# Delay-Limited Rate-Compatible Protograph LDPC Codes

Thuy V. Nguyen, Hung N. Dang and Hieu T. Nguyen

***Abstract*—This paper produces a family of rate-compatible protograph-based low-density parity-check (PLDPC) codes that have superior performance under strict constraints of short block length and low decoding iterations. The design of such practical codes is a challenging task since the constraints imposed by structured designs (PLDPC), rate-compatibility, small block length, as well as a small number of decoding iterations, are hard to meet simultaneously. As the block length and the number of decoding iterations decrease, the typical LDPC design based only on the coding threshold is no longer effective due to imperfection in the modeling of LDPC decoders in the short block length regime. We propose a code design method that takes the code simulations and the number of the decoding iterations as inputs to optimize the new codes. Analytical and simulation results confirm that the new codes produced by the proposed approach outperform the state-of-the-art codes in a wide range of code rates. None of the newly optimized codes has the error-floor behavior even below the frame error rate of $10^{-5}$ or bit error rate of $10^{-6}$.**

## I. INTRODUCTION

### A. Motivations

In most of today's communication and storage systems, for example mobile 5G networks and flash memory devices, there are heterogeneous demands on latency, throughput, and reliability [2]. The varied demands on reliability can be directly translated to the diverse requirements on frame error rate (FER) and bit error rate (BER) performance at the physical layer [3], [4]–[9]. To meet the diverse requirement on FER/BER performance, a practical solution is to use different channel coding and modulation configurations accordingly to channel scenarios. This technique is often referred to as adaptive coded modulation (ACM) [8], [10]–[13].

The ACM architecture is implemented by designing either a set of separate channel codes or a family of punctured codes. The first choice can, nevertheless, be impractical when the low hardware complexity is a crucial design parameter since a pair of encoder and decoder is needed for a particular code rate. The second choice of using the punctured code can probably cause either the degradation in performance [14] or the slow iterative decoding convergence. To tackle the disadvantages of the two aforementioned approaches, we use a rate-compatible framework to design a family of nested protograph based LDPC codes (PLDPC) that facilitates only one pair of encoder/decoder structure and the same information block length for all designed code rates.

When designing a PLDPC code, there are two fundamental issues that need to take into account: 1) good performance in both the water-fall region and the error-floor region; 2) low decoding complexity. In previous works, most of protograph LDPC designs focused on finding the codes whose iterative decoding thresholds are as close to the capacity limits as possible based on either density evolution [15] or extrinsic information transfer (EXIT) chart [16]. This led to the fact that the reported codes usually require a large number of decoding iterations [17], [18] in order to provide satisfactory performance. As a result, the iterative decoders for such protograph codes are often complex since the complexity level of an iterative decoder is proportional to the number of maximum decoding iterations [19].

In addition to the complexity, using a large number of iterations can result in long processing time for the receiver, one of the reasons causing the overall long delay for communication systems. The long processing delay makes the previously designed protograph-based LDPC codes incompatible to some modern communication systems, e.g., the next generation 5G networks in which ultra-low latency is a key requirement when delivering information from one point to another [9], [20]. For those reasons, the applications of the previous optimized protograph codes are limited in many practical scenarios where low hardware complexity and delay are critical design requirements.

The contribution of the paper is to produce a family of rate-compatible PLDPC codes, which has superior performance in short block length and a small number of iteration regime, providing low delay and low complexity system. The designed codes are nested in a rate-compatible family of multiple rates within one common hardware platform and able to provide ultra-low delay performance. Before presenting our design approach in more detail, the literature review in the protograph design related to our works in the section below.

### B. Previous Works on Protograph Codes

Protograph codes [9], [17] have been an active topic in recent years as the practical coding method for next-generation communication systems. To design a good P-LDPC code, ones usually focused on optimizing iterative decoding threshold, resulting in many capacity-approaching codes [17]. However, those codes often required a large number of decoding iterations and a very large block length to provide satisfactory

Thuy V. Nguyen, Hung N. Dang is with Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam. Email: thuynv@ptit.edu.vn, hungnd@ptit.edu.vn

Hieu T. Nguyen is with Dept. of Science and Industry Systems, Faculty of Technology, Natural Sciences and Maritime Sciences, University of Southeast Norway, NO 3603 Kongsberg, Norway. Email: hieu.nguyen@usn.no

This work has been partially presented at Int. Conf. Advanced Technologies for Communications conference 2016 [1]

performance. These lead to high latency and power consumption in the receiver.

For finite-length codes, the designs are not that simple since the decoder model is not valid. There were many designs attempted to address this problem, for example, finite geometry (FG) [21], nonbinary (NB) codes [22], and generalized LDPC (GLDPC) codes [23]. Several protograph designs were proposed for finite-length protograph codes. However, these codes can be improved further to deal with issues caused by short block length.

The key contributions of our papers are summarized below: We propose a design framework to produce a good protograph code in short-to-medium length. The proposed codes have superior performance when the number of iterative decoding iterations is small. Based on the design framework, we then produce a family of nested codes that can provide multiple rates within the same encoding/decoding architecture. Since the proposed codes are designed to work best with short-to-medium length and small decoding iterations, the coding family is suitable for applications with ultra-low delay requirements.

Some of new research works on protograph codes have been recently reported in [3], [14], [24]. The newest rate-compatible protograph LDPC code was optimized for multi-level-cell flash memory for the first time [3]. The authors utilized the unbalance in raw BER over different program/erase cycles and different types of bits stored in a memory cell to assign the degree across the variable nodes in the protograph. In that work, a family of the rate-compatible protograph codes with code rates ranging from $0.5$ to $0.93$ were found by extending the number of variable nodes while keeping the number of the check nodes unchanged. The proposed codes have superior error rate performance and fast decoding convergence in comparison with irregular LDPC codes. Tang et al. [24] introduced a design method to find good protograph codes where both the number of nonzero elements in the protograph matrices and the number of decoding iterations were strictly limited. This work only reported punctured protograph codes, and none of the rate-compatible codes was found. Uchikawa [14] used the lengthening technique from [18] to search for a family of non-punctured codes whose code rates are expressed by the formula $R = (n+1)/(n+2)$, for $n = 1, 2, \cdots, 8$. All codes were reported to possess good performance with a small number of decoding iterations. Nevertheless, a limited number of code rates were reported.

Motivated by the remarkable progress in protograph code design and strong demands for low complexity and low delay services in many new communication systems, we propose a framework to design good rate-compatible protograph codes taking account to such strict constraints. The objective is to produce a new family of codes that performs well with the small number of decoding iterations at both water-fall and error-floor regions.

The remainder of the paper is organized as follows. Section II introduces the concept of a protograph code and its iterative decoding threshold behavior in two types of protograph codes: punctured and non-punctured codes for a specific code rate of $1/2$. Section III presents the framework to design daughter codes and a family of rate-compatible protograph codes. Both analytical and simulation results are included to confirm the advantage of the design approach. Finally, Section IV concludes the paper.

## II. Proposed Protograph Code Design

In this section, a design framework is proposed to search for a good performing protograph which possesses a low iterative decoding threshold in AWGN channel with a predefined number of iterative decoding iterations. An iterative decoding threshold of a protograph is the minimum channel quality that supports reliable iterative decoding. This new iterative decoding threshold is computed based on PEXIT method [16] with the predefined number of iterations. In our method, we only focus on graphs that provide excellent coding properties, such as a low decoding threshold and the linear minimum distance growth property [17] that guarantees no error floor if random circulants are assigned when constructing the protograph code. In the following, we follow the design guideline described in [18], but with an additional constraint on the predetermined number of iterations.

In the literature, there are two types of protograph codes reported, i.e., punctured and non-punctured protograph code structures. The former was mainly based on works of Divsalar and others in several years which were summarized in a highly cited paper [17] after implementing many different protograph designs. Since the protograph has a simple structure, a high degree punctured node is needed to normalize connectivities in the graph which had positive impacts in iterative message passing decoder, yielding good performance [17], [18], [25]. However, these optimized protograph structures required many decoding iterations to produce satisfactory performance. In practical communications systems where the number of decoding iterations is limited, a high degree punctured node might not be a good design solution. Thus, un-punctured protograph was the other structure that was studied to address this practical design problem [14]. This is also our design topic presented in this paper.

In the following, let us revise these two graph structures, i.e., punctured and non-punctured protographs, and describe our proposed coding design solution for practical communications systems.

### A. Punctured Protograph

One of the most popular protograph designs was proposed by Divsalar et al. in [17]. They invented the family of accumulate-repeat-4-jagged-accumulate (AR4JA) codes, whose minimum distances grow linearly with the block length. The iterative decoding thresholds approach the Shannon limit on binary-input additive white Gaussian noise (BI-AWGN) channels. The structure includes a degree-one variable node connected with a highly connected punctured variable node. Extending this structure, Nguyen et al. [18] proposed a reference design to produce a good protograph code. The proposed code has the lowest iterative decoding threshold and linear minimum distance growth property, which facilitates excellent error performance. This design produced one of the best performing protograph codes so far in the literature [18], [26]. Since these codes were mostly optimized for iterative decoding thresholds, they required a large number of decoding iterations to have the best performance. An optimized rate-$1/2$ code in [18] has the following proto-matrix:

$$H_{Code1}^{1/2} = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 2 & 1 & 2 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 2 \end{pmatrix} \tag{1}$$

where the second column is punctured. This protograph has 7 variable and 4 check nodes, so-called $7 \times 4$ protograph structure. The threshold of this code in the AWGN channel is $0.395$ dB which shows a gap of $0.208$ dB to the capacity, which is one of the best-structured rate $1/2$ LDPC codes so far. This code has the linear minimum distance growth property that facilitates excellent error floor performance.

As reported in [17], [18], a good protograph code, which has the decoding threshold close to the capacity, would have a degree-1 variable node, a high-degree punctured node, and a fraction of degree-2 variable nodes. In their works, the code performance was reported with a large number of decoding iterations. That is not suitable for delay-constraint communications systems, which are the subject of our paper. In the following, let us design our first coding scheme with the constraints of a predefined number of decoding iterations.

First, we study the same structure, as used in [18], which has 7 variable nodes, and 4 check nodes. Using the same search space as given in [18] with the constraints of maximum 3 parallel edges and linear minimum distance growth property, we find the new code that is optimized to work with the maximum number of 25 iterations only. The new code has the following proto-matrix:

$$H_{Code3}^{1/2} = \begin{pmatrix} 1 & 3 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 3 & 2 & 2 & 1 & 1 \\ 0 & 3 & 0 & 1 & 0 & 2 & 0 \end{pmatrix} \tag{2}$$

where the second column is punctured. This protograph has 7 variable and 4 check nodes, the so-called $7 \times 4$ protograph structure. The protograph of Code 1 is plotted in Figure 1, in which the labeled number indicates the number of parallel edges. The threshold of the code with 25 decoding iterations is $1.048$ dB, which is $0.236$ dB better than that of Code 1 computed at the same number of iterations. This suggests that Code 1 is not suitable for time-sensitive applications that can only support a small number of decoding iterations. Since the code is optimized with 25 decoding iterations, the code is not performing well when the number of decoding iterations increases to hundreds. We will have a close look at this behavior in Figure 3 that will be described in subsection underneath.

### *B. Non-punctured Protograph*

These above capacity approaching protograph codes used punctured codes, typically requiring a large number of decoding iterations to have excellent performance. One of the reason

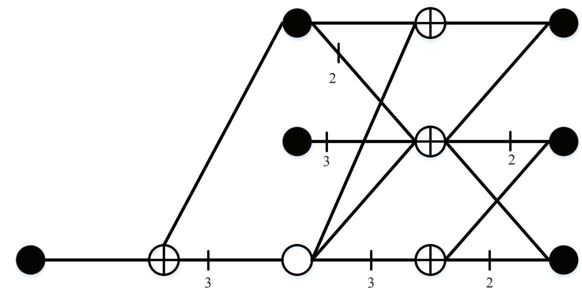


Fig. 1. The rate-$1/2$ punctured protograph (Code-3) optimized for 20 decoding iterations

might be punctured nodes since they are untransmitted and have the highest degree in the protograph. They might slow down the decoder convergence, resulting in a large number of decoding iterations needed to decode successfully. Keeping that argument in mind, Uchikawa [14] studied a design method of non-punctured protograph codes that have better performance in a small number of decoding iterations than the rate-$1/2$ NND code reported in [18]. The idea of the design proposed in [14] was using the non-punctured protograph structure, and was optimized based on decoding threshold only the same as in [17], [18]. The rate-$1/2$ protograph code reported in [14] has the following proto-matrix:

$$H_{Code2}^{1/2} = \begin{pmatrix} 3 & 3 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 1 & 0 & 1 & 0 & 0 \\ 3 & 2 & 1 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 2 & 2 & 2 & 1 \end{pmatrix} \tag{3}$$

where there are 8 variable nodes and 4 check nodes, so-called $8 \times 4$ protograph structure. This code has the decoding threshold of $0.501$ dB, which is higher than that of the NND code [18], in the case of a large number of decoding iterations. As reported in [14], this code had outstanding performance when the number of decoding iterations is limited. This phenomenon will be further studied in the subsection below.

In the following, we present our code design platform to take into account the delay/complexity constraints of a practical communications system. We present a design platform using the idea reported firstly in [18] and adding one more design dimension, i.e., the predefined number of decoding iterations. To illustrate our design, we use the non-punctured protograph structure as studied [14]. First, we represent the protograph by its $8 \times 4$ proto-matrix. The matrix contains 32 elements, each showing how many parallel edges connect the respective variable node (column) and check node (row). Second, we optimize the proto-matrix of over 32 variables which lead to a highly computationally complex. Therefore a search space reduction is needed to have a feasible solution. As discussed in [18], the good protograph should contain several degree-2 variable nodes and variable nodes with degrees 3 or higher. Thus, to reduce the search space further, we start by a search structure with two degree-2 and one degree-3 variable nodes in the following form

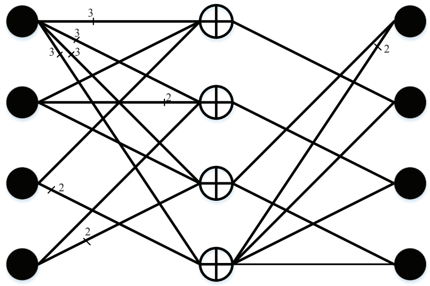


Fig. 2. The rate-$1/2$ non-punctured protograph (Code-4) optimized for 25 decoding iterations

$$H^{1/2}_{search} = \begin{pmatrix} x_1 & x_5 & x_9 & x_{13} & 0 & 1 & 0 & 0 \\ x_2 & x_6 & x_{10} & x_{14} & 0 & 0 & 1 & 0 \\ x_3 & x_7 & x_{11} & x_{15} & 1 & 0 & 0 & 1 \\ x_4 & x_8 & x_{12} & x_{16} & 2 & 1 & 1 & 1 \end{pmatrix} \quad (4)$$

where $x_i$, $i = 1, \ldots, 16$, are the number of edges connecting their associated variable nodes (column) and check nodes (row). To have the linear minimum distance growth property within the code structure, the edge summation over first 4 columns should be 3 or higher [27]. We can further simplify the problem by limiting $x_i \in \{1, 2, 3\}$ for $1 \leq i \leq 4$ as the one node that has the highest degree connecting to all check nodes, and $x_i \in \{0, 1, 2, 3\}$ for $5 \leq i \leq 16$.

In this specific example, our goal is to find a proto-matrix that has the lowest iterative decoding threshold with a maximum of 25 iterations. The reason behind the choice of 25 iterations is to guarantee that the message is fully propagated through all the nodes when we design rate-adaptive protograph codes using the lengthening method.

After a simple search, the resulting protograph, so-called Code-4, is in the form of

$$H^{1/2}_{Code4} = \begin{pmatrix} 3 & 3 & 1 & 0 & 0 & 1 & 0 & 0 \\ 3 & 2 & 0 & 1 & 0 & 0 & 1 & 0 \\ 3 & 1 & 0 & 2 & 1 & 0 & 0 & 1 \\ 3 & 0 & 2 & 0 & 2 & 1 & 1 & 1 \end{pmatrix} \quad (5)$$

The resulted graph has 8 variable nodes, and 4 check nodes, the so-called $8 \times 4$ protograph structure and rate of $1/2$. The protograph of Code-4 is plotted in Figure 2, in which the labeled number indicates the number of parallel edges. This code has the decoding threshold of 0.964 dB with 25 decoding iterations. The threshold is 0.06 dB lower than that of the Code-2 reported in [14] with the same protograph structure. In the following, the relationship between the decoding threshold and the number of decoding iterations is studied in detail to support our design approach.

### C. Number of Decoding Iterations vs. Decoding Threshold

In this subsection, the behavior of the decoding threshold of protograph codes for the number of decoding iterations is investigated. Decoding threshold reflects a limit at which an iterative message-passing decoder can decode codewords successfully if the codeword length is large enough. If the codes have minimum distance linear growth property [17], [18], the decoding threshold could be a good measure to evaluate the code performance in comparison with other protograph codes.

Figure 3 shows the behavior of the iterative decoding threshold of protograph codes with the number of iterations of 10, 25, 50, 100, 200, and 1000 respectively. The four protograph codes in the previous subsections are studied in detail. From the figure, Code 1, the solid blue line with a circle marker, has the largest threshold (resulting in the worse water-fall performance) with the number of decoding iterations 40 or lower. However, the code threshold is getting lower significantly as the iteration number increases. If the number of decoding iterations is large enough (100 and higher), this code has the smallest decoding threshold among 4 codes studied in this paper. This code is one of the best rate-$1/2$ LDPC codes reported in the literature in high decoding iteration regime.

Based on the decoding threshold behavior of Code 1 in Figure 3, we can predict that Code 1 has the worse error performance among 4 codes in the small number of iteration regime. However, in the high number of iteration regime, Code 1 provides the best performance among studied codes. This insight was also reported in [14] and confirmed in the numerical result section in this paper.

Moreover, the author of [14] reported Code 2 that has a good performance in both small and high number of iterations. The behavior of Code 2 is plotted in the solid red curve with square in Figure 3. Code-2 was optimized with decoding iteration threshold only, it has a better performance at the number of iterations small than that of Code 1, thanks to its non-punctured protograph structure. In the high number of decoding iteration regime, Code-2 has a higher decoding threshold than that of Code-1, resulting in a numerical performance worse than that of Code-1 as reported in [14]. The behavior of two reported codes study supported our analysis intuition that the decoding threshold is still a good measure to find the excellent protograph code in both small and large numbers of decoding iterations. However, in the small number of decoding iteration scenarios, the code design need to have a predefined number of decoding iterations that is the extra design dimension to search for a good protograph code, working best in this domain.

The behaviors of two new rate-$1/2$ codes in both punctured (Code 3) and non-punctured (Code 4) structure are plotted as pink and black curves in Figure 3. These codes outperform old codes in the same structure type. In the figure, Code 4 performs the best among 4 codes studied, proving that the non-punctured protograph structure is the most suitable coding structure of delay-limited communications systems. The next subsection describes the numerical performance of these codes that confirm our design insights described in this section.

### D. Numerical Results

To this point, we have only represented codes in the form of proto-matrices (or protographs). As discussed in Section I-B, a protograph code (or an equivalent LDPC code) is constructed by copy-and-permutation operation on a protograph, a process known as *protograph lifting*. In the following, we will report some numerical results to support the analytic results and insights as shown in Figure 3.

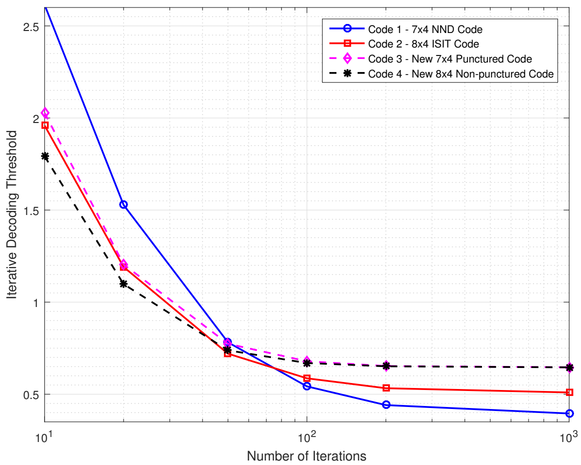


Fig. 3. Iterative decoding threshold of protograph codes vs. Number of Iterations

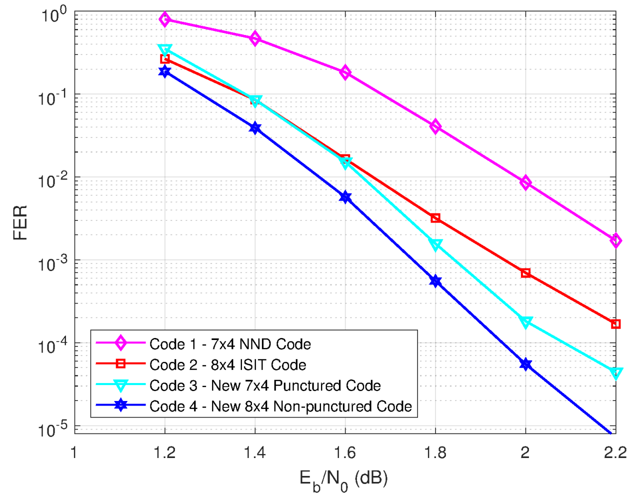


Fig. 4. FER performance comparison

Our protograph codes are derived from protographs in two lifting steps. First, the protograph is lifted by a factor of $4$ using the progressive edge growth (PEG) algorithm [28] to remove all multiple parallel edges. Then, the second lifting of $64$ or $85$ for $8 \times 4$ proto-structures, and $85$ for $7 \times 4$ structure, respectively, using the PEG algorithm. This algorithm was applied to determine a circulant permutation of each edge class to avoid short-length cycles occurred with the desired code block length of approximately $1024$ bits. The decoder is a standard message-passing decoder, in which the maximum number of iterations is set to $25$. LLR clipping and other decoding parameters are set according to [29].

The FER performance of our two new codes is shown in Figure 4. No error floor is observed down to FER $10^{-4}$. The figure shows that Code-$3$ outperforms Code-$1$ with a gap of approximately $0.5$ dB. These two codes have the same protograph structure. The similar observation is seen with the unpunctured protograph structure; Particularly, Code-$4$ is $0.23$ dB better than Code-$2$. Furthermore, Code-$4$ has a gain of approximately $0.72$ dB over Code-$1$. Performances of these codes are compared at $FER = 4 \times 10^{-4}$. These performance gains highlight the effectiveness of our code design method with a small number of decoding iterations. From Figure 4, since Code-$4$ also performs better than Code-$3$ about $0.1$ dB. It once confirms that the non-punctured protograph structure is preferable to the punctured one with a small number of decoding iterations.

### *E. Case study: Channel Efficiency of the New Protograph Code in Satellite Communications*

In this subsection, we further investigate the benefit of our coding design in terms of channel efficiency via a practical automatic repeat request (ARQ) in satellite communication systems. In fact, protograph codes have been used in space communications [30], [31]. We now investigate how the FER improvement in our coding design is translated to the improvement of the channel efficiency in the satellite communication context. The channel efficiency is defined as [32]:

$$\eta = \frac{\text{Throughput}}{R_b}, \tag{6}$$

where $R_b$ is the data bit rate. The channel efficiency depends on data bit rate, information block length, frame error rate, and round-trip time of the communication channel. When the automatic repeat request go-back-$N$ (ARQ-GB) protocol is used, the expression for the channel efficiency is given below [32]:

$$\eta_{GB} = \frac{k(1 - FER) \times 100}{k(1 - FER) + R_b T_{RT} FER}, \tag{7}$$

where $k$ is the information block length, and $T_{RT}$ is round-trip time of the channel. For detailed derivation of (7) and the round-trip time, readers refer to [32].

Assume that the decoder was designed to operate at signal-to-noise (SNR) ratio level of 2 dB, the distance from an earth station to a geostationary satellite is approximated 36000 km [33], and the data rate $R_b$ is equal $2.048$ Mbps and information block-length is $1024$ bits, the channel efficiency of for codes is given in Table I. Code $4$, which is designed for a small number of decoding iterations can achieve the channel efficiency of $91\%$, which is almost two and three times higher than that of Code $3$ and Code $2$, respectively. When comparing with Code $1$, both Code $2$, Code $3$ and Code $4$ have much better channel efficiency than that of Code $1$. The significant improvement in the channel efficiency of Code $4$ is important for today's communication systems where efficient use of the channel is a vital factor to meet the high traffic demand. This suggests that it is worth redesigning protograph codes for a small number of decoding iterations.

## III. Low Complexity Rate-Compatible Code Design

In today's communication system, it often happens that one communication platform is utilized to deliver different types of services and applications. Each of which usually has its own sets of quality of service (QoS) requirements, including delay, data bit rate, and bit error rate [2]. The rate-

| Protograph codes | FER | Channel efficiency $\eta_{GB}$ |
|---|---|---|
| Code 1 | 9e-2 | 2% |
| Code 2 | 4e-3 | 34% |
| Code 3 | 2e-3 | 51% |
| Code 4 | 2e-4 | 91% |

TABLE I
CHANNEL EFFICIENCY COMPARISON.

compatible channel code is one of the essential components to support multiple services in one hardware platform. The rate-compatible protograph codes were found by Nguyen et al. [18] for a wide range of code rates. However, the rate-compatible family was optimized for long block length and a large number of decoding iterations. Although a new set of rate-compatible punctured codes was reported in [26] for the finite block length, it was only optimized for a large number of decoding iterations. To authors' best knowledge, low complexity rate-compatible non-punctured protograph codes with a limited number of decoding iterations for communication channels such as BI-AWGN have not been reported in the literature although the demand for such codes is high, particularly for 5G mobile networks with ultra-low delay applications and services. For such a reason, this section presents the rate-compatible low-iteration protograph codes which perform well at both the water-fall and error-floor regions. The design of the rate-compatible protograph codes, which have fixed information block-length, consists of two phases:

**1) Daughter Code Design:** In our design framework, we first need to design a high-rate code called a daughter code. This code is designed using a technique called the lengthening technique. The technique was widely used in protograph designs, thank to its simple protograph structure, yet producing highly-performing codes [18]. To start with, one must choose a base protograph that usually has a low code rate, e.g., rate-$1/2$ code. Based on the base protograph, the parity check proto-matrix of a higher-rate code is constructed by adding one or more variable nodes to the base code while keeping the number of check nodes unchanged.

**2) Rate-Compatible Code Design:** The protograph of a high-rate daughter code found above is in turn used to design the lower-rate protographs by adding the same number of variable nodes and check nodes. These lower rate codes are built in an embedded rate-compatible structure from the daughter code, having the same information bit block. This rate-compatible structure makes the codes suitable for many practical applications where low complexity is a crucial technical requirement.

In the following, the detailed design of daughter codes and the rate-compatible codes are described in detail.

### A. Daughter Codes

In this section, we present one code family construction. However, the design process ca ben extended to other code families. We start with rate-$1/2$ protograph; nevertheless, we can start from any applicable rate code. To design a high-rate daughter code, one can use the method as described in the previous section. But, it is much more difficult, since the search space contains many more elements. The difficulty can be managed by using the code lengthening design approach. Here, we pick the best rate-$1/2$ protograph studied in the previous subsection whose proto-matrix is shown in Eq. (5), so-called Code $4$ protograph.

In the lengthening technique, a higher-rate protograph is constructed from the lower-rate protograph by adding one or more variable nodes while keeping the number of the check nodes unchanged. The structure of the proto-matrix for a certain daughter code is given

$$H_H = [H_E \mid H_L] \tag{8}$$

where $H_L$ is the parity check matrix of the low-rate code, $H_E$ is an extension matrix and $H_H$ is the parity check matrix of a high-rate code.

In the following, we give an example of designing some daughter codes, starting from Code $4$, whose code rates are written in the following formula:

$$R = \frac{n+1}{n+2}, \tag{9}$$

where $n = 1, 2, 3$. To achieved this rate formula, for each value of $n$ the extension matrix $H_E$ must have $4$ more variables nodes for each code rate. This means that a new protograph is obtained by adding $4$ variable nodes at a time while keeping the number of check nodes the same as the number of check nodes of the base protograph. The structure of the proto-matrix for the rate $R = \frac{n+1}{n+2}$ is given in equation (10):

$$H_{\frac{n+1}{n+2}} = \left(\begin{array}{cccc|c} x_1 & x_5 & x_9 & x_{13} & \\ x_2 & x_6 & x_{10} & x_{14} & \\ x_3 & x_7 & x_{11} & x_{15} & H_{\frac{n}{n+1}} \\ x_4 & x_8 & x_{12} & x_{16} & \end{array}\right). \tag{10}$$

Using the same constraints on $x_i, i = 1, 2, \cdots, 16$ as in Section II, we can search for the higher rate codes based on the base matrix $H_{1/2}$ in (5). The protographs for the daughter codes with rates $R = 2/3, 3/4, 4/5$ are given below:

$$H_{\frac{2}{3}} = \left(\begin{array}{cccc|c} 3 & 0 & 0 & 1 & \\ 2 & 1 & 1 & 1 & \\ 3 & 0 & 0 & 1 & H_{\frac{1}{2}} \\ 3 & 2 & 2 & 0 & \end{array}\right), \tag{11}$$

$$H_{\frac{3}{4}} = \left(\begin{array}{cccc|c} 3 & 0 & 0 & 1 & \\ 2 & 2 & 2 & 0 & \\ 3 & 0 & 1 & 0 & H_{\frac{2}{3}} \\ 3 & 1 & 0 & 2 & \end{array}\right), \tag{12}$$

$$H_{\frac{4}{5}} = \left(\begin{array}{cccc|c} 3 & 0 & 0 & 1 & \\ 3 & 0 & 1 & 1 & \\ 3 & 1 & 1 & 0 & H_{\frac{3}{4}} \\ 2 & 2 & 1 & 1 & \end{array}\right). \tag{13}$$

*Threshold and Frame Error Rate Evaluations of Daughter Codes* The most recent non-punctured low-iteration protograph codes of size $8 \times 4$ were reported by Uchikawa [14]. The iterative decoding thresholds of the proposed codes and the codes by Uchikawa are given in Table II. It is seen that the newly found protographs all have lower iterative decoding

| Rate | Uchikawa's codes [14] | New codes |
|---|---|---|
| 1/2 | 1.03 | 0.96 |
| 2/3 | 1.65 | 1.58 |
| 3/4 | 2.14 | 2.06 |
| 4/5 | 2.49 | 2.43 |

TABLE II
ITERATIVE DECODING THRESHOLD COMPARISON.

thresholds in comparison with those of Uchikawa's codes. It is proven via simulation that the code with a lower decoding threshold has better FER performance and BER performance as shown in Figure 5 and Figure 6, respectively. The proposed codes outperform 0.3 dB Uchikawa's codes at FER level of $10^{-4}$. Moreover, the curves of the proposed codes have steep slopes in the water-fall region, and none of the proposed codes has the error-floor behavior even below the FER level of $10^{-4}$. These attributes are crucial for many applications in which low frame error rates are required.

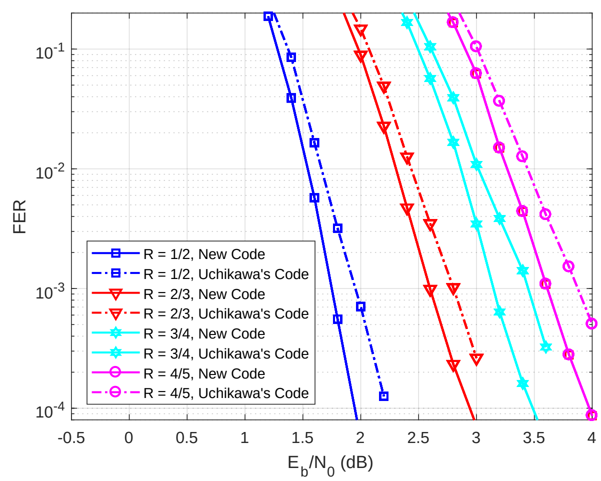


Fig. 5. Frame error rate, 25 iterations, block length 1024 bits.

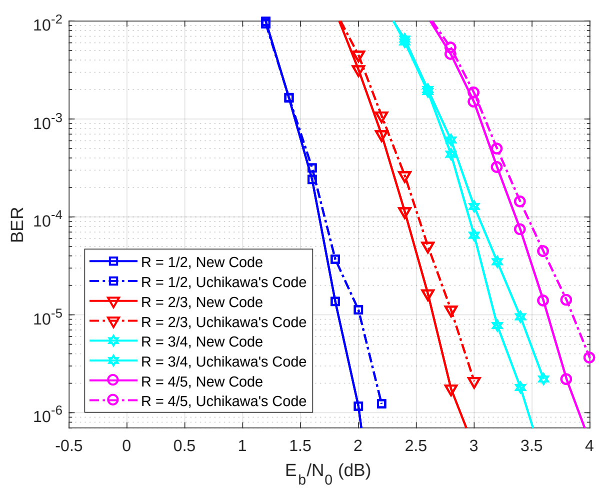


Fig. 6. Bit error rate, 25 iterations, block length 1024 bits.

*B. Rate-Compatible Protograph Codes*

As shown in Subsection III-A, one can proceed to use the lengthening method to obtain the protograph codes of various code rates. However, the drawback of the lengthening method is that the information block lengths of the daughter codes vary from one code rate to another. As a consequence, the transmission scheme using rate-compatible techniques becomes highly complex to implement. The design idea of the rate-compatible codes is to use the protograph of a daughter code as the base, and then the base protograph is enlarged by adding the same number of rows and columns to obtain a new proto-matrix for a new rate-compatible code. By doing this, we guarantee that the whole family of the rate-compatible protograph codes has a fixed information block length. As a result, the complexity of the encoding/decoding structure is kept low. The structure of the rate-compatible protograph is given in Figure 7.

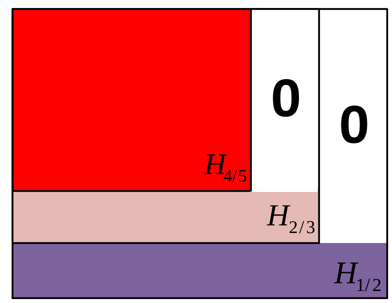


Fig. 7. The structure of rate-compatible protographs.

For demonstration purpose, we will use the proto-matrix of rate $4/5$ with proto-matrix size of $20 \times 4$ in (13) to develop a new family of the rate-compatible protograph codes which have the rates in the form $R = 16/(n+20)$, where $n$ is the number of extra rows and columns added to the base matrix.

For example, by adding one extra row and one extra column to proto-matrix of the base protograph $H_{4/5}$ in (13), the structure for the protograph of a lower rate $R = 16/21$ is shown in (14):

$$H_{\frac{16}{21}} = \left(\begin{array}{cccc|c} & & & & 0 \\ & & H_{\frac{4}{5}} & & 0 \\ & & & & 0 \\ & & & & 0 \\ & & & & 0 \\ \hline x_1 & x_2 & \cdots & x_{20} & 1 \end{array}\right) \quad (14)$$

The last variable node in column 21 in (14) is fixed and has degree 1. The choice is based on the design guidelines that a good protograph LDPC code should have some degree-1 variable nodes [18]. The newly added $5^{th}$ row has 20 variables $x_i, i = 1, 2, \cdots, 20$ that are optimized such that the resulting protograph has both the low iterative decoding threshold and linear minimum distance growth, and thus the code has good FER performance. We set some constraints on these variables to narrow the search space when searching for a good protograph code of rate $R = 16/21$ as: $x_i \in \{0, 1\}, i = 1, 2, \cdots, 19$ and $x_{20} \in \{0, 1, 2\}$.

Even we reduce the search space for the variables $x_i \in \{0, 1\}, i = 1, 2, \cdots, 19$, we still expect that the resulting rate-compatible protograph codes still have good performance since the daughter code protographs are designed by the method following the guidelines to produce a good code [18]: several degree-2 variable nodes and the other variable nodes have degree greater than 3 for linear minimum distance growth. Therefore, variable $x_i \in \{0, 1\}$ does not destroy the nice properties of the daughter codes. The search results for 20 rate-compatible protograph codes with rates from $R = 16/21$ to $R = 16/40$ are found. The proto-matrix of the protograph with the lowest rate of $16/40$ is shown in Eq. (15), from which all proto-matrices of the other code rates are deduced. All of these codes are capacity approaching with their decoding thresholds approaching capacity as described in the following praragraphs.

*Iterative decoding threshold and FER performance* The iterative decoding thresholds of the proposed rate-compatible protograph codes are listed in Table III. The threshold gap to the Shannon threshold is at most 0.739 dB. This gap is more significant compared to that of the rate-compatible code family, whose maximum gap is 0.533 dB, in [26]. The exact difference is 0.206 dB. The reason for the more significant gap of the proposed code family is that the proposed code family is optimized for a small number of decoding iterations. Whereas the protograph codes in [26] were optimized for a large number of decoding iterations (200 iterations). It is also noted that the block lengths of the two code families are equal 1024 bits.

The FER curves for some selected codes in the family of rate-compatible codes are plotted in Figure 8 via computer simulations with the information blocklength 1k to support the analytical results. The simulation procedure is similar to the one in Section II-D. As expected, the FER curves of the proposed rate-compatible codes have a steep slope at the water-fall region and none of them has the error-floor behavior as low as FER = $10^{-5}$. This attribute is vitally important for ultra-reliable communication applications, especially for 5G communications system [9].

$$H_{16/40} =$$
$$\begin{pmatrix}
3333301110000000100100000000000000000000\\
3232211001101201120000000000000000000000\\
3333120000011010111000000000000000000000\\
3323001222221121001100000000000000000000\\
2111110010011000000010000000000000000000\\
2111110001000100000001000000000000000000\\
2111100000000111000000100000000000000000\\
2111101010000100000000010000000000000000\\
2111001100001000000000001000000000000000\\
2111100001000010000000000100000000000000\\
2111000000110000000000000010000000000000\\
2111100000000001000000000001000000000000\\
2111100000000000001000000000100000000000\\
2110001000010010000000000000010000000000\\
2111000010100000000000000000001000000000\\
2101000000000000010000000000000100000000\\
2111000110000000000000000000000010000000\\
2110000000000001100000000000000001000000\\
2110000001000000100000000000000000100000\\
2010000101000000000000000000000000010000\\
2110010000000001000000000000000000001000\\
2100010100000000000000000000000000000100\\
2100001000000000001000000000000000000010\\
2100100000100000000000000000000000000001
\end{pmatrix} \tag{15}$$

| n | Rate | Threshold | Shannon Threshold | Gap |
|---|---|---|---|---|
| 1 | 16/21 | 2.168 | 1.719 | 0.449 |
| 2 | 16/22 | 1.947 | 1.460 | 0.487 |
| 3 | 16/23 | 1.756 | 1.235 | 0.521 |
| 4 | 16/24 | 1.594 | 1.058 | 0.536 |
| 5 | 16/25 | 1.453 | 0.900 | 0.553 |
| 6 | 16/26 | 1.334 | 0.762 | 0.572 |
| 7 | 16/27 | 1.229 | 0.639 | 0.590 |
| 8 | 16/28 | 1.133 | 0.529 | 0.604 |
| 9 | 16/29 | 1.046 | 0.430 | 0.616 |
| 10 | 16/30 | 0.976 | 0.341 | 0.635 |
| 11 | 16/31 | 0.905 | 0.259 | 0.646 |
| 12 | 16/32 (1/2) | 0.842 | 0.185 | 0.657 |
| 13 | 16/33 | 0.786 | 0.117 | 0.669 |
| 14 | 16/34 | 0.731 | 0.054 | 0.677 |
| 15 | 16/35 | 0.685 | -0.004 | 0.689 |
| 16 | 16/36 | 0.644 | -0.057 | 0.701 |
| 17 | 16/37 | 0.604 | -0.107 | 0.711 |
| 18 | 16/38 | 0.570 | -0.153 | 0.723 |
| 19 | 16/39 | 0.536 | -0.197 | 0.733 |
| 20 | 16/40 | 0.503 | -0.236 | 0.739 |

TABLE III
ITERATIVE DECODING THRESHOLDS FOR RATE-COMPATIBLE FAMILY WITH CODE RATE FROM $16/40$ TO $16/21$.

## IV. CONCLUSION

This paper presents an approach for designing a family of the low complexity and low delay rate-compatible protograph codes whose code rates can vary in the range from 0.4 to 0.8. Both analytical and simulation results prove that the proposed codes outperform the state-of-the-art codes when the number of decoding iterations is limited. All selected code members in the rate-compatible family have steep FER curves at the water-fall region and none of them have error-floor behavior at FER as low as $10^{-5}$. The low complexity and low delay properties together with excellent FER/BER performance make the proposed rate-compatible codes applicable to new communication networks where delivering ultra-low delay and ultra-reliable is one of essential requirements such as next 5G mobile networks [9].

## ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2016.23

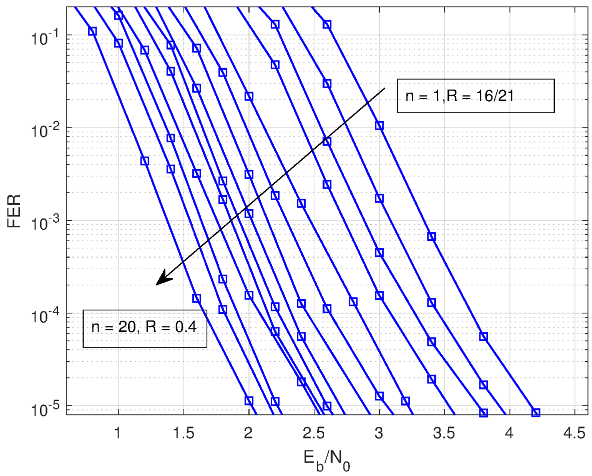


Fig. 8. The FER performance of rate-compatible protograph code $R = 16/(n+20), n = 1, 2, \cdots, 20$, block length 1024 bits.

## REFERENCES


[1] T. V. Nguyen and H. T. Nguyen, "The design of optimized fast decoding protograph LDPC codes," in *Proc. Int. Conf. Advanced Technologies for Communications (ATC)*, Oct. 2016, pp. 282–286.

[2] I. S. Comsa, A. De-Domenico, and D. Ktenas, "QoS-driven scheduling in 5G radio access networks - A reinforcement learning approach," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–7.

[3] P. Chen, K. Cai, and S. Zheng, "Rate-adaptive protograph LDPC codes for multi-level-cell nand flash memory," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1112–1115, June 2018.

[4] T. P. Fowdur and B. N. Furzun, "Performance of IEEE 802.11n LDPC codes with modified reliability based hybrid ARQ and unequal error protection," in *Int. Conf. on Computer as a Tool (EUROCON)*, Sept. 2015, pp. 1–6.

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

[6] G. Ricciutelli, M. Baldi, N. Maturo, and F. Chiaraluce, "LDPC coded modulation schemes with largely unequal error protection," in *IEEE Int. Black Sea Conf. on Commun. and Networking (BlackSeaCom)*, May 2015, pp. 48–52.

[7] L. Xu, L. Wang, S. Hong, and H. Wu, "New results on radiography image transmission with unequal error protection using protograph double LDPC codes," in *Int. Symp. on Medical Information and Commun. Technology (ISMICT)*, Apr. 2014, pp. 1–4.

[8] B. Rajkumarsingh and K. Nuckchady, "LDPC coded adaptive multiresolution modulation for unequal error protection," in *2013 Africon*, Sept 2013, pp. 1–5.

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

[10] A. Goldsmith, *Wireless Communications*, 2005.

[11] I. B. Djordjevic and D. Zou, "FPGA-based rate-adaptive LDPC-coded modulation for the next generation of optical communication systems," in *Int. Conf. on Transparent Optical Networks (ICTON)*, July 2017, pp. 1–6.

[12] B. Li, C. Ju, H. Yang, and G. Liu, "Adaptive coded modulation based on LDPC codes," in *Int. Conf. on Commun. and Networking in China (ChinaCom)*, Aug. 2015, pp. 648–651.

[13] M. Mazzotti, S. Moretti, and M. Chiani, "Multiuser resource allocation with adaptive modulation and LDPC coding for heterogeneous traffic in ofdma downlink," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2915–2925, October 2012.

[14] H. Uchikawa, "Design of non-precoded protograph-based LDPC codes," in *Proc. IEEE ISIT*, June 2014, pp. 2779–2783.

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

[16] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 3250–3254.

[17] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Select. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[18] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "The design of rate-compatible protograph LDPC codes," *IEEE Trans. Commun.*, Oct. 2012.

[19] T. Koike-Akino, D. S. Millar, K. Kojima, K. Parsons, Y. Miyata, K. Sugihara, and W. Matsumoto, "Iteration-aware LDPC code design for low-power optical communications," *Journal of Lightwave Technology*, vol. 34, no. 2, pp. 573–581, Jan. 2016.

[20] L. Buccheri, S. Mandelli, S. Saur, L. Reggiani, and M. Magarini, "Hybrid retransmission scheme for QoS-defined 5G ultra-reliable low-latency communications," in *IEEE Wireless Commun. and Net. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[21] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[22] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.

[23] G. Liva, W. E. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 49–57, January 2008.

[24] C. Tang, M. Jiang, C. Zhao, and H. Shen, "Design of protograph-based LDPC codes with limited decoding complexity," *IEEE Commun. Lett.*, vol. PP, no. 99, p. 1, 2017.

[25] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate-repeat-accumulate codes," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 692–702, Apr. 2007.

[26] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "Rate-compatible short-length protograph LDPC codes," *IEEE Commun. Lett.*, May 2013.

[27] S. Abu-Surra, D. Divsalar, and W. Ryan, "On the existence of typical minimum distance for protograph-based LDPC codes," in *Information Theory and Applications Workshop (ITA)*, Jan. 2010, pp. 1–7.

[28] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, pp. 386–398, 2003.

[29] J. Hamkins, "Performance of low-density parity-check coded modulation," IPN Progress Report 42-184, Feb. 2011.

[30] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, "A survey on protograph ldpc codes and their applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1989–2016, Fourthquarter 2015.

[31] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The development of turbo and LDPC codes for deep-space applications," *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2142–2156, Nov 2007.

[32] G. Maral, *VSAT Networks*, 2003.

[33] ——, *Sattelite communications systems: Systems, Techniques and Technology*, 2009.