

Article

# Adaptive Rate-Compatible Non-Binary LDPC Coding Scheme for the B5G Mobile System

Dan-feng Zhao, Hai Tian \*  and Rui Xue

College of Information & Communication Engineering, Harbin Engineering University, Harbin 150001, China; zhaodanfeng@hrbeu.edu.cn (D.-f.Z.); xuerui@hrbeu.edu.cn (R.X.)

\* Correspondence: tianhai@hrbeu.edu.cn; Tel.: +86-177-0364-9232

Received: 4 January 2019; Accepted: 26 February 2019; Published: 2 March 2019



**Abstract:** This paper studies an adaptive coding scheme for B5G (beyond 5th generation) mobile system-enhanced transmission technology. Different from the existing works, the authors develop a class of rate-compatible, non-binary, low-density parity check (RC-NB-LDPC) codes, which expresses the strong connection between the algebra-based and graph-theoretic-based constructions. The constructed codes can not only express rate-compatible (RC) features, but also possess a quasi-cyclic (QC) structure that facilitates the encoding implementation. Further, in order to achieve the code rate-adaptive allocation scheme, the authors propose using the K-means++ clustering algorithm to cluster different channel environments, considering various factors that affect channel characteristics. Finally, in order to present the advantages of the adaptive coding scheme, the authors construct a coding scheme for image transmission. The numerical results demonstrate that the developed code can obtain better waterfall performance in a larger code rate range, which is more suitable for data transmission; the adaptive coding transmission scheme can obtain higher reconstructed image quality compared to the fixed code rate-coding scheme. Moreover, when considering unequal error protection (UEP), the proposed scheme can further improve the reconstructed image quality.

**Keywords:** B5G; adaptive coding scheme; channel clustering; k-means++ algorithm; rate-compatible LDPC

## 1. Introduction

With the advent of the 5G communication era, the demand for wireless communication is still strong, and the increased spectrum and power resources are insufficient. Therefore, new transmission technology needs to be updated to achieve rapid growth of the wireless communication capacity [1,2]. The B5G mobile system performs well in terms of wireless coverage and user experience. To overcome the challenge, one should improve the error-correcting, anti-jamming, and efficient transmission capabilities for B5G. Among the communication technologies for error correction and detection, channel coding can significantly improve the reliability of communication systems and ensure efficient transmission—this has become one of the focuses among the communication technologies [3–9].

In order to achieve significant improvements with key performance parameters, the channel-coding technique for B5G should have features such as large coding gain, low-complexity, low-latency, high-throughput, and flexible code parameters [10,11]. The conventional single channel-coding schemes can hardly meet all of the requirements of B5G. The B5G system will select different channel-coding schemes for different scenarios and serves to meet the corresponding technical requirements.

For the B5G mobile system, the channel-coding scheme for a high data rate is pursued; however, the wireless channel environment is time-varying and fading. High-data-rate signals are prone

to inter-symbol interference (ISI) in frequency-selective fading channels, which affects the overall transmission performance of the system [12]. In order to improve the throughput of the system, the adaptive coding method is proposed [13]. This method dynamically adjusts the coding mode of the transmitted signal according to the channel state information (CSI) fed back from the receiving end, and then finds a balance between the transmission rate and the transmission reliability to improve the throughput of the system. The DVB-S2 standard adopts an adaptive coding modulation (ACM) scheme in which channel coding comprises LDPC codes [14,15]. However, the parameters such as the code length and the rate of LDPC codes in the DVB-S2 standard are fixed, and LDPC codes of different parameters need to design corresponding check matrices, resulting in the consumption of hardware resources.

LDPC codes are currently among the most promising coding techniques to achieve the channel capacities for a wide range of channels. LDPC codes were first proposed by Gallager in 1962 [16], and developed in the late 1990s [17,18]. Since their rediscovery, a great deal of research has been conducted in design, construction, encoding, and decoding algorithms. Many LDPC codes have been adopted as standard codes for current and next-generation communication systems, such as wireless, optical, satellite, deep space, multi-media broadcast (MMB), digital video broadcast (DVB) systems [8–10], and so on. This rapid dominance of LDPC codes in applications is due to their capacity-approaching performance, and they can be achieved with practically implementable iterative decoding algorithms. However, further research is needed to better understand the structural properties and performance characteristics of these codes. QC-LDPC codes, a subclass of LDPC codes [11], can be efficiently encoded and decoded by shift register adder accumulator (SRAA) [19,20] and semi-parallel decoding architectures [21,22], respectively. Thus, QC-LDPC codes can greatly reduce the consumption of hardware resources, and solve the complicated problem of implementing the codec from a certain angle.

Rate-compatible LDPC (RC-LDPC) codes are a special set of LDPC codes with different code rates. These LDPC codes have good performance at each rate and can be implemented using the same set of codecs. The communication system using the RC-LDPC code can adjust the code rate according to the current channel condition, so that the communication system can improve the transmission efficiency while satisfying the transmission reliability. In 2002, J Li and KR Narayanan combined the puncturing and extension methods to generate a series of RC-LDPC codes, which expanded the range of dynamic code rates [23]. In 2006, J. Ha selected the location of the puncturing variable node from the perspective of the recovery tree, and proposed a grouping/sorting algorithm to sort the variable nodes according to the size of the recovery tree. The node with the smallest number of layers is used as the puncturing node [24]. In 2007, H.Y Park proposed another perforation measure [25]. The non-greedy criterion puncturing algorithm proposed in [26] shows better performance than the recovery tree-based puncturing algorithm. In [27], the shortening priority of variable nodes is divided according the column weight of each column in the check matrix, and a shortening algorithm suitable for whether irregular or regular quasi-cyclic LDPC codes was proposed.

The traditional ACM technology selects the corresponding modulation and coding schemes according to the threshold calculated by the CSI fed back from the receiver, which depends largely on the assumed channel model [28,29]. In fact, the real channel is not Gaussian-distributed; the amplifier incurs nonlinear effects, etc. Using the traditional ACM algorithm, it is often difficult to meet the frame error rate (FER) requirement [12,30]. With the rise of machine learning algorithms, more simple and effective methods are used to reduce the complexity of the traditional methods [6,31,32]. The clustering algorithm is an important unsupervised learning algorithm in machine learning. Due to its flexibility, it is widely used in various technical fields [33,34].

This paper studies an adaptive coding scheme for the B5G mobile system. We first develop a class of RC-NB-LDPC codes based on masking and puncturing techniques to construct the RC-LDPC codes. The masking technique converts the check matrix of the regular non-binary LDPC (NB-LDPC) code into one of irregular NB-LDPC code, and the puncturing technology constructs RC-NB-LDPC code. Then, we design a rate allocation scheme. Different from the existing works, we propose using the

k-means++ clustering algorithm to cluster different channel environments, considering various factors that affect channel characteristics. This method relies on a perfect mathematical model and can learn interactively in the wireless channel. According to the actual error rate performance of the system, the correspondence between the signal-to-noise ratio and the modulation and coding mode is determined, which helps to select a more accurate coding rate. Finally, in order to present the advantages of the adaptive coding scheme in compressed image processing, we construct a coding scheme for image transmission. The numerical results demonstrate that the developed code can obtain better waterfall performance in a larger code rate range, which is more suitable for data transmission; the adaptive coding transmission scheme can obtain higher reconstructed image quality compared to the fixed code rate-coding scheme. Moreover, when considering UEP, the proposed scheme can further improve the reconstructed image quality.

The remainder of this paper is organized as follows. The construction method and notations of RC-NB-LDPC codes are described in Section 2. In Section 3, we propose using the channel-clustering algorithm to cluster different channel parameters. In Section 4, we discuss the BER performance and decoding thresholds of RC-NB-LDPC codes. We also construct an adaptive coding scheme for image transmission to evaluate the performance of reconstructed image quality. We present our conclusions in the final section.

## 2. The Construction of RC-NB-LDPC Codes

In this section, we first briefly review a general matrix-theoretic method for constructing QC-LDPC codes based on superposition. Then, we develop a class of RC-NB-LDPC codes based on masking and puncturing techniques.

### 2.1. QC-LDPC Codes and Superposition Construction

The check matrix  $H$  corresponding to a class of quasi-cyclic LDPC codes can be expressed by

$$H = \begin{bmatrix} I(P_{11}) & I(P_{12}) & \cdot & \cdot & \cdot & I(P_{1n}) \\ I(P_{21}) & I(P_{22}) & \cdot & \cdot & \cdot & I(P_{2n}) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ I(P_{m1}) & I(P_{m2}) & \cdot & \cdot & \cdot & I(P_{mn}) \end{bmatrix} \quad (1)$$

where  $I(P_{ij})$  ( $1 \leq i \leq m, 1 \leq j \leq n$ ) is a  $b \times b$ -dimensional circulant matrix, and is referred to as the circulant permutation matrix [35]. This matrix can be obtained by rotating each row vector of a  $b \times b$ -dimensional unit matrix  $P_{ij}$  element to the right, and  $P_{ij} \in \{0, 1, 2, \dots, b-1, \infty\}$ . Herein,  $P_{ij} = 0$  means that  $I(P_{ij})$  is a unit matrix, and  $P_{ij} = \infty$  means that  $I(P_{ij})$  is a zero matrix. For any  $i, j$ , replacing the circulant permutation matrix  $I(P_{ij})$  in the check matrix  $H$  by value  $P_{ij}$ , one can obtain a new matrix  $P$ , referred to as the circulation coefficient-matrix [36], which is shown by

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdot & \cdot & \cdot & P_{1n} \\ P_{12} & P_{22} & \cdot & \cdot & \cdot & P_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ P_{m1} & P_{m2} & \cdot & \cdot & \cdot & P_{mn} \end{bmatrix} \quad (2)$$

$H_{ij} = 0$  when  $P_{ij} = \infty$  and  $H_{ij} = 1$ ; otherwise, for any  $i, j$ , one can obtain the corresponding base matrix  $H_b$ , which is given by

$$H_b = \begin{bmatrix} H_{11} & H_{12} & \dots & H_{1n} \\ H_{12} & H_{22} & \dots & H_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ H_{m1} & H_{m2} & \dots & H_{mn} \end{bmatrix} \tag{3}$$

where  $H_{ij} \in \{0,1\}, (1 \leq i \leq m, 1 \leq j \leq n)$ .

Thus, if we obtain a small  $m \times n$  base matrix  $H_b$  and a set of  $b \times b$  circulant permutation matrices, we can construct an  $m \times n$  array  $H$  of  $b \times b$  matrices by replacing each 1-entry in  $H_b$  by a circulant permutation matrix and each 0-entry by a zero matrix. This replacement of entries in  $H_b$  by the circulant permutation matrices and/or zero matrices expands the matrix  $H_b$  into a  $mb \times nb$  sparse matrix  $H$ .

### 2.2. The Construction of RC-NB-LDPC Codes

Most methods for constructing LDPC codes can be classified into two general categories: the algebraic-based and the graph-theoretic-based constructions [36]. This paper combines these two methods to construct the RC-LDPC codes, as shown in Figure 1.

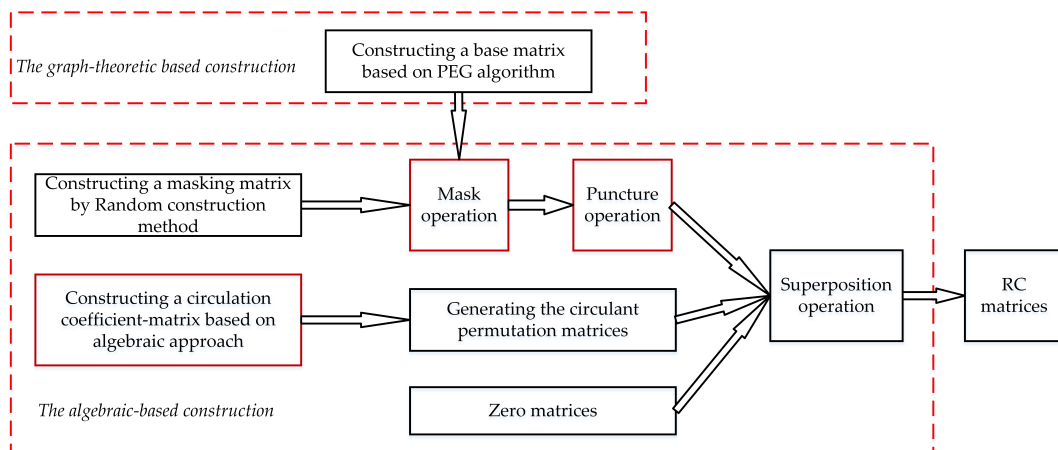


Figure 1. RC-NB-LDPC codes construction process.

The best-known graph-theoretic-based construction method is the progressive edge-growth (PEG) algorithm [37,38]. The goal of this algorithm is to construct a graph that adds one edge each time and maximizes the local girth. Therefore, the PEG algorithm is a greedy algorithm for constructing a Tanner graph with a large girth. In this paper, we use the PEG algorithm to construct a small base matrix for future research.

The algebraic-based methods for constructing LDPC codes have been developed using mathematical tools such as finite geometries [39], finite fields [40], and combinatorial designs [41,42]. In general, Algebraic LDPC codes have much lower error-floors than randomly constructed LDPC codes. Most of the algebraic constructions involve several key ingredients including the circulant permutation matrix, masking matrix, and superposition operation [43]. In this paper, the key operations for constructing RC-NB-LDPC codes are described as follows.

- Mask operation

The masking technique can selectively mask some sub-circulant matrixes in order to reduce the density of non-zero elements in the check matrix  $H$ , so that fewer small cycles or longer girth could



be obtained. Moreover, the performance of the LDPC code is improved. Through selecting a suitable masking matrix, we can obtain any rate code according to the same base matrix.

The masking technique can be described as special matrix product operations [44]. We assume a masking matrix  $H_m = [m_{ij}]$  as a binary matrix with size of  $m \times n$ , then the definition of the product of  $H_m$  and  $H$  is shown as (4).

$$H = H_m \otimes H_b \equiv \begin{bmatrix} m_{1,1}I(P_{11}) & m_{1,2}I(P_{12}) & \dots & m_{1,n}I(P_{1n}) \\ m_{2,1}I(P_{21}) & m_{2,2}I(P_{22}) & \dots & m_{2,n}I(P_{2n}) \\ \vdots & \vdots & \dots & \vdots \\ m_{m,1}I(P_{m1}) & m_{m,2}I(P_{m2}) & \dots & m_{m,n}I(P_{mn}) \end{bmatrix} \quad (4)$$

In Equation (4), when  $m_{i,j} = 1$ ,  $m_{i,j}A_{s_{ij}} = A_{s_{ij}}$  and when  $m_{i,j} = 0$ ,  $m_{i,j}A_{s_{ij}} = 0$ . We define  $H_b$  as the base matrix and  $H$  as the masked matrix. If  $H_m$  and  $H_b$  are sparse matrices, then  $H$  is also a sparse matrix. In addition, sub-matrices in  $H$  can be replaced with zero arrays of the same size by using the masking technique, which can reduce the density of non-zero elements in  $GF(q)$ . Thus, a sparser matrix  $H$  degrades the probability of small cycles.

When constructing LDPC codes, if we obtain the degree distributions  $\lambda(x)$  and  $\rho(x)$  of the variable node and the check node, then we can obtain the check matrix  $H$ . We assume that the corresponding degree distribution of code words with code rate  $r$  and code length  $n$  are  $\lambda(x)$  and  $\rho(x)$ ; if the masking matrix  $H_m$  obeys the same degree distribution, the masked check matrix  $H$  will also obey the same distribution.

- Puncture operation

The puncturing technique is the most common rate adaptive technique [24,45,46], the implementation process of which first selects a high rate mother code with excellent performance, then punctures the check bit in the code word, thus deleting a number of columns in the check matrix, resulting in a low rate code word. The high rate QC-LDPC code may be used at higher SNR. On the contrary, when the channel environment is poor, a low rate QC-LDPC code may be used, thereby improving the average effectiveness of the communication system. Using the puncturing technique to change the degree distribution of the masking matrix, an adaptive-rate LDPC code can be obtained. Since the dimension of the masking matrix is much smaller than the check matrix of the QC-LDPC code, it can greatly reduce the hardware complexity of the rate-adaptive communication system.

The NB-LDPC code defined on the finite field  $GF(q)$  has  $q = 2^m$  elements; the mapping relationship is as follows:

$$\mathbf{F}_q \rightarrow \mathbf{F}_2^m \quad (5)$$

We refer to  $(x_0, \dots, x_{m-1}) \in \mathbf{F}_2^m$  as a binary mapping of the symbol  $X \in \mathbf{F}_q$ . The NB-LDPC code can be mapped into binary code for transmission in a channel. Therefore, the NB-LDPC code can be punctured at both the symbol level and bit level.

For example, for 8-ary LDPC codes, the degree distribution function is  $\lambda(x) = 1/3x + 2/3x^3$ ,  $\rho(x) = x^5$ . The bit-level puncturing algorithm is intuitively represented by Figure 2.

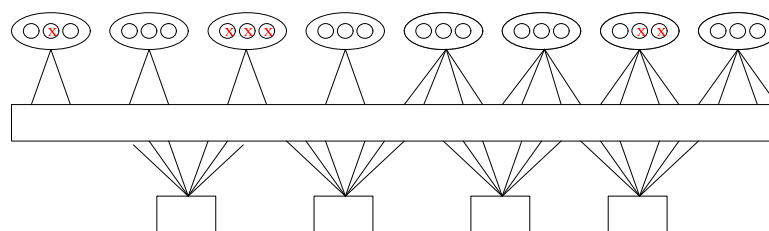


Figure 2. The bit-level puncturing pattern of the NB-LDPC code.

- The circulation coefficient-matrix is designed using an algebraic-based method

The circulation coefficient-matrix of size  $m \times n$  is designed using an algebraic-based method, assuming that  $GF(q)$  is a finite field containing  $q$  elements,  $\alpha$  is a primitive of  $GF(q)$ , and all elements in  $GF(q)$  can be represented as a form of  $\alpha$  power, i.e.,  $\alpha^{-\infty}, \alpha^0, \alpha^1, \dots, \alpha^{q-2}$ , where  $0 = \alpha^{-\infty}, \alpha^0 = 1$ .  $S_1 = \{\alpha^{i_0}, \alpha^{i_1}, \dots, \alpha^{i_{m-1}}\}$  and  $S_2 = \{\alpha^{j_0}, \alpha^{j_1}, \dots, \alpha^{j_{n-1}}\}$  are two subsets of  $GF(q)$ , where  $i_k, j_l \in \{0, 1, \dots, q-2\}, 0 \leq k < m, 0 \leq l < n$ . Based on the closeness of the computation of the finite field, a finite field matrix on  $GF(q)$  can be constructed from  $S_1$  and  $S_2$ . The specific construction method is given by

$$P = \begin{bmatrix} \alpha^{i_0} - \alpha^{j_0} & \alpha^{i_0} - \alpha^{j_1} & \dots & \alpha^{i_0} - \alpha^{j_{n-1}} \\ \alpha^{i_1} - \alpha^{j_0} & \alpha^{i_1} - \alpha^{j_1} & \dots & \alpha^{i_1} - \alpha^{j_{n-1}} \\ \vdots & \vdots & \dots & \vdots \\ \alpha^{i_{m-1}} - \alpha^{j_0} & \alpha^{i_{m-1}} - \alpha^{j_1} & \dots & \alpha^{i_{m-1}} - \alpha^{j_{n-1}} \end{bmatrix} = \begin{bmatrix} \alpha^{c_{0,0}} & \alpha^{c_{0,1}} & \dots & \alpha^{c_{0,n-1}} \\ \alpha^{c_{1,0}} & \alpha^{c_{1,1}} & \dots & \alpha^{c_{1,n-1}} \\ \vdots & \vdots & \dots & \vdots \\ \alpha^{c_{m-1,0}} & \alpha^{c_{m-1,1}} & \dots & \alpha^{c_{m-1,n-1}} \end{bmatrix} \quad (6)$$

where  $\alpha^{c_{k,l}} = \alpha^{i_k} - \alpha^{j_l}, 0 \leq k < m, 0 \leq l < n$ . Matrix  $P$  is a circulation coefficient-matrix.

In summary, the steps of constructing the RC-NB-LDPC codes are as follows:

1. Construct a matrix  $H_b$  of size  $m \times n$  based on the PEG algorithm as the base matrix.
2. Construct a matrix  $H_m$  of size  $m \times n$  based on the random construction as the mask matrix.
3. Mask and puncture the base matrix  $H_b$ .
4. Calculate a circulation coefficient-matrix  $P$  of size  $m \times n$  using an algebraic-based method and generate the circulant permutation matrices.
5. Superposition operation. Replacing the entries in  $H_b$  by the circulant permutation matrices and/or zero matrices expands the matrix  $H_b$  into a  $mb \times nb$  sparse matrix  $H$ .

### 2.3. Design and Implementation of the RC-NB-LDPC Code

The check matrix of the RC-NB-LDPC code has a QC structure, so the coding algorithm can adopt the QC-structured fast-coding method. The FPGA implementation based on the QC-structured fast-coding method will be described in detail below.

#### 2.3.1. Matrix of RC-NB-LDPC Codes

The fast-coding method uses the special structure of the QC-LDPC code for encoding, so the check matrix and the generator matrix need to be set. The structure of the check matrix is shown in Figure 3. The base matrix size is  $4 \times 16$ , and the check matrix is divided into 64 blocks, and the size of each block matrix is  $48 \times 48$ . The numbers in the above matrix represent the element coefficients of the current matrix, and the matrix with a coefficient of 0 is the all-zero matrix. The numbers in the below matrix represent the circulant coefficients of the current matrix, and a coefficient of 0 indicates that the current matrix is not shifted.

Since the check matrix of the NB-QC-LDPC code has a QC structure, the generator matrix also has a QC structure. The generation matrix is as shown in Figure 4. The right side of the matrix is a unit matrix of size  $576 \times 576$ , and each  $G_{i,j}$  on the left side is a circulant of size  $48 \times 48$ . Therefore, the entire matrix can be stored by simply storing the first row of each  $G_{i,j}$  matrix, greatly reducing resource usage. The source symbol is multiplied by the matrix of its corresponding code rate to obtain the coding result. Since the right side of the matrix is a unit matrix, it is only necessary to calculate the result of multiplying the source symbol by the  $G_{i,j}$ , and then stitching the source symbol to obtain the final coding result. The source symbol and  $G_{i,j}$  are encoded to obtain a check sequence, wherein  $G_{i,j}$  is divided into four columns, which can be encoded simultaneously by four channels, so that the coding efficiency can be improved four-fold.

5	2	5	2	6	3	0	7	4	1	7	0	1	0	0	6
3	3	0	7	2	0	6	0	2	6	4	2	1	4	0	0
7	0	3	0	3	3	6	0	6	7	0	4	0	2	2	0
4	5	7	0	4	3	7	4	2	0	2	0	0	0	5	5

$\overbrace{\hspace{16em}}^{1/2}$ $\overbrace{\hspace{16em}}^{2/3}$ $\overbrace{\hspace{16em}}^{3/4}$															
5	35	30	16	25	7	Inf	29	19	7	14	Inf	38	Inf	Inf	2
21	20	Inf	20	18	Inf	10	Inf	46	22	45	33	28	9	Inf	Inf
14	Inf	0	Inf	30	9	20	Inf	9	43	Inf	21	Inf	15	44	Inf
16	35	25	Inf	0	32	28	5	24	Inf	43	Inf	Inf	Inf	38	29

Figure 3. Check matrix H structure of the NB-LDPC codes.

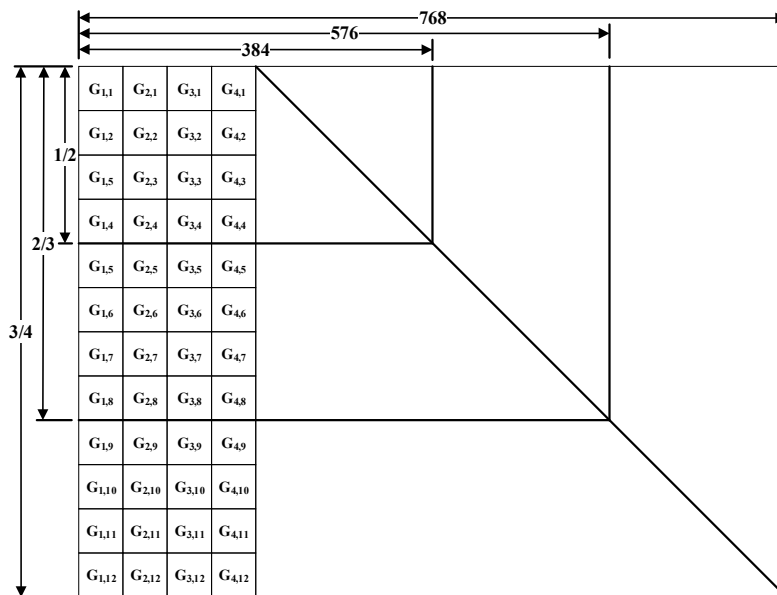


Figure 4. Generation matrix G structure of NB-LDPC codes.

### 2.3.2. Encoder Architecture

The encoder-coding block diagram is shown in Figure 5. It is divided into three parts. The input buffer part is used to buffer the information transmitted by the upper level and output the data when the encoding part is needed. The second part is the encoding part, which is used to calculate the encoding result and output the calculation result to the output buffer module. The output buffer part is used to store the encoding result and the communication with the downstream module.

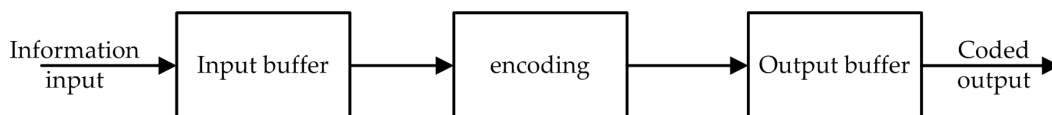


Figure 5. The coding block diagram.

The hardware structure of the NB-QC-LDPC code encoder is shown in Figure 6, which includes an input module, an encoding module, and an output module.

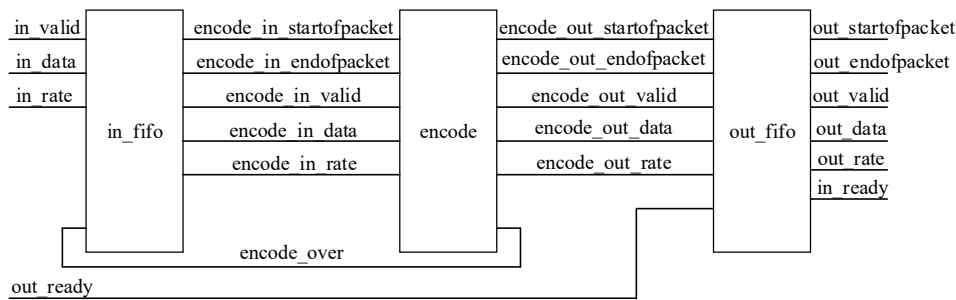


Figure 6. The hardware structure of the NB-QC-LDPC code encoder.

In addition to buffering the source symbol function passed by the previous level, the input module also completes the cross-clock domain operation of the data. Furthermore, according to the currently input code rate information, the source symbol is truncated into a sequence of corresponding lengths, so that the coding module performs coding. Therefore, the input module mainly includes a fifo module and a corresponding control module. When the “encoding\_over” signal of the encoding module is high, it indicates that the encoding module has been encoded, and can receive a new source frame.

The coding module is a core module of the NB-LDPC encoder, and its function is to perform a matrix multiplication operation on the source information and the generation matrix in the Galois field to obtain a coding result. The coding module mainly includes an operation module, a RAM storage module and a corresponding control module.

The output module mainly performs the function of storing the encoded result and communicating with the downstream module. The encoding module inputs the encoding result to the output module. When the “out\_ready” signal is high, it indicates that the next level has been prepared for reception, and the encoding result may be output; otherwise, the output module temporarily stores the encoding result. When the output module is full, the output “in\_ready” signal is turned low, indicating that new data cannot be accepted. Therefore, the output module mainly contains the fifo module and the corresponding control module.

### 2.3.3. Coding Module

The coding module is the core module of the encoder. The block diagram of the structure is shown in Figure 7. It includes the SSRAA (serial shift register accumulator unit) operation module, the source information cache module, the parallel-to-serial conversion module, the output module and the control module.

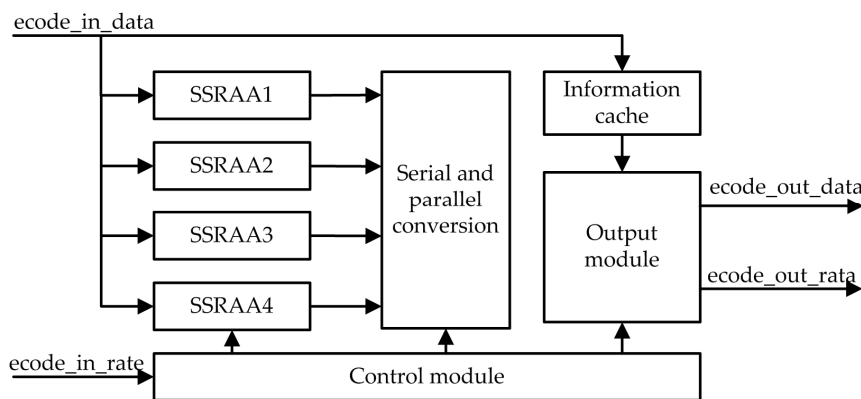
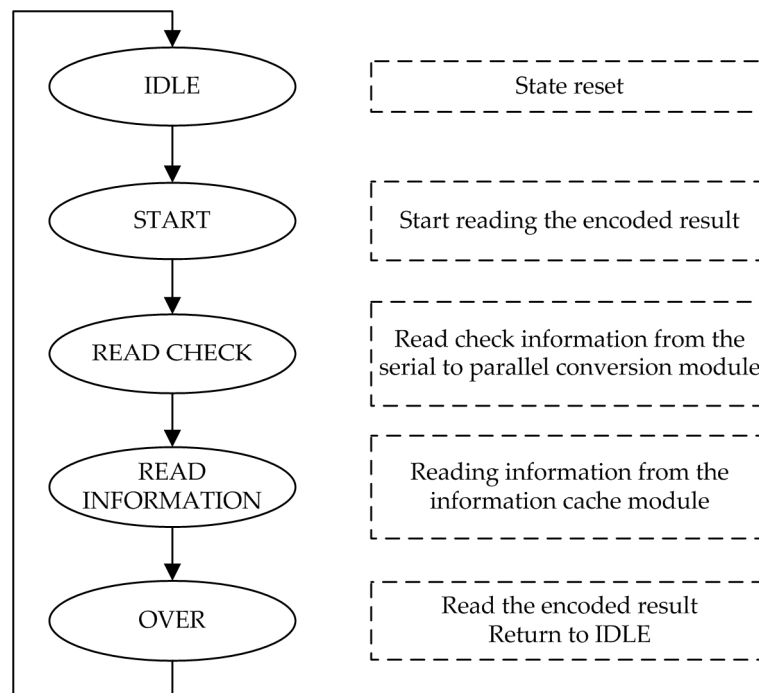


Figure 7. Coding module structure.

The SSRAA operation module completes the coding operation process, and the four sets of SSRAA operation modules work in parallel, which improves the coding efficiency. The parallel–serial

conversion module converts the verification sequence encoded by the four parallel SSRAA circuits into a serial form for sequential output. The source information cache module is used to store the source sequence. The output module mainly completes the output of the encoded result.

After the processing of the upper input module, the input data of the encoding module is a piece of truncated source information, and the data length corresponds to the current encoding rate. At the time of data entry, the SSRAA module is also constantly calculating the check sequence. When a piece of information is entered, the SSRAA module also completes the calculation of the check sequence. The output of the encoded result is then performed, and the state machine of the output module is shown in Figure 8.



**Figure 8.** The state machine of the output module.

After the encoding is completed, the state machine jumps from the original IDLE state to the SRTART state and begins reading the encoded result. In the READ\_CHECK state, the output module reads the parallel-converted 192-bit check sequence from the parallel-to-serial conversion module and then jumps to the READ\_INFORMATION state. In the READ\_INFORMATION state, the output module reads the source information of the corresponding length from the information cache module according to the code rate information of the current frame, and completes the output of the encoded result. The state machine then jumps back from the OVER state to the IDLE state, waiting for the encoding of the next frame to complete.

#### 2.3.4. SSRAA Module

The encoder of the QC-LDPC code can be completed by the SSRAA circuit, and its implementation structure is shown in Figure 9.

The multiplication and addition operations in the encoding of the NB-LDPC code are operations of the Galois field. For an 8-ary LDPC code, each symbol consists of three bits. The addition of the Galois field is essentially a bitwise XOR operation, as shown in the right part of Figure 10. The multiplication of the Galois field is designed by the looking-up table method, as shown in the left part of Figure 10. First, the input data  $d_{in1}$  and  $d_{in2}$  are respectively looked up in the ROMA, the result is added in the eight-element domain, and the addition result is checked in the ROMB. The result is the multiplication of  $d_{in1}$  and  $d_{in2}$  in  $GF(8)$ .

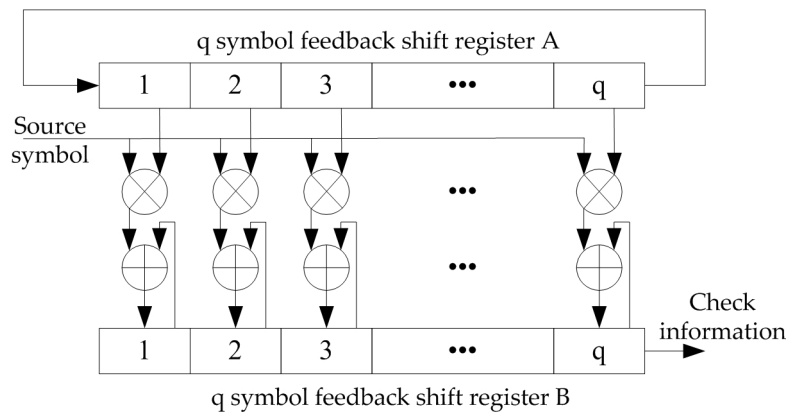


Figure 9. Serial shift register accumulator unit.

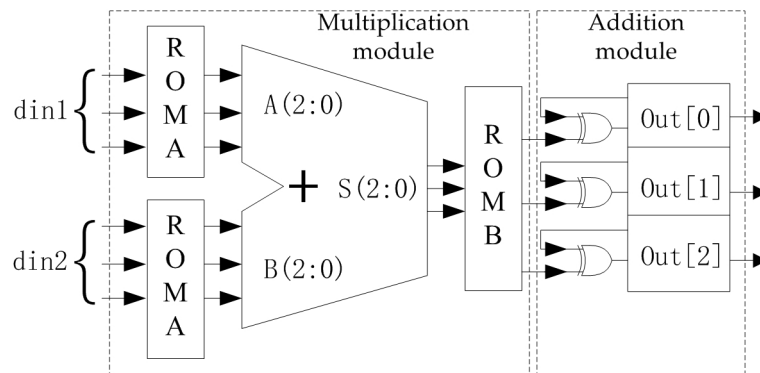


Figure 10. 8-ary LDPC code coding unit.

#### 2.4. Complexity Analysis

The coding complexity is shown in Table 1. Since the check matrix construction algorithm for constructing the QC-LDPC code based on the masking and puncturing technique has a quasi-cyclic structure, an encoding method of a quasi-cyclic code with low coding complexity can be employed. From the perspective of coding complexity, adopting the QC-encoding algorithm, only needs to multiply  $O((L - J)^2 p)$  times in  $GF(q)$ , and there is no need to add operations throughout the coding process— $J$  and  $L$  are the number of rows and columns of the mask matrix  $H_m$ , respectively, and  $p$  is the dimension of the sub matrix. Linear complexity coding is achieved when high-speed coding is employed. The coding complexity calculation of the random construction algorithm of the NB-LDPC code check matrix consists of two parts: one is to Gaussian eliminate the check matrix into the structure of the lower triangular matrix, that is, pre-process the check matrix; the operation complexity is  $O(n^3)$ , where  $n$  is the code length. The other part is that the coding complexity is  $O(n^2)$ . This is because the coding complexity of the coding algorithm depends on the sparsity of the generator matrix. Let the average column weight of the generator matrix be  $w$ ; approximately,  $wn$  multiplication operations and  $(w - 1)n$  sub-operations in  $GF(q)$  are required throughout the encoding process. Although the check matrix of a NB-LDPC code is very sparse, its generator matrix is not sparse. Usually, the ratio of  $w$  and  $n$  is a non-negligible constant, which often makes the coding complexity proportional to the square of its code length. From the hardware storage capacity of the check matrix, when storing the matrix with this construction algorithm, only one  $p \times p$  unit array  $I$ , one  $J \times L$  non-binary coefficient matrix  $G_{C_{J \times L}}$ , and one  $J \times L$  circulation coefficient matrix  $S_{J \times L}$  are stored. However, to construct a parity check matrix of the same size using a random construction method, it is necessary to store a  $p \times J \times p \times L$  check matrix. The hardware storage resource is largely saved compared with the storage of the complete NB-LDPC code check matrix.



**Table 1.** Coding complexity analysis.

Matrix Construction Algorithm	Pre-Process	Add Operation Times in GF( $q$ )	Multiplication Operation Times in GF( $q$ )	Coding Complexity
Random construction	$O(n^3)$	$(w - 1)n$	$wn$	$O(n^2)$
Masking and puncturing construction	No	No	$o((L - J)^2 p)$	$O(n)$

In addition, the masking and puncturing techniques are used to overcome the shortcomings of the search time when the NB-LDPC code of the long code length is randomly constructed. This is because the dimension of the masking matrix  $H_m$  is a small constant compared to a randomly constructed matrix. It is easy to construct a check matrix of the NB-LDPC code of the given degree distribution,  $\lambda(x)$  and  $\rho(x)$ , with the row weight and the column weight.

In the rate-adaptive communication system, the NB-LDPC code of the appropriate code rate is selected according to the channel environment for encoding. Most of the actual communication systems require real-time performance. Improving the effectiveness of the communication system and the speed of hardware processing are two effective ways. With adaptive-rate technology, the average effectiveness of the communication system can be improved. The puncturing of the masking matrix can greatly improve the speed of the hardware processing. Let the masking matrix  $H_m$  be a binary matrix of  $J \times L$  and the matrix  $H$  be a non-binary matrix of  $m \times n$ , where  $m = p \times J$ ,  $n = p \times L$ . For example, if  $J = 8$ ,  $L = 16$ , the mother code rate is  $r = 1/2$ , and the puncturing code rate is  $r_m = 3/4$ . To delete the check bits in  $H$ ,  $4 \times p$  check bits need to be deleted. However, for the check matrix constructed by the algorithm based on masking and puncturing techniques, only 4 check bits of the masking matrix  $H_m$  need to be deleted. The processing speed of the hardware is increased by  $p$  times. Obviously, the longer the codeword, that is, the larger the  $p$  value, the more the processing speed of the hardware is improved, thereby greatly improving the real-time performance of the communication system.

### 3. Channel Clustering Based on the K-Means++ Algorithm

Wireless signals are susceptible to the effect of a variety of physical factors during transmission, such as signal source position, obstacles, antenna gain, transmit power, and so on. In order to ensure the validity and reliability of information transmission, it is necessary to change the transmission code rate according to different channel conditions. We usually use channel estimation techniques to judge the quality of channel states. However, the channel estimation algorithm with computational complexity would reduce the effectiveness of the communication system.

With the rise of machine learning, more simple and effective methods are used to reduce the complexity [6,31,32]. The clustering algorithm is an important unsupervised learning algorithm in machine learning. Because of its flexibility, rapid classifications based on different similarities of data are widely used in various technical fields [47,48].

The partition-based clustering algorithm is one of the most commonly used algorithms. Usually, the method based on distance division is used to divide the objects with approximate characteristics into the same cluster. The distances of objects in the same cluster are as close as possible. The distance between objects in different clusters is as far as possible. Partition-based clustering algorithms often need to enumerate the possibilities of all categories in order to obtain an optimal solution.

#### 3.1. Channel Environment Clustering Based on the K-Means++ Algorithm

The k-means algorithm is one of the most common partition-based clustering algorithms. K-means clustering aims to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The k-means algorithm is simple and flexible, but sensitive to outliers and isolated points.

In addition to the impact of the physical factors, the number of users under the same hotspot, as well as the user usage behavior (occupied bandwidth), will have an impact on the actual available

channel state. In this paper, we abstract these influencing factors into data vectors and form the data vectors into a sample set, and then cluster them with the k-means algorithm.

We assume that the inputs are the sample set  $D = \{x_1, x_2, \dots, x_m\}$ , the cluster number  $k$ , the maximum number of iterations  $N$ , and the output is cluster partition  $C = \{C_1, C_2, \dots, C_k\}$ . The k-means algorithm flow is as follows:

1. We randomly select  $k$  samples from data set  $D$  as the initial cluster center vector  $\{\mu_1, \mu_2, \dots, \mu_k\}$ .
2. For  $n = 1, 2, \dots, N$ 
  - (a) Initialize cluster partition  $C$  to  $C_t = \emptyset, t = 1, 2, \dots, k$ .
  - (b) For  $i = 1, 2, \dots, m, j = 1, 2, \dots, k$ , calculate the distance between the sample  $x_i$  and cluster center vector  $\mu_j$ :  $d_{ij} = \|x_i - \mu_j\|^2$ . Mark  $x_i$  as the category  $\lambda_i$  corresponding to the smallest  $d_{ij}$  and update  $C_{\lambda_i} = C_{\lambda_i} \cup \{x_i\}$ .
  - (c) For  $j = 1, 2, \dots, k$ , recalculate the new cluster center for all sample points in  $C_j$  by  $\mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$ .
  - (d) If all cluster center vectors have not changed, go to step 3.
3. Output cluster division  $C = \{C_1, C_2, \dots, C_k\}$ .

The k-means algorithm is sensitive to the initially selected cluster center, and the position selection of the  $k$  initial cluster centers has a great influence on the final clustering result. Therefore, it is necessary to select the appropriate  $k$  cluster centers. The k-means++ algorithm randomly initializes the cluster center, which is the optimization of the k-means algorithm.

The optimization strategy of k-means++ for initializing the cluster center is as follows:

According to the process of the algorithm, different initial clustering centers may result in different results. In this paper, we select the k-means++ algorithm in order to optimize the initial clustering center. The k-means++ algorithm is an improvement of the k-means algorithm. Try to choose the k-measuring point distance as the initial cluster center. The specific steps are as follows:

1. Randomly select sample data from the data set as the first cluster center  $\mu_1$  in  $k$  clusters.
2. Calculate the distance  $D(x)$  between the data sample  $x_i$  in the dataset and the nearest cluster center.
 
$$D(x) = \operatorname{argmin} \|x_i - \mu_r\|^2, r = 1, 2, \dots, k_{\text{selected}}$$
3. According to the value of  $D(x)$ , select a new data point in the remaining sample data as a new cluster center. The larger the  $D(x)$ , the greater the probability of being selected;
4. Repeat steps 2 and 3 until  $k$  cluster centers are selected;
5. Implement the standard k-means algorithm by using the  $k$  centers as the initial cluster centers.

### 3.2. Complexity Optimization of the End-to-End Distortion Model Based on the Clustering Algorithm

Traditional rate allocation schemes often need to traverse all combinations of coding parameters to achieve adaptive-rate allocation by iterating all combinations. In this paper, the clustering algorithm is introduced to quickly cluster the channel environment to obtain the channel-coding scheme, so as to achieve the goal of adaptive-rate allocation. Compared with the traditional scheme, the proposed method can effectively reduce the computational complexity.

The joint source channel-coding design based on rate allocation is to obtain the optimal video reconstruction quality through bit allocation between the channel and the source in the case of limited bandwidth and channel interference. A block diagram of the bit allocation strategy for joint source channel coding is shown in Figure 11.

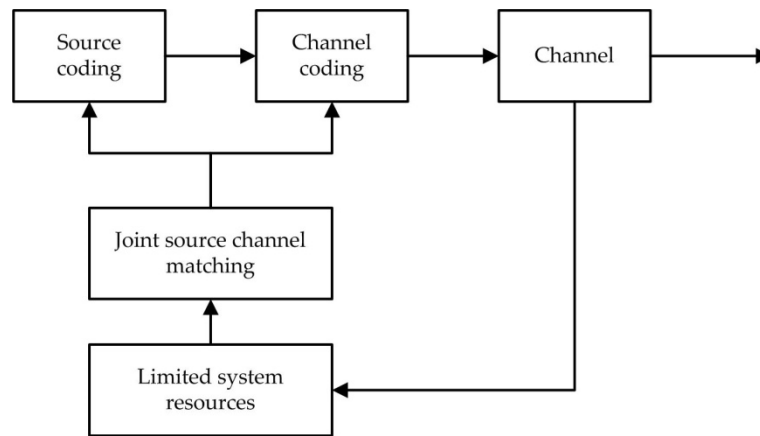


Figure 11. The bit allocation strategy for joint source channel coding.

In the joint source channel-coding system, end-to-end distortion is one of the important means to measure the quality of the system. The performance of the transmission system is measured by calculating the mean square error between the reconstructed video image and the original video image at the encoder end. The end-to-end distortion model is derived below.

First, define the pixel in the  $n^{\text{th}}$  frame as  $i$ , and the pixel of the reference frame  $ref$  is  $j$ . Let  $f_n^i$  be the original value of the pixel, let  $\hat{f}_n^i$  and  $\tilde{f}_n^i$  be the reconstructed values of the encoder and decoder, respectively, and let  $\hat{r}_n^i$  be the reconstructed residual of the encoder, i.e.,  $\hat{f}_n^i = \hat{f}_{ref}^j + \hat{r}_n^i$ . If the current pixel is lost at the decoder side, it will copy the pixels of the  $(n-1)^{\text{th}}$  frame. Assuming that the transmission error rate is  $p$ , the pixel value of the reconstructed image can be expressed by:

$$\tilde{f}_n^i = \begin{cases} \hat{f}_{ref}^j + \hat{r}_n^i & 1 - p \\ \tilde{f}_{n-1}^k & p \end{cases} \quad (7)$$

Therefore, the estimated end-to-end distortion of the decoder can be expressed as:

$$\begin{aligned} d(n, i) &= E \left\{ \left( f_n^i - \tilde{f}_n^i \right)^2 \right\} \\ &= (1 - p) E \left\{ \left( f_n^i - \left( \hat{f}_{ref}^j + \hat{r}_n^i \right) \right)^2 \right\} + p E \left\{ \left( f_n^i - \tilde{f}_{n-1}^k \right)^2 \right\} \\ &= (1 - p) E \left\{ \left( f_n^i - \hat{f}_n^i \right)^2 \right\} + (1 - p) E \left\{ \left( \hat{f}_{ref}^j - \tilde{f}_{ref}^j \right)^2 \right\} + p E \left\{ \left( f_n^i - \tilde{f}_{n-1}^k \right)^2 \right\} \\ &= (1 - p) d_s(n, i) + (1 - p) d_{ep}(ref, j) + p d_{ec}(n, i) \end{aligned} \quad (8)$$

It can be seen that the end-to-end distortion can be represented by the sum of the three items  $d_s(n, i)$ ,  $d_{ep}(ref, j)$  and  $d_{ec}(n, i)$ , where  $d_s(n, i)$  represents source distortion,  $d_{ep}(ref, j)$  represents error propagation distortion from the reference frame, and  $d_{ec}(n, i)$  represents error concealment distortion. Error propagation distortion refers to the use of predictive coding in HEVC coding. If an error occurs in a reference frame or a referenced module, the distortion is propagated to the corresponding frame or pixel with reference to it. Error concealment refers to the distortion caused by the recovery of errors or lost modules generated by transmissions in the channel using temporal or spatial correlation. Usually, in the time domain, the motion-compensated content of the previous frame is used to replace the lost portion of the current frame, and the spatial correlation is used to recover the lost portion in the spatial domain.

Since  $d_s(n, i)$  can be obtained by source-encoded parameters at the encoder side, end-to-end distortion mainly depends on the calculation of  $d_{ep}(ref, j)$  and  $d_{ec}(n, i)$ . The result shown in Equation (9) can be obtained by calculating  $d_{ec}(n, i)$ :

$$\begin{aligned}
d_{ec}(n, i) &= E \left\{ \left( f_n^i - \tilde{f}_{n-1}^k \right)^2 \right\} \\
&= E \left\{ \left( f_n^i - \hat{f}_{n-1}^k + \hat{f}_{n-1}^k - \tilde{f}_{n-1}^k \right)^2 \right\} \\
&= E \left\{ \left( f_n^i - \hat{f}_{n-1}^k \right)^2 \right\} + E \left\{ \left( \hat{f}_{n-1}^k - \tilde{f}_{n-1}^k \right)^2 \right\} \\
&= d_{ec\_o}(n, i) + d_{ep}(n-1, k)
\end{aligned} \tag{9}$$

As can be seen from the above formula,  $d_{ec}(n, i)$  consists of two items,  $d_{ec\_o}(n, i)$  and  $d_{ep}(n-1, k)$ , where  $d_{ec\_o}(n, i)$  represents the mean square error between the original image and the encoder-side error-coded pixels, called the original error concealment distortion.  $d_{ec\_o}(n, i)$  can also be obtained by parameter calculation on the encoder side.

It can be seen that  $d_{ep}(n-1, k)$  in Equation (9) and  $d_{ep}(ref, j)$  in Equation (8) represent error propagation distortion. The calculation method of  $d_{ep}(n, i)$  is as shown in:

$$\begin{aligned}
d_{ep}(n, i) &= E \left\{ \left( \hat{f}_n^i - \tilde{f}_n^i \right)^2 \right\} \\
&= (1-p)E \left\{ \left( \hat{f}_n^i - \left( \tilde{f}_{ref}^i + \hat{r}_n^i \right) \right)^2 \right\} + pE \left\{ \left( \hat{f}_n^i - \tilde{f}_{n-1}^k \right)^2 \right\} \\
&= (1-p)E \left\{ \left( \hat{f}_{ref}^j - \tilde{f}_{ref}^j \right)^2 \right\} + pE \left\{ \left( \hat{f}_n^i - \hat{f}_{n-1}^k + \hat{f}_{n-1}^k - \tilde{f}_{n-1}^k \right)^2 \right\} \\
&= (1-p)E \left\{ \left( \hat{f}_{ref}^j - \tilde{f}_{ref}^j \right)^2 \right\} + pE \left\{ \left( \hat{f}_n^i - \hat{f}_{n-1}^k \right)^2 \right\} + pE \left\{ \left( \hat{f}_{n-1}^k - \tilde{f}_{n-1}^k \right)^2 \right\} \\
&= (1-p)d_{ep}(ref, j) + pd_{ec\_r}(n, i) + pd_{ep}(n-1, k)
\end{aligned} \tag{10}$$

It can be seen that  $d_{ep}(n, i)$  consists of three parts, where  $d_{ec\_r}(n, i)$  represents the reconstructed image and the error concealment distortion at the encoder end, which is called reconstructed error concealment distortion. The expression  $d_{ec\_r}(n, i)$  can be obtained by parameter calculation at the decoder end. Therefore, A is mainly generated by the error propagation distortion of its reference frame and the previous frame. In addition, since the first frame in the video sequence is usually an intra-coded frame, the value  $d_{ep}$  of the first frame can be directly obtained without considering error propagation and the values  $d_{eps}$  of the remaining frames can be recursively calculated by Equation (10) on a frame-by-frame basis. In the end-to-end distortion model, the end-to-end distortion of the current frame is obtained by referring to the error propagation distortion of the previous frame, and the subsequent frame is updated by the error propagation distortion of the current frame.

Therefore, from the above analysis, the end-to-end distortion estimation is related to the selection of source-coding, channel-coding, and reference frames.

According to the derivation of the end-to-end distortion estimation method, in the JSCC algorithm based on the rate allocation, in order to select the optimal rate allocation strategy, it is necessary to comprehensively consider the parameters of the source, the channel and the like, so it will be introduced. There is a large number of iterative calculations, so how to choose an effective means to reduce the computational complexity is also a problem worth studying.

Since the end-to-end distortion of the video transmission system is related to the source-coding parameters, the channel-coding parameters, and the reference frame, let  $S$  denote the set of all source-coding options,  $C$  denote the set of all channel-coding options, and  $M$  denote the set of all reference frames. The candidate parameters of the source-coding, channel-coding, and reference frame are  $u, r, ref$ , respectively. A system description that minimizes distortion at the decoder is shown in:

$$\min_{u \in S, r \in C, ref \in M} E[d(u, r, ref)] \tag{11}$$

The traditional JSCC algorithm is based on the principle of minimizing end-to-end distortion as a scheme for selecting a code rate allocation. It is often necessary to traverse all combinations of coding

parameters, and the worst case computational complexity is  $O(|S| \times |C| \times |M|)$ . However, the JSCC scheme based on rate allocation proposed in this paper divides the selection method of the parameter combination into two steps. Firstly, the channel environment is estimated by the clustering algorithm, that is, the code rate  $r(k)$  of the channel coding can be obtained by simple clustering before encoding, and then the obtained channel-coding parameters are brought into the minimum end-to-end distortion estimation to obtain the source-coding parameters and the choice of the best reference frame. In the worst case, the computational complexity is  $O(|S| \times |M|)$ . In addition, in practice, since not all of the source-coding options can be used in any of the FCC coding modes of the channel-coding candidate set  $C$ , the computational complexity may be even lower.

#### 4. Design and Performance Simulation of the Adaptive RC-NB-LDPC Coding Scheme

This section presents the simulation results, which are described in three parts. The first part presents the performance comparison of the constructed RC-NB-LDPC codes with the DVB-S2 short codes. The second part presents the clustering results by the k-mean++ algorithm. The third part presents the advantages of the adaptive coding scheme in compressed image processing.

##### 4.1. Design, Analysis, and Simulation of RC-NB-LDPC Codes

The multi-rate LDPC scheme of DVB-S2 is a typical paradigm of the LDPC codes applied in communication standards, which is designed with full consideration of business diversity requirements and has good adaptability. It supports 11 code rates from  $1/4$  to  $9/10$ , in both normal and short FEC frame modes. The use of ACM is another significant feature of DVB-S2, which provides more accurate channel protection and dynamic connection adaptability for specific receiving terminals according to specific propagation conditions. While improving the original channel transmission capacity, it greatly expands the service range and directly improves network coverage and the user transmission rate. This provides a reference for the ultimate user experience of B5G in the future.

In this section, we first discuss the construction method of RC-NB-LDPC codes, and some parameters such as code length and code rate all refer to the DVB-S2 standard [49]. In addition, we analyze the decoding threshold of the LDPC code by using the EXIT chart [50]. Finally, we simulate the BER performance of the proposed LDPC codes.

##### 4.1.1. Design of RC-NB-LDPC Codes

The LDPC code in the DVB-S2 standard has two kinds of code length, 64800-bit long code, and 16200-bit short-code, and 11 and 10 code rates are specified for long-code and short-code, respectively. The information bit length, check bit length and code-word length of the LDPC code in the DVB-S2 standard are fixed and cannot be arbitrarily selected. Regardless of long-code or short-code, they have the same code-word length and different information  $n$  bit length, respectively. Table 2 shows the specific parameters of six typical rates of short-code in the DVB-S2 standard.

**Table 2.** Parameters of the short codes in the DVB-S2 standard.

Code Rate	Information Bitlength	Code-Word Length	Variable Node Degree	Variable Node Degree Distributions
$4/9$ ( $\approx 1/2$ )	7200	16,200	(1, 2, 3, 8)	(1, 8999, 5400, 1800)/16,200
$2/3$	10,800		(1, 2, 3, 13)	(1, 5399, 9720, 1080)/16,200
$11/15$ ( $\approx 3/4$ )	11,880		(1, 2, 3, 12)	(1, 4319, 11,520, 360)/16,200
$7/9$ ( $\approx 4/5$ )	12,600		(1, 2, 3)	(1, 3599, 12,600)/16,200
$37/45$ ( $\approx 5/6$ )	13,320		(1, 2, 3, 13)	(1, 2879, 12,960, 360)/16,200

Through density evolution optimization analysis, the distribution of variable node degrees for all rate designs is (2, 3, 4), which can achieve excellent waterfall performance. On the finite fields, the designed mother matrix rate is  $7/8$ , the size is  $4 \times 32$ , and the expansion factor is  $p = 127$ .

The RCNBQC-LDPC code matrix is designed based on the concealment puncturing technique, and other sub-rate matrixes are obtained from the mother matrix. The particular design is shown in Figure 12. Each time the four columns of the matrix of the current rate are deleted, the number of columns at the same time as 2, 3, and 4 is reduced by 1, 2, and 1. The above process makes the designed matrix of each rate have a superior degree distribution, and the structure is similar, so that codec is simple to implement. The specific parameters of the designed code are shown in Table 3.

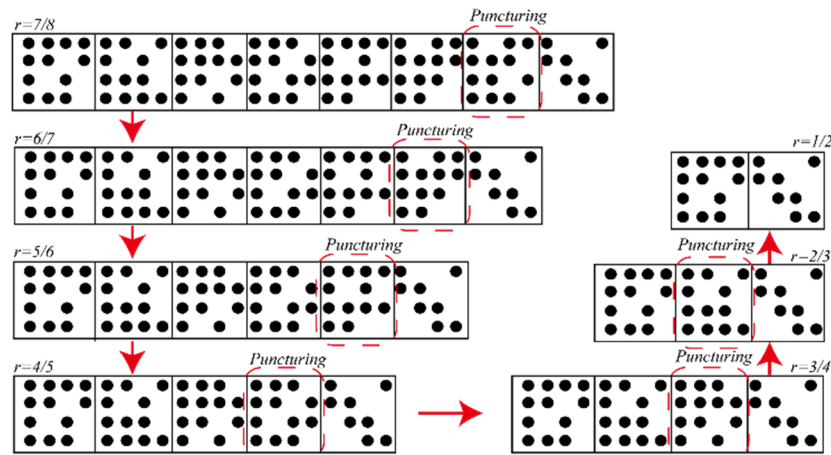


Figure 12. RC-LDPC base matrix scatter diagram.

Table 3. Parameters of the designed RC-NB-LDPC codes.

Code Rate	Information Length (Bit/Symbol)	Code Length (Bit/Symbol)	Base Matrix Size	Expansion Factor	Variable Node Degrees	Variable Node Degree Distributions
7/8	14,224/3556	16,256/4064	$4 \times 32$	127	(4, 3, 2)	(7, 14, 11)/32
6/7	13,920/3480	16,240/4060	$4 \times 28$	145	(4, 3, 2)	(6, 12, 10)/28
5/6	13,440/3360	16,128/4032	$4 \times 24$	169	(4, 3, 2)	(5, 10, 9)/24
4/5	12,992/3248	16,240/4060	$4 \times 20$	203	(4, 3, 2)	(4, 8, 8)/20
3/4	12,144/3036	16,192/4048	$4 \times 16$	253	(4, 3, 2)	(3, 6, 7)/16
2/3	10,816/2704	16,224/4056	$4 \times 12$	338	(4, 3, 2)	(2, 4, 6)/12
1/2	8096/2024	16,192/4048	$4 \times 8$	506	(4, 3, 2)	(1, 2, 5)/8

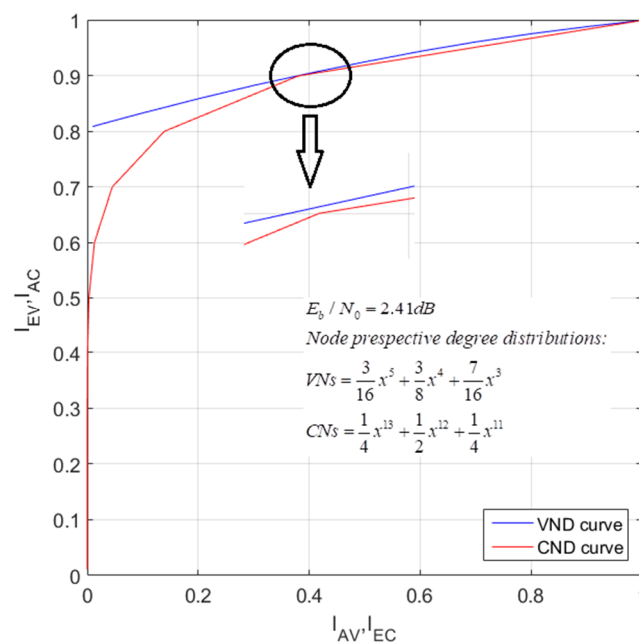
#### 4.1.2. EXIT Chart Analysis of RC-NB-LDPC Codes

The Extrinsic-Information-Transfer (EXIT) chart technique is a graphical tool for estimating the decoding threshold of the LDPC code, which is based on Gaussian approximation and can provide intuitive information for the dynamics and convergence of iterative decoding [50]. The EXIT chart is based on the principle that Variable-Node Processors (VNPs) and Check-Node Processors (CNPs) work together and iteratively make bit decisions, with each half-iteration improving the performance metrics. For VNP and CNP, we can obtain the transfer curve of the input metrics and output metrics. The transfer curve of VNP depends on the SNR of the channel. Since the output metric of a processing unit is the input metric of its neighboring processing unit, we can draw two transfer curves in the same coordinate system, with the abscissa and ordinate of a processing unit being exchanged. Such a graph can help predict the code set and decoding threshold for a given variable node and check node degree distribution: the decoding threshold is the signal-to-noise ratio at the focal point of the VNP and CNP transfer curves, which impede the convergence of the two processing units.

The LDPC decoder can be thought of as a bipartite graph. Its parity relationship corresponds to the check nodes. The transmission code-word corresponds to the left  $n$  variable nodes. If there is an edge between the  $i$  check node and the  $j$  variable node in the bipartite graph, the element  $H_{i,j}$  in the corresponding check matrix is a non-zero element. The decoding process of the LDPC code is as







**Figure 14.** EXIT chart of the designed degree distributions.

**Table 4.** Threshold Predictions of Designed Codes.

Code Rates	Thresholds under AWGN Channel (dB)	
	DVB-S2 Scheme	RC-NB-LDPC Codes
7/8	----	2.87
6/7	----	2.81
5/6	2.86	2.72
4/5	2.70	2.59
3/4	2.58	2.41
2/3	2.38	2.12
1/2	2.02	1.80

#### 4.1.3. Simulation of RC-NB-LDPC Codes

This section performs performance simulation and analysis of the RC-NB-LDPC code. The parameters are as follows: the decoding algorithm is a Mixed-LOG-FFT-BP decoding algorithm, the maximum number of iterations is 20, the modulation mode is BPSK, and the channel is an AWGN channel. Monte Carlo simulation was performed and the following results were obtained.

Figures 15 and 16 simulate the bit error rate (BER) performance of the RC-NB-LDPC codes with different code rates. As can be seen from the figure, the constructed LDPC code achieves better waterfall performance in a larger code rate range. Moreover, the constructed RC-NB-LDPC code achieves a performance gain of about 0.2–0.5 dB compared to the LDPC code in the DVB-S2 standard.

Figure 17 simulates the frame error rate (FER) performance of the RC-NB-LDPC codes with different code rates. It can be seen that the RC-NB-LDPC code achieves better anti-interference performance in a larger code rate range.

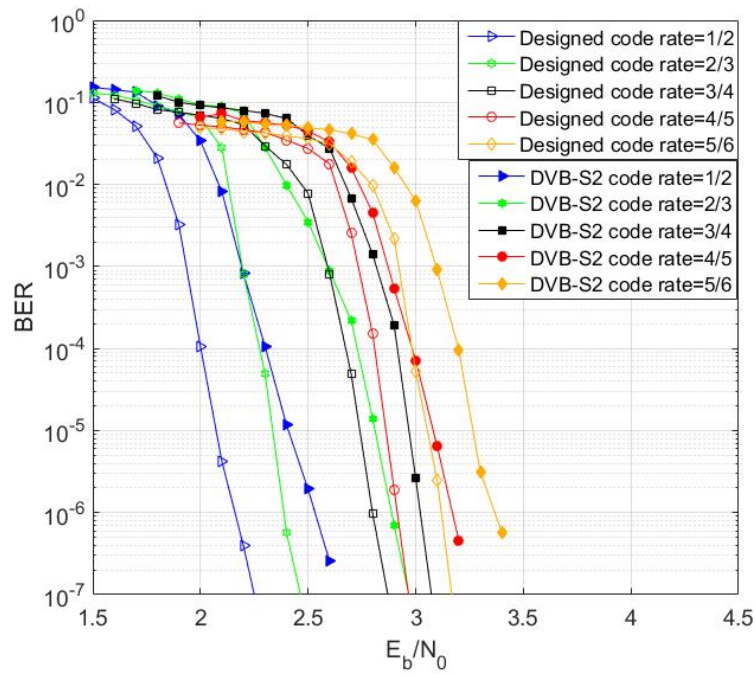


Figure 15. BER simulation comparison result.

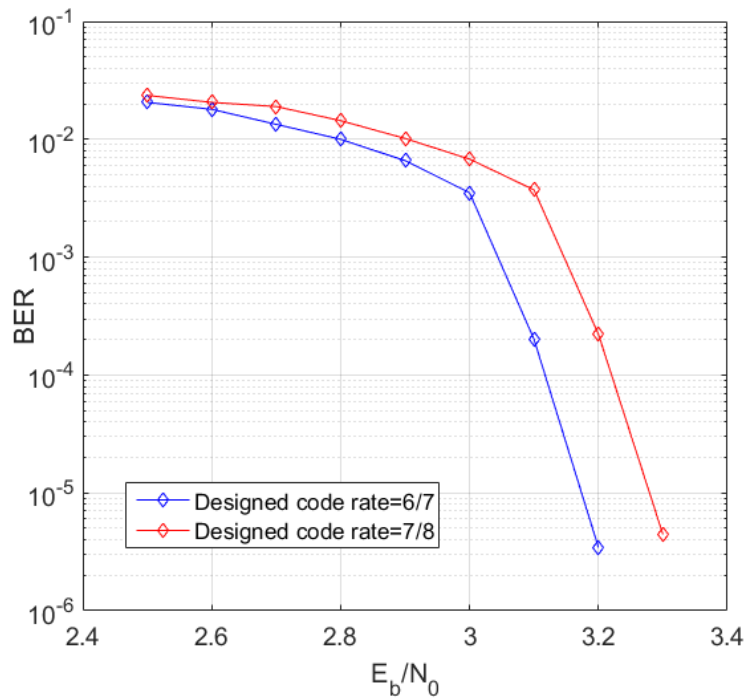


Figure 16. Designed 7/8 rate code performance.

Figure 18 simulates the average number of iterations of the RC-NB-LDPC code with a code rate of 1/2. As can be seen from the figure, the higher the  $E_b/N_0$ , the smaller the average number of iterations. In addition, the RC-NB-LDPC codes have low decoding delay and are suitable for real-time communication scenarios.

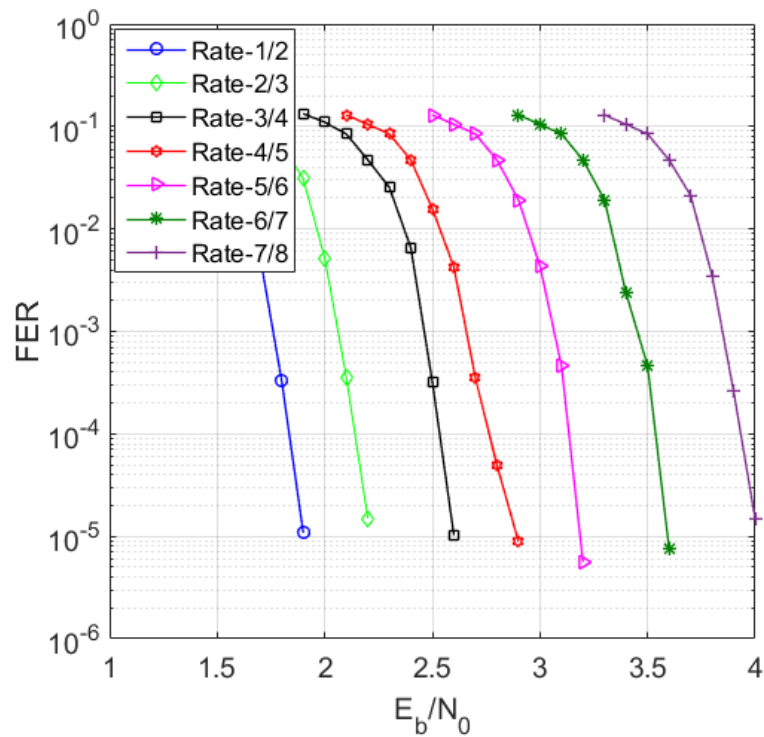


Figure 17. FER simulation result.

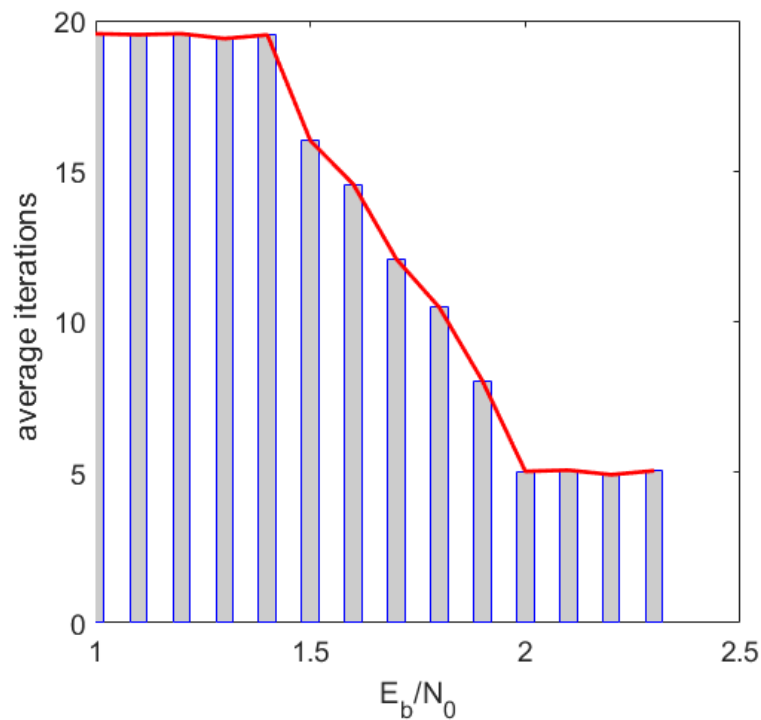


Figure 18. Average iterations simulation result at R = 0.5.

#### 4.2. Simulation of the Channel Clustering Algorithm

In this section, we combine the physical factors that affect the state of the wireless channel with the channel bandwidth to form a training sample. The physical factors in the wireless channel are represented by the signal-to-noise ratio (SNR), and the value range is [0, 4.5] dB. The transmission bandwidth is [1000, 8000] KHz. The training sample data set consists of a data set that conforms to a

Gaussian distribution and contains 2000 data points. The k-means++ algorithm is used to cluster data, and  $k = 3$ . The clustering result is shown in Figure 10.

The data points with different symbols in Figure 19 indicate different categories. It can be seen that a series of unordered channel data is divided into three categories by the clustering algorithm.

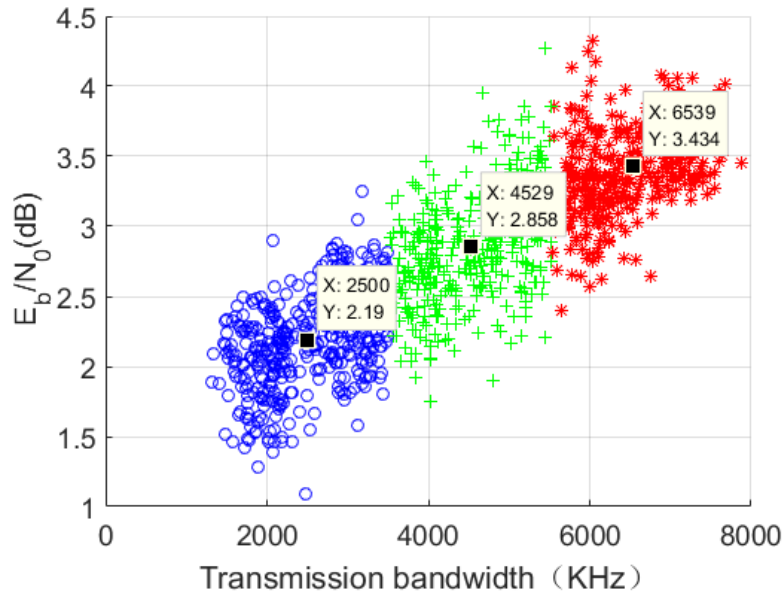


Figure 19. Clustering of simulated wireless channel conditions.

### 4.3. Design and Performance Simulation of the Adaptive RC-LDPC Coding Scheme

#### 4.3.1. The Design of the Adaptive RC-LDPC Coding Scheme

In this section, we take the image transmission as an example, and propose a rate-compatible LDPC coding scheme based on channel clustering, as shown in Figure 20. The basic process of the program is divided into three parts:

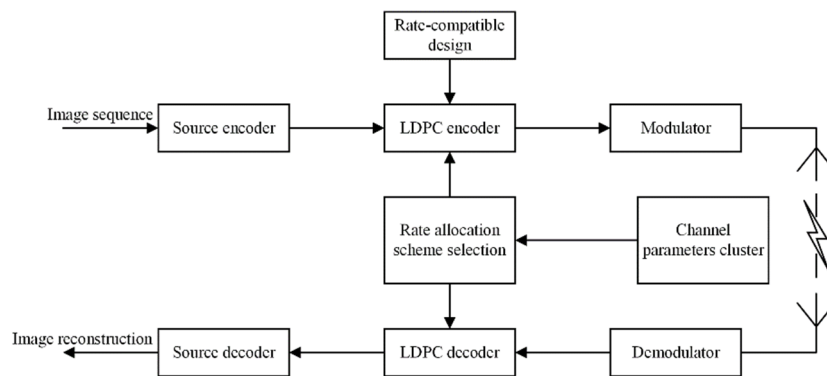


Figure 20. The adaptive RC-LDPC coding scheme block diagram.

#### 1. Channel clustering

Using the data set composed of different channel environment parameters and user usage behavior as training data, the k-means++ algorithm is used to divide the wireless channel situation into  $k$  different classes.

#### 2. Rate allocation

We use the set partitioning in hierarchical trees (SPIHT) algorithm as the image compression algorithm. The SPIHT algorithm has extremely low computational complexity and high-quality recovery characteristic and breaks the boundary between coding efficiency and complexity in traditional coding algorithms [51]. It makes reasonable use of multi-resolution characteristics after wavelet decomposition and has good coding performance. The higher compression rate causes the output code stream to be very sensitive to channel errors. Therefore, the worst one of each channel class is selected as the basis for designing the code rate allocation, ensuring that under all channel conditions in the channel class, the compressed code stream is transmitted as accurately as possible. A corresponding code rate allocation scheme is designed according to the worst channel condition in each cluster.

### 3. Encode and update the cluster center

When the image is transmitted in the wireless channel, the code rate allocation scheme is adaptively adjusted by the clustering of the current channel data, and the corresponding source and channel coding are performed by using the code rate allocation scheme. Finally, the real-time data of the channel at this time is used as training data to update the cluster center.

#### 4.3.2. Simulation Results and Analysis

##### 1. Image quality evaluation

We evaluate the performance of the transmission scheme by comparing the quality of the reconstructed image with the original image. The evaluation methods include subjective evaluation and objective evaluation.

Subjective evaluation is a direct and important evaluation method for subjective judgment by observing and comparing the difference between the reconstructed image and the original image. However, in practice, individual differences in observers are likely to cause subjective bias. Moreover, subjective evaluation often requires many labor and material resources, and it is impossible to perform real-time detection.

The objective evaluation method usually uses the error between the reconstructed image and the original image to measure the quality of the reconstructed image. Common evaluation methods include Mean Squared Error (MSE) and Peak Signal Noise Ratio (PSNR) [51]. The calculation formula for MSE is:

$$\text{MSE} = \frac{\sum_{i=1}^M \sum_{j=1}^N (f(i,j) - f(i,j'))^2}{M \times N} \quad (16)$$

where  $f(i,j)$  and  $f(i,j)'$  represent the pixel values of the original image and the reconstructed image at  $(i,j)$ , respectively, and  $M \times N$  is the size of the image. MSE reflects the quality of image restoration. It can be seen that the smaller the MSE value, the smaller the deviation between the reconstructed image and the original image, and vice versa. The formula for calculating PSNR is:

$$\text{PSNR} = 10 \log_{10} \frac{255}{\text{MSE}} \quad (17)$$

PSNR and MSE are essentially the same. The larger the PSNR value, the smaller the deviation between the restored image and the original image, and vice versa. PSNR is often viewed as a quality metric and MSE as a distortion metric.

##### 2. Reconstructed image quality simulation

In this paper, we select the "Lena"  $512 \times 512$  grayscale image as the test image. The source-coding method is the SPIHT algorithm, the channel-coding method is rate-compatible LDPC code, and the



channel model is the AWGN channel. We select three different channel conditions as test conditions. The specific parameters of the channel are shown in Table 5.

**Table 5.** Test channel parameter.

Channel Types	Test Parameters [ $E_b/N_0$ , Bandwidth]
Class 1	[2.2, 2500 KHz]
Class 2	[2.8, 4500 KHz]
Class 3	[3.5, 6500 KHz]

We performed a performance analysis comparison of the following three transmission schemes:

- Scheme 1, the fixed-rate transmission scheme. That is, the channel coding uses a fixed code rate of 3/4 for three types of test channels in this paper.
- Scheme 2, the adaptive-rate transmission scheme. That is, after channel clustering, the corresponding rate is selected for each type of test channel. Corresponding to three types of test channels, the coding rates used in this paper are 1/2, 4/5, and 7/8, respectively.
- Scheme 3, the adaptive-rate transmission scheme with UEP. According to different parts of the code stream that have different effects on image recovery, we adopt different levels of the channel protection mechanism, that is, the high-level protection for the important code stream and the low-level protection for the secondary code stream. Corresponding to the three types of test channels, the coding rate groups in this paper are (1/2, 3/4), (2/3, 5/6) and (3/4, 7/8), respectively.

The scheme affects the different degrees of the restored image according to different parts of the code stream, adopts different levels of channel protection mechanisms, protects the important code stream to a high degree, and protects the secondary code stream to a low degree to achieve different degrees of differentiation.

In this paper, the PSNR of the reconstructed image is chosen as the criterion for evaluating image quality. The simulation results are shown in Table 6. The “fail” in the table indicates that the decoding is unsuccessful or the reconstructed image quality is extremely poor due to severe error. In order to ensure the correctness of the results, we repeat the transmission process 50 times and take the average result.

**Table 6.** Reconstructed image PSNR for different schemes.

Compression Ratio	Test Schemes	PSNR (dB)		
		Class1	Class2	Class3
1	Scheme 1	Fail	33.53	35.94
	Scheme 2	31.79	35.24	37.68
	Scheme 3	32.38	35.75	38.23
0.5	Scheme 1	Fail	30.71	32.77
	Scheme 2	28.86	32.47	34.64
	Scheme 3	29.33	32.91	35.14
0.1	Scheme 1	Fail	27.82	29.24
	Scheme 2	27.26	29.01	29.32
	Scheme 3	27.68	29.28	29.32

As shown in Table 6, with the fixed bit rate scheme, in the case of poor channel conditions, a decoding failure occurs. This is because the fixed bit rate employed does not provide sufficient channel protection under poor conditions. For image compression streams, if the error occurs at a critical location and there is no error retransmission or other means of recovery, it will result in decoding failure or poor reconstructed image quality.

It can also be seen that compared to scheme 1, scheme 2 achieves an average gain of about 2 dB. The reason is that scheme 2 adopts different code rates according to different channel conditions. When the channel condition is poor, low code rate coding is used to enhance channel protection. When the channel environment is good, high code rate coding is used to improve the overall efficiency of the system.

In addition, by comparing the data of scheme 2 and scheme 3, we can also see that since scheme 3 adopts UEP, the reconstructed image quality can be further improved compared with scheme 2, and the PSNR of the reconstructed image can obtain an average of about 0.5 dB.

At an image compression ratio of 0.5, Figure 21 shows reconstructed images obtained by using the three different schemes under the conditions of the channel environment of Class 2. Image (a) is the original image, and image (b), image (c) and image (d) are reconstructed images using the three schemes, respectively. We can see that the image quality of image (b), image (c) and image (d) is sequentially improved. Moreover, image (d) and the original image (a) are almost identical.



**Figure 21.** Reconstruction of images under different schemes. (a) is the original image, and image (b–d) are reconstructed images using the three schemes, respectively.

## 5. Conclusions

For the B5G mobile system, this paper first proposed a construction method of the RC-NB-LDPC code, which is realized by combining masking and puncturing technology. The RC-LDPC code constructed by the method has an easy hardware implementation of the check matrix structure and better waterfall performance in a larger code rate range. Compared with the LDPC codes in the

DVB-S2 standard, the RC-NB-LDPC codes achieved a performance gain of about 0.2–0.5 dB, which is suitable for data transmission. Then, according to the influencing factors of the wireless channel, this paper proposed using the k-means++ algorithm to cluster the wireless channel environment to obtain the code rate allocation scheme. Finally, this paper took compressed image transmission as an example, applied the clustering algorithm to the fast classification of channel environment, combined the characteristics of the SPIHT code stream structure, and studied the problem of compressed image transmission in the wireless channel. In the process of transmitting the compressed image code stream, the proposed adaptive-rate transmission scheme can effectively utilize the limited bandwidth resources of the channel to ensure reliable transmission of image information, and obtain higher reconstructed image quality than the fixed code rate-coding scheme. Further enhancements can be obtained by combining the UEP strategy. Through simulation, the adaptive-rate transmission scheme can achieve a performance improvement of about 2 dB over the fixed-rate transmission scheme. Moreover, using the adaptive-rate transmission scheme with UEP, the PSNR of the reconstructed image can be additionally increased by about 0.5 dB on average.

This paper only provides an adaptive channel-coding scheme, but the other advanced source-coding techniques can be used to further design a better joint source channel-coding scheme. In addition, channel-coding technology can be combined with modulation, power control and other technologies to design a more complete adaptive transmission scheme.

**Author Contributions:** D.-f.Z. and H.T. conceived the ideas and concept. H.T. implemented the software, carried out the experiments, and wrote the manuscript. H.T. and R.X. carried out simulation experiments and further verified the experimental results. D.-f.Z. and R.X. critically reviewed the paper.

**Funding:** This research work was supported in part by the ZTE Cooperation Research Forum Project under Grant No. KY10800160020, and in part by Foundation of China Shipbuilding Industry Corporation 722 Research Institute under Grant KY10800170051.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gawas, U. An Overview on Evolution of Mobile Wireless Communication Networks: 1 g–6 g. *Int. J. Recent Innov. Trends Comput. Commun.* **2015**, *3*, 3130–3133.
2. Li, B.; Fei, Z.; Zhang, Y. UAV Communications for 5G and Beyond: Recent Advances and Future Trends. *IEEE Internet Things J.* **2018**. [CrossRef]
3. Li, H.; Bai, B.; Mu, X.; Zhang, J.; Xu, H. Algebra-Assisted Construction of Quasi-Cyclic LDPC Codes for 5G New Radio. *IEEE Access* **2018**, *6*, 50229–50244. [CrossRef]
4. Hui, D.; Sandberg, S.; Blankenship, Y.; Andersson, M.; Grosjean, L. CHANNEL CODING IN 5G NEW RADIO. *IEEE Veh. Technol. Mag.* **2018**, *13*, 60–69. [CrossRef]
5. Richardson, T.; Kudekar, S. Design of Low-Density Parity Check Codes for 5G New Radio. *IEEE Commun. Mag.* **2018**, *56*, 28–34. [CrossRef]
6. Wu, X.; Jiang, M.; Zhao, C. Decoding Optimization for 5G LDPC Codes by Machine Learning. *IEEE Access* **2018**, *6*, 50179–50186. [CrossRef]
7. Wu, X.; Jiang, M.; Zhao, C.; Ma, L.; Wei, Y. Low-Rate PBRL-LDPC Codes for URLLC in 5G. *IEEE Wirel. Commun. Lett.* **2018**. [CrossRef]
8. Zhang, Y.; Peng, K.; Song, J. Enhanced IDMA with Rate-Compatible Raptor-Like Quasi-Cyclic LDPC Code for 5G. In Proceedings of the 2017 IEEE Globecom Workshops, Singapore, 4–8 December 2017.
9. Zhang, Y.; Peng, K.; Wang, X.; Song, J. Performance Analysis and Code Optimization of IDMA With 5G New Radio LDPC Code. *IEEE Commun. Lett.* **2018**, *22*, 1552–1555. [CrossRef]
10. Promotion Group. *White Paper on 5G Wireless Technology Architecture*; Promotion Group IMT-2020(5G): 2015. Available online: <http://www.imt-2020.cn/en/documents/3> (accessed on 4 January 2019).
11. Promotion Group. *White Paper on 5G Network Technology Architecture*; Promotion Group IMT-2020(5G): 2016. Available online: <http://www.imt-2020.cn/en/documents/3> (accessed on 4 January 2019).
12. Torrance, J.M.; Hanzo, L. Upper bound performance of adaptive modulation in a slow Rayleigh fading channel. *Electron. Lett.* **1996**, *32*, 718–719. [CrossRef]

13. Bender, P.; Black, P.; Grob, M.; Padovani, R.; Sindhusyana, N.; Viterbi, S. CDMA/HDR: A bandwidth efficient high speed wireless data service for nomadic users. *IEEE Commun. Mag.* **2000**, *38*, 70–77. [[CrossRef](#)]
14. Abusedra, L.F.; Daeri, A.M.; Zerek, A.R. Implementation and performance study of the LDPC coding in the DVB-S2 link system using MATLAB. In Proceedings of the 2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 19–21 December 2016; pp. 669–674.
15. Doost, R.; Kashani, Z.G.; Mohammadi, L. BER investigation in satellite DVB-S2 service by modeling the transponder non-idealities. In Proceedings of the 7th International Symposium on Telecommunications (IST'2014), Tehran, Iran, 9–11 September 2014; pp. 1050–1053.
16. Gallager, R. Low Density Parity-Check Codes. Ph.D. Thesis, MIT Press, Cambridge, MA, USA, 1963.
17. MacKay, D.J.; Neal, R.M. Near Shannon limit performance of low density parity check codes. *Electron. Lett.* **1996**, *32*, 1645–1646. [[CrossRef](#)]
18. MacKay, D.J. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory* **1999**, *45*, 399–431. [[CrossRef](#)]
19. Zongwang, L.; Lei, C.; Lingqi, Z.; Shu, L.; Wai, F. Efficient encoding of quasi-cyclic low-density parity-check codes. In Proceedings of the IEEE Global Telecommunications Conference, St. Louis, MO, USA, 28 November–2 December 2005; p. 6.
20. Zhixing, Y.; Qiuliang, X.; Kewu, P.; Jian, F. A fast and efficient encoding structure for QC-LDPC codes. In Proceedings of the 2008 International Conference on Communications, Circuits and Systems, Xiamen, China, 25–27 May 2008; pp. 16–20.
21. Verdier, F.; Declercq, D. A low-cost parallel scalable FPGA architecture for regular and irregular LDPC decoding. *IEEE Trans. Commun.* **2006**, *54*, 1215–1223. [[CrossRef](#)]
22. Karkooti, M.; Cavallaro, J.R. Semi-parallel reconfigurable architectures for real-time LDPC decoding. In Proceedings of the International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004, Las Vegas, NV, USA, 5–7 April 2004; Volume 571, pp. 579–585.
23. Li, J.R.; Narayanan, K.R. Rate-compatible Low Density Parity Check Codes for Capacity-approaching ARQ Schemes in Packet Data Communications. In Proceedings of the IASTED International Conference on Communications, Internet, and Information Technology, St. Thomas, US Virgin Islands, 18–20 November 2002.
24. Ha, J.; Kim, J.; Klinc, D.; McLaughlin, S.W. Rate-compatible punctured low-density parity-check codes with short block lengths. *IEEE Trans. Inf. Theory* **2006**, *52*, 728–738. [[CrossRef](#)]
25. Hyo, Y.P.; Jae, W.K.; Kwang, S.; Keum, C.W. Efficient puncturing method for rate-compatible low-density parity-check codes. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 3914–3919. [[CrossRef](#)]
26. Wu, S.X.; Mow, W.H. Efficient Ranking of Rate-Compatible Puncturing Patterns for a Given LDPC Code Matrix. In Proceedings of the GLOBECOM 2009—2009 IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
27. Liu, X.; Wu, X.; Zhao, C. Shortening for irregular QC-LDPC codes. *IEEE Commun. Lett.* **2009**, *13*, 612–614. [[CrossRef](#)]
28. Angelone, M.; Ginesi, A.; Re, E.; Cioni, S. Performance of a combined dynamic rate adaptation and adaptive coding modulation technique for a DVB-RCS2 system. In Proceedings of the 2012 6th Advanced Satellite Multimedia Systems Conference, ASMS 2012 and 12th Signal Processing for Space Communications Workshop, SPSC 2012, Baiona, Spain, 5–7 September 2012; pp. 124–131.
29. Chen, F.; Meiya, C.; Lijun, S.; Xin, Z.; Dacheng, Y. Research on threshold adjustment algorithm in Adaptive Modulation and Coding. In Proceedings of the 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, Helsinki, Finland, 11–14 September 2006.
30. Floor, P.A.; Kim, A.N.; Ramstad, T.A.; Balasingham, I.; Wernersson, N.; Skoglund, M. On Joint Source-Channel Coding for a Multivariate Gaussian on a Gaussian MAC. *IEEE Trans. Commun.* **2015**, *63*, 1824–1836. [[CrossRef](#)]
31. Kurniawan, E.; Zhiwei, L.; Sun, S. Machine Learning-Based Channel Classification and Its Application to IEEE 802.11ad Communications. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.



32. Lee, C.; Yilmaz, H.B.; Chae, C.B.; Farsad, N.; Goldsmith, A. Machine learning based channel modeling for molecular MIMO communications. In Proceedings of the 2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Sapporo, Japan, 3–6 July 2017; pp. 1–5.
33. Condrobimo, A.R.; Abbas, B.S.; Trisetyarso, A.; Suparta, W.; Kang, C.H. Data mining technique with cluster analysis use K-means algorithm for LQ45 index on Indonesia stock exchange. In Proceedings of the 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 6–7 March 2018; pp. 885–888.
34. De Carvalho, F.D.T.; Simoes, E.C.; Santana, L.V.C.; Ferreira, M.R.P. Gaussian kernel c-means hard clustering algorithms with automated computation of the width hyper-parameters. *Pattern Recognit.* **2018**, *79*, 370–386. [[CrossRef](#)]
35. Yazdani, M.R.; Banihashemi, A.H. On construction of rate-compatible low-density parity-check codes. *IEEE Commun. Lett.* **2004**, *8*, 159–161. [[CrossRef](#)]
36. Li, J.; Liu, K.; Lin, S.; Abdel-Ghaffar, K.; Ryan, W.E. An unnoticed strong connection between algebraic-based and protograph-based LDPC codes, Part I: Binary case and interpretation. In Proceedings of the Information Theory and Applications Workshop, ITA 2015, San Diego, CA, USA, 1–6 February 2015; pp. 36–45.
37. Xiao-Yu, H.; Eleftheriou, E.; Arnold, D. Progressive edge-growth Tanner graphs. In Proceedings of the GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270), San Antonio, TX, USA, 25–29 November 2001; Volume 1002, pp. 995–1001.
38. Xiao-Yu, H.; Eleftheriou, E.; Arnold, D.M. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inf. Theory* **2005**, *51*, 386–398. [[CrossRef](#)]
39. Yu, K.; Shu, L.; Fossorier, M.P.C. Low density parity check codes based on finite geometries: A rediscovery. In Proceedings of the 2000 IEEE International Symposium on Information Theory (Cat. No.00CH37060), Sorrento, Italy, 25–30 June 2000; p. 200.
40. Lan, L.; Zeng, L.; Tai, Y.Y.; Chen, L.; Lin, S.; Abdel-Ghaffar, K. Construction of Quasi-Cyclic LDPC Codes for AWGN and Binary Erasure Channels: A Finite Field Approach. *IEEE Trans. Inf. Theory* **2007**, *53*, 2429–2458. [[CrossRef](#)]
41. Vasic, B. Combinatorial constructions of low-density parity check codes for iterative decoding. In Proceedings of the IEEE International Symposium on Information Theory, Lausanne, Switzerland, 30 June–5 July 2002; p. 312.
42. Ammar, B.; Honary, B.; Kou, Y.; Lin, S. Construction of low density parity check codes: A combinatoric design approach. In Proceedings of the IEEE International Symposium on Information Theory, Lausanne, Switzerland, 30 June–5 July 2002; p. 311.
43. Li, J.; Liu, K.; Lin, S.; Abdel-Ghaffar, K. Algebraic Quasi-Cyclic LDPC Codes: Construction, Low Error-Floor, Large Girth and a Reduced-Complexity Decoding Scheme. *IEEE Trans. Commun.* **2014**, *62*, 2626–2637. [[CrossRef](#)]
44. Xu, H.; Feng, D.; Luo, R.; Bai, B. Construction of Quasi-Cyclic LDPC Codes via Masking With Successive Cycle Elimination. *IEEE Commun. Lett.* **2016**, *20*, 2370–2373. [[CrossRef](#)]
45. Asvadi, R.; Banihashemi, A.H. A Rate-Compatible Puncturing Scheme for Finite-Length LDPC Codes. *IEEE Commun. Lett.* **2013**, *17*, 147–150. [[CrossRef](#)]
46. Mitchell, D.G.M.; Lentmaier, M.; Pusane, A.E.; Costello, D.J. Randomly Punctured LDPC Codes. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 408–421. [[CrossRef](#)]
47. Oktar, Y.; Turkan, M. A review of sparsity-based clustering methods. *Signal Process.* **2018**, *148*, 20–30. [[CrossRef](#)]
48. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [[CrossRef](#)] [[PubMed](#)]
49. EBU. *Digital Video Broadcasting (DVB-S2) Standard Draft ETSI EN 302 307-1*; EBU: 2014. Available online: <https://www.etsi.org/> (accessed on 4 January 2019).

50. El-Hajjar, M.; Hanzo, L. EXIT Charts for System Design and Analysis. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 127–153. [[CrossRef](#)]
51. Li, Q.; Chen, D.; Jiang, W.; Liu, B.; Gong, J. Generalization of SPIHT: Set Partition Coding System. *IEEE Trans. Image Process.* **2016**, *25*, 713–725. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).