IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 August 2024, accepted 12 October 2024, date of publication 24 October 2024, date of current version 4 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3485875

RESEARCH ARTICLE

Highly Reliable and Secure System With Multi-Layer Parallel LDPC and Kyber for 5G Communications

LINH NGUYEN, (Member, IEEE), QUOC BAO PHAN, (Member, IEEE), AND TUY TAN NGUYEN, (Member, IEEE)

School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011, USA

Corresponding author: Tuy Tan Nguyen (tuy.nguyen@nau.edu)

This work was supported by the National Science Foundation under Grant No. 2348464.

ABSTRACT The development of fifth-generation (5G) technology marks a significant milestone for digital communication systems, providing substantial improvements in data transmission speeds and enabling enhanced connectivity across a wider range of devices. However, this rapid increase in data volume also introduces new challenges related to transmission latency, reliability, and security. This paper introduces KyMLP-LDPC, a novel approach that integrates a multi-layer parallel LDPC (MLP-LDPC) algorithm with Kyber, a post-quantum cryptography scheme, to accelerate and enable reliable and secure transmission. MLP-LDPC partitions the LDPC parity check matrix into processing groups to streamline parallel decoding and minimize message collisions during transmission, thereby accelerating error correction operations. Kyber encrypts data preemptively to safeguard against potential attacks. The effectiveness of our proposed method is evaluated using both image data and signals transmitted through an additive white Gaussian noise communication channel. Evaluation results demonstrate that the proposed method achieves superior performance in terms of error correction capabilities and data security compared to existing approaches.

INDEX TERMS Digital communication systems, LDPC codes, quantum computing, Kyber, 5G communication, security, reliability.

I. INTRODUCTION

Digital communication systems (DCSs) play a key role in the rapid growth of internet services, social media platforms, streaming services, and cloud-based applications, causing a sharp increase in data traffic [1], [2]. These systems provide flexible and efficient data processing options for large amounts of data, meeting the growing demand for seamless connectivity and high-speed internet access. The advancements in fifth-generation (5G) technology further enhance this capability. With higher data transfer speeds and lower latency than previous generations, 5G facilitates faster and more effective access and interaction with online services on mobile devices [3], [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Shuangqing Wei.

A basic DCS consists of a data source, source encoding, encryption, channel encoding, modulation, channel, demodulation, channel decoding, decryption, source decoding, and finally, the data sink [5]. Two important processes that ensure reliable, secure, and efficient communications are channel coding and cryptography. Cryptographic techniques are particularly crucial as quantum technology advances. These techniques need to be resistant to attacks from quantum computers to safeguard the security of DCSs [6], [7]. Additionally, channel coding techniques, when combined with cryptography, must still maintain system performance. This ensures that DCSs remain secure and efficient even in the face of evolving threats, including those posed by quantum computing.

Channel coding, employing error correction algorithms such as polar codes, low-density parity-check (LDPC) codes, turbo codes, safeguards data integrity during transmission

over noisy channels. Among these, LDPC codes, a pioneering invention by Gallager [8], have gained recognition for their remarkable ability to correct errors during data transmission [9], [10]. Widely employed in telecommunications, data storage, and digital broadcasting [11], [12], LDPC codes are valued for their exceptional error correcting capabilities. They can correct a large number of errors and achieve performance closer to the Shannon limit for additive white Gaussian noise (AWGN) channels compared to turbo codes, despite having similar decoding complexity [13], [14].

In 5G new radio (NR), LDPC codes have been specifically selected due to their robust error correction capabilities, which meet the demands for high data rates and low latency [15], [16], [17], [18]. LDPC codes enable reliable data transmission at the core of 5G NR systems. Their flexibility in code rate and block size adaptation is crucial for fulfilling the diverse requirements of 5G applications, spanning enhanced mobile broadband (eMBB) [19], ultra-reliable low-latency communication (URLLC) [20], [21], and massive machine-type communication (mMTC) [22].

LDPC codes achieve impressive error correction with efficient decoding algorithms such as belief propagation (BP), also known as the sum-product algorithm (SPA) [14], [23]. The decoding process is performed on a special type of graph called a bipartite graph as known as a Tanner graph [24], [25]. In this graph, variable nodes (VNs) represent individual code bits and check nodes (CNs) represent parity-check constraints. Beginning the decoding process, the VNs are updated with a log-likelihood ratio (LLR) based on the received channel information. This LLR reflects the likelihood that each bit is '0' or '1'. The LLR values are then propagated to CNs to calculate new LLR based on parity-check equations. These updated values are then sent back to the VNs. This iterative exchange of information (message passing) continues until the decoding process converges or reaches a predetermined number of iterations. Based on the last updated LLR, the bit values are then determined.

To achieve faster LDPC decoding in practical applications, simplifying decoding algorithms is essential. The min-sum algorithm (MSA) provides a computationally efficient by focusing on minimal values, MSA streamlines calculations, making it ideal for real-time scenarios [18], [26]. For further efficiency gains while maintaining good error correction, algorithms including offset min-sum (OMS) and normalized min-sum (NMS) have been developed [27]. Scheduling algorithms such as flooding and layered scheduling introduce another layer of efficiency improvement [28]. Layered scheduling updates nodes sequentially, often leading to faster convergence and better error correction by utilizing the latest information. However, this sequential approach introduces a trade-off. While it simplifies decoding, it can also lead to significant decoding latency per iteration, as indicated in [29]. To address the limitations of sequential processing, layered parallel LDPC (LP-LDPC) decoding, as presented in [30], allows concurrent processing of all layers. In LP-LDPC, each layer can exchange messages with others simultaneously. This parallelism comes at a cost, though, as it introduces data conflicts during LLR updates because multiple check nodes might connect to the same variable node during concurrent processing, posing new challenges for LDPC decoding.

Cryptography secures data during transmission, particularly over wireless channels susceptible to eavesdropping and information loss, referenced in studies [17], [31], [32]. Various cryptographic techniques, including the data encryption standard (DES) and advanced encryption standard (AES) [33], [34], have been instrumental in safeguarding data privacy and confidentiality. However, the emergence of quantum computers poses a significant threat to traditional public-key cryptography, rendering these techniques vulnerable to attacks [35]. In August 2024, Kyber was selected by the National Institute of Standards and Technology (NIST) as a key-encapsulation mechanism (KEM) standard to enhance data security for the quantum era [36]. Built upon the hardness of the learning with errors (LWE) problem on module lattices, Kyber offers robust IND-CCA2 security [37], guaranteeing protection against adaptive chosen ciphertext attacks, a powerful attack strategy [38]. Kyber demonstrates robust security features alongside practical advantages in real-world applications. It is computationally efficient, offering high performance without sacrificing security. Its scalability facilitates seamless integration into DCSs, adapting flexibly to diverse security needs.

In this paper, we focus on developing a new decoding method to address latency in traditional layered methods and message passing conflicts in parallel decoding. In addition, we evaluate the possibility of integrating the new decoding method with the Kyber algorithm. Our contributions can be summarized as follows:

1) We propose a novel method for constructing multi-layered set, replacing the conventional layer set in layered LDPC decoding algorithms. It partitions layers into concurrent processing sets while maintaining seamless message passing, similar to traditional layered LDPC decoding. By partitioning layers into sets, our method enables efficient concurrent processing. This strategy significantly improves computational resources, enhances decoding speed, and maintains effective message passing between layer sets.
2) We propose a specialized multi-layer parallel LDPC (MLP-LDPC) decoding method for 5G NR communications that utilizes a multi-layered set structure to optimize LDPC decoding efficiency, addressing latency issues and minimizing message passing conflicts.
3) We introduce KyMLP-LDPC, an integrated encoding system integrating MLP-LDPC decoding with the Kyber post-quantum cryptography algorithm. This approach enhances data security while efficiently handling transmission conflicts in LP-LDPC decoding.
4) We assess the proposed system by testing how it performs with both image and signal data, demonstrating its effectiveness across different scenarios and offering

detailed insights into its practical applications and benefits.

The subsequent sections of the paper are organized as follows: Section II introduces the mathematical background of the employed techniques. Section III illustrates the architecture and optimization strategies implemented in the proposed system. Section IV presents the results obtained from simulations performed on signal and image data. Lastly, Section V encapsulates the conclusions drawn from the study.

II. BACKGROUND

A. 5G QC-LDPC CODES

In 5G NR communication, quasi-cyclic low-density parity-check (QC-LDPC) codes is a crucial component, having been accepted by the 3rd generation partnership project (3GPP) TS 38.212 [16] as the channel coding scheme for the enhanced mobile broadband (eMBB) data channel. QC-LDPC codes, a class of structured LDPC codes that allow low-complexity encoding and decoding, are widely used due to their low implementation complexity [39], [40], [41], [42].

Two base graph (BG) matrices, BG1 and BG2, are used in 5G NR [43], each optimized for different block lengths K and code rates R. BG1 handles larger block lengths ranging from 500 to 8448 and higher code rates between 1/3 and 8/9, while BG2 targets smaller block lengths in the range of 40 to 2560 and lower code rates between 1/5 and 2/3. Both BG1 and BG2 are structured matrices comprised of five submatrices: **A**, **B**, **O**, **C**, and **I**. Each submatrix has a specific function within the overall LDPC code construction. Submatrix **A**, responsible for information bits, maintains a fixed size of 22 columns ($k_b = 22$) in BG1. However, for BG2, the number of information bit columns k_b dynamically adjusts based on the block length K of the information bits. For larger block lengths, specifically those in the range greater than 640, k_b is set to 10 information bit columns. As K progressively decreases, k_b also gradually diminishes. It takes on values of 9 for K in the range 560 to 640, inclusive, then 8 for K in the range 192 to 560, inclusive, and finally 6 for K in the range 192 or less. Any remaining columns within submatrix **A** are filled with zeros (zero-padding) to ensure consistent matrix dimensions. Submatrix **B**, a square matrix, contains the parity bits and exhibits a specific bi-diagonal structure. The first column of **B** has a weight of 3. Subsequent columns display an upper bi-diagonal pattern, meaning they connect to information bits diagonally above them in the matrix. Alongside the main submatrices **A** and **B**, there are additional submatrices known as extensions. These include the **O** submatrix, a zero submatrix that serves a structural purpose; the **I** submatrix, an identity matrix used to identify information bits; and the **C** submatrix, which has a specified structure contributing to the overall matrix. The values within the BG1 (BG2) matrices are populated by manual insertion based on the corresponding entries our Table 5 (5.3.2-3) of the 3GPP document [16].

The shift matrix **P** is constructed from the base graph matrix using modulo Z_c arithmetic, where Z_c represents the size of the circulant permutation submatrices. These submatrices are square matrices where each row is a cyclic shift of the row above it. The value $P_{m,n}$ in the base graph matrix dictates the amount of shift applied to the corresponding circulant permutation matrix. When $P_{m,n} \geq 0$, the permutation matrix is obtained by shifting the identity matrix $\mathbf{I}_{Z_c}$ by $P_{m,n}$ positions. If $P_{m,n} = -1$, the corresponding submatrix is a zero matrix. The lifting size Z_c is given by $Z_c = a \times 2^z$, where a takes specific values (2, 3, 5, 7, 9, 11, 13, or 15) and z ranges from 0 to 7, depending on the value of a, corresponding to the exponent matrices set shown in Table 1. The combination of these values of a and z results in 51 distinct values of Z_c, as indicated in Table 2. These Z_c values ensure there are enough columns of information bits k_b to accommodate the desired information bits length K. The critical relationship here is $k_b \times Z_c \geq K$. Selecting the smallest Z_c that meets this criterion is crucial for constructing efficient codes while satisfying to the thorough LDPC code requirements established by 3GPP for 5G NR.

TABLE 1. Relationship between exponent matrices and lifting size Z_c.

Exponent Matrix	Lifting Size Set $\{a \times 2^z\}$	Exponent Matrix	Lifting Size Set $\{a \times 2^z\}$
Set 1 (a=2)	$\{z = 0, 1, 2, 3, 4, 5, 6, 7\}$	Set 5 (a=9)	$\{z = 0, 1, 2, 3, 4, 5\}$
Set 2 (a=3)	$\{z = 0, 1, 2, 3, 4, 5, 6, 7\}$	Set 6 (a=11)	$\{z = 0, 1, 2, 3, 4, 5\}$
Set 3 (a=5)	$\{z = 0, 1, 2, 3, 4, 5, 6\}$	Set 7 (a=13)	$\{z = 0, 1, 2, 3, 4\}$
Set 4 (a=7)	$\{z = 0, 1, 2, 3, 4, 5\}$	Set 8 (a=15)	$\{z = 0, 1, 2, 3, 4\}$

TABLE 2. Lifting size Z of standard 5G QC-LDPC codes.

Z		z							
		0	1	2	3	4	5	6	7
a	2	2	4	8	16	32	4	128	256
	3	3	6	12	24	48	96	192	384
	5	5	10	20	40	80	160	320	_
	7	7	14	28	45	112	224	_	_
	9	9	18	36	72	144	288	_	_
	11	11	22	44	88	176	352	_	_
	13	13	26	52	104	208	_	_	_
	15	15	30	60	120	240	_	_	_

After determining the Z_c, the next step involves constructing the parity check matrix, denoted **H**. Each element of **P** is expanded into the corresponding circulant permutation matrix, resulting in the structure of **H** as shown in Fig. 1.

To ensure efficient data transmission over the encoded channel, LDPC codes undergo a final optimization step called rate matching. This process adjusts the code rate to match the channel capacity by employing two techniques shortening and puncturing [44]. The first step involves puncturing (removing) the first two sets of $2Z_c$ columns. These initial columns are densely packed with information, meaning errors in these bits can easily cascade and disrupt the entire message. By removing them, we prioritize the integrity of the remaining data for more reliable communication. The second step removes a portion of the parity bits on the right side of the codeword. The shortening process can be performed by

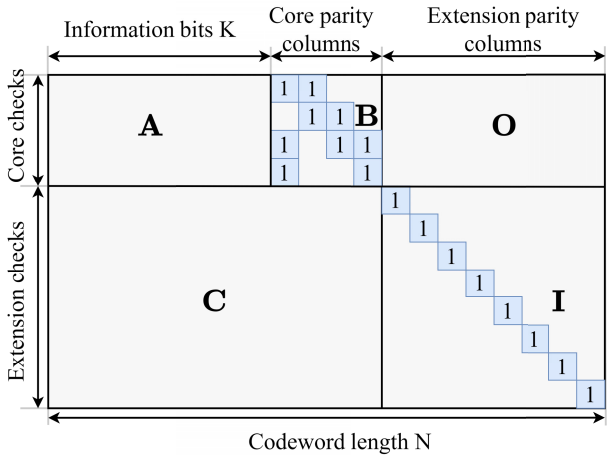


FIGURE 1. Illustration of the fundamental parity-check structure for the 5G NR QC-LDPC codes.

removing $k_b \times Z_c - K$ zero-padding columns within submatrix **A**. This targeted puncturing reduces the overall codeword length while simultaneously increasing the code rate.

TABLE 3. Relationship between base matrix dimensions and code rate in 5G QC-LDPC.

Code Rate	BG1		BG2		Code Rate	BG1		BG2	
(R)	Rows	Cols	Rows	Cols	(R)	Rows	Cols	Rows	Cols
$\frac{1}{5}$	_	_	42	52	$\frac{2}{3}$	13	35	7	17
$\frac{1}{3}$	46	68	22	32	$\frac{22}{30}$ ($\sim\frac{3}{4}$)	10	32	_	_
$\frac{2}{5}$	35	57	17	27	$\frac{22}{27}$($\sim\frac{5}{6}$)	7	29	_	_
$\frac{1}{2}$	24	46	12	22	$\frac{22}{25}$($\sim\frac{8}{9}$)	5	27	_	_

Table 3 shows the dimension of the base matrix corresponding to the coding rate. The amount of punctured parity bits depends on the desired coding rate. Since the dimension of the shift matrix **P** is $m_b \times n_b$, the number of first punctured bits N_{p_1} is $2Z_c$. The number of shortening bits N_s is calculated as $k_b \times Z_c - K$. Finally, the number of second punctured bits N_{p_2} is $n_b \times Z_c - 2Z_c - N - N_s$ where N is the total codeword length after rate matching.

B. LAYERED LDPC DECODING ALGORITHM

Assuming we have a matrix **H** of size $M \times N$, partitioned into $\mathcal{L}$ horizontal decoding layers. Each layer contains Z_c consecutive rows of matrix **H**. As a result, any variable node is connected at most once to any layer. We denote $\mathcal{R}_l$ as the set of consecutive rows of **H** corresponding to layer l (where l ranges from 1 to $\mathcal{L}$).

We consider a binary codeword $\mathbf{c} = (c_1, c_2, \ldots, c_N)$. Let y_i (where i ranges from 1 to N) represent the corresponding received bit from the channel. For j going from 1 to M, let $r_{j,i}^k$ denote the check-to-variable (CTV) message from check node j to variable node i during the k-th iteration. Similarly, let $q_{j,i}^k$ denote the variable-to-check (VTC) message from variable node i to check node j. Let $\mathcal{V}(j)$ denote the set of VNs connected to check node j by parity check constraints, and $\mathcal{C}(i)$ denote the set of CNs connected to variable node i. The set $\mathcal{V}(j) \setminus i$ denotes $\mathcal{V}(j)$ excluding variable node i, and $\mathcal{C}(i) \setminus j$ denotes $\mathcal{C}(i)$ excluding check node j.

The following outlines the main steps of the layered LDPC decoding process using the MSA algorithm [44]:

1) *Initialization:* Each variable node i load the a priori probability p_i, which is computed by:
$$p_i = \log \frac{P(c_i = 0 \mid y_i)}{P(c_i = 1 \mid y_i)} = \frac{2}{\sigma^2} y_i. \tag{1}$$
Each CTV message $r_{j,i}$ where $i \in \mathcal{V}(j)$ is initially set to zero.
2) *Iteration loop:* These steps below are performed sequentially for each layer in the LDPC code, starting from the first layer. With $j \in \mathcal{R}_l$ and $i \in \mathcal{V}(j)$:
   a) *Variable node update:* Calculate the VTC message $q_{j,i}^k$ at the $k-$th iteration by:
   $$q_{j,i}^k = p_i + r_{j,i}^{k-1}. \tag{2}$$
   b) *Check node update:* Calculate the CTV message $r_{j,i}^k$ with $i' \in \{\mathcal{V}(j) \setminus i\}$ at the $k-$th iteration by:
   $$r_{j,i}^k = \left(\prod_{i'} \operatorname{sgn}(q_{j,i'}^k) \right) \times \left(\min_{i'} \{ |q_{j,i'}^k| \} \right). \tag{3}$$
   c) *Posteriori information update:* Calculate the a-posteriori probability (APP) as follows:
   $$p_i^k = q_{j,i}^k + r_{j,i}^{k-1}. \tag{4}$$
3) *Decision:* Decide the $i-$th bit of the decoded codeword $c_i = 0$ if $p_i > 0$ and $c_i = 1$ otherwise. The decoding process terminates when the entire codeword $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ satisfy the parity check equations: $\mathbf{H} \times \mathbf{c}^T = 0^T$, or the preset maximum number of iterations is reached.

C. KYBER

Kyber [36] caters to diverse security needs by offering three distinct security levels: Kyber-512, Kyber-768, and Kyber-1024. Kyber-512 provides a security level comparable to AES-128, offering adequate protection against current classical attacks. However, its strength might not be sufficient to withstand the potential power of quantum computers. Kyber-768 strikes a commendable balance between performance and security. Its security level aligns with AES-192, and it offers more than 128 bits of security against potential quantum attacks. This makes it a compelling choice for many applications. Finally, Kyber-1024 delivers the highest level of security, comparable to AES-256. This option is ideal for scenarios demanding the utmost protection, such as safeguarding highly sensitive information. Its robust security, efficiency, scalability, and flexibility make it a valuable tool for securing communication in the quantum age.

The Kyber encryption algorithm involves three main steps:

1) *Key Generation*:

- Generate public-private key pair: (pk, sk).
- Compute inverses modulo q: u^{-1}, v^{-1}, and w^{-1} where u, e, v, and w are random polynomials.
- Public key: pk $= (v, w)$, private key: sk $= u$.

2) *Encryption*:

- Add error terms: $d = (vr+e) \mod q$ with random r and q.
- Ciphertext: $h = (d, w)$.

3) *Decryption*:

- Use the sk to decode: $g = (hu^{-1}) \mod q$.
- Recover the original polynomial: $g' = (gw^{-1}) \mod q$.

In this paper, we suggest partitioning the data into segments of length $K = 256$, which matches the input length for Kyber encryption [37]. This approach enables us to generate a key directly from the data string, eliminating the need for subdividing it into smaller components. This ensures both the integrity and efficiency of the encryption process during system performance testing.

III. PROPOSED SYSTEM

A. PROPOSED MULTI-LAYER PARALLEL LDPC DECODING ALGORITHM

A comparison of message passing flow between three approaches—layered LDPC decoding, LP-LDPC decoding, and proposed MLP-LDPC decoding—is presented in Fig. 2. With the traditional layered LDPC algorithm, the message passing process starts from the first layer containing the information bits and then updates the LLR values after calculations to the subsequent layers. During this process, the LLR values of the layers depend on the values computed from the previous layers, creating a tight linkage between the layers.

In contrast, with the LP-LDPC algorithm, the layers are processed independently and simultaneously. The dependency between the layers is maintained by immediately sending and updating the LLR values as soon as any layer computes a p_i value, creating a new LLR that contains the newly calculated values. These updated values are then used as input for layers that have not yet calculated the LLR values at position i. As we mentioned above, conflicts can arise when updating a message passing LLR between two or more different layers in the decoding process. Fig. 3 exemplifies such a conflict, where a variable node i (represented by LLR value p_i) is connected to check nodes from both layers l_1 and l_2.

For the proposed MLP-LDPC algorithm, we retain the advantage of message passing from the upper layers to the subsequent layers of layered-LDPC and the independent processing of LP-LDPC by dividing the layers into appropriate sets, creating a multi-layered set $\mathcal{S}$ as shown in Algorithm 1. For each set, we apply the LP-LDPC decoding algorithm, starting from the first set of $\mathcal{S}$, then taking the computed LLR results from set $\mathcal{S}_1$ as input for set $\mathcal{S}_2$. This process

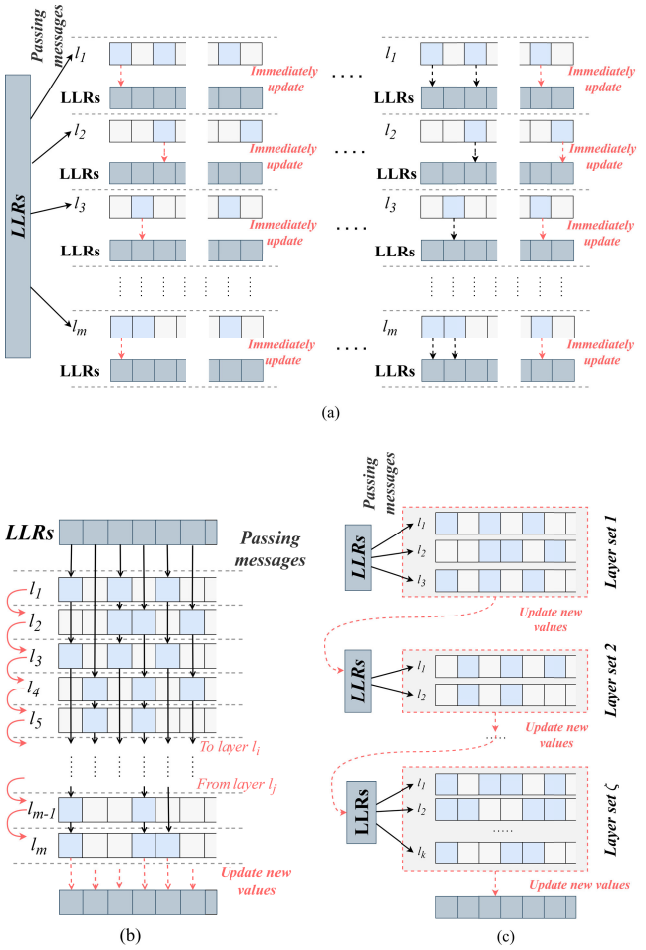


FIGURE 2. LDPC message passing flow (a) LP-LDPC, (b) layered LDPC, and (c) proposed MLP-LDPC.

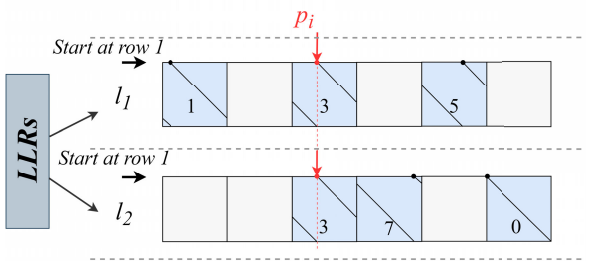


FIGURE 3. An example of the data conflicts when updating LLRs for two layers.

continues until all sets in $\mathcal{S}$ have been processed, as presented in Algorithm 2.

1) MULTI-LAYERED SET DETERMINATION

The main ideas of the multi-layered set $\mathcal{S}$ are as follows:

To begin, $\mathcal{S}$ is initialized as an empty set, which will eventually store the multi-layered set. $\mathcal{M} = \{1, 2, \ldots, m_b\}$ represents the set of all row indices of the shift matrix $\mathbf{P}$. Initially, $\mathcal{L}$ is set to $\mathcal{M}$, indicating the rows that have not yet been assigned to any layer set. The column index n starts at 1 ($n = 1$). The algorithm proceeds with an outer while loop, which continues while n is less than or equal to the total

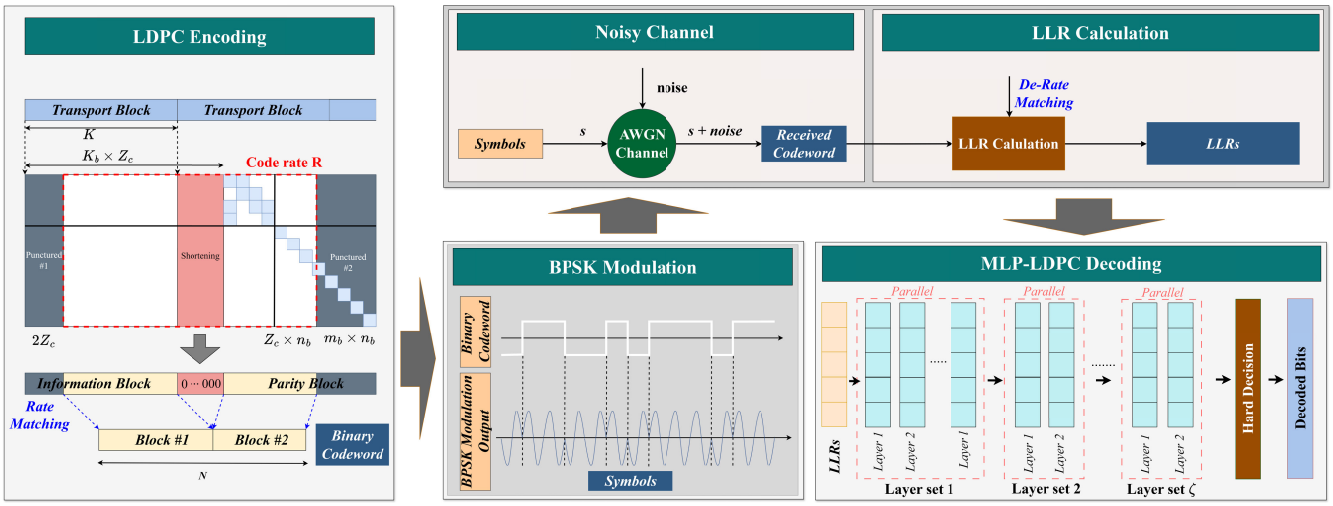


FIGURE 4. LDPC process with proposed MLP-LDPC decoding algorithm.

Algorithm 1 Multi-Layered Set Determination

Input: Shift matrix **P**
Output: Multi-layered set $\mathcal{S}$

Initialization:
$\mathcal{S} \leftarrow \emptyset$
$\mathcal{M} \leftarrow \{1, 2, \ldots, m_b\}$ ▷ Set of all row indices
$\mathcal{L} \leftarrow \mathcal{M}$
$n \leftarrow 1$
while $n \leq n_b$ and $\mathcal{L} \neq \emptyset$ **do**
  $\mathcal{L}_{\text{temp}} \leftarrow \emptyset$
  for each $m \in \mathcal{L}$ **do**
    if $P_{m,n} \neq -1$ **then**
      $\mathcal{L}_{\text{temp}} \leftarrow \mathcal{L}_{\text{temp}} \cup \{P_{m,n}\}$
    end if
  end for
  if $\mathcal{L}_{\text{temp}} \neq \emptyset$ **then**
    $\mathcal{L}_{\text{temp}} \leftarrow \text{Distinct}(\mathcal{L}_{\text{temp}})$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{L}_{\text{temp}}\}$
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{m \mid P_{m,n} \in \mathcal{L}_{\text{temp}}\}$
  end if
  $n \leftarrow n + 1$
end while

number of columns n_b, and $\mathcal{L}$ is not empty, indicating there are still rows to process. Within this loop:

- $\mathcal{L}_{\text{temp}}$ is initialized as an empty set to store distinct non-negative values encountered in the current column.
- The inner for loop iterates through each row m in $\mathcal{L}$. If $P_{m,n}$ is not -1, the value $P_{m,n}$ is added to $\mathcal{L}_{\text{temp}}$.
- After populating $\mathcal{L}_{\text{temp}}$, if it contains any elements:
  - $\mathcal{L}_{\text{temp}}$ is converted into a set of distinct values.
  - This distinct set $\mathcal{L}_{\text{temp}}$ is added to $\mathcal{S}$.
  - Rows in $\mathcal{L}$ where $P_{m,n}$ matches any value in $\mathcal{L}_{\text{temp}}$ are removed from $\mathcal{L}$.
- Finally, the column index n is incremented by 1 to proceed to the next column.

The algorithm terminates when all columns have been processed ($n > n_b$) or all rows have been assigned to a layer set ($\mathcal{L} = \emptyset$). This systematic approach ensures that rows in matrix **P** are categorized into distinct layer sets based on their non-negative values in each column, consolidating these sets into $\mathcal{S}$.

2) MLP-LDPC DECODING PROCESS

Fig. 4 shows the LDPC process with the proposed MLP-LDPC decoding algorithm.

- Within each layer set in $\mathcal{S}$, the algorithm further iterates through each layer l in the first layer set $\mathcal{S}_1$. For each layer l at the j-th position:

Algorithm 2 Proposed MLP-LDPC Decoding Algorithm

Input: $y = (y_1, y_2, \ldots, y_N) \in Y^N$. ▷ Received word
Output: $\hat{c} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N) \in \{0, 1\}^N$ ▷ Estimated word

Initialization:
  for all $i = 1$ to N **do** $p_i = \log \frac{P(c_i=0|y_i)}{P(c_i=1|y_i)} = \frac{2}{\sigma^2} y_i$
  for all $j = 1$ to M and $i \in \mathcal{V}(j)$ **do** $r_{j,i} = 0$
Iteration Loop:
  for all $\mathcal{S}_\zeta \in \mathcal{S}$ **do** ▷ Loop over horizontal layer set
    for $l \in \mathcal{S}_\zeta$ **do in parallel** ▷ Parallel process
      for all $j \in \mathcal{M}_l$, $i \in \mathcal{V}(j)$ **do**
        $q^k_{j,i} = p^k_i - r^{k-1}_{j,i}$
      for all $j \in \mathcal{M}_l$, $i \in \mathcal{V}(j)$ and $i' \in \mathcal{V}(j) \setminus i$ **do**
        $r^k_{j,i} = \min |q^k_{j,i'}| \cdot \prod \text{sgn}(q^k_{j,i'})$
        $p^k_i = q^k_{j,i} + r^k_{j,i}$ ▷ Immediately update
  end (horizontal layer set loop)
  for all $i = 1$ to N **do**
    $\hat{c}_i = \frac{1-\text{sgn}(p^k_i)}{2}$ ▷ Hard decision
  if $\mathbf{H} \times \hat{\mathbf{c}}^T = \mathbf{0}^T$ **then** ▷ Syndrome check
    exit iteration loop
  end if
End Iteration Loop

- VTC message $q_{j,i}|i \in \mathcal{V}(j)$ is calculated within the same row.
- Horizontally, new CTV messages $r_{j,i}|i \in \mathcal{V}(j)$ are computed based on the received VTC messages.
- APP value $p_i|i \in \mathcal{V}(j)$ are immediately updated.
- The updated variable p_i is then propagated to the same location in another layer.

- After processing all layers in $\mathcal{S}_1$, the algorithm proceeds to utilize the APP results obtained from $\mathcal{S}_1$ to handle the layer set $\mathcal{S}_2$.
- This sequential processing continues until all layer sets in $\mathcal{S}$ are completed, ensuring comprehensive decoding of the MLP-LDPC system.

This iterative approach ensures that the decoding process progressively refines variables and checks node messages across different layers, utilizing both horizontal and vertical relationships within the LDPC structure to iteratively improve decoding accuracy.

B. PROPOSED KYMLP-LDPC SYSTEM

This section introduces the proposed system for guaranteeing the integrity and confidentiality of data, such as images, during transmission over communication channels, as illustrated in Fig. 5. Key features of the proposed system are summarized as follows:

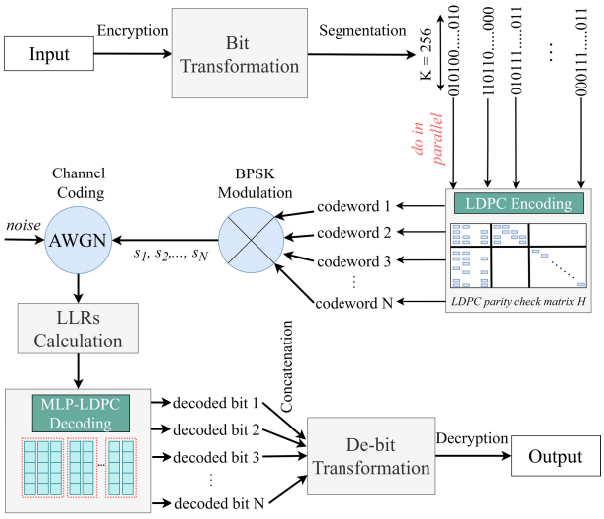


FIGURE 5. Data transmission process with proposed KyMLP-LDPC system.

1) *Encryption:* The input data is encrypted using the Kyber encryption algorithm with the public key pk to obtain the ciphertext string $\mathcal{C}$.
2) *Bit Transformation and Splitting:* Implement a bit transformation to convert the ciphertext string $\mathcal{C}$ to a binary string $\mathcal{B}$. Then, split the binary string into $\mathcal{N}$-blocks with each block size K bits to prepare for the LDPC encoding process. This process can be represented mathematically as follows:

$$\mathcal{B} = \text{bit\_transform}(\mathcal{C})$$
$$\text{blocks} = \text{split\_into\_blocks}(\mathcal{B}, \mathcal{N}, K)$$

Algorithm 3 Proposed KyMLP-LDPC process

Input: $\mathcal{T}$ ▷ Original message, pk ▷ Public key, $\mathcal{N}$ ▷ Number of blocks, K ▷ Block size, σ^2 ▷ Noise variance, sk ▷ Secret key

Output: $\mathcal{T}_{\text{original}}$ ▷ Decrypted original message

Step 1: Encryption
  $\mathcal{C} \leftarrow \text{Encrypt}(\mathcal{T})$
Step 2: Bit Transformation and Splitting
  $\mathcal{B} \leftarrow \text{bit\_transform}(\mathcal{C})$
  $\text{blocks} \leftarrow \text{split\_into\_blocks}(\mathcal{B}, \mathcal{N}, K)$
Step 3: LDPC Encoding
for $\eta = 1$ to $\mathcal{N}$ **do in parallel**
    $\mathbf{c}_\eta \leftarrow \text{LDPC\_encode}(\text{blocks}[\eta])$
end for
Step 4: BPSK Modulation
for $\eta = 1$ to $\mathcal{N}$ **do in parallel**
    $\mathbf{s}_\eta \leftarrow \text{BPSK\_modulation}(\mathbf{c}_\eta)$
end for
Step 5: LLRs Calculation
for $\eta = 1$ to $\mathcal{N}$ **do in parallel**
    $\mathbf{y}_\eta \leftarrow \mathbf{s}_\eta + \mathbf{w}_\eta$ ▷ Received signal
    $\mathbf{LLR}_\eta \leftarrow \text{LLRs\_calculation}(\mathbf{y}_\eta, \sigma^2)$
end for
Step 6: MLP-LDPC Decoding
  $\hat{\mathbf{c}} \leftarrow \text{MLP\_LDPC\_decode}(\{\mathbf{LLR}_1, \mathbf{LLR}_2, \ldots, \mathbf{LLR}_\mathcal{N}\})$
  $\hat{\mathbf{C}} \leftarrow \text{concat}(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \ldots, \hat{\mathbf{c}}_\mathcal{N})$
Step 7: De-bit Transformation
  $\hat{\mathbf{B}} \leftarrow \text{Debit\_transform}(\hat{\mathbf{C}})$
Step 8: Decryption
  $\mathcal{T}_{\text{original}} \leftarrow \text{Decrypt}(\hat{\mathbf{B}}, \text{sk})$
Return $\mathcal{T}_{\text{original}}$

where $\text{bit\_transform}(\mathcal{C})$ represents the transformation of the ciphertext string $\mathcal{C}$ into a binary string $\mathcal{B}$, and $\text{split\_into\_blocks}(\mathcal{B}, \mathcal{N}, K)$ denotes the splitting of the binary string $\mathcal{B}$ into $\mathcal{N}$ blocks of size K bits each.

3) *LDPC Encoding:* Conduct parallel encoding of the $\mathcal{N}$-block bits by using the LDPC encoding algorithm to obtain $\mathcal{N}$ codewords $\mathbf{c}_1 = \{c_{1_1}, \ldots, c_{1_N}\}, \mathbf{c}_2 = \{c_{2_1}, \ldots, c_{2_N}\}, \ldots, \mathbf{c}_\mathcal{N} = \{c_{\mathcal{N}_1}, \ldots, c_{\mathcal{N}_N}\}$. This parallel encoding process can be expressed as:

$$\mathbf{c}_\eta = \text{LDPC\_encode}(\text{blocks}[\eta])$$

where $\text{LDPC\_encode}(\cdot)$ denotes the LDPC encoding algorithm applied to each $\mathcal{N}$-block represented by $\text{blocks}[\eta]$.

4) *BPSK Modulation:* We consider binary phase-shift keying (BPSK) transmission over a binary-input AWGN channel. Each codeword is mapped into a BPSK sequence $\mathbf{s}_\eta = (s_{\eta_1}, s_{\eta_2}, \ldots, s_{\eta_N})$. Specifically, bit '0' is represented by $+1$ and bit '1' is represented by -1. Mathematically, this is given by:

$$s_{\eta_\theta} = 1 - 2c_{\eta_\theta}$$

where c_{η_θ} is the θ-th bit in the codeword $\mathbf{c}_\eta$ sequence.

5) *LLRs Calculation:* Each BPSK signal $\mathbf{s}_\eta = (s_{\eta_1}, s_{\eta_2}, \ldots, s_{\eta_N})$ is transmitted over an AWGN channel, resulting in the received signals $\mathbf{y}_\eta$.

The LLR for each received signal y_{η_θ} is calculated as:

$$\text{LLR}(y_{\eta_\theta}) = \log \frac{P(y_{\eta_\theta} \mid c_{\eta_\theta} = 0)}{P(y_{\eta_\theta} \mid c_{\eta_\theta} = 1)} = \frac{2}{\sigma^2} y_{\eta_\theta}$$

where $y_{\eta_\theta} = s_{\eta_\theta} + w_{\eta_\theta}$ is the received signal, w_{η_θ} is the Gaussian noise with zero mean and variance σ^2. These LLR values are then used as input for the MLP-LDPC decoding process.

6) *MLP-LDPC Decoding:* After the MLP-LDPC decoding process is completed, let $\hat{\mathbf{c}} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_{\mathcal{N}})$ represent the $\mathcal{N}$ binary decoded bits. These decoded bits are concatenated into a binary string:

$$\hat{\mathcal{C}} = \text{concat}(\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_{\mathcal{N}})$$

7) *De-bit Transformation:* The binary string $\hat{\mathcal{C}}$ is then transformed from bits to its corresponding message. Let $\hat{\mathcal{B}}$ represent the message obtained from $\hat{\mathcal{C}}$.

$$\hat{\mathcal{B}} \leftarrow \text{Debit\_transform}(\hat{\mathcal{C}})$$

8) *Decryption:* Finally, the transfer message $\hat{\mathcal{B}}$ is decrypted using the secret key sk, resulting in the original input message $\mathcal{T}_{\text{original}}$:

$$\mathcal{T}_{\text{original}} = \text{Decrypt}(\mathcal{T}, \text{sk})$$

Details of the proposed system process are presented in Algorithm 3.

IV. EXPERIMENT RESULTS

We first evaluate the performance of the proposed MLP-LDPC algorithm and compare it with the existing LP-LDPC and layered LDPC algorithms. We focus on the ability of the proposed algorithm to reduce latency and minimize collisions during message transmission compared to other algorithms. Then, we examine the performance of the proposed KyMLP-LDPC coding system using the parameter set with the optimal coding rate. We evaluate the performance of the system with image and signal data, which are two common types of data used in communications.

A. MLP-LDPC DECODING EVALUATIONS

First, we compare the ability of the proposed MLP-LDPC decoding algorithm to reduce conflicts during message passing with the LP-LDPC algorithm across four different code rates (1/3, 2/5, 1/2, and 2/3). For this evaluation, we fix the block length K at 256, which necessitates the use of base graph BG2 as specified in the 5G NR LDPC code standard. With this base graph, the lifting size Z_c is calculated to be 32, and the number of information bit columns k_b is 8.

Table 4 presents the detailed results of the conflict comparison. As results from the table, the MLP-LDPC algorithm exhibits a significant reduction in message passing conflicts compared to LP-LDPC across all code rates. For code rate $R = 1/3$, the MLP-LDPC algorithm shows a significant conflict reduction. Conflicts occur only in two layer sets, {1, 2, 3, 5, 7, 9, 11, 19, 20, 22} and {4, 8, 10, 13, 15}, with a total number of conflicts of 4. In contrast, the LP-LDPC algorithm has 11 conflicts. For code rate $R = 2/5$, the MLP-LDPC algorithm again outperforms LP-LDPC, recording 1 conflict in set {1, 2, 3, 5, 7, 9, 11} and 2 conflicts in set {4, 8, 10, 13, 15}, while LP-LDPC has 11 conflicts. For code rate $R = 1/2$, the MLP-LDPC algorithm shows minimal conflicts, with only 1 collision in set {1, 2, 3, 5, 7, 9, 11}, compared to 5 conflicts for LP-LDPC. With code rate $R = 2/3$, the proposed algorithm has only 1 conflict in set {1, 2, 3, 5, 7} and no collisions in other sets, while LP-LDPC records 3 conflicts.

TABLE 4. Analyzing conflict position between MLP-LDPC and LP-LDPC decoding algorithms.

Code Rate	Decoding Algorithm	Layer Sets	No. of Conflicts
$R = \frac{1}{3}$	MLP-LDPC	{1, 2, 3, 5, 7, 9, 11, 19, 20, 22}	2
		{6, 12, 14}	0
		{16}	0
		{4, 8, 10, 13, 15}	2
		{17}	0
	LP-LDPC	{1, 2, 3, …, 22}	13
$R = \frac{2}{5}$	MLP-LDPC	{1, 2, 3, 5, 7, 9, 11}	1
		{6,12,14}	0
		{16}	0
		{4, 8, 10, 13, 15}	2
		{17}	0
	LP-LDPC	{1, 2, 3, …, 17}	11
$R = \frac{1}{2}$	MLP-LDPC	{1, 2, 3, 5, 7, 9, 11}	1
		{6, 12}	0
		{8, 10, 4}	0
	LP-LDPC	{1, 2, 3, …, 12}	5
$R = \frac{2}{3}$	MLP-LDPC	{1, 2, 3, 5, 7}	1
		{6}	0
		{4}	0
	LP-LDPC	{1, 2, 3, 4, 5, 6, 7}	3

The consistent reduction in message passing conflicts achieved by the MLP-LDPC algorithm across all code rates underlines its potential for improved decoding reliability.

Next, we evaluate the bit error rate (BER) and average execution time (AET) per block of three LDPC decoding algorithms: layered LDPC, LP-LDPC, and the proposed MLP-LDPC. The evaluation is conducted as a function of the signal-to-noise ratio (SNR). To allow performance comparisons across different code rates R, we use the normalized SNR per information bit, given by $\text{SNR} = 2 \times R \times E_b/N_0$ where E_b/N_0 is the energy bit to noise power spectral density ratio of AWGN. The BER value is calculated as follows:

$$\text{BER} = \frac{N_{\text{err}}}{N_{\text{tb}} \times K} \tag{5}$$

where N_{err} and N_{tb} are the number of bits with errors and the total number of transport blocks, respectively.

This evaluation continuously employs information bits of length $K = 256$ and investigate performance across four code rates (1/3, 2/5, 1/2, 2/3) with varying transport block

TABLE 5. AET per block for various SNR values (0.5 db to 3.0 db).

Code Rate	Number of Blocks	Proposed (s)	LP-LDPC (s)	Layered LDPC (s)
$R = \frac{1}{3}$	128	0.37	0.47	0.78
	256	0.37	0.49	0.67
	512	0.36	0.49	0.67
	1024	0.38	0.48	0.69
$R = \frac{2}{5}$	128	0.35	0.45	0.62
	256	0.35	0.47	0.61
	512	0.35	0.44	0.55
	1024	0.37	0.47	0.64
$R = \frac{1}{2}$	128	0.31	0.43	0.53
	256	0.33	0.42	0.55
	512	0.32	0.42	0.51
	1024	0.35	0.44	0.59
$R = \frac{2}{3}$	128	0.21	0.41	0.51
	256	0.23	0.4	0.52
	512	0.17	0.41	0.49
	1024	0.3	0.41	0.55

lengths (128, 256, 512, 1024). All algorithms are subjected to a maximum of 15 iterations, and experiments span 15 SNR values ranging from 0.5 dB to 3.0 dB.

Our new MLP-LDPC decoder design shows a significant improvement in LDPC decoding efficiency. Table 5 clearly demonstrates that the MLP-LDPC decoder consistently achieves a lower AET per block compared to both LP-LDPC and layered LDPC decoding methods. This means that the MLP-LDPC decoder can process data faster and use resources more effectively, making it a more efficient solution overall.

Additionally, Fig. 6 supports our findings by illustrating the superior convergence behavior of the MLP-LDPC decoder across a range of code rates. Traditional decoders often experience performance drops when dealing with different coding schemes, but the MLP-LDPC decoder maintains its efficiency regardless of these variations. This consistent performance makes the MLP-LDPC decoder a versatile and reliable option for various communication scenarios.

Through the evaluation results, it is evident that the proposed MLP-LDPC algorithm significantly minimizes conflicts during the decoding process. Furthermore, it achieves a lower AET, and better BER compared to the other two algorithms. This demonstrates the algorithm's efficiency and effectiveness. The reduced conflicts and lower AET indicate that the MLP-LDPC algorithm is not only theoretically sound but also practically viable. Consequently, it holds great potential for application in real-world scenarios where efficient and reliable decoding is crucial.

B. KYMLP-LDPC PERFORMANCE EVALUATION

To evaluate the performance of the KyMLP-LDPC system, we conducted tests using image and signal data, which are common types of data in wireless channel transmission.

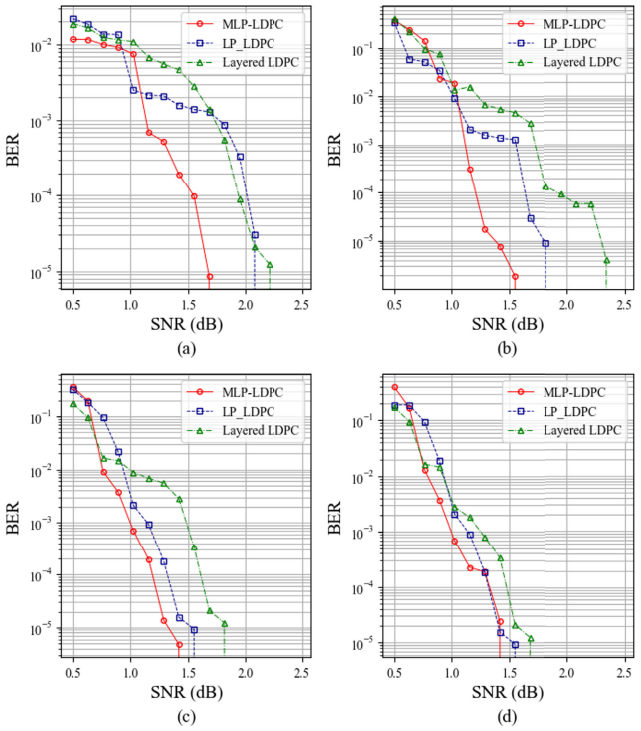


FIGURE 6. Performance comparison of three algorithms with 1024 blocks and code rate (a) $R = 1/3$, (b) $R = 2/5$, (c) $R = 1/2$, (d) $R = 2/3$.

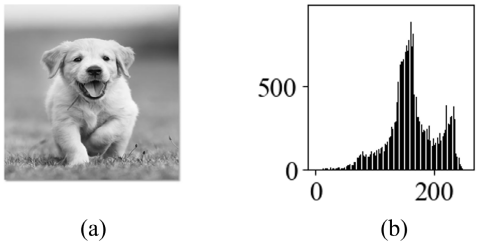


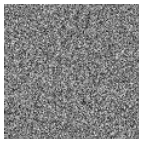
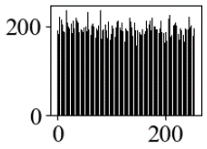
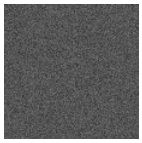
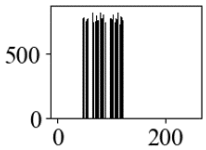
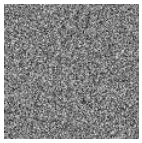
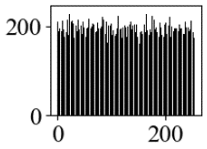
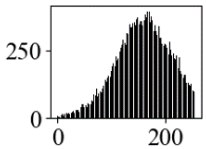
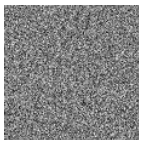
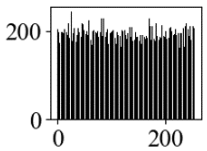
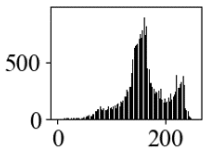
FIGURE 7. (a) Input image and (b) its histogram.

1) IMAGE DATA

Fig. 7 shows the input data and its histogram. The data is simulated and transmitted through the AWGN channel with SNR values of 0.5 dB, 1.5 dB, and 2 dB respectively. As mentioned before, we divided the input data after the bit-transformation step into blocks of length $K = 256$ as input for the encoding and decoding process using MLP-LDPC. Based on the comparison results in the previous section, a code rate of $R = 2/3$ gives the optimal result for input block $K = 256$. Therefore, in this test, we will use $R = 2/3$. The detailed results of the system performance are presented in Table 6. The key metrics evaluated include the entropy of the encrypted and decrypted images, their histograms, and the correlation score of the decrypted images with the original image.

The entropy analysis reveals interesting trends. For the encrypted images, entropy remains relatively constant across different SNR levels (0.5 dB: 7.9964, 1.5 dB: 7.9937, 2.0 dB: 7.9968). This consistency suggests that the encryption process effectively maintains the complexity and randomness

TABLE 6. Performance of the system across various SNR levels with image data.

SNR	Encrypted			Decrypted			Correlation Score vs. Original Image
	Image	Histogram	Entropy	Image	Histogram	Entropy	
0.5 dB			7.9964			7.8974	0.0012
1.5 dB			7.9937			7.5142	0.7797
2.0 dB			7.9968			7.2315	0.9973

of the image data, regardless of the noise level. On the other hand, the entropy of the decrypted images shows a notable decrease as the SNR increases (0.5 dB: 7.8974, 1.5 dB: 7.5142, 2.0 dB: 7.2315). This trend indicates that higher noise levels impact the decrypted image's complexity, making it less random and more structured.

Histograms of both encrypted and decrypted images provide a detailed visual representation of the pixel value distribution, important for analyzing the effectiveness of the encryption and decryption processes. The histograms of encrypted images maintain a uniform distribution, which is a clear indicator that the encryption algorithm has effectively obscured the original image data, rendering it unrecognizable and ensuring data security. This uniform distribution suggests that each pixel value is equally probable, a desirable trait in secure encryption as it minimizes any discernible patterns that could be exploited by unauthorized parties. Conversely, the histograms of decrypted images reveal the varying impact of noise on the image reconstruction process. As the SNR increases, these histograms start to closely resemble the histogram of the original image, indicating a more accurate reconstruction. Specifically, at lower SNR levels, the histograms of decrypted images appear more distorted and less similar to the original, reflecting the higher level of noise interference. However, with higher SNR levels, the histograms of decrypted images align more closely with the original image, demonstrating reduced noise influence and improved reconstruction quality.

The correlation score between the decrypted images and the original image improves significantly with increasing SNR (0.5 dB: 0.0012, 1.5 dB: 0.7797, 2.0 dB: 0.9973), which is a crucial metric as it directly indicates the quality of the decrypted image. At a low SNR of 0.5 dB, the correlation is almost negligible, highlighting poor reconstruction quality.

TABLE 7. Performance comparison between the original signal and decrypted signal across different SNR levels.

SNR Value	MSE	PSNR	Correlation
0.5	95245	14.943	0.8195
1.5	39734	17.945	0.9089
2.5	507.84	56.076	0.9997

However, as the SNR improves to 1.5 dB and further to 2.0 dB, the correlation scores approach near-perfect values, indicating that the decrypted image is nearly identical to the original image at higher SNRs.

2) SIGNAL DATA

The experimental results presented in Fig. 8 and Table 7 demonstrate the performance of the system under varying SNR conditions. The figure illustrates how the system's performance changes across SNR levels of 0.5 dB, 1.5 dB, and 2.5 dB using signal data. From Table 7, it is evident that as the SNR increases, there is a noticeable improvement in several key metrics. Specifically, the mean squared error (MSE) decreases significantly from 95245 at 0.5 dB SNR to 507.84 at 2.5 dB SNR. This reduction indicates that the fidelity of the decrypted signal improves with higher SNR levels, leading to a more accurate reconstruction of the original signal. Similarly, the peak signal-to-noise ratio (PSNR) shows a consistent improvement as SNR increases, rising from 14.943 dB to 56.076 dB across the same SNR range. This metric quantifies the quality of the decrypted signal relative to the original, with higher values indicating better quality and less distortion. Moreover, the correlation coefficient, which measures the similarity between the original and decrypted signals, approaches unity as SNR increases, indicating a stronger linear relationship

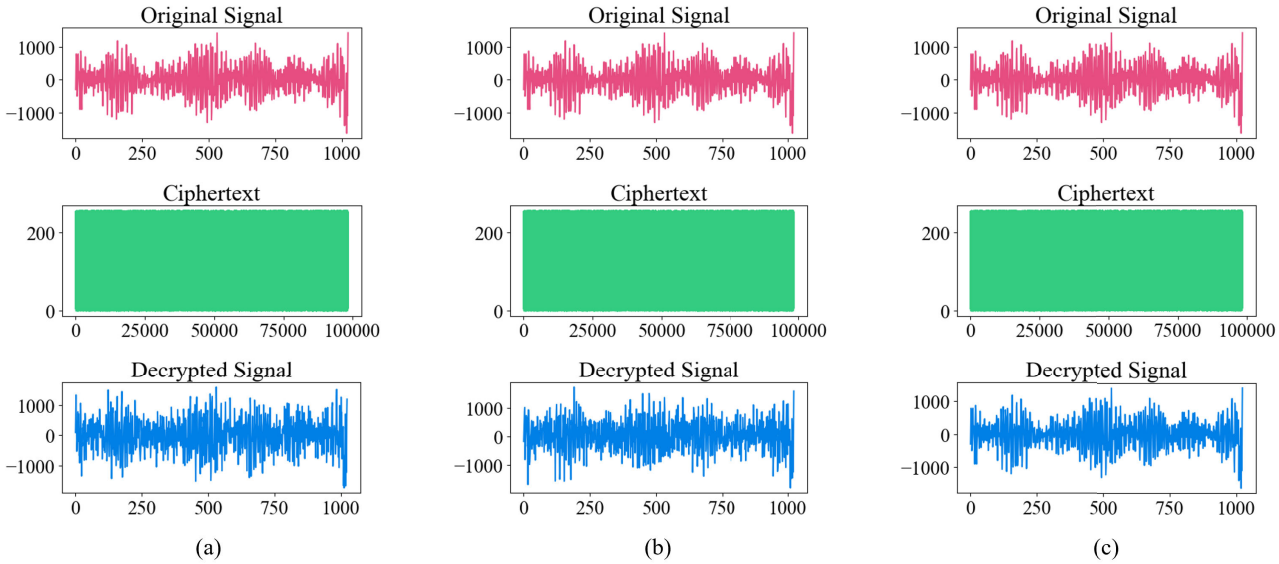


FIGURE 8. The system's performance with signal data varies across different SNR levels (a) 0.5 dB, (b) 1.5 dB, and (c) 2.5 dB.

between the two signals. At 2.5 dB SNR, the correlation coefficient reaches 0.9997, demonstrating almost perfect agreement between the original and decrypted signals. These findings highlight the system's robustness and effectiveness in handling signal decryption under varying SNR conditions. The improvements observed in MSE, PSNR, and correlation underscore the system's capability to achieve high-fidelity signal recovery when operating in environments with higher SNR.

V. CONCLUSION

In this paper, we introduce an MLP-LDPC decoding method for error correction in 5G NR communications to optimize decoding efficiency, address latency issues, and minimize message-passing conflicts. Additionally, we integrate the proposed MLP-LDPC approach with the Kyber post-quantum cryptography algorithm to enhance data security and integrity in 5G communications. We assessed the effectiveness of our method by comparing the average execution time per block across three decoding techniques: LDPC, LP-LDPC, and MLP-LDPC. The results demonstrate that our algorithm achieves faster decoding times and lower BER values across four code rates. Furthermore, the integration of Kyber algorithm with MLP-LDPC ensures robust security against potential cyberattacks, making it well-suited for secure 5G communications.

REFERENCES

[1] B. Sklar. (2020). *Digital Communications: Fundamentals and Applications*. [Online]. Available: https://api.semanticscholar.org/CorpusID:62253895

[2] Y. Yang, W. Xue, J. Sun, G. Yang, Y. Li, H. Hwa Pang, and R. H. Deng, "PkT-SIN: A secure communication protocol for space information networks with periodic k-time anonymous authentication," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 6097–6112, 2024.

[3] U. Bucci, D. Cassioli, and A. Marotta, "Performance of spatially diverse URLLC and eMBB traffic in cell free massive MIMO environments," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 1, pp. 161–173, Feb. 2024.

[4] K. Ali and M. Jammal, "Proactive VNF scaling and placement in 5G O-RAN using ML," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 1, pp. 174–186, Feb. 2024.

[5] O. Adamo, S. Fu, and M. Varanasi, "Hardware-efficient encryption encoder and decoder unit," in *Proc. MILCOM IEEE Mil. Commun. Conf.*, San Diego, CA, USA, Nov. 2008, pp. 1–6.

[6] M. Mehic, L. Michalek, E. Dervisevic, P. Burdiak, M. Plakalovic, J. Rozhon, N. Mahovac, F. Richter, E. Kaljic, F. Lauterbach, P. Njemcevic, A. Maric, M. Hamza, P. Fazio, and M. Voznak, "Quantum cryptography in 5G networks: A comprehensive overview," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 302–346, 1st Quart., 2024.

[7] R. Ma, J. Cao, S. He, Y. Zhang, B. Niu, and H. Li, "A UAV-assisted UE access authentication scheme for 5G/6G network," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 2, pp. 2426–2444, Apr. 2024.

[8] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[9] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[10] C. Condo, G. Masera, and P. Montuschi, "Unequal error protection of memories in LDPC decoders," *IEEE Trans. Comput.*, vol. 64, no. 10, pp. 2981–2993, Oct. 2015.

[11] Q. Li, L. Shi, Y. Cui, and C. J. Xue, "Exploiting asymmetric errors for LDPC decoding optimization on 3D NAND flash memory," *IEEE Trans. Comput.*, vol. 69, no. 4, pp. 475–488, Apr. 2020.

[12] S. P. Tera, R. Chinthaginjala, P. Natha, G. Pau, C. Dhanamjayulu, and F. Mohammad, "CNN-based approach for enhancing 5G LDPC code decoding performance," *IEEE Access*, vol. 12, pp. 89873–89886, 2024.

[13] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[14] B. Jang, H. Jang, S. Kim, K. Choi, and I.-C. Park, "Area-efficient QC-LDPC decoding architecture with thermometer code-based sorting and relative quasi-cyclic shifting," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 71, no. 6, pp. 2897–2910, Jun. 2024.

[15] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, Mar. 2018.

[16] *Technical Specification Group Radio Access Network; NR; Multiplexing and Channel Coding (Release 16)*, Standard 3GPP TS 38.212, 2017. [Online]. Available: https://portal.3gpp.org/#/55936-specifications

[17] S. Li, Y. Zhang, Y. Song, N. Cheng, K. Yang, and H. Li, "Blockchain-based portable authenticated data transmission for mobile edge computing: A universally composable secure solution," *IEEE Trans. Comput.*, vol. 73, no. 4, pp. 1114–1125, Apr. 2024.

[18] Y. Ren, H. Harb, Y. Shen, A. Balatsoukas-Stimming, and A. Burg, "A generalized adjusted min-sum decoder for 5G LDPC codes: Algorithm and implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 6, pp. 2911–2924, Jun. 2024.
[19] K. D. Rao and T. A. Babu, "Performance analysis of QC-LDPC and polar codes for eMBB in 5G systems," in *Proc. Int. Conf. Electr., Electron. Comput. Eng. (UPCON)*, Aligarh, India, Nov. 2019, pp. 1–6.
[20] J.-C. Liu, H.-C. Wang, C.-A. Shen, and J.-W. Lee, "Low-complexity LDPC decoder for 5G URLLC," in *Proc. IEEE Asia–Pacific Conf. Postgraduate Res. Microelectron. Electron. (PrimeAsia)*, Chengdu, China, Oct. 2018, pp. 43–46.
[21] M. Rahim, T. L. Nguyen, T. Nhu Do, and G. Kaddoum, "Joint power and user allocation in coexistence of eMBB and URLLC services," *IEEE Commun. Lett.*, vol. 28, no. 9, pp. 2186–2190, Sep. 2024.
[22] M. A. Jadoon, A. Pastore, M. Navarro, and A. Valcarce, "Learning random access schemes for massive machine-type communication with Marl," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, pp. 95–109, 2024.
[23] Z. Yan, W. Guan, and L. Liang, "List-based residual belief-propagation decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 28, no. 5, pp. 984–988, May 2024.
[24] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
[25] Z. Li, Y. Shen, Y. Ren, Y. Huang, X. You, and C. Zhang, "Belief propagation decoding for short-length codes based on sparse tanner graph," *IEEE Commun. Lett.*, vol. 28, no. 5, pp. 969–973, May 2024.
[26] T. Kim and J. S. Park, "Neural self-corrected min-sum decoder for NR LDPC codes," *IEEE Commun. Lett.*, vol. 28, no. 7, pp. 1504–1508, Jul. 2024.
[27] X. Wu, Y. Song, M. Jiang, and C. Zhao, "Adaptive-normalized/offset min-sum algorithm," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 667–669, Jul. 2010.
[28] B. Wang, Y. Zhu, and J. Kang, "Two effective scheduling schemes for layered belief propagation of 5G LDPC codes," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1683–1686, Aug. 2020.
[29] X. Wang, Z. He, H. He, and J. Wang, "Exploiting the joint sparsity for LDPC-enhanced delay-Doppler multicarrier modulation: A parallel belief propagation-based approach," *IEEE Commun. Lett.*, vol. 28, no. 4, pp. 907–911, Apr. 2024.
[30] K. Zhang, X. Huang, and Z. Wang, "High-throughput layered decoder implementation for quasi-cyclic LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 985–994, Aug. 2009.
[31] X. Ji, J. Dong, T. Deng, P. Zhang, J. Hua, and F. Xiao, "HI-Kyber: A novel high-performance implementation scheme of Kyber based on GPU," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 6, pp. 877–891, Jun. 2024.
[32] W. Xia, B. Liu, J. Ren, Y. Mao, X. Wu, R. Ullah, L. Zhao, S. Chen, Y. Wan, Y. Ma, Y. Li, Z. Qi, Y. Wu, and X. Guo, "High-security transmission scheme of secure key generation and distribution based on polling-permutation encryption," *J. Lightw. Technol.*, vol. 42, no. 1, pp. 149–157, Jan. 15, 2024.
[33] A. Cohen, R. G. L. D'Oliveira, K. R. Duffy, J. Woo, and M. Médard, "AES as error correction: Cryptosystems for reliable communication," *IEEE Commun. Lett.*, vol. 27, no. 8, pp. 1964–1968, Aug. 2023.
[34] M. Bhavitha, K. Rakshitha, and S. M. Rajagopal, "Performance evaluation of AES, DES, RSA, and Paillier homomorphic for image security," in *Proc. IEEE 9th Int. Conf. Converg. Technol. (I2CT)*, Pune, India, Apr. 2024, pp. 1–5.
[35] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The impact of quantum computing on present cryptography," 2018, *arXiv:1804.00200*.
[36] *Module-Lattice-Based Key-Encapsulation Mechanism Standard*, Standard FIPS 203, Federal Information Processing Standards Publication, NIST, Aug. 2024, doi: 10.6028/NIST.FIPS.203.
[37] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber algorithm specifications and supporting documentation," *NIST PQC Round, Version*, vol. 2, no. 4, pp. 1–43, Aug. 2021.
[38] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, Jan. 1999.
[39] T. T. Nguyen, T. T. B. Nguyen, and H. Lee, "Low-complexity multi-size circular-shift network for 5G new radio LDPC decoders," *Sensors*, vol. 22, no. 5, p. 1792, Feb. 2022.
[40] J. Kang, J. An, and Y. Zhu, "Universal high-throughput and low-complexity LDPC decoder for laser communications," *IEEE Access*, vol. 12, pp. 33328–33336, 2024.
[41] Y. Zhang, K. Peng, Z. Chen, and J. Song, "Construction of rate-compatible raptor-like quasi-cyclic LDPC code with edge classification for IDMA based random access," *IEEE Access*, vol. 7, pp. 30818–30830, 2019.
[42] T. Lin, S. Cao, S. Zhang, and S. Xu, "A unified reconfigurable datapath for 5G compatible LDPC decoding," in *Proc. IEEE Asia–Pacific Conf. Circuits Syst. (APCCAS)*, Chengdu, China, Oct. 2018, pp. 215–218.
[43] *Document 3GPP Chairman'Notes 3GPP TSG RAN WG1 Meeting 89, 3GPP*. Accessed: Aug. 22, 2024. [Online]. Available: https://www.3gpp.org
[44] T. T. B. Nguyen, T. Nguyen Tan, and H. Lee, "Efficient QC-LDPC encoder for 5G new radio," *Electronics*, vol. 8, no. 6, p. 668, Jun. 2019.

LINH NGUYEN (Member, IEEE) received the bachelor's and M.S. degrees in mathematics and informatics engineering from Hanoi University of Science and Technology, Vietnam, in 2020 and 2021, respectively. She is currently pursuing the Ph.D. degree in informatics and computing with the School of Informatics, Computing, and Cyber Systems, Northern Arizona University, USA. Her research interests include post-quantum cryptography, error correction codes, and artificial intelligence.

QUOC BAO PHAN (Member, IEEE) received the bachelor's degree in electrical engineering from Hanoi University of Science and Technology, Vietnam, in 2022. He is currently pursuing the Ph.D. degree in informatics and computing with the School of Informatics, Computing, and Cyber Systems, Northern Arizona University, USA. His research interests include post-quantum cryptography, homomorphic encryption, and artificial intelligence.

TUY TAN NGUYEN (Member, IEEE) received the M.S. and Ph.D. degrees in information and communication engineering from Inha University, South Korea, in 2016 and 2019, respectively. Then, he was a Senior Research Engineer with Conextt Inc., South Korea, from August 2019 to April 2021. From May 2021 to July 2022, he was a Lecturer with the School of Global Convergence Studies and a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Inha University. He is currently an Assistant Professor with the School of Informatics, Computing, and Cyber Systems, Northern Arizona University, USA. His research interests include post-quantum cryptography, homomorphic encryption, error correction codes, and applied artificial intelligence. He serves as a Technical Committee Member of the IEEE Circuits and Systems Society—Circuits and Systems for Communications.