

Weighted BATS Codes with LDPC Precoding

Wenyue Zhang¹  and Min Zhu^{1,2,*} 

¹ The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
² Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081, China
* Correspondence: zhunanzhumin@gmail.com

Abstract: Batched Sparse (BATS) codes are a type of network coding scheme that use a combination of random linear network coding (RLNC) and fountain coding to enhance the reliability and efficiency of data transmission. In order to achieve unequal error protection for different data, researchers have proposed unequal error protection BATS (UEP-BATS) codes. However, current UEP-BATS codes suffer from high error floors in their decoding performance, which restricts their practical applications. To address this issue, we propose a novel UEP-BATS code scheme that employs a precoding stage prior to the weighted BATS code. The proposed precoding stage utilizes a partially regular low-density parity-check (PR-LDPC) code, which helps to mitigate the high error floors in the weighted BATS code. We derive the asymptotic performance of the proposed scheme based on density evolution (DE). Additionally, we propose a searching algorithm to optimize precoding degree distribution within the complexity range of the precoding stage. Simulation results show that compared to the conventional weighted BATS codes, our proposed scheme offers superior UEP performance and lower error floor, which verifies the effectiveness of our scheme.

Keywords: network coding; BATS codes; LDPC codes; unequal error protection; density evolution



Citation: Zhang, W.; Zhu, M. Weighted BATS Codes with LDPC Precoding. *Entropy* **2023**, *25*, 686. <https://doi.org/10.3390/e25040686>

Academic Editors: Yunjiang Wang, Junbin Fang, Ke Xiong and Fan Cheng

Received: 6 March 2023

Revised: 5 April 2023

Accepted: 11 April 2023

Published: 19 April 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Data packets may be lost during transmission in wireless network communication due to path fading and interference. Network coding [1], which allows relay nodes to encode packets, is an effective forward error correction (FEC) code in wireless erasure networks. Random linear network coding (RLNC) [2] is a typical network coding scheme, whose coding coefficients are randomly selected over a finite field. RLNC can improve throughput effectively. However, conventional RLNC has high decoding complexity and needs big cache capacity, which restrict its application. As a new network coding scheme, the BATS code [3] has been extensively studied in recent years due to its low complexity and rateless characteristic.

Input packets are encoded in batches by the BATS code, which is composed of the outer code and the inner code. The outer code adopts the matrix form of the Luby Transform (LT) code, which is a kind of fountain codes. The BATS code is rateless because the outer code can produce any number of batches. The inner code adopts RLNC in the same batch. Designing the degree distribution function for BATS codes can effectively enhance their decoding performance when employing the belief propagation (BP) algorithm. In [3,4], the design methods of degree distribution of the BATS code are given in the case of infinite code length and finite code length, respectively. The benefit of the BATS code is that only a fixed number of packets need to be buffered and processed by the relay node for the fixed batch size of the BATS code.

In order to improve the performance of BATS codes, Zhou et al. [5] investigated the design of the inner code to maximize the expected batch transfer matrix rank normalized by the total number of packets transmitted by both the source and intermediate nodes. Juan Yang et al. [6] proposed an improved algorithm which iteratively performs BP decoding and incremental Gaussian elimination for decoding finite-length BATS codes for improving

the decoding performance. To achieve lower latency, a sliding window framework was analyzed in [7], which divides the information blocks into smaller sub-blocks and jointly optimizes the degree distribution and window selection probability of each sub-window. Wang et al. [8] proposed cascading LDPC codes before BATS codes.

BATS code has great application prospects in many scenarios. Gao et al. [9] proposes a joint infrastructure-to-vehicle and vehicle-to-vehicle communication scheme using batched sparse coding to efficiently distribute content to vehicles passing by roadside units, reducing transmission delay and traffic overhead. Yeung et al. [10] discussed the potential and prospects of using BATS codes for space communication. Wang et al. [11] investigated the use of batched sparse (BATS) codes in a butterfly network for multicast communication.

Conventional BATS codes can only provide equal error protection (EEP) for all data. However, in the scenarios of image and video data transmission, some data require higher reliability than others. So unequal error protection (UEP) technology is needed. Xu et al. proposed the weighted BATS code [12] and the expanding window BATS (EW-BATS) code [13] with UEP characteristics. Xiang et al. [14] proposed the feedback expanding window BATS (FEW-BATS) code, which decreases average overhead of successful decoding in the less important packets (LIP) and has no effect on the decoding performance of the more important packets (MIP) by adding a single feedback on the basis of the EW-BATS code.

The conventional UEP-BATS codes have a high error floor. This is because BATS codes randomly select input packets during encoding, which leads a small number of input packets participating in encoding only a few times or even never participating in encoding. To solve this problem, we propose a new UEP-BATS code scheme which performs LDPC precoding before the weighted BATS code. In order to obtain better UEP, we adopt partially regular LDPC (PR-LDPC) [15] as the precoding scheme. We derive the density evolution (DE) analysis when the code proposed in this paper is decoded by BP algorithm on binary erasure channel (BEC) and optimize the relevant parameters according to the DE analysis. Finally, we verify the performance of the proposed code through simulations.

2. BATS Codes and Weighted BATS Codes

2.1. BATS Code

The BATS code consists of the outer code and the inner code. We assume that there are K input packets, and the set of input packets is denoted by $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K\}$ where \mathbf{s}_k is the k th input packet and each packet consists of L bits. The outer code encodes the input packets into batches. We assume that the batch size is M and the degree distribution is denoted by $\Omega(x) = \sum_{d=1}^D \Omega_d x^d$, where Ω_d is the probability of generating a degree value of d and D is the maximum degree. The encoding process in i th batch of the BATS code is given as follows:

- (1) A degree value d_i is generated according to $\Omega(x)$.
- (2) d_i input packets in \mathbf{S} are randomly selected to generate \mathbf{S}_i .
- (3) Batch $\mathbf{C}_i = \mathbf{S}_i \mathbf{G}_i$ is generated, where \mathbf{G}_i is a random matrix with its elements on a finite field \mathbb{F}_q with size q and dimension $d_i \times M$.
- (4) The inner code of the BATS code is RLNC in the batch.

Then the encoded packets are sent to the downstream node. Assuming that the transmission matrix of the i th batch is \mathbf{A}_i , the packets of the i th batch received by the destination node are

$$\mathbf{Y}_i = \mathbf{C}_i \mathbf{A}_i = \mathbf{S}_i \mathbf{G}_i \mathbf{A}_i, \quad (1)$$

where \mathbf{A}_i has dimension $M \times M_i'$, and M_i' denotes the number of packets received in the i th batch.

The number of batches generated is $n = \frac{(1+\gamma)K}{M}$ where γ is the encoding redundancy. We let $\mu = \Omega'(1)$ and $\theta = \frac{1+\gamma}{M}\mu$ denote the average degree of output batches and input packets, respectively. $\Lambda(x) = \sum_i \Lambda_i x^i$ is the degree distribution of the input packets, where

$$\Lambda_i = \binom{n}{i} \left(\frac{\mu}{K}\right)^i \left(1 - \frac{\mu}{K}\right)^{n-i} \approx \frac{e^{-\theta} \theta^i}{i!}. \tag{2}$$

BATS codes can be decoded using the BP algorithm. In [16], the AND-OR tree analysis of BATS codes using the BP algorithm is introduced in detail. In the AND-OR tree, the OR-nodes denote the input data packets, and the AND-nodes denote the output batches. The root node of AND-OR tree is an OR-node. The children of OR-nodes are AND-nodes and the children of AND-nodes are OR-nodes. Suppose $\mathbf{h} = [h_1, h_2, \dots, h_M]$ denotes the channel rank distribution and h_r denotes the probability that the rank of the batch received by the destination node is r . There are the following conclusions about AND-OR tree:

- The probability that an OR-node has i children is $\delta_i = \frac{(i+1)\Lambda_{i+1}}{\theta}, i = 0, 1, \dots, n - 1$.
- The probability that an AND-node has i children is $\omega_i = \frac{(i+1)\Omega_{i+1}}{\mu}, i = 0, 1, \dots, D - 1$.
- An OR-node is decodable only when any of its children are decodable.
- An AND-node is associated with rank r with probability h_r where $r = 1, 2, \dots, M$ and it is decodable if less than $r - 1$ children are undecodable.

The probability that the root node is undecodable in the l th iteration is

$$\begin{cases} y_0 = 1, \\ y_l = \delta(1 - \omega(1 - y_{l-1})), \end{cases} \tag{3}$$

where

$$\delta(x) = \sum_{i=0}^{n-1} \delta_i x^i = e^{\theta(x-1)}, \tag{4}$$

and

$$\omega(1 - x) = \mu \sum_{d=1}^D \omega_{d-1} \sum_{r=1}^M h_r \sum_{j=1}^{r-1} \binom{d-1}{j} (1 - x)^{d-1-j} x^j. \tag{5}$$

2.2. Weighted BATS Codes

Weighted BATS codes change the probability of being selected for encoding by setting different weight factors for input packets of different importance. We assume that there are K input packets and they are divided into m important levels. The j th level of input packets is denoted by IP_j . The number of packets with the j th important level is $\alpha_j K$ and $\sum_{j=1}^m \alpha_j = 1$. The probability of j th level packets being selected is $p_j = T_j / K$ where T_j is the weight factor of the j th important packets. The value of T_j has to satisfy $\sum_{j=1}^m \alpha_j T_j = 1$ and $p_i > p_j (i < j)$. Figure 1 illustrates the encoding process of weighted BATS codes at the source node.

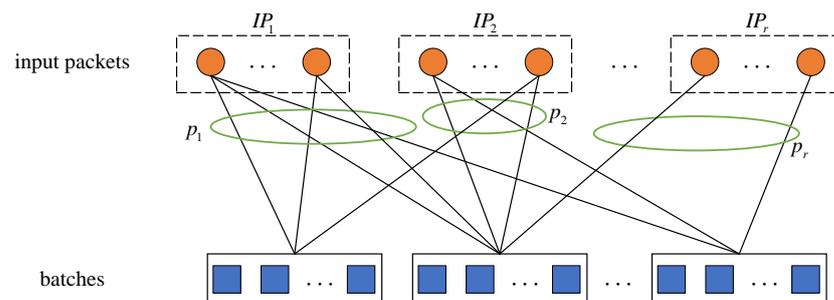


Figure 1. The encoding process of weighted BATS codes.

According to the analysis of weighted BATS codes in [12], the probability of a j th important packet undecodable after l iterations of BP is

$$\begin{cases} y_{0,j} = 1, \\ y_{l,j} = \delta^{(j)}(1 - \omega(1 - \sum_{u=1}^m \alpha_u T_u y_{l-1,u})) \\ = \exp[-\theta_j \omega(1 - \sum_{u=1}^m \alpha_u T_u y_{l-1,u})], \end{cases} \tag{6}$$

where $\theta_j = \frac{(1+\gamma)\mu T_j}{M}$ is the average degree of the input packets with j th-level importance.

3. Weighted BATS Code with LDPC Precoding

3.1. Encoding Scheme

In the conventional weighted BATS coding scheme, the input packets are randomly selected with different probabilities when encoding. Therefore, some input packets are rarely selected or even never selected, which would lead to high error floor at the receiving side. In order to lower the packet error rate, we propose a novel UEP-BATS code, which performs PR-LDPC codes precoding before the weighted BATS code, and precoding phase can be utilized to mitigate errors and recover input packets that cannot be restored by weighted BATS codes. In our scheme, the encoding process of the source node is divided into a precoding phase and a weighted BATS code encoding phase. The encoding process of these two phases is as follows:

Precoding phase: The PR-LDPC code [15] with UEP characteristics is used for precoding in our coding scheme. We assume that the number of the input packets is K , and N intermediate packets are generated by precoding. Therefore, the number of parity packets (PP) is $N - K$, and the code rate of precoding is $R = K/N$. Let the precoding check matrix be $\mathbf{H} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \cdots \ \mathbf{H}_m | \mathbf{H}_P]$, where \mathbf{H}_j ($1 \leq j \leq m$) is the sub-matrix corresponding to the IP_j , \mathbf{H}_P is the sub-matrix corresponding to PP and \mathbf{H}_P is a non-singular matrix. Let d_j be the degree of the IP_j , d_P be the degree of PP, and d_C be the degree of check nodes. So the column weight of \mathbf{H}_j is d_j , the column weight of \mathbf{H}_P is d_P , and the row weight of \mathbf{H} is d_C . According to the analysis in [15], the decoding error probability of the IP_i is lower than that of the IP_j when $d_i > d_j$. Since PP are not included in the packets to be recovered, $d_1 > d_2 > \cdots > d_m > d_P$ should be satisfied when d_C is determined. The input packets are denoted by $\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2 \ \cdots \ \mathbf{B}_m]$, where \mathbf{B}_j denotes the sub-matrix formed by the IP_j , so the dimension of \mathbf{B}_j is $L \times \alpha_j K$. The intermediate packets are denoted by $\mathbf{B}' = [\mathbf{B} | \mathbf{B}_P]$, where \mathbf{B}_P is the sub-matrix formed by PP, so the dimension of \mathbf{B}_P is $L \times (N - K)$. According to $\mathbf{H} \cdot (\mathbf{B}')^T = \mathbf{0}$, we can obtain

$$\mathbf{H}_1 \cdot \mathbf{B}_1^T + \mathbf{H}_2 \cdot \mathbf{B}_2^T + \cdots + \mathbf{H}_m \cdot \mathbf{B}_m^T + \mathbf{H}_P \cdot \mathbf{B}_P^T = \mathbf{0}. \tag{7}$$

The precoding process is performed on the finite field GF(2), so the PP generated by precoding is

$$\mathbf{B}_P = (\mathbf{B}_1 \cdot \mathbf{H}_1^T + \mathbf{B}_2 \cdot \mathbf{H}_2^T + \cdots + \mathbf{B}_m \cdot \mathbf{H}_m^T) \cdot (\mathbf{H}_P^T)^{-1} \tag{8}$$

Weighted BATS phase: Weighted BATS coding is performed on the intermediate packets in this phase. The intermediate packets are divided into m levels, where the i th level of intermediate packets are consistent with the i th level of input packets, $1 \leq i \leq m - 1$, and the m th level of intermediate packets includes the m th level of input packets and the PP. Therefore, the proportion of intermediate packets of i th level is $\alpha'_i = \alpha_i K/N = \alpha_i R$, and the proportion of intermediate packets of m th level is $\alpha'_m = (\alpha_m K + N - K)/N = \alpha_m R + (1 - R)$. The weight factor set for the j th level of intermediate packets is T_j where $1 \leq j \leq m$. Therefore, when weighted BATS coding is performed on the intermediate packet, the probability that the j th level of the intermediate packet is selected is $p_j = T_j/N$.

Figure 2 illustrates the encoding procedure of weighted BATS codes based on PR-LDPC precoding at the source node with $m = 2$ as an example. Input packets can be classified as MIP and LIP. Intermediate packets can be classified as MIP, LIP, and PP, or can be further classified as IMIP (Intermediate more important packets) and ILIP (Intermediate less important packets).

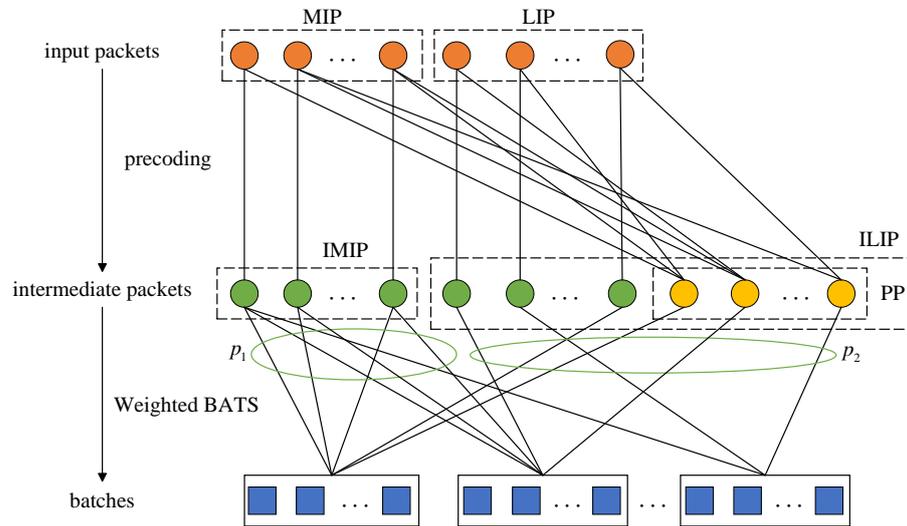


Figure 2. The encoding process of weighted BATS code with precoding.

3.2. Decoding Scheme

At the destination node, The receiver uses the BP algorithm for decoding and the bipartite graph of the BP algorithm is shown in Figure 3. The decoder of receiver consists of a weighted BATS code decoder and a precoding decoder. The specific decoding processes of these two decoders are as follows:

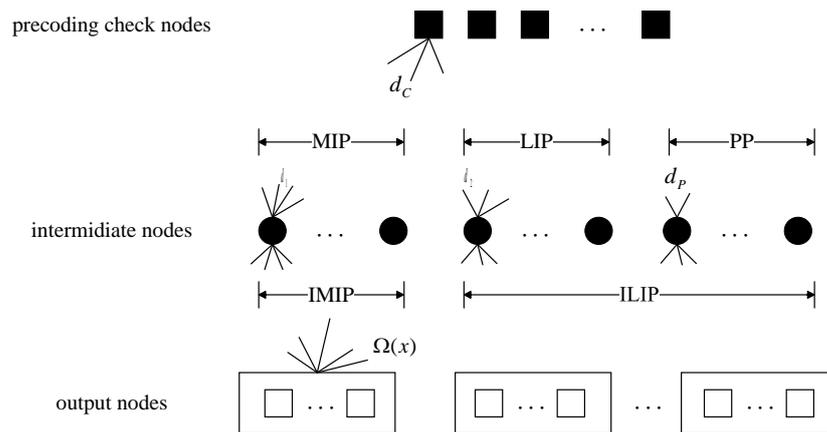


Figure 3. Bipartite graph of the weighted BATS code with PR-LDPC precoding.

Weighted BATS decoder: Similar with the BP algorithm in the conventional BATS codes, each output node in Figure 3 corresponds to a batch of weighted BATS codes. Due to the packet loss in the transmission, the number of receiving packets in the i th batch $M'_i \leq M$. The rank of the i th batch is $r_i = \text{rank}(\mathbf{G}_i \mathbf{A}_i)$. Let d_i and S_i denote the degree and the input packets set of i th batch, respectively. If $r_i = d_i$, S_i can be decoded by Gaussian elimination and delete all edges connected to the intermediate nodes corresponding to S_i in the bipartite graph. If the k th ($k \neq i$) batch contains intermediate node $s_i (s_i \in S_i)$,

remove the row corresponding to s_i in \mathbf{G}_k . Repeat the process until no batch satisfies that the degree is equal to the rank.

Precoding decoder: In each iteration, the precoding check nodes with degree of 1 are decodable. If the degree of j th check node is 1 and it is connected only to intermediate node s_j , the input packet corresponding to s_j is decodable. Then all edges connected to s_j in the graph are deleted. If s_j is selected to encode the k th batch, the row corresponding to s_j in \mathbf{G}_k is removed. The process is repeated until there is no check node with degree of 1.

The decoding process is shown in Figure 4, where U and V represent the index of the received packets and recovered packets, respectively. The receiver side performs the weighted BATS decoder and the precoding decoder in a cyclic manner. At the beginning of each cycle, the sets I_B and I_P are initialized as empty sets. Subsequently, the weighted BATS code decoding and the precoding decoding are executed. The index of the intermediate data packets decoded by weighted BATS decoder and precoding decoder are added to the sets I_B and I_P , respectively. The intermediate packet indexes in sets I_B and I_P , but not belonging to PP, are inserted into set V . The receiver side restarts weighted BATS decoding because there may still exist some decodable batches after parts of the intermediate data packets are decoded by precoding decoder. The decoding process is stopped when set I_B is empty.

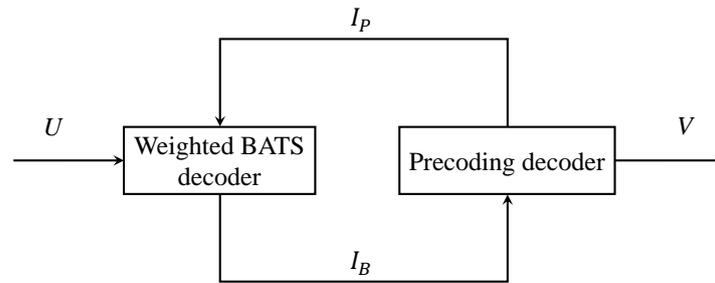


Figure 4. The decoding process of the of the weighted BATS code with LDPC precoding.

4. Performance Analysis and Parameter Optimization

4.1. Decoding Error Probability

In this subsection, we use the AND-OR tree analysis to derive the DE formulas of the weighted BATS code with LDPC precoding under BP decoding on BEC according to the conclusions in [15,17]. During the BP decoding process, erasure information is repeatedly passed between precoding check nodes, intermediate nodes, and output nodes. Each node updates its information upon receiving information from other nodes and passes the updated information to other nodes. Let $\bar{\Lambda}^{(j)}(x) = \sum_i \bar{\Lambda}_i^{(j)} x^i$ denotes the input degree distributions of j th intermediate nodes, and $\bar{\delta}^{(j)}(x) = \sum_i \bar{\delta}_i^{(j)} x^i$ represents the degree distribution of edges connected to j th intermediate nodes. So we can obtain $\bar{\delta}^{(j)}(x) = \bar{\Lambda}^{(j)'}(x) / \bar{\Lambda}^{(j)'}(1)$. Let E be the set of edges between the intermediate nodes and the precoding check nodes. Let λ_j and λ_P respectively denote the probability of selecting an edge from E at random, which corresponds to the j th level input packets and PP. So we have

$$\lambda_j = \frac{\alpha_j K d_j}{|E|}, \tag{9a}$$

$$\lambda_P = \frac{(1 - R) N d_P}{|E|}, \tag{9b}$$

where $|E| = \sum_{j=1}^m \alpha_j K d_j + (1 - R) N d_P$, indicating the number of edges between the intermediate nodes and the precoding check nodes.

At i th BP iteration, let $P_{j,i}$ and $P_{P,i}$ respectively denote the probability of passing erasure information to the precoding check nodes from the intermediate nodes corresponding

to IP_j and PP; $Q_{j,i}$ and $Q_{P,i}$ respectively denote the probability of passing erasure information to the output nodes from the corresponding intermediate nodes to IP_j and PP; U_i denotes the probability of passing erasure information to the intermediate nodes from the precoding check nodes; V_i denotes the probability of transmitting erasure information to the intermediate nodes from the output nodes. Subsequently, the respective mathematical expressions for the aforementioned probabilities will be presented.

When $i = 0$, all intermediate nodes are initialized to 1, which means that $P_{j,0} = P_{P,0} = 1$. When $i \geq 1$, $P_{j,i}$ is equivalent to the probability that the intermediate nodes corresponding to IP_j receive erasure messages from the other $d_j - 1$ edges connected to the precoding check nodes and all edges connected to the output nodes. The former probability is $(U_{i-1})^{d_j-1}$, and the latter probability is $\sum_i \bar{\Lambda}_i^{(j)} (V_{i-1})^i = \bar{\Lambda}^{(j)} (V_{i-1})$. Therefore, the formula for calculating $P_{j,i}$ is

$$P_{j,i} = (U_{i-1})^{d_j-1} \bar{\Lambda}^{(j)} (V_{i-1}). \tag{10}$$

Similarly, the formula for calculating $P_{P,i}$ is

$$P_{P,i} = (U_{i-1})^{d_P-1} \bar{\Lambda}^{(m)} (V_{i-1}). \tag{11}$$

When $i = 0$, all intermediate nodes are initialized to 1, which means that $Q_{j,0} = Q_{P,0} = 1$. When $i \geq 1$, $Q_{j,i}$ is equivalent to the probability that the intermediate nodes corresponding to IP_j receives the erasure messages from other output nodes and all precoding check nodes. The former probability is $(U_{i-1})^{d_j}$, and the latter probability is equivalent to the probability that all children of the intermediate node in the AND-OR tree pass the erasure messages to it. So we have

$$Q_{j,i} = (U_{i-1})^{d_j} \bar{\delta}^{(j)} (V_{i-1}). \tag{12}$$

Similarly, we can calculate $Q_{P,i}$ by

$$Q_{P,i} = (U_{i-1})^{d_P} \bar{\delta}^{(m)} (V_{i-1}). \tag{13}$$

The probability that the precoding check node passes on the erasure message to the intermediate node is denoted as U_i , which is equivalent to the probability that the check node receives the erasure messages from other $d_C - 1$ intermediate nodes, so

$$U_i = 1 - (1 - \sum_{j=1}^m \lambda_j P_{j,i} - \lambda_P P_{P,i})^{d_C-1}. \tag{14}$$

The probability that the output node passes on the erasure message to the intermediate node is denoted as V_i

$$V_i = 1 - \omega(1 - y), \tag{15}$$

where y denotes the probability that a child node of the AND node in the AND-OR tree transmits an erasure message, $\omega(1 - y)$ denotes the probability that less than $r - 1$ child nodes transmit erasure messages to the AND node, and r is the rank of the coding coefficient matrix corresponding to the AND node, so

$$y = \sum_{j=1}^m \alpha_j R T_j Q_{j,i} + (1 - R) T_m Q_{P,i}, \tag{16}$$

and

$$\omega(1 - y) = \mu \sum_{d=1}^D \omega_{d-1} \sum_{r=1}^M h_r \sum_{j=1}^{r-1} \binom{d-1}{j} (1 - y)^{d-1-j} y^j. \tag{17}$$

The above is the calculation process for the probability of erasure message transmitted between nodes. Based on these probabilities, we can make an asymptotic estimate of the decoding error probabilities of IP_j which is given by:

$$P_i^{(j)} = \frac{d_j P_{j,i} + \bar{\theta}_j Q_{j,i}}{d_j + \bar{\theta}_j}, \tag{18}$$

where $\bar{\theta}_j = n\mu p_j$ is the average degree of the intermediate packets with j th level importance.

4.2. Complexity

In the conventional weighted BATS code, the M encoded packets in the batch is generated by linearly combining μ input packets on average. Therefore, the encoding complexity of generating n batches in the conventional weighted BATS code is $O(n\mu LM)$. The encoding complexity is slightly elevated in our scheme due to the addition of the precoding phase. During the precoding phase, the encoding complexity is related to the row weight d_C of the parity matrix and the number of parity packets. A packet XOR operation must be performed during encoding for each element in the parity matrix with a value of "1". Therefore, the encoding complexity in the precoding phase is $O(Ld_C(N - K))$. The additional complexity introduced in the precoding phase is constant and independent of the number of batches generated.

At the destination node, the decoding process is divided into two stages, where the average complexity of the weighted BATS decoder is $O(nM^3 + n\mu LM)$, and the complexity of the precoding decoder is $O(Ld_C(N - K))$. Therefore, the decoding complexity of this scheme is $O(nM^3 + n\mu LM + Ld_C(N - K))$.

In summary, compared to conventional weighted BATS codes, our scheme only add a marginal and fixed increase in the complexity of the encoding process and decoding process.

4.3. Optimization of Precoding Degree Distribution

This subsection proposes an optimization algorithm for the precoding degree distribution under the constraint of limited precoding complexity. Based on the complexity analysis presented in the previous subsection, it can be concluded that, under the assumption of data packet length L and precoding rate R , the complexity of the precoding phase is solely determined by the row weight d_C of the parity check matrix \mathbf{H} . Therefore, the limitation of complexity in the precoding phase refers to its complexity being constrained by the maximum value of d_C . In other words, the maximum allowable value of d_C restricts the complexity of this phase. So we need to find the optimal values of d_j and d_p in the case where d_C is fixed.

According to the fact that the sum of the number of "1" elements in all columns of the parity check matrix \mathbf{H} is equal to the sum of the number of "1" elements in all rows, we can obtain

$$\sum_{j=1}^m \alpha_j R d_j + (1 - R) d_p = (1 - R) d_C. \tag{19}$$

Suppose that d_j and d_p have lower bounds of 1, and D_j and D_p denote the maximal values of d_j and d_p , respectively. Due to $d_1 > d_2 > \dots > d_m > d_p$, we sequentially determine d_1, d_2, \dots, d_m and d_p . According to (19), D_j and D_p can be calculated using the following methodology.

- (1) When $j = 1$, it can be inferred that the variable d_1 reaches its maximum value under the circumstance of $d_2 = d_3 = \dots = d_m = d_p = 1$, so

$$D_1 = \frac{(1 - R) d_C - \sum_{i=2}^r \alpha_i R - (1 - R)}{\alpha_j R}. \tag{20}$$

- (2) When $2 \leq j \leq m$, it can be inferred that the variable d_j reaches its maximum value under the circumstance of $d_{j+1} = d_{j+2} = \dots = d_m = d_p = 1$, so

$$D_j = \min \left\{ d_{j-1} - 1, \frac{(1 - R)d_C - \sum_{i=1}^{j-1} \alpha_i R d_i - \sum_{i=j+1}^m \alpha_i R - (1 - R)}{\alpha_j R} \right\}. \tag{21}$$

(3) When d_j is completely determined, the maximum possible value of d_p is

$$D_p = \min \left\{ d_m - 1, \frac{(1 - R)d_C - \sum_{i=1}^m \alpha_i R d_i - (1 - R)}{1 - R} \right\}. \tag{22}$$

Given the aforementioned expression, upon specifying the value of d_C , one can determine the values of D_1, D_2, \dots, D_m and D_p in a sequential manner. Thereafter, the search process can be employed to obtain the optimal precoding degree distribution, as presented in Algorithm 1, where the symbol “ \leftarrow ” signifies assignment.

Algorithm 1 Precoding degree distribution optimization algorithm

Input: $K, \alpha_1, \dots, \alpha_m, R, \Omega(x), d_C$

Output: d_1, \dots, d_m, d_p

- 1: $P_1 \leftarrow 1, P_2 \leftarrow 1, \dots, P_m \leftarrow 1;$
 - 2: Calculate D_1 based on (20);
 - 3: **for** $d_{(1)} = 1$ to D_1 **do**
 - 4: Calculate D_2 based on (21);
 - 5: **for** $d_{(2)} = 1$ to D_2 **do**
 - 6: \vdots
 - 7: Calculate D_m based on (21);
 - 8: **for** $d_{(m)} = 1$ to D_m **do**
 - 9: Calculate D_p based on (22);
 - 10: **for** $d_p = 1$ to D_p **do**
 - 11: Calculate the decoding error probability $P^{(j)}$ for the IP_j based on (18);
 - 12: **if** $P^{(1)} < P_1 \ \&\& \ \dots \ \&\& \ P^{(m)} < P_m \ \&\& \ P^{(p)} < P_p$ **then**
 - 13: $P_1 \leftarrow P^{(1)}, \dots, P_m \leftarrow P^{(m)};$
 - 14: $d_1 \leftarrow d_{(1)}, \dots, d_m \leftarrow d_{(m)}, d_p \leftarrow d_p;$
 - 15: **end if**
 - 16: **end for**
 - 17: **end for**
 - 18: \vdots
 - 19: **end for**
 - 20: **end for**
 - 21: **return** d_1, \dots, d_m, d_p
-

5. Numerical Results

In this section, we present some simulation results to illustrate the performance of our proposed coding scheme. As is shown in Figure 5, we consider the two-hop line-erasure network with one relay node r between the source s and destination t where the channel erasure probability of each hop is 0.2.

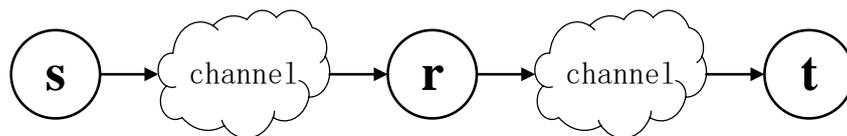


Figure 5. A two-hop line-erasure network with packet erasure probability 0.2 on each link.

Unless explicitly stated, we use the following parameters: the number of source packets is $K = 10,000$ where MIP occupies 0.1, the weight factor of MIP is $T_1 = 2$, and the

size of a batch is $M = 16$. The channel rank distribution and degree distribution of the BATS code are shown in Table 1 and Table 2 [3], respectively.

Table 1. The rank distribution of the two-hop erasure network when channel erasure probability is 0.2.

r	1	2	3	4	5	6	7	8
h_r	0	0	0	0	0.0001	0.0004	0.0025	0.0110
r	9	10	11	12	13	14	15	16
h_r	0.0387	0.1040	0.2062	0.2797	0.2338	0.1038	0.0190	0.0008

Table 2. The BATS codes degree distribution when $M = 16$.

d	14	15	20	21	28	38	39
Ω_d	0.0478	0.2665	0.1012	0.0977	0.1411	0.0899	0.0122
d	51	52	73	74	111	113	199
Ω_d	0.0034	0.0734	0.0579	0.0061	0.0251	0.0286	0.0491

We set $\gamma = 0.45$ and $R = 0.95$. Using Algorithm 1, we find that when $d_C = 100$, the optimal degree distributions of precoding are $d_1 = 23$, $d_2 = 3$, $d_p = 2$, while for $d_C = 80$, they are $d_1 = 13$, $d_2 = 3$, $d_p = 2$. We randomly generate the precoding check matrix $\mathbf{H} = [\mathbf{H}_1 \quad \mathbf{H}_2 | \mathbf{H}_p]$ according to the degree distribution, where \mathbf{H}_p is a non-singular matrix.

Figure 6 shows the packet error rate of the weighted BATS code based on PR-LDPC precoding at different precoding rates. From the simulation results, it can be observed that when the coding redundancy γ is small ($\gamma < 0.45$), the packet error rate of using precoding at different rates is very close. Because with a high-rate precoding, the receiver mainly relies on the weighted BATS decoder to recover the input packets, while the precoding decoder almost does not recover input packets. Therefore, the performance of using different precoding rates is similar to that of the conventional weighted BATS code in this case. When the coding redundancy is small, the packet error rate slightly increases as the precoding rate decreases. Because under the condition of constant coding redundancy, lower precoding rates introduce more redundancy in the precoding stage and less redundancy in the weighted BATS code stage. Since the decoding error probability mainly depends on the performance of the weighted BATS code when the coding redundancy is small, the lower precoding rates result in higher overall packet error rate. However, as the coding redundancy γ increases and the receiver recovers enough input packets, the number of input packets recovered in the precoding decoding stage increases. Moreover, the lower the precoding rate, the more input packets are recovered in the precoding decoder. Thus, the packet error rate decreases faster, and the error floor is lower.

Figure 7 demonstrates the performance of weighted BATS codes with PR-LDPC precoding using various degree distributions in terms of packet error rate. The simulation results indicate that when the encoding redundancy is small, the packet error rate is nearly identical across different precoding degree distributions. This can be attributed to the fact that, at this stage, the decoding of the weighted BATS codes is primarily responsible for recovering input packets. However, as the encoding redundancy increases, the packet error rate is highest when utilizing regular LDPC codes as precoding. This is due to the higher degree of LIP in regular LDPC codes with the same check node degree, and the higher error rate of LIP. As a result, in BP decoding, the probability of deleting edges connected to check nodes with LIP is lower, resulting in a lower probability of check nodes with degree 1. Consequently, the packet error rate is higher when employing regular LDPC codes as precoding.

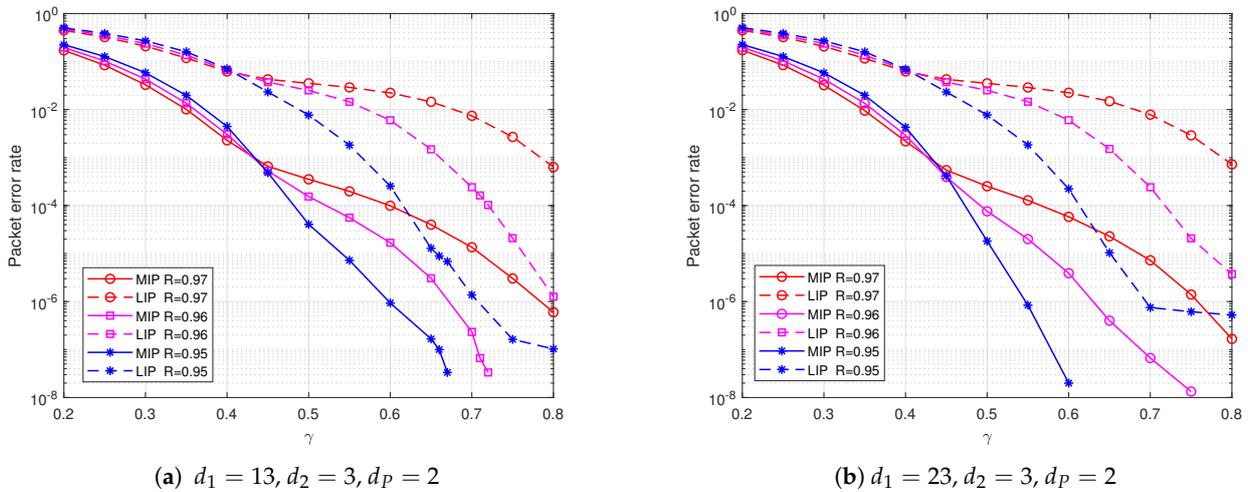


Figure 6. The packet error probability of weighted BATS codes based on PR-LDPC precoding at different precoding code rates.

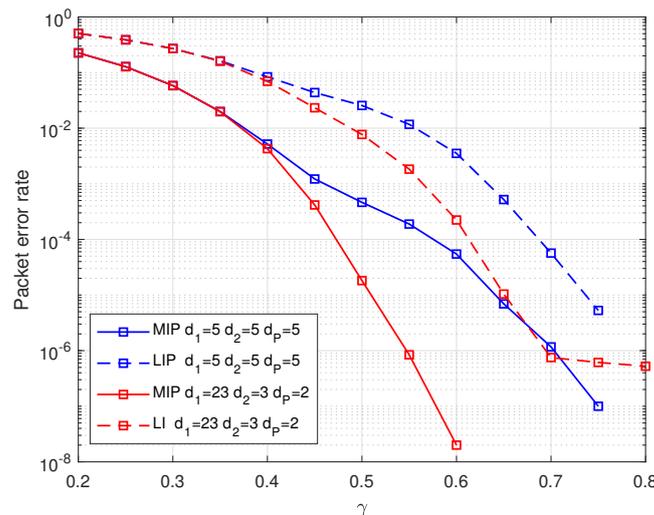


Figure 7. The packet error rate of weighted BATS codes based on PR-LDPC precoding with different precoding degree distributions.

Given the precoding degree distribution and γ , we can optimize T_M with the DE analysis in Sec. III B. Figure 8 shows the performance of MIP and LIP with different T_1 when $\gamma = 0.48$ and precoding degree distribution is $d_1 = 23, d_2 = 3, d_p = 2$. According to the asymptotic results, the error probability of MIP is minimized when $T_1 = 1.9$. The simulation result in Figure 9 shows the packet error rate of weighted BATS codes and our proposed scheme when $T_1 = 1.9$ and $T_1 = 2$. As is shown in Figure 9, when the coding redundancy is small, the error rate of our proposed scheme is slightly higher than that of the weighted BATS codes. That is because precoding can hardly lower the packet error rate but increases the overall encoding redundancy. As the coding redundancy increases, the conventional weighted BATS codes have high error floor. The packet error rate of our proposed scheme drops sharply, which is significantly lower than that of the conventional weighted BATS codes. Meanwhile, in our scheme, the packet error rate when $T_1 = 1.9$ is lower than that when $T_1 = 2$, which verifies the optimization results above.

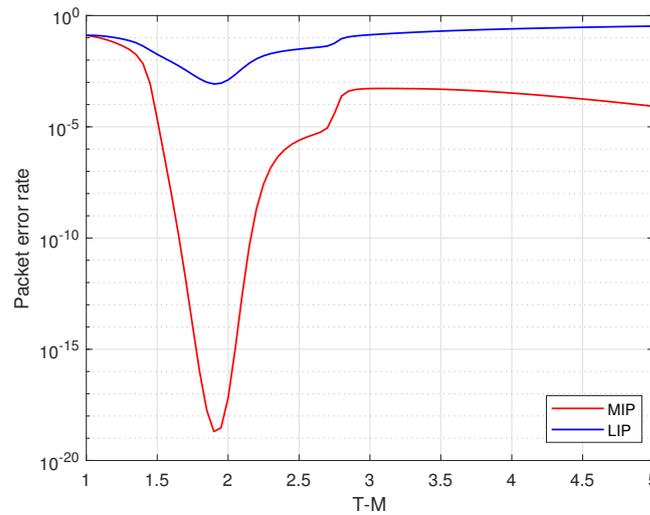


Figure 8. The packet error rate of proposed scheme with different T_M when $\gamma = 0.48$.

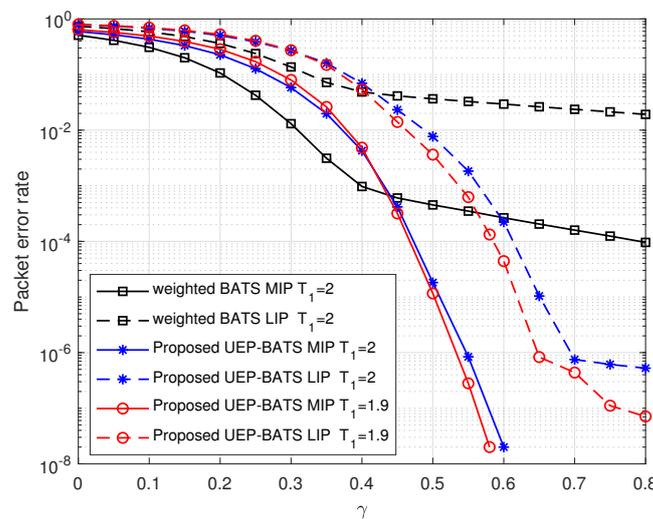


Figure 9. The packet error rate comparison of the proposed coding scheme and conventional weighted BATS codes with different encoding redundancy γ when $R = 0.95$.

6. Conclusions

In this paper, we propose a novel UEP-BATS coding scheme which performs UEP-LDPC precoding before the weighted BATS code. Compared with conventional weighted BATS codes, our proposed coding scheme has lower packet error rate. We analyze the situation where the input packets have two important levels in detail and derive the DE formulas of the proposed coding scheme decoded using the BP algorithm on BEC. We also optimize precoding degree distribution and weight factor of weighted BATS with the help of DE analysis. Finally, simulation results show that compared with the conventional weighted BATS code, our coding scheme has lower packet error rate with the increase of encoding redundancy.

Author Contributions: Methodology, W.Z. and M.Z.; Validation, M.Z.; Formal analysis, W.Z.; Writing—original draft, W.Z.; Writing—review & editing, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by National Natural Science Foundation of China under Grant No. 62271380.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Dana, A.F.; Gowaikar, R.; Palanki, R.; Hassibi, B.; Effros, M. Capacity of wireless erasure networks. *IEEE Trans. Inf. Theory* **2006**, *52*, 789–804. [[CrossRef](#)]
2. Ho, T.; Médard, M.; Koetter, R.; Karger, D.R.; Effros, M.; Shi, J.; Leong, B. A random linear network coding approach to multicast. *IEEE Trans. Inf. Theory* **2006**, *52*, 4413–4430. [[CrossRef](#)]
3. Yang, S.; Yeung, R.W. Batched sparse codes. *IEEE Trans. Inf. Theory* **2014**, *60*, 5322–5346. [[CrossRef](#)]
4. Yang, S.; Ng, T.C.; Yeung, R.W. Finite-length analysis of BATS codes. *IEEE Trans. Inf. Theory* **2017**, *64*, 322–348. [[CrossRef](#)]
5. Zhou, Z.; Li, C.; Yang, S.; Guang, X. Practical Inner Codes for BATS Codes in Multi-Hop Wireless Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2751–2762. [[CrossRef](#)]
6. Yang, J.; Shi, Z.P.; Xiong, J.; Wang, C.X. An Improved BPss Decoding of BATS Codes With Iterated Incremental Gaussian Elimination. *IEEE Commun. Lett.* **2020**, *24*, 321–324. [[CrossRef](#)]
7. Yongning, Z.; Xiaoxue, Z.; Qin, W.; Zongzheng, Z.; Si, W.; Yun, L. Research on a Sliding Window BATS Code Method. In Proceedings of the 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 18–20 December 2020; pp. 404–409. [[CrossRef](#)]
8. Wang, S.; Liu, H.; Ma, Z.; Xiao, M. Precoded Batched Sparse Codes Transmission Based on Low-Density Parity-Check Codes. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–5.
9. Gao, Y.; Xu, X.; Guan, Y.L.; Chong, P.H.J. V2X content distribution based on batched network coding with distributed scheduling. *IEEE Access* **2018**, *6*, 59449–59461. [[CrossRef](#)]
10. Yeung, R.W.; Dong, G.; Zhu, J.; Li, H.; Yang, S.; Chen, C. Space communication and BATS codes: A marriage made in heaven. *J. Deep Space Explor.* **2018**, *5*, 129–139.
11. Wang, S.; Zhou, Q.; Yang, S.; Bai, C.; Liu, H. Wireless communication strategy with BATS codes for butterfly network. In Proceedings of the Journal of Physics: Conference Series, 2021 3rd International Conference on Computer, Communications and Mechatronics Engineering (CCME 2021), Online, 17–18 December 2021; IOP Publisher: Bristol, UK, 2021; Volume 2218, p. 012003.
12. Xu, X.; Zeng, Y.; Guan, Y.L.; Yuan, L. BATS code with unequal error protection. In Proceedings of the 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, China, 14–16 December 2016; pp. 1–6.
13. Xu, X.; Zeng, Y.; Guan, Y.L.; Yuan, L. Expanding-window BATS code for scalable video multicasting over erasure networks. *IEEE Trans. Multimed.* **2017**, *20*, 271–281. [[CrossRef](#)]
14. Xiang, M.; Yi, B.; Qiu, K.; Huang, T. Expanding-window BATS code with intermediate feedback. *IEEE Commun. Lett.* **2018**, *22*, 1750–1753. [[CrossRef](#)]
15. Rahnavard, N.; Pishro-Nik, H.; Fekri, F. Unequal Error Protection Using Partially Regular LDPC Codes. *IEEE Trans. Commun.* **2007**, *55*, 387–391. [[CrossRef](#)]
16. Yang, S.; Zhou, Q. Tree Analysis of BATS Codes. *IEEE Commun. Lett.* **2016**, *20*, 37–40. [[CrossRef](#)]
17. Yuan, L.; An, J. Design of UEP-Raptor codes over BEC. *Eur. Trans. Telecommun.* **2012**, *21*, 30–34. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.