A}_{3,4j+1} & \mathbf{A}_{3,4j+2} & \mathbf{A}_{3,4j+3} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{A}_{0,4j+1} & \mathbf{A}_{0,4j+2} & \mathbf{A}_{0,4j+3} & \mathbf{A}_{0,4j} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{A}_{1,4j+2} & \mathbf{A}_{1,4j+3} & \mathbf{A}_{1,4j} & \mathbf{A}_{1,4j+1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A}_{2,4j+3} & \mathbf{A}_{2,4j} & \mathbf{A}_{2,4j+1} & \mathbf{A}_{2,4j+2} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{A}_{3,4j} & \mathbf{A}_{3,4j+1} & \mathbf{A}_{3,4j+2} & \mathbf{A}_{3,4j+3} \end{bmatrix}$$

$\mathbf{D}_j$ is called a *dispersion* of $\mathbf{M}_j$, and we can readily see that $\mathbf{D}_j$ also satisfies the RC constraint. Each submatrix in $\mathbf{D}_j$ is either a $(q^m - 1) \times (q^m - 1)$ circulant permutation matrix or a$(q^m - 1) \times (q^m - 1)$ zero matrix. $\mathbf{D}_j$ is an $8(q^m - 1) \times 8(q^m - 1)$ matrix over $GF(2)$ with both column and row weights four. Since each circulant permutation matrix in $\mathbf{D}_j$ is followed by four $(q^m - 1) \times (q^m - 1)$ zero matrices (including the end-around case), the zero span of $\mathbf{D}_j$ is at least $4(q^m - 1)$. Replacing each subarray $\mathbf{M}_j$ in $\mathbf{H}_{\mathrm{EG}}(4{,}4k)$ by its dispersion $\mathbf{D}_j$, we obtain an $8 \times 8k$ array of $(q^m - 1) \times (q^m - 1)$ circulant permutation and zero matrices, $\mathbf{H}_{\mathrm{EG,d}}(8{,}8k) = [\mathbf{D}_0\, \mathbf{D}_1 \cdots\, \mathbf{D}_{k-1}]$. $\mathbf{H}_{\mathrm{EG,d}}(8{,}8k)$ is an $8(q^m - 1) \times 8k(q^m - 1)$matrix over GF(2) with column and row weights 4 and , respectively, that satisfies the RC constraint and has a zero-covering span of length at least $4(q^m - 1)$ bits. $\mathbf{H}_{\mathrm{EG,d}}(8{,}8k)$ is called the dispersion of $\mathbf{H}_{\mathrm{EG}}(4{,}4k)$. The null space of $\mathbf{H}_{\mathrm{EG,d}}(8{,}8k)$ gives a QC-LDPC code $C_{qc,d}$ whose Tanner graph has a girth of at least six. The code is capable of correcting any erasure burst of length at least up to $4(q^m - 1) + 1$ bits.

3.3 Algebraic Construction of QC-LDPC codes: Rakibul Code

The H matrix for the QC-LDPC code proposed in [13] is written as

$$H = [H_{L-1} \cdots H_2\, H_1\, H_2 \cdots H_L] \qquad (10)$$

The Quasi-Cyclic Generator matrix of rate $R = 1/2$ has the following structure:

$$G = \begin{bmatrix} 0 & \cdots & 0 & P_2^T & I_m & 0 & \cdots & 0 \\ 0 & \cdots & P_3^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & \cdots & 0 & 0 & 0 & 0 & \cdots & I_m \end{bmatrix}$$

As one of the requirements is $GH^T = 0$, we can write

$$\begin{bmatrix} 0 & \cdots & 0 & P_2^T & I_m & 0 & \cdots & 0 \\ 0 & \cdots & P_3^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & \cdots & 0 & 0 & 0 & 0 & \cdots & I_m \end{bmatrix} \times \begin{bmatrix} H_{L-1}^T \\ \vdots \\ H_2^T \\ H_1^T \\ H_2^T \\ \vdots \\ H_L^T \end{bmatrix} = 0 \qquad (11)$$

From the above equation, several relations can be written

$$\begin{aligned} P_2^T H_1^T &= H_2^T \\ P_3^T H_2^T &= H_3^T \\ &\vdots \\ P_L^T H_{L-1}^T &= H_L^T \end{aligned} \qquad (12)$$

The previous equation concludes $P_i = H_{i-1}^{-1} H_i$, where $i = 2 \cdots L$. The inverse of a circulant matrix is circulant, and the product of two circulant matrices is also a circulant matrix. By using this construction, the quasi-cyclic nature of generator matrix is preserved. Since the generator matrix is quasi-cyclic, the first row of each circulant sub-matrix is stored, and successive rows can be generated by a shift register generator. The Bresnan code and Rakibul code can be compared and shown in Fig. 10.

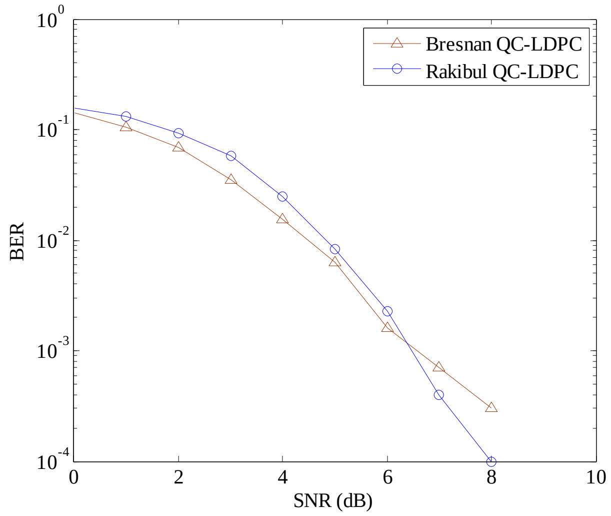

Fig. 10 Comparison between Bresnan and Rakibul QC-LDPC codes

4 Other Encoding Schemes

4.1 Encoding of QC-LDPC Codes Related to Cyclic MDS Codes

The authors in [33] presented an efficient systematic encoding algorithm for QC-LDPC codes that are related to cyclic maximum-distance separable (MDS) codes. They showed that the algebraic nature of the QC-LDPC codes related to cyclic MDS codes makes it possible to design a systematic encoding algorithm with linear time complexity. The algorithm can be easily implemented by using polynomial multiplication and division circuits. The division polynomials can be completely characterized by their zeros, and the sum of the respective numbers of their zeros will be equal to the parity-length of the codes.

The encoding procedure is shown below.

1) Input $c^{(0)}, c^{(1)}, \cdots, c^{(q)}$.

2) For $i = 0, 1, \cdots, r - 1$, compute $p^{(i)}$ as $p(i) \equiv \sum_{j=0}^{q-r} c^{(j)} \cdot \psi_{q-j}^{(i)} \; mod \; (x^{q-1} - 1)$, where $\deg(p^{(i)}) < q - 1$.

3) For $i = r - 1, r - 2, \cdots, 1, 0$, update $c^{(\mathrm{q-i})}$ by $c^{(q-i)} \leftarrow c^{(q-i)} + ((c^{(q-i)} + p^{(i)}) \; mod \; \psi_I^{(i)})$, and then update $p^{(0)}, p^{(1)}, \cdots, p^{(i-1)}$ by $p^{(j)} \leftarrow \left(p^{(j)} + c^{(q-i)} \psi_I^{(i)}\right) \mathrm{mod} \; (x^{q-1} - 1)$, where $j = 0, 1 \cdots, i - 1$.

4) Output $c^{(0)}, c^{(1)}, \cdots, c^{(q)}$.

4.2 Efficient Encoding of IEEE 802.11n LDPC Codes

Given a (sparse) parity check matrix H, the goal of encoding is to compute the systematic codeword c from the input sequence m. Owing to the special structure of the IEEE 802.11n LDPC parity check matrices, the encoding process can be done very efficiently. The IEEE 802.11n LDPC codes are based on block-structured LDPC codes with circular block matrices [28], i.e. the entire parity check matrix can be partitioned into an array of block matrices; each block matrix is either a zero matrix or a right cyclic shift of an identity matrix. The parity check matrix designed in this way can be conveniently represented by a base (block) matrix. The base matrix $\mathrm{H_b}$ for an IEEE 802.11n LDPC code with code length $N = 1944$ bits and $rate = 1/2$ can be seen from the standard. The block size is $Z = 81$ bits with $m_b = 12$ and $n_b = 24$. In this matrix, each entry represents a circular right shift of the identity matrix $\mathrm{I_Z}$. For example if $Z = 3$ and the entry is 1, then the corresponding block is [0 1 0; 0 0 1; 1 0 0]. The -1 entry means a null (all zero) block. In this way the above matrix is a compact expression of a binary $2D[M = 12 \times 81, N = 24 \times 81]$ matrix. Note that in the above matrix there are always three non -1 elements at the $\mathrm{k_b}$th column (usually they are 1 0 1). This property holds for all 12 LDPC codes defined in IEEE 802.11n. This observation, together with the (dual) diagonal parity check sub-matrix (the right-hand side of $\mathrm{H_b}$), can be explored to encode IEEE 802.11n LDPC codes efficiently.

The input information sequence is denoted as $\mathbb{m}$ and it is divided into $k_b = n_b - m_b$ groups of $\mathbb{Z}$ bits, i.e. $m = [m_0, m_1, \ldots, m_{k_b - 1}]$, where each element of $\mathbb{m}$ is a vector of length $\mathbb{Z}$. The parity sequence can also be grouped as length of $\mathbb{Z}$ bits. The codeword is denoted as

$$c_b = [mp] = [m_0, m_1, \ldots, m_{k_b - 1}, p_0, p_1, \ldots, p_{m_b - 1}]$$

Recall that a codeword has to satisfy $\mathrm{H_b c_b} = 0$. Expanding the above equation, the following equations hold:

$$\sum_{j=0}^{k_b-1} h_{0,j}m_j + \pi_1 p_0 + p_1 = 0\ (0^{th}\ row)$$
$$\sum_{j} h_{i,j}m_j + p_i + p_{i+1} = 0\ (i \neq 0, x, m_b - 1)$$
$$\sum_{j} h_{x,j}m_j + p_0 + p_x + p_{x+1} = 0\ (x^{th}\ row)$$
$$\sum_{j} h_{m_b-1,j}m_j + \pi_1 p_0 + p_{m_b-1} = 0\ ((m_b-1)^{th}\ row) \tag{13}$$

where $\pi_1 p_0$ makes p_0 circular shift 1-cycle. Adding up all the above equations, we have

$$p_0 = \sum_{i=0}^{m_b-1}\sum_{j=0}^{k_b-1} h_{i,j}m_j$$

$\lambda_i = \sum_{j=0}^{k_b-1} h_{i,j}m_j$ for $i = 0, \cdots, m_b - 1$, the above equation becomes $p_0 = \sum_{i=0}^{m_b-1}\lambda_i$. With p_0 in hand, p_1 and p_{m_b-1} can be easily obtained from (13)

$$p_1 = \lambda_0 + \pi_1 p_0$$

$$p_{m_b-1} = \lambda_{m_b-1} + \pi_1 p_0$$

Other parity sub-vectors can be solved by upward and downward recursions, according to (13). In summary, $\lambda_i = \sum_{j=0}^{k_b-1} h_{i,j}m_j$ and $\sum_{i=0}^{m_b-1}\lambda_i$ are needed to get the codeword c. Since $h_{i,j}m_j$ is nothing but a circular shift of m_j, the resource requirement is trivial.

4.3 Encoding of Array LDPC Codes

The authors in [30] started with a code that satisfies the condition $Hv^T = 0$ and therefore defined

$$C_A := \{v = (v_0, v_1, \cdots, v_{k-1}) \in F_2^{pk} \,|Hv^T = 0\,\}. \tag{14}$$

They called the code C_A an *array LDPC code*. The code C_A is a (*j*, *k*)-regular LDPC code.

The array LDPC code C_A has an algebraic characterization similar to that of RS codes. Although an RS code is defined over a finite field, the array LDPC code can be defined over a ring. Then they defined a subcode $C_A^{'} \subset CA$ as follows:

$C_A^{'} := \{A(z)G(z)\ |A(z) \in Rp[z]$ s.t. $\deg A(z) < k - j\}$.

$G(z)$ can be considered as a "generator polynomial" of the sub-code $C_A^{'}$. Note that $C_A^{'}$ has length $N = pk$, dimension $K1 := p(k - j)$ and rate $R := K1/N = 1 - j/k$, which is the so-called design rate of (j, k)-regular LDPC codes. The dimension of $C_A^{'}$ is smaller than that of C_A by $j - 1$, but in practice j is small, e.g., $3 \leq j \leq 6$, so that the loss of the information rate is negligible.

Let $\boldsymbol{u} := (\boldsymbol{u_0}, \boldsymbol{u_1}, \ldots, \boldsymbol{u}_{k-j-1})$ be an information vector, where$\boldsymbol{u_i} = (u_{i,0},\ u_{i,1}, \ldots, u_{i,p-1}) \in F_p^2, i = 0, 1, \ldots, k - j - 1$. For each $\boldsymbol{u_i}$, we define $u_i\ (\alpha) := \sum_{s=0}^{p-1} u_{i,s}\ \alpha s \in Rp$. First, construct the information polynomial $U(z)$ as follows:

$$U(z) = \sum_{i=0}^{k-j-1} u_i(\alpha)z^i.$$

Next, compute the residue $R(z)$ of $z\,jU(z)$ modulo $G(z)$, i.e.,

$$R(z) \equiv z^j\ U(z)\ mod\ G(z).$$

Finally, set $V(z) := z^j\ U(z) - R(z)$. Then $V(z) \in C_A^{'}$. Note that since the leading coefficient of $G(z)$ is 1, no divisions in the ring Rp are required. This encoding algorithm can be implemented on digital circuits.

5 Conclusion

QC-LDPC code has been the focus of interest for the last few years. Being the low complexity counterpart of the LDPC code, QC-LDPC has successfully drawn the attention of the potential researchers. Encoding in the LDPC code has been the most critical part in low complexity applications. Decoding can be performed at the fixed node and encoding is crucial for multi-hop transmission. This paper discusses several encoding techniques which may be considered for energy aware low complexity applications, such as in wireless sensor network. The future work of this paper is to develop an energy efficient encoding scheme for energy constraint wireless sensor network.

References:

[1] R. G. Gallager, “Low Density Parity Check Codes,” IRE transactions on Information Theory, IT-8: 21-28, January 1962.

[2] R. G. Gallager, Low-Density Parity-Check Codes, Cambridge, MA: MIT Press, 1963.

[3] D. J. C. Mackay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” IEE Electron Letter, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.

[4] D. J. C. Mackay, “Good error-correcting codes based on very sparse matrices, “IEEE Trans. Inform. Theory, vol. IT-45, no. 2, pp. 399-431, March 1999.

[5] N. Wiberg, “codes and decoding on general graphs,” Linkoeping studies in science and technology, No. 440, 1996.

[6] T. J. Richardson, A. Shokrollahi, and R. Urbanke, “Design of capacity approaching low-density parity-check codes,” IEEE Trans. Inform. Theory, vol. 47, pp. 619-637, Feb. 2001.

[7] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” Proc. 30th Annu. ACM Symp. Theory of computing, 1998, pp. 249-258.

[8] F. R. Kschischang, “Codes defined of graphs,” IEEE Commun. Mag, Vol. 41, no. 8, pp. 118-125, Aug. 2003.

[9] T. J. Richardson, and R. Urbanke, “Efficient encoding of low-density parity-check codes,” IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 638-656, Feb. 2001.

[10] Richard Bresnan, “Novel code construction and decoding techniques for LDPC codes”, Master’s thesis, Dept. of Elec. Eng., UCC Cork, 2004.

[11] M.P.C. Fossorier, “Quasi-cyclic low density parity check codes from circulant permutation matrices,” IEEE Trans. Inform. Theory, vol.50, pp. 1788-1794, Aug. 2004.

[12] M. R. Islam and J. Kim, “Linear encoding of LDPC codes using approximate lower triangulation with postprocessing”, Personal, Indoor and Mobile Radio Communications Symposium (PIMRC), Tokyo, Japan, September 13-16, 2009

[13] M. R. Islam and J. Kim, “Quasi Cyclic Low Density Parity Check Code for High SNR Data Transfer,” Journal of Radio Engineering, vol. 19, no. 2, 2010

[14] C. Yoon, J. Oh, M. Cheong and S. Lee, “A hardware efficient LDPC encoding scheme for exploiting decoder structure and resources”, pp 2445-2449, VTC 2007-spring.

[15] C. Yoon, J. Oh, M. Cheong and S. Lee, “Arbitrary Bit Generation and Correction Technique for Encoding QC-LDPC Codes with Dual-Diagonal Parity Structure”, pp 663-667, WCNC 2007.

[16] M. Jayabalan and H. M. Kwon, “An improved quasi-cyclic low-density parity-check code for memory channels,” VTC 2004-fall.

[17] L. Lan, L. Zeng, Y. Y. Tai, L. Chen, S. Lin, and K. A. Ghaffar, “Construction of Quasi-Cyclic LDPC Codes for AWGN and Binary Erasure Channels: A Finite Field Approach” IEEE Transactions on Information Theory, vol. 53, no. 7, July 2007.

[18] M. Arabaci, and I. Djordjevic, “An Alternative FPGA Implementation of Decoders for Quasi-Cyclic LDPC Codes”, TELFOR, 2008.

[19] Y. Sun, M. Karkooti and J. R. Cavallaro, “VLSI Decoder Architecture for High Throughput, Variable Block-size and Multi-rate LDPC Codes”, ISCAS 2007.

[20] M. Hagiwara and H. Imai, “Quantum Quasi-Cyclic LDPC Codes”, IEEE International Symposium on Information Theory, June 2007.

[21] M. Hsieh, T. Brun, and I. Devetak, “Quantum Qusi-Cyclic Low-Density Parity-Check Codes”, 2008. http://arxiv.org/abs/0803.0100v1

[22] S. Zhao, B. Zheng, and W. Wang, “Construction of Quantum Low Density Parity Check Code Based on Quasi-cyclic Sparse Sequence”, International Conference on Communications and Networking in China, 2008.

[23] X. Wu, X. You, and C. Zhao, “A necessary and sufficient condition for determining the girth of Quasi-Cyclic LDPC Codes”, IEEE Transactions on communications, vol. 56, no. 6, pp. 854-857, June 2008.

[24] G. Malema and M. Liebelt, “Quasi-Cyclic LDPC Codes of Column-Weight Two Using a Search Algorithm”, EURASIP Journal on Advances in Signal Processing, 2007. doi:10.1155/2007/45768

[25] Y. Wang, J. S. Yedidia, and S. C. Draper, “Construction of High-Girth QC-LDPC Codes”, International Symposium on Turbo Codes and Related Topics, 2008.

[26] S. Kim, J. S. No, H. Chung and D. J. Shin, “Cycle Analysis and Construction of

Protographs for QC LDPC Codes with girth larger than 12", IEEE International Symposium on Information Theory, June 2007.

[27] Z. Cai, J. Hao, P.H. Tan, S. Sun and P.S. Chin, "Efficient encoding of IEEE 802.11n LDPC codes", Electronics Letters , Volume 42, Issue 25, pp. 1471-1472, December 2006. doi:10.1049/el:20063126

[28] H. Zhong and T. Zhang, "Block-LDPC: a practical LDPC coding system design approach," IEEE Trans. Circuits Syst., 2005, 52, (4), pp. 766–775.

[29] S. Lin, D. J. Costello, Error control coding, Pearson prentice hall, 2004.

[30] H. Fujjita and K. Sakaniwa, "Some Classes of Quasi-Cyclic LDPC Codes: Properties and Efficient Encoding Method ," IEICE Transaction on Fundamentals, vol.E88–A, no.12 December 2005

[31] Y. Y. Tai, L. Lan, L. Zeng, S. Lin and K. A. S. Abdel-Ghaffar, "Algebraic Construction of Quasi-Cyclic LDPC Codes for the AWGN and Erasure Channels," IEEE Transaction on Communications, vol. 54, no. 10, pp. 1765-1774, October 2006.

[32] Hanghang Qi, Norbert Goertz, "Low-Complexity Encoding of LDPC Codes: A New Algorithm and its Performance,“ available at publik.tuwien.ac.at/files/PubDat_166941.pdf, (06. 04. 2011).

[33] Tong Zhang and Keshab K. Parhi, "A class of efficient-encoding generalized low-density parity-check codes," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001

[34] Y. Kaji, "Encoding LDPC codes using the Triangular Factorization", IEICE Transaction on Fundamentals, vol. E-89 A, no. 10, pp. 2510-2518, October 2006

Mohammad Rakibul Islam received the B.Sc.Engg. and M.Sc.Engg. degree in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 1998 and 2004 respectively. He also received MBA degree in Marketing from the Institute of Business Administration (IBA) under the University of Dhaka in 2006. He received his PhD degree in the department of Electronics and Radio Engineering from Kyung Hee University, South Korea in the year 2010. He joined the Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT) as a faculty on 1999 and serving as a Professor there. His research interests include cooperative technique for wireless sensor networks, LDPC and QC-LDPC codes, secrecy capacity and other wireless applications.

Syed Iftekhar Ali received his B.Sc. and M.Sc. engineering degrees in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in 1999 and 2002 respectively. He also received Master of Applied Science (MASc) in Electrical and Computer Engineering from University of Waterloo, Waterloo, Canada in 2004. Currently he is an Assistant Professor in Electrical and Electronic Engineering Department, Islamic University of Technology (IUT), Gazipur, Bangladesh. He is also a part-time PhD student in the Department of Electrical and Electronic Engineering, BUET, Dhaka.