Proximity Link Throughput Enhancements via Raptor Code Technology

Amogh Rajanna*, Clay Okino†, and Ken Andrews†

This version of the article incorporates errata described at https://ipnpr.jpl.nasa.gov/progress_report/42-229/42-229C.pdf. The original version of the article is available at https://ipnpr.jpl.nasa.gov/progress_report/42-224/42-224A-orig.pdf.

ABSTRACT. — Proximity link space communications, such as between a lander and an orbiter, uses the Consultative Committee for Space Data Systems (CCSDS) Proximity-1 space link protocol standard, which specifies a low-density parity-check (LDPC) code of rate 1/2, and a Go-Back-N automatic repeat request (ARQ) protocol in the data link layer. It also includes an optional feedback mechanism by which the receiver can request transmitter changes; this was first used by the Mars Science Laboratory to implement an Adaptive Data Rate (ADR) mechanism based on the received signal quality. If such an ADR mechanism is not used, the Proximity-1 standard can have severely limited throughput due to an inefficient retransmission protocol. In this article, we develop a new adaptive transmissions technology based on Raptor codes for the additive white Gaussian noise channel, with modifications for spacecraft proximity links. In the new design, transfer frames are Raptor encoded, and in place of retransmissions, additional code symbols are generated and transmitted as needed. This new technology matches the coding rate to the instantaneous channel conditions, thereby maximizing the throughput at both high and low signal-to-noise ratio (SNR). By simulation, we show that Raptor codes provide greater throughput than the Proximity-1 (Prox-1) standard operating without ADR over a wide range of SNRs. At high SNR, this scheme improves the throughput by up to 90%, and at low SNR, it provides reliable communication even when the throughput of the Prox-1 standard essentially drops to zero. Our analysis also shows that this Raptor code approach reduces the computational costs for both the encoder and decoder.

---

*Formerly with the Communications Architectures and Research Section as a Visiting Researcher, University of Bristol, UK.

†Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

I. Introduction

In this article, we focus on proximity links, i.e., space communication links designed for the purpose of data transfer among probes, landers, rovers, orbiting satellite relays, and orbiting constellations. Design and operation of proximity link communication systems are based on communications standards developed by the Consultative Committee for Space Data Systems (CCSDS). Proximity-1 (Prox-1) standard-based communications technology is used for reliable operation of proximity links. In the current Prox-1 Coding and Synchronization (C&S) standard, each message block is encoded by an $R = 1/2$ low-density parity check (LDPC) code in the C&S sublayer. A Go-Back-N automatic repeat request (ARQ) protocol is present in the data link layer (DLL). The performance of the $R = 1/2$ LDPC code and the Go-Back-N protocol determine the overall throughput of a proximity link for a given operating point. When used with fixed transmission parameters, the current Prox-1 standard severely limits the throughput when the environment differs from the optimal operating point such as when the channel has higher signal-to-noise ratio (SNR).

The goal of this article is to explore a rateless coding approach that will increase the throughput of proximity links relative to the current Prox-1 standard. In this article, we develop a rateless code technology that CCSDS could consider as an enhancement to the Prox-1 C&S standard for proximity links. A very interesting use case for the proposed rateless code technology is the CubeSat relays. Science missions with CubeSat constellations are among the strategic technology directions of NASA space exploration. Unlike larger spacecraft, CubeSats have a much smaller energy budget. Enhancing the performance of a CubeSat proximity link under a very limited energy budget is of crucial importance for NASA space missions.

Rateless codes being a new class of variable-length codes have the innate property to adapt both the parity bit construction and the number of parity bits in response to the time-varying channel conditions. In this article, we will be using rateless codes designed for the additive white gaussian noise (AWGN) channel. The rateless code we consider is the Raptor code, which has a high-rate outer code, i.e., $R = 0.95$ LDPC code and an inner Luby transform (LT) code. The parity symbols for a Raptor encoded message block are incrementally transmitted until the receiver decoding succeeds. As such, each message block (or frame) has the potential for a variable amount of coding resulting in a variable code-rate. N_{cb} is the number of parity bits per code block. We evaluate the Raptor code system for three values of $N_{cb} = \{450, 250, 150\}$. The key aspect of the LT encoder design is the generation of the parity bit degrees d. In this article, we generate the parity bit degrees d based on two types of approaches, namely a) degree distribution and b) descending order degrees.

To perform throughput simulation under both the Prox-1 standard and the Raptor addition, K-bit message blocks are transmitted until successful decoding at the receiver for a wide range of SNRs. From a throughput standpoint, the optimal SNR for the Prox-1 standard-based system is 2 dB. In the low SNR regime, the Prox-1 standard has near-

zero throughput while the Raptor addition has good throughput ensuring the transfer of a substantial number of message blocks even in the presence of intermittent links. Also in the high SNR regime, there is a throughput gain of the Raptor addition over the Prox-1 standard. Specifically, we demonstrate through simulations enhancements in throughput performance on the order of 30% to 90% relative to the current Prox-1 over varying SNR scenarios.

In terms of the computational costs, the Raptor addition has an advantage over the Prox-1 standard. The encoder costs of the Raptor scheme with a sparse generator matrix is much smaller than the Prox-1 LDPC, which has a high density generator matrix. From a decoder costs perspective, the relative computational advantage of the Raptor scheme is valid for low SNRs. At high SNRs, one of the Raptor approaches has decoding costs in a similar range as the current Prox-1 scheme.

Section II discusses the salient features of the currently used Prox-1 standard specifications. We introduce the novel Raptor code-based transmission model in Section III. In Sections IV and V, we provide a detailed description of the new Raptor code-based transmissions technology developed. We present the performance results of the new approach by means of simulation results in Section VI. An insightful comparison of the two transmission technologies from a computational cost standpoint is presented in Section VII. A summary and conclusion of the article are provided in Section IX.

II. Current Prox-1 Standard - CCSDS 211.0-B-5

CCSDS 211.0-B-5 is the Proximity-1 space link protocol standard for the data link layer DLL [2]. It specifies the data communication operations procedures for transfer of Service Data Units (SDUs, typically packets) from the sender to the receiver. On the sender side, the DLL accepts SDUs from the network layer and delivers a coded symbol stream to the physical layer. On the receiver side, the DLL accepts a stream of noisy demodulated symbols from the physical layer, and delivers SDUs to the network layer.

We now describe the key aspects of the CCSDS 211.0-B-5 standard, which are relevant to our data retransmission modelling approach. Figure 1 shows a simplified layered model of the DLL in the CCSDS 211.0-B-5. The DLL of the protocol stack has input/output (I/O), data services, frame, and C&S sublayers. In the data services sublayer of the transmit side, the version 3 transfer frame (V3TF) to be transmitted is selected from either the new frame queue or the sent frame queue. In the frame sublayer, the 24-bit attached sync marker (ASM), the V3TF, and 32-bit cyclic redundancy check (CRC) are concatenated to form the proximity link transmission unit (PLTU). In a proximity space link, the data packets/units to be transmitted are in the form of PLTUs, which can be up to 16384 bits long. Multiple PLTUs are appended along with the idle and tail sequences to form a data stream. In the C&S sublayer, each PLTU is divided into several 1024-bit message blocks. Each message block is encoded by a $R = 1/2$ LDPC code in the C&S sublayer [3]. The LDPC codewords corresponding to a PLTU are

transmitted over the physical layer.

In the C&S sublayer of the receive side, the LDPC codewords are decoded into data blocks and appended to form the PLTUs. In the frame sublayer, each of the PLTUs are split into ASM, V3TF, and CRC. In the data services sublayer, each V3TF undergoes a CRC error detection. If the CRC results in a zero syndrome, the V3TF is passed on to higher layers as a valid frame. If the CRC results in a non-zero syndrome, the V3TF is discarded. One bit of the 16-bit proximity link control word (PLCW) is reserved for the retransmit flag, i.e, acknowledgement/negative acknowledgement (ACK/NACK). The PLCW is part of the P-frame sent by the receive side back to the transmit side.

A Go-Back-N ARQ protocol operating on the V3TF level is present in the data services sublayer [2, p. 2-8, p. 7-6]. The ARQ protocol is activated for every NACK. The performance of the $R = 1/2$ LDPC code and the Go-Back-N protocol determine the throughput of a proximity link.

III. Proposed Rateless Code Model

In this section, we introduce the concept of the rateless code technology addition to the CCSDS C&S standard for proximity links. The Prox-1 standard has the Go-Back-N ARQ protocol, which triggers the retransmission of the entire LDPC codewords of the V3TFs. When used with fixed transmission parameters, this retransmission approach is bandwidth-inefficient and suffers from throughput loss due to inefficient retransmissions. In this section, we seek to enhance the throughput of a proximity link by proposing a new retransmission model in which the code bits are transmitted to the receiver on an as-need basis. The idea is to optimize the throughput of a K-bit message block [3, section 3.4.4.1(c)] transmission by only sending the necessary parity bits, i.e., minimizing the parity overhead.

Figure 2 shows a simplified layered model of the DLL for Proximity-1 space links with the Raptor code option in the C&S sublayer. The nature of code construction for the Raptor code method is bound to substantially influence the sequence of operations in the frame and data services sublayers of the DLL. The Raptor decoder in the C&S sublayer interfaces with the CRC error detection in the frame sublayer. The non-zero syndrome of CRC detection triggers the Raptor decoding based on incremental parity symbols. The parity generation nature of the Raptor coder and its decoder operation have implications for the ACK/NACK signal and the Go-Back-N ARQ protocol of data services sublayer. In Figure 2, the protocol stack is an idealized version and the goal is to design a very simple layered model with potentially no ARQ protocol for the Raptor codes-based transmission.

The message blocks of a PLTU are Raptor encoded. The parity symbols for a Raptor encoded message block are incrementally transmitted until the receiver side decoding succeeds, i.e., zero syndrome for the CRC detection. Each message block is potentially transmitted by a variable code rate. All the message blocks and thus, the PLTUs can be

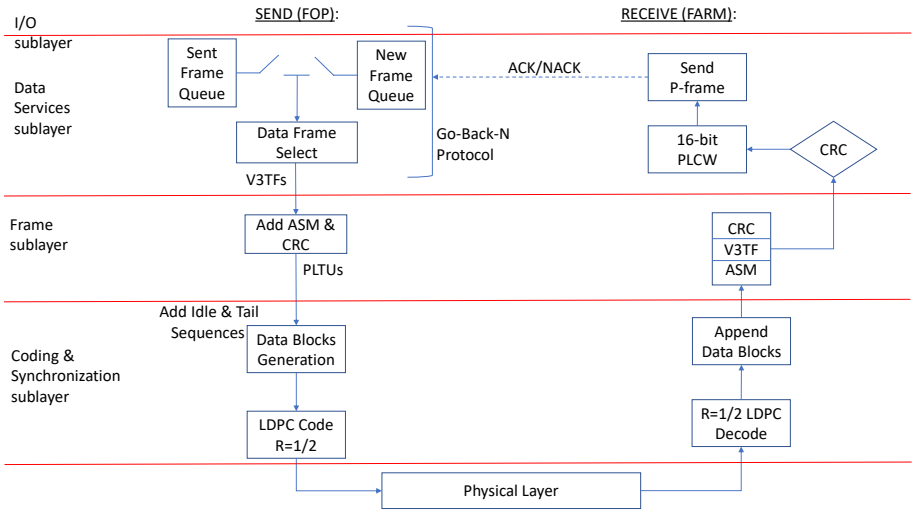

Figure 1. Simplified layered model of the DLL of CCSDS 211.0-B-5 Proximity-1 standard. FOP stands for Frame Operation Procedure and FARM stands for Frame Acceptance and Reporting Mechanism.

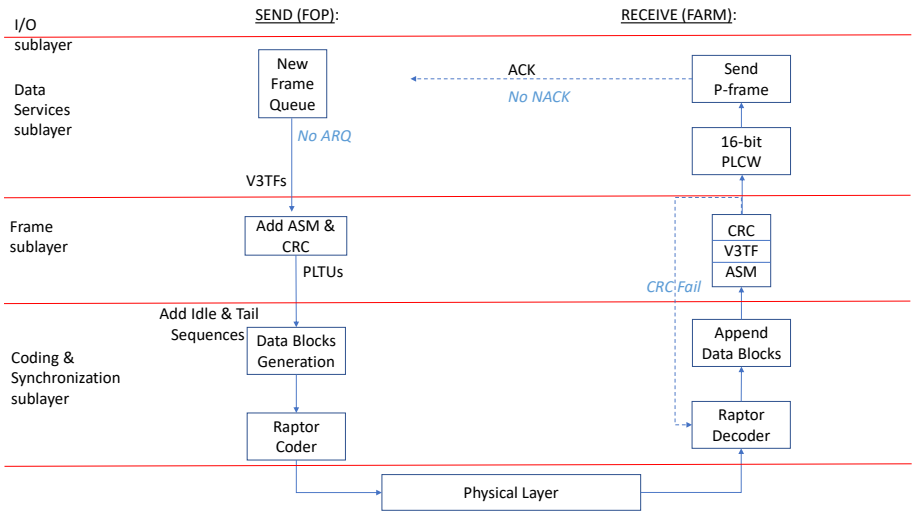

Figure 2. Simplified layered model of the DLL with Raptor code option for the Prox-1 standard.

successfully communicated by variable-rate transmission. Effectively, there is no need for a V3TF-level ARQ protocol in the data link layer.

Rateless codes being a new class of variable-length codes have the innate property to adapt both the code/parity bit construction and the number of code/parity bits in response to the time-varying channel conditions. Rateless codes were originally developed for the erasure channel around 15 years ago. However, due to the above two desirable properties, rateless codes have been investigated for the AWGN channel too[8]-[10].

Rateless codes have a high-rate outer code and an inner LT code. Raptor codes are a special class of rateless codes in which the outer code is a LDPC code [8][9]. (BCH codes are also an option for the outer code. See Analog Fountain Codes [10].) In this article, we develop a new transmissions technology based on the Raptor code with several key enhancements relative to the current Prox-1 standard. Figure 3 shows a block diagram view of the Raptor encoder and decoder to be used in the C&S sublayer of the DLL of the Prox-1 standard as part of the proposed Raptor addition. The design of the Raptor encoder and decoder is discussed in the next two sections.

IV. Raptor Encoder for the AWGN Channel

In this section, we describe in detail the design of the Raptor encoder. First, we discuss the design of the outer code, i.e., LDPC code and then, move onto the design of the inner LT code.

A. Outer Code - LDPC Encoder

The outer code is a high-rate LDPC primarily used for clean up of the error-floors of the inner LT code. Below, we describe the salient features of the outer LDPC code used.

The code-rate is $R = \frac{K}{N} = 0.95$, where K is the number of information bits and N is the number of parity bits. For the parity check matrix H, we adopt a regular design with variable-node degree $d_v = 4$ and check-node degree $d_c = 80$. Let the parity check matrix H be partitioned as $H \triangleq [Q|P]$. Note that the dimensions of the matrices are $\mathbf{H} : (N-K) * N$; $\mathbf{Q} : (N-K) * K$; $\mathbf{P} : (N-K) * (N-K)$.

Now, we describe the design of matrix Q. Initially, Q is a $d_v * (d_c - d_v)$ matrix of ones.

$$Q = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix} \tag{1}$$

We focus on generating the matrix Q by the random permutation method [11]. Each 1 in Q is replaced by a π_i, a random permutation block of size $M * M$, i.e., π_i is obtained by a random permutation of the $M * M$ identity matrix, where $M = (N-K)/d_v =$

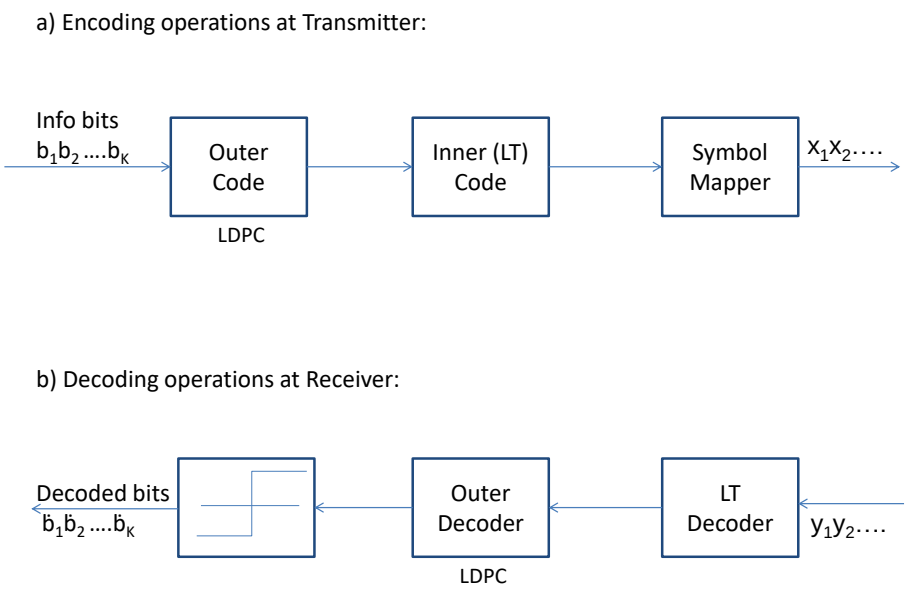

Figure 3. Block diagram view of the Raptor encoder and decoder.

$\frac{K}{4} \cdot \frac{1}{19} = \frac{K}{76}$. Then, the matrix Q expands to $(N-K) * K$. Note that for a choice of $d_v = 4$ and code-rate $R = 0.95$, we are required to choose K as a multiple of 76.

We now describe the design of the square matrix P. Initially, P is a $d_v * d_v$ matrix of ones. Similar to Q, each 1 in P is replaced by a π, a random permutation block of size $M * M$. However, we have to construct P such that P is invertible in GF(2) (Galois Field (2), i.e., modulo-2 arithmetic). This translates to the condition that no set of rows of P should add to zero in GF(2). To satisfy this condition, any three of the four π_i's on the main diagonal are replaced by π_i^o's. Alternatively, all the π_i's on the lower/upper diagonal are replaced by π_i^o's. We do this so that no two rows of P add up to an all-zero row. Note that π_i^o is the same as π_i with one column replaced by an all-zero column. Currently, we replace the last column of π_i by an all-zero column, although a random column can be chosen.

$$P = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \\ \pi_5^o & \pi_6 & \pi_7 & \pi_8 \\ \pi_9 & \pi_{10}^o & \pi_{11} & \pi_{12} \\ \pi_{13} & \pi_{14} & \pi_{15}^o & \pi_{16} \end{bmatrix} \tag{2}$$

Since P is invertible in GF(2), the generator matrix G_o is constructed from parity check matrix H as described in [6]. In this article, we focus on obtaining the matrix H via the method of random permutations. The method of structured permutations might improve the LDPC code performance although it is more complex.

B. Inner Code - LT Encoder

In this subsection, the inner LT code design is presented. Figure 4 shows a block diagram view of the LT encoder. Each parity/code bit of the LT code has a degree d. To generate a code bit of degree d based on a K-bit information vector, uniformly select d out of K bits and perform eXclusive OR (XOR) operation on them. This procedure is repeated for every new parity bit. In this way, an infinite number of parity bits can be generated.

The key aspect of the LT encoder design is the generation of degree d of the parity bits. In this article, we generate the degree d of the parity bits based on two types of approaches. In the first approach, only the number of parity bits generated is adaptive to the channel conditions. The parity bits are assigned random degrees. For both low SNR and high SNR channels, random degree parity bits are generated. In the second approach, both the number of parity bits generated and the parity bit degrees are adaptive to the channel conditions. For high SNR channels, the parity bits are assigned high degrees. As the estimate of channel SNR decreases, the degree of the parity bits is gradually decreased.

1. Degree Distribution

In this approach, each code bit c_i has a degree d_i that is random valued and sampled according to a distribution, which is called degree distribution $\Omega(x)$[8]. Since the Prox-1 standard has $R = 1/2$ LDPC code, the degree distribution $\Omega(x)$ has been optimized for rate $R = 1/2$ for a fair comparison.

$$\Omega(x) = .00477x^1 + .26101x^2 + .0924x^3 + .06913x^8 + .51223x^9 + .06046x^{60} \quad (3)$$

The degree distribution $\Omega(x)$ is the key to good performance. This degree distribution minimizes the bit error rate (BER) at code-rate $R = 1/2$ even though it can generate an

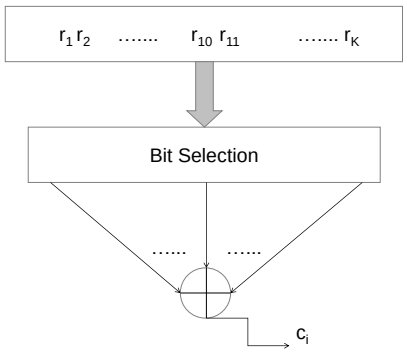

Figure 4. A graphical representation of the LT encoder.

infinite number of parity bits. In the parity bit stream sent by the encoder, we observe that the consecutive parity bits will have random degrees from $\Omega(x)$.

2. Descending Order Degrees

In this approach, the parity bits are assigned degrees according to a structure [9]. The first group of parity bits are assigned degree d_1, the second group of parity bits are assigned degree d_2, the third group is assigned degree d_3 and so on. The parity bits within each group are adjacent to one another. The degrees are in descending order, i.e., $d_1 \geq d_2 \geq d_3 \geq d_4 \cdots$. The number of parity bits in group $\#i$ is L_i and each of them have degree d_i. This approach is termed descending order degrees (DoD). Table 1 tabulates the proposed approach [9].

Table 1. Descending order degrees (DoD) approach to generate the LT code bits. Note N is the number of intermediate code bits output by the R=0.95 LDPC outer code.

Systematic	$d_1 = 12$	$d_2 = 6$	$d_3 = 4$	$d_4 = 2$
N	$L_1 = \alpha_1 N$	$L_2 = \alpha_2 N$	$L_3 = \alpha_3 N$	$L_4 = \alpha_4 N$
	$\alpha_1 = .27$	$\alpha_2 = .146$	$\alpha_3 = 1.39$	$\alpha_4 = \infty$

The codeword $\mathbf{c}$ for the Raptor encoder is given by

$$\mathbf{c} = \mathbf{b}\mathbf{G_o}\mathbf{G_{lt}}, \tag{4}$$

where G_o and G_{lt} are the LDPC and LT code generator matrices, respectively. The generator matrix G_{lt} is of the form $G_{lt} \triangleq [\mathbf{g_1}\mathbf{g_2}\cdots\mathbf{g_i}\cdots]$, where the N element column vector $\mathbf{g_i}$ corresponds to the bit selection for the generation of code bit c_i.

The codeword $\mathbf{c}$ is quadrature phase shift keying (QPSK) modulated to generate the transmit signal vector $\mathbf{x} = [x_1 x_2 \cdots x_i \cdots]$.

V. Raptor Decoder for the AWGN Channel

In this section, the Raptor decoder is described in detail. The inner LT decoder is presented first followed by the outer decoder, i.e., LDPC code decoder.

On the receiver side, the received signal vector $\mathbf{y}$ is given by

$$\mathbf{y} = \mathbf{x} + \mathbf{z}, \tag{5}$$

where $\mathbf{z}$ is the AWGN vector with independent and identically distributed $\mathcal{N}(0, \sigma^2)$ entries. σ^2 is the noise variance.

A. Inner LT Decoder

The received signal $\mathbf{y}$ is input to the inner LT decoder. A bipartite graph is constructed for decoding the inner LT code. The graph has variable nodes denoting the input

bits and the check nodes denoting the parity bits. The decoding graph is based on the generator matrix G_{lt}. Figure 5 shows a sample decoding graph to illustrate the concept. The received samples y_i corresponding to the systematic bits are input to the variable nodes and the samples corresponding to the parity bits are input to the check nodes.

Once the decoding graph is constructed, the standard message passing algorithm, i.e., the belief propagation (BP) or the sum product algorithm (SPA) is used for decoding the LT codes. Below, we provide the equations pertaining to the SPA.

1. Variable nodes: K

$$L_{v_i c_j} = L_{v_i} + \sum_{N_v(i)\setminus\{j\}} L_{c_{j'} v_i}, \tag{6}$$

   where $N_v(i)$ is the set of check nodes connected to the variable node v_i and the summation domain for j' is of the form $B \setminus A = \{x \in B, x \notin A\}$.

2. Check nodes: N'

$$L_{c_j v_i} = 2\ atanh\Big[tanh\left(\frac{L_{c_j}}{2}\right) \prod_{N_c(j)\setminus\{i\}} tanh\left(\frac{L_{v_{i'} c_j}}{2}\right)\Big], \tag{7}$$

   where $N_c(j)$ is the set of variable nodes connected to the check node c_j.

3. Output:

$$\underbrace{L^o_{v_i}}_{APP\ LLR} = \underbrace{L_{v_i}}_{Channel\ LLR} + \underbrace{\sum_{N_v(i)} L_{c_{j'} v_i}}_{Code\ LLR}. \tag{8}$$

Note that $N_v(i)$ can be obtained from the rows of generator matrix G_{lt}. Similarly, $N_c(j)$ can be obtained from the columns of generator matrix G_{lt}.

The SPA is initiated as per Equation (6). The channel log-likelihood ratio (LLR) for the variable nodes, i.e., L_{v_i} are given by

$$L_{v_i} = \frac{2}{\sigma^2} y_i. \tag{9}$$

Initially, the LLR messages from the check nodes $L_{c_{j'} v_i}$ are set to zero. The channel LLR for the check nodes L_{c_j} are computed similar to Equation (9). The a posteriori probability (APP) LLR of the variable nodes $L^o_{v_i}$ are computed as per Equation (8). After the last iteration, the APP LLRs $L^o_{v_i}$ are passed on to the outer LDPC decoder as soft input.

B. Outer LDPC Decoder

The decoding of LDPC codes is described in standard references on digital communications such as in Reference [4]. For completeness, we briefly review it here. A Tanner graph is constructed for decoding the outer LDPC code. The graph has variable nodes denoting the code bits including the systematic bits and the check nodes denoting the

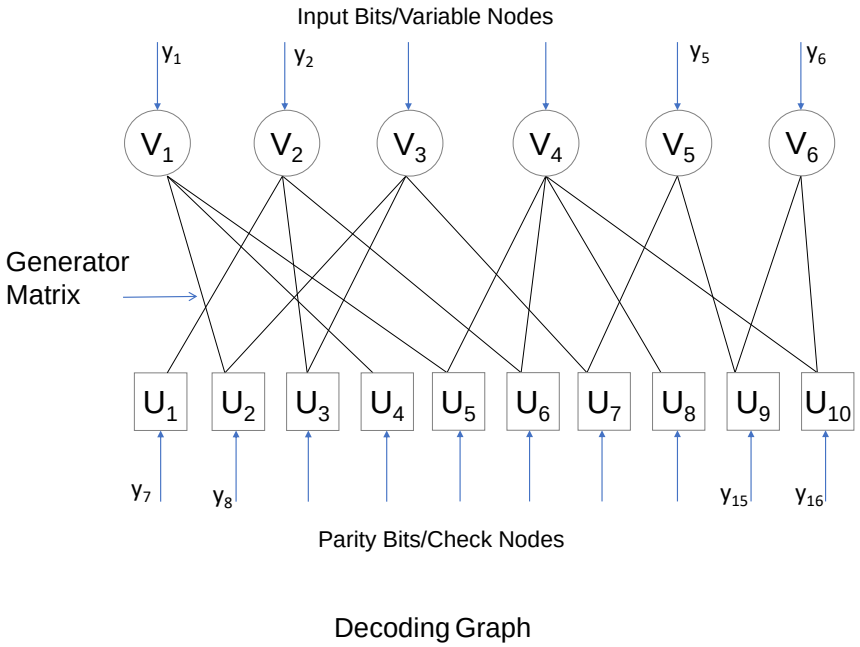

Figure 5. Belief propagation decoding for LT codes. In this decoding graph example, K=6 and N'=10.

parity check equations. The Tanner graph is based on the parity check matrix H. Figure 6 shows a sample Tanner graph to illustrate the concept. Note that the received samples y_i in Figure 6 are set equal to the output LLRs of the inner LT decoder.

Once the Tanner graph is constructed, the standard message passing algorithm, i.e., the BP/SPA is used for decoding the LDPC codes. The SPA for LDPC is given below.

1. Variable nodes: N

$$L_{v_i c_j} = L_{v_i} + \sum_{N_v(i)\backslash\{j\}} L_{c_{j'} v_i}. \tag{10}$$

2. Check nodes: $N - K$

$$L_{c_j v_i} = 2\ atanh\Big[\prod_{N_c(j)\backslash\{i\}} tanh\left(\frac{L_{v_{i'} c_j}}{2}\right)\Big]. \tag{11}$$

3. Output:

$$L^o_{v_i} = L_{v_i} + \sum_{N_v(i)} L_{c_{j'} v_i}. \tag{12}$$

Note that $N_v(i)$ can be obtained from the columns of parity check matrix H. Similarly, $N_c(j)$ can be obtained from the rows of parity check matrix H. The iterations are executed until the parity check equations $\mathbf{Hr} = \mathbf{0}$ are satisfied. The information bits are obtained by applying hard decision to the output LLRs in Equation (12).

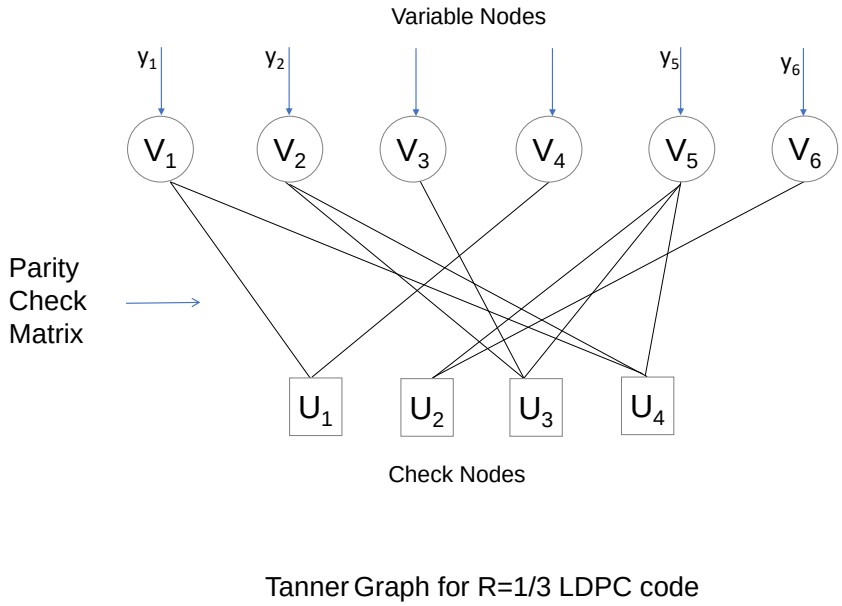

Figure 6. Belief propagation decoding for LDPC codes. In this Tanner graph example, K=2 and N=6.

The simplest decoding architecture for the Raptor decoder has been presented. Enhancements such as joint decoding between the LT and LDPC components may improve the performance albeit with increased complexity.

VI. Simulation Results and Analysis

In this section, the numerical results that illustrate the performance of the Prox-1 standard and the proposed Raptor addition are presented. An end-to-end communication system from transmitter to receiver was simulated in Matlab for both the Prox-1 LDPC and the Raptor schemes. The key characteristics of the communication system are QPSK modulation, AWGN channel, and error-free feedback channel. The effect on throughput of dynamically changing transmission parameters as allowed by the prox-1 standard was not included in this study. We also assume that the feedback link round-trip delay (RTD) is known to both the sender and receiver. In the Go-Back-N protocol, we assume the value of N is chosen such that N is greater than the integer quotient of RTD divided by the time it takes for the encoder to transmit a V3TF.

The key performance metrics being considered in this article are BER, frame error rate (FER), and throughput. For the sake of exposition, we refer to the Raptor schemes presented in Sections IV.B.1 and IV.B.2 as Approach 1 and Approach 2, respectively.

A. BER

Figure 7 shows a plot of the BER versus the E_b/N_o for both the Prox-1 standard and the Raptor addition (Approach 1). For the Prox-1 LDPC, the curve is for a blocklength of $K = 1024$ while for the Raptor scheme, the BER curves for four different blocklengths are shown. For every E_b/N_o point, enough K-bit message blocks are transmitted over the communication system such that the number of codeword errors obtained is at least 100. The degree distribution $\Omega(x)$ presented in Equation (3) has been optimized for $R = 1/2$ via the density evolution. Hence, the BER curve of the Raptor scheme for the $K = 1064$ case closely follows the Prox-1 LDPC curve. Jointly optimizing the inner and outer parts of the Raptor codes may produce better BER curves relative to the ones shown in Figure 7.

Figure 8 shows a similar plot of the BER for the Raptor Approach 2 scheme. The degrees of the parity bits are chosen from the set $d_i = \{12, 6, 4, 2\}$ in descending order. The Approach 2 assignment of parity degrees performs better in the low SNR cases while the same descending order approach has BER loss as the SNR increases.

B. FER

Figure 9 shows a plot of the FER versus the E_b/N_o for both the Prox-1 standard and the Raptor addition (Approach 1). The Prox-1 LDPC code has been designed for $R = 2/5$ and after puncturing the $R = 2/5$ codeword, the $R = 1/2$ codeword is transmitted. However, the degree distribution $\Omega(x)$ in Equation (3) has been optimized for BER, not FER. Since the FER curve measures the number of codeword errors, the Prox-1 LDPC design has an advantage in this metric over the Raptor scheme. Given that there is no universal degree distribution for Raptor codes on the AWGN channel, i.e., the optimal degree distribution depends on the SNR, $\Omega(x)$ presented in Equation (3) produces acceptable FER curves. Figure 10 shows a similar plot of the FER versus the E_b/N_o for the Raptor Approach 2 scheme. The performance difference between the Prox-1 LDPC and Approach 2 can be explained by similar insights as discussed for Figure 9. The Approach 1 is BER optimized for code rate $R = 1/2$, whereas the Approach 2 is not optimized for any code rate. However, the key benefit of Approach 2 will be clear in the following section focused on throughput.

C. Throughput

As we mentioned in Section I, our goal is to explore a Raptor coding approach to enhance the throughput of a proximity link relative to the throughput offered by the current Prox-1 standard. Although the BER and FER curves document the performance, the complete picture or *system view* of V3TFs transmission from sender to receiver is captured by the throughput plot [1].

To simulate the throughput performance of the proximity link communication system

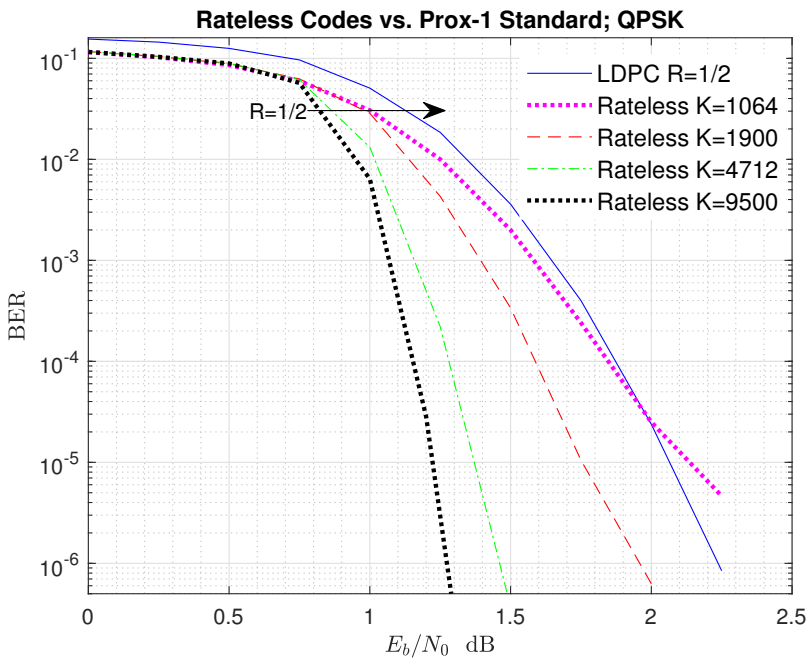

Figure 7. BER comparison between Raptor codes (Approach 1) and Prox-1 LDPC. The R=1/2 arrow indicates that all the curves have code rate 1/2. Note that the curves have varying information blocklengths K.

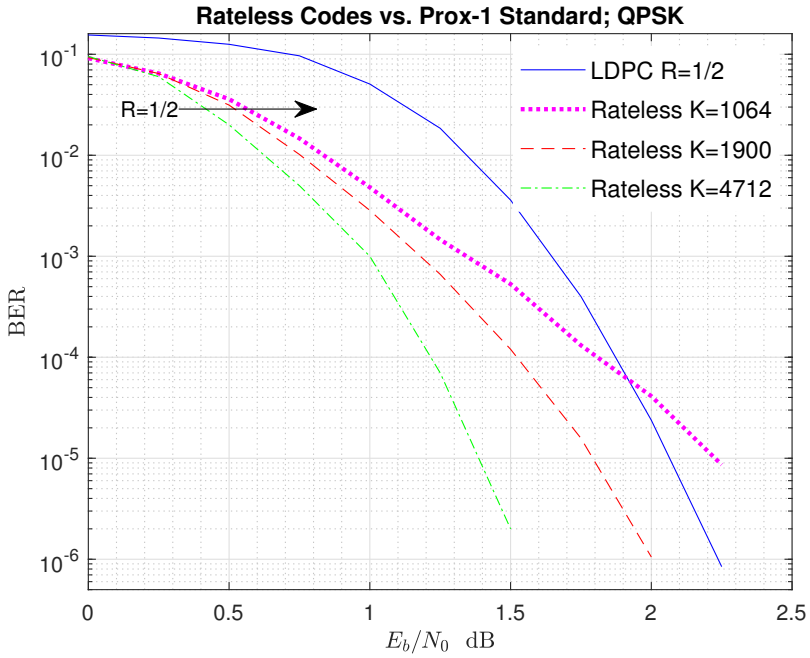

Figure 8. BER comparison between Raptor codes (Approach 2) and Prox-1 LDPC.

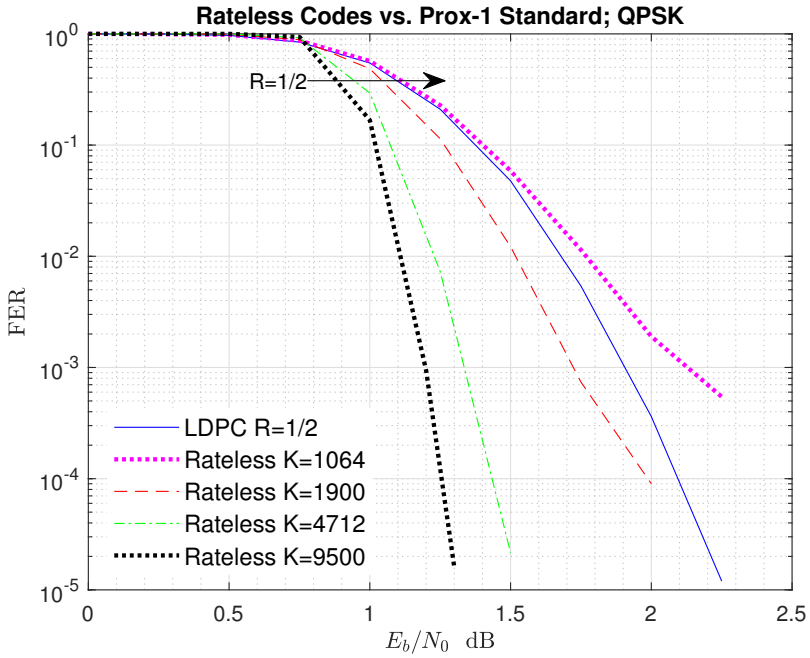

Figure 9. FER comparison between Raptor codes (Approach 1) and Prox-1 LDPC.

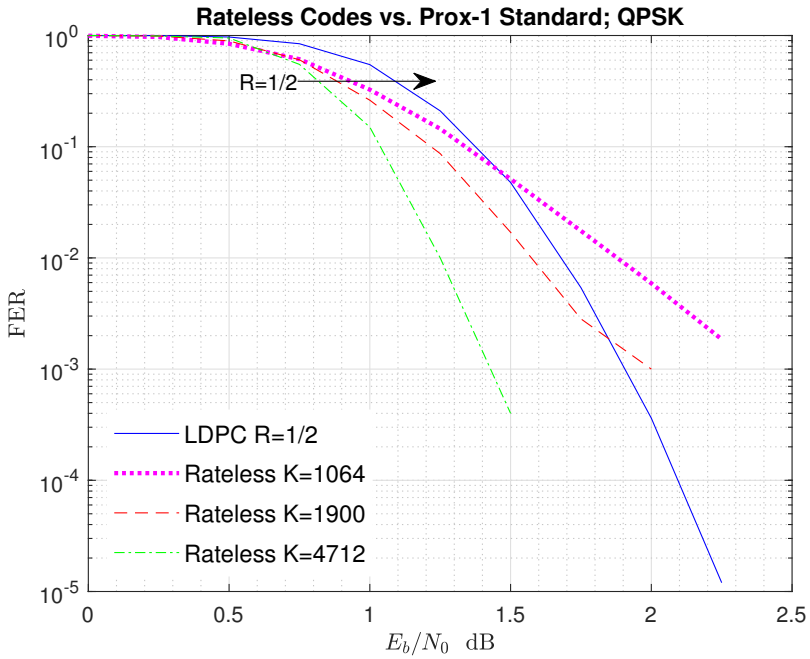

Figure 10. FER comparison between Raptor codes (Approach 2) and Prox-1 LDPC.

under both the Prox-1 standard and the Raptor addition, K-bit message blocks are transmitted from sender to receiver until successful decoding is achieved. The throughput performance is evaluated at a wide range of SNRs. Note that the previous BER and FER curves were plotted as a function of E_b/N_o, which is given by (see [5])

$$\frac{E_b}{N_o} = \frac{E_s}{N_o}\frac{1}{R\log_2 M} = \frac{SNR}{R\log_2 M}, \tag{13}$$

where M is the quadrature amplitude modulation (QAM) constellation size. For the BER and FER curves, the coding rate was fixed at $R = 1/2$ and hence, the E_b/N_o is the independent variable of interest. However, for the throughput curves, K-bit message blocks are transmitted until successful decoding with each K-bit message block potentially having a variable coding rate R. Hence from Equation (13), it is clear that E_b/N_o is also variable. Thus for the throughput curves, the SNR is the independent variable of interest.

For each SNR point, close to 1000 K-bit message blocks are transmitted until successful decoding. Let T denote the transmission time for the K-bit message block. T is the number of channel uses, i.e., QPSK symbols for transmitting a K-bit message block. Note that T is a random variable (RV) and its value depends on the SNR of the AWGN channel. For each SNR point, the throughput is calculated as $\eta = \mathbb{E}[\frac{K}{T}]$. The throughput η is computed by a long run time-average of $\frac{K}{T}$.

For the Prox-1 LDPC, $K = 1024$ bit message blocks were transmitted and for the Raptor scheme, $K = 1900$ bit message blocks were transmitted. Figure 11 shows a plot of the throughput η for both the Prox-1 standard and the Raptor addition. We observe that from a throughput standpoint, the optimal E_b/N_o for the Prox-1 standard curve is 2 dB.

Let N_{cb} denote the number of parity bits per codeblock. In this article, a codeblock refers to the group/set of parity bits the decoder needs to receive in order to perform a decoding attempt. For example, the decoder performs decoding attempt #1 when it receives the codeblock #1, attempt #2 for codeblock #2, and so on. N_{cb} refers to the parity resolution, i.e., number of parity bits output by the encoder at a time. We evaluate the Raptor Approach 1 for three values of $N_{cb} = \{450, 250, 150\}$. Note that the value of N_{cb} is determined by the complexity/costs consideration for the decoder at the receiver. An extreme value of $N_{cb} = 1$ provides high throughput resolution albeit with very high decoder complexity/costs since the BP decoding is restarted for each new code bit.

As before, N is the number of intermediate code bits output by the $R = 0.95$ LDPC outer code. Since $K = 1900$, for the $R = 0.95$ LDPC outer code we have $N = 2000$. For the Raptor Approach 2 scheme, the parity bits are chosen from the universe of $L_1 = 540$ degree $d_1 = 12$ parity bits, $L_2 = 292$ degree $d_2 = 6$ parity bits, $L_3 = 2780$ degree $d_3 = 4$ parity bits, and $L_4 = \infty$ degree $d_4 = 2$ parity bits on an as-need basis. For the Approach 2 scheme, the value of N_{cb} was chosen heuristically within each parity group. Optimizing the N_{cb} value for the Approach 2 might enhance the system throughput.

1. Low SNR Throughput

Figure 12 shows a plot of the throughput η in the low SNR regime for both the Prox-1 LDPC and the Raptor schemes. In the low SNR regime, the Prox-1 standard has near-zero throughput while the Raptor addition has good throughput ensuring the transfer of a substantial number of message blocks even in the presence of intermittent links.

2. High SNR Throughput

In the high SNR regime also, there is a throughput gain of the Raptor addition over the Prox-1 LDPC. Specifically, we observe the enhancements in throughput performance on the order of 30% to 90% relative to the current Prox-1 over varying SNR scenarios. For example, a 30% gain at 2 dB above optimal SNR and a 90% gain at 8 dB above optimal SNR.

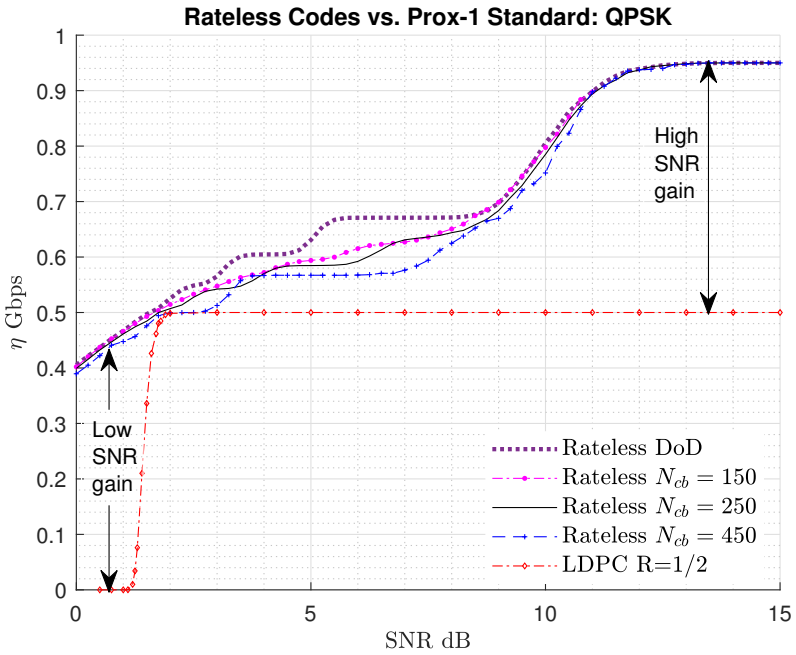

Figure 11. A plot of throughput $\eta = \mathbb{E}[\frac{K}{T}]$ versus SNR. A 1 GHz clock is assumed to calculate the throughput η. Due to QPSK modulation, the throughput η is normalized by a factor of 2 to make the throughput comparison independent of the QAM constellation size.

VII. Computational Costs

In this section, we aim to quantify the computational costs associated with the encoding and decoding of V3TFs. From a processor/FPGA perspective, the computational costs translate to the speed of processing and power consumption of encoders and decoders. The goal is to compare the computational costs associated with the Raptor processing

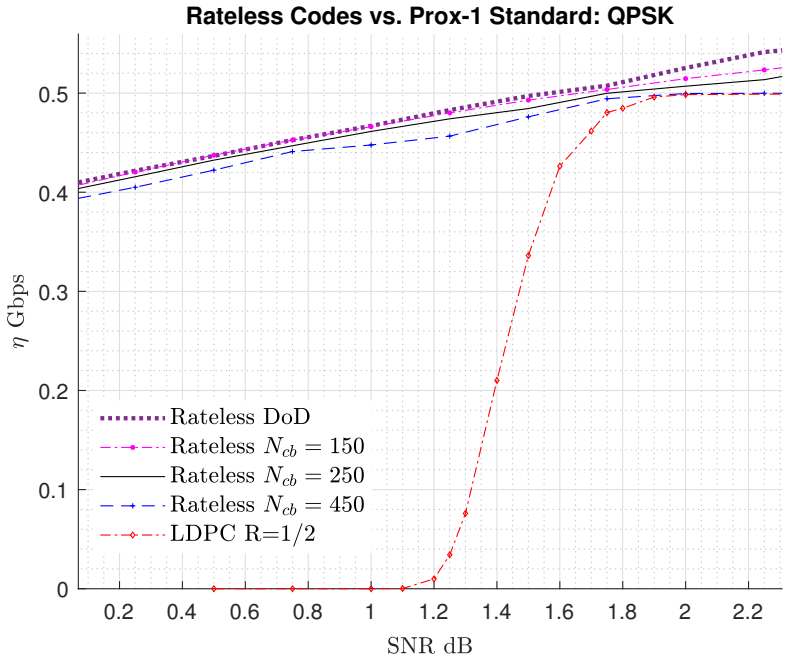

Figure 12. Throughput η comparison in the low SNR regime. A 1 GHz clock is assumed to calculate the throughput η.

vs. the standard Prox-1 LDPC code processing. The analysis will give insights into the speed of processing and the power consumption to process M number of V3TFs or their codewords for typical SNRs.

A. Encoder Comparison

Now, we quantify the computational costs associated with the two encoder types.

1. LDPC Encoder

For the Prox-1 LDPC case, the codeword c is given by

$$c = bG_1, \tag{14}$$

where G_1 is the generator matrix with dimensions $\dim G_1 = 1024 * 2560$. The generator matrix G_1 has the structure

$$G_1 \triangleq [I|Q_1], \tag{15}$$

where the sub-matrix Q_1 is a *non-sparse matrix* with $\dim Q_1 = 1024 * 1536$ and column weights w_c satisfying

$$501 \leq w_c \leq 537. \tag{16}$$

2. Raptor Encoder

The blocklength K of the information vector b for the Raptor case is fixed to $K = 1064$. The codeword c is given by

$$c = bG_oG_{lt}. \tag{17}$$

The outer code generator matrix G_o has dimensions $\dim G_o = 1064 * 1120$ and the structure $G_o \triangleq [I|Q_o]$, where $\underline{\dim Q_o = 1064 * 56}$.

The LT code generator matrix G_{lt} has dimensions $\dim G_{lt} = 1120*(1120 + N_{lt})$ and the structure $G_{lt} \triangleq [I|Q_{lt}]$. Note that Q_{lt} is a *low density (sparse) matrix* with dimensions $\dim Q_{lt} = 1120 * N_{lt}$. The column weights w_c of matrix Q_{lt} satisfy

$$\begin{cases} 2 \leq w_c \leq 9, & \textit{Approach } 1 \\ 2 \leq w_c \leq 12, & \textit{Approach } 2. \end{cases} \tag{18}$$

Approach 1 can have a column weight of 60. But this event has very small probability as per the degree distribution in Equation (3).

From the preceding discussion, the main component of the operations to generate codewords is the multiplication of the information vector b with matrices Q_1, Q_o, and Q_{lt}. From (16), (18), and also by comparing the dimensions of Q_1, Q_o, and Q_{lt}, it is clear that the computational costs, i.e., number of XOR operations required for the Prox-1 LDPC encoding of V3TFs, is very high relative to the costs required for the Raptor encoding of V3TFs.

Let "CPL" be an acronym for denoting the costs, power consumption and latency for the encoding and decoding of V3TFs. Our analysis clearly shows that the Prox-1 LDPC encoder has higher CPL than the Raptor encoder. Also, the CPL scales for multiple encoding/transmission rounds.

B. Decoder Operations

In this section, we quantify the computational costs associated with the decoders. Table 2 quantifies the number of operations required for a *single iteration* of the SPA for the Prox-1 LDPC decoder and the outer LDPC and LT decoders of the Raptor addition.

In Table 2, we focus on the second column as the benchmark since it corresponds to the Prox-1 standard. We observe that the number of operations for the Approach 1 LT encoder is higher relative to the operations for the Prox-1 standard. It is also observed that the Approach 2 LT encoder has fewer costs relative to the Prox-1 standard. Note that the number of operations tabulated in Table 2 is for a single iteration of the SPA. The computational costs, i.e., the total number of operations required to decode M number of V3TFs or their codewords, depends on the number of iterations and the operating SNR. However, the calculations in Table 2 offer basic insights into the speed of processing and the power consumption for decoding of V3TFs.

Table 2. Number of operations required for a single iteration of the SPA. $\mathbf{Nadd_{vc}}$ captures the number of additions in (6) and (10). The operations in (7)-(12) are similarly captured.

Property	Prox-1 LDPC	Outer LDPC	Approach 1	Approach 2
Blocklength K	1024	1064	1120	1120
$Nadd_{vc}$	17290	10080	22466	15568
$Nmul_{cv}$	7680	4480	11793	8344
$Ndiv_{cv}$	7680	4480	10673	7224
$Natan_{cv}$	7680	4480	10673	7224
$Nadd_{v}$	10240	5600	11793	8344
Retransmissions	*Linear* scaling	N/A	Incremental	Incremental

The Prox-1 standard performs brand new decoding of the K-bit message block for every retransmission. Hence in Table 2, the number of operations tabulated in the second column scales *linearly* with the number of retransmissions. On the other hand, the Raptor addition only sends incremental parity symbols N_{lt} for decoding the K-bit message block. In the SPA for decoding LT codes as per Equations (6)-(8), the computational costs increase linearly with N_{lt} only in Equation (7) whereas the costs in Equations (6) and (8) increase linearly with K, which is fixed here. Hence in the case of Raptor decoding with retransmission, the computational costs increase linearly with the number of parity symbols N_{lt} only in the message passing calculations from check-to-variable nodes. It is fair to say that the overall costs of the SPA is only *incrementally* increasing. In low SNR, the Prox-1 standard has higher CPL due to retransmissions. In high SNR scenarios with only one transmission round, the CPL of the Raptor addition may be higher or lower relative to the Prox-1 standard depending on whether the Approach 1 or Approach 2 LT encoder is being used.

VIII. Deep Learning Aided BP Decoder Enhancements

In this section, we introduce the concept of adding deep learning-based enhancements to the BP decoder. At a certain CPL cost, we illustrate that the BP decoder enhancements provide a benefit that would outweigh the cost. The BP algorithm is used in decoding linear block codes such as LDPC, BCH, Polar and Raptor codes. BP is used on a Tanner graph or decoding graph. BP is suboptimal due to the presence of loops on the Tanner graph. In Reference [7], deep learning-based solutions have been proposed to enhance the BP algorithm.

The focus is on a solution that is based on both *subject knowledge* and *deep learning* tools. One approach involves the scaling of the channel LLRs to compensate for the performance degradation due to loops. In this section, we focus on a decoder enhancement termed *neural BP*, where the weights are assigned to the edges of the Tanner graph or decoding graph. The neural network (NN) is composed of an input layer, output layer, and many hidden layers. The variable-to-check node messages on the Tanner graph are

represented by hidden layer i on the NN. Similarly, the check-to-variable node messages on the Tanner graph are represented by hidden layer $i+1$ on the NN.

The SPA for the neural BP decoder is provided.

1. Variable nodes:
$$L_{v_i c_j} = tanh\Big[\frac{1}{2}\big(w_i^n L_{v_i} + \sum_{N_v(i)\backslash j} w_{j',i}^n L_{c_{j'} v_i}\big)\Big]. \tag{19}$$

2. Check nodes:
$$L_{c_j v_i} = 2\ atanh\Big[\underbrace{tanh\left(\frac{w_j^n L_{c_j}}{2}\right)}_{only\ for\ LT} \prod_{N_c(j)\backslash i} L_{v_{i'} c_j}\Big]. \tag{20}$$

3. Output:
$$o_i = \sigma\Big(w_i^{2L} L_{v_i} + \sum_{N_v(i)} w_{j',i}^{2L} L_{c_{j'} v_i}\Big), \tag{21}$$
where $\sigma(x) = 1/(1+e^{-x})$.

The weight $w_{j,i}^n$ applies to the edge between c_j and v_i in hidden layer n. Note that the above SPA applies to both LT and LDPC code decoders with the key differences between LT and LDPC codes as described in Sections V.A and V.B still applicable.

Min Sum Algorithm: The Min-Sum version of the above SPA can be obtained by replacing the check node message computations in Equation (20) with the following relevant equations.

- Normalized version:
$$L_{c_j v_i} = w_{j,i}^n \cdot \min_{N_c(j)\backslash i} L_{v_{i'} c_j} \cdot \prod_{N_c(j)\backslash i} sign(L_{v_{i'} c_j}). \tag{22}$$

- Offset version:
$$L_{c_j v_i} = \big(\min_{N_c(j)\backslash i} L_{v_{i'} c_j} - \beta_{j,i}^n\big)^+ \cdot \prod_{N_c(j)\backslash i} sign(L_{v_{i'} c_j}), \tag{23}$$
where $(a)^+ = \max(a, 0)$.

Note that in the case of a recurrent neural network (RNN), the weights are set to be equal for all hidden layers, i.e., $w_{j,i}^n$ or $\beta_{j,i}^n$ are the same for all layers n. A training database is used to learn the weights. The training database consists of noisy versions of the all-zero codeword.

The objective for the training process is cross-entropy minimization
$$H_{cr}(o, y) = -\frac{1}{N_v} \sum_{i=1}^{N_v} y_i \log o_i + (1 - y_i) \log(1 - o_i), \tag{24}$$

where N_v is the number of variable nodes.

The goal is to obtain a Tensorflow and Google Colab (see [12]) implementation of the neural BP decoder outlined in Equations (19)-(21). Initially for short blocklength $K = 100$, neural BP decoder designs have been obtained. Figures 13 and 14 show plots of the BER curves obtained with the neural BP decoder for the rate $R = 1/2$ LDPC code. The goal in the follow-on project would be to extend this design to the blocklength $K = 1064$ for LT codes to obtain a 0.5-1 dB improvement. The extent of the benefit from deep learning tools depends on the depth of subject knowledge and the use of relevant neural network architectures. The key features of the new technology such as energy efficiency, lower computational costs, and low-latency can be enhanced through the use of deep learning tools in the new design.

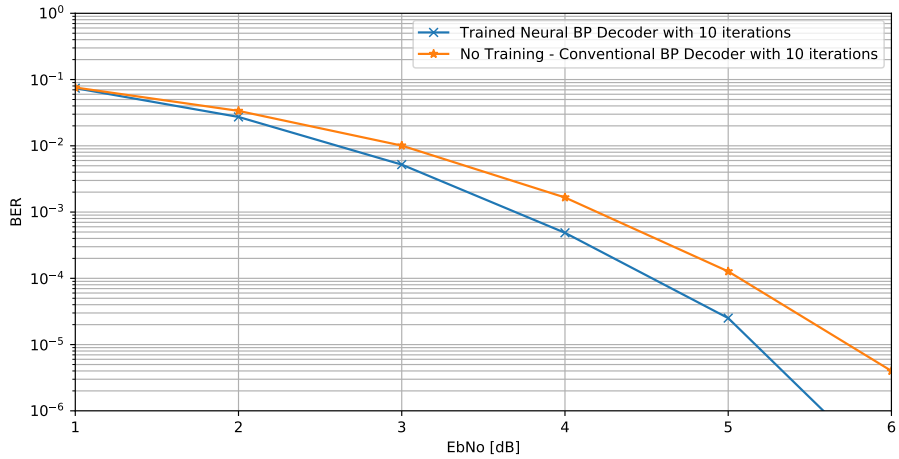

Figure 13. BER comparison between the neural BP and traditional BP decoders for R=1/2 LDPC code. The neural BP decoder uses an RNN. The figure was obtained using Google Colab and appears different from the previous figures, which were obtained using Matlab.

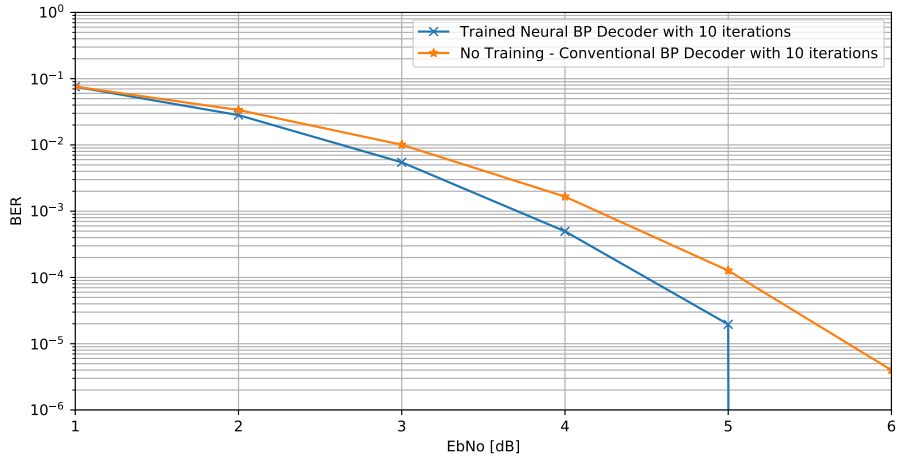

Figure 14. BER comparison between the neural BP and traditional BP decoders for R=1/2 LDPC code. The neural BP decoder uses a feed forward neural network.

IX. Conclusions

In this article, we presented an approach for using rateless codes over the AWGN channel as an additional option for the Prox-1 standard to increase the throughput of proximity space links. We discussed two types of Raptor coding schemes that can be used in the C&S sublayer of the DLL of proximity link communication system. One approach is based on degree distributions and the other is based on descending order degrees. We show that over a wide range of SNR, the Raptor coding schemes lead to an enhanced throughput over the current Prox-1 LDPC code option. For example, at 3 dB above the Prox-1 optimal SNR, the throughput gain is 30%. An initial comparison also shows that the Raptor addition has a computational gain in terms of the processing speed and the power consumption of encoders and decoders. The takeaway messages are summarized in Table 3.

Table 3. Throughput and CPL comparison between the Prox-1 standard and the Raptor addition. LDGM stands for low density generator matrix.

Prox-1 Standard:

Regime	Throughput	Encoder CPL	Decoder CPL
Low SNR	Near-zero	***High***: Dense matrix multiplication	***High***: Due to retransmissions
High SNR	Good	***High***: Dense matrix multiplication	***Moderate***

Raptor Addition:

Regime	Throughput	Encoder CPL	Decoder CPL
Low SNR	Good	***Low***: LDGM multiplication	***Incremental***: Fresh decoding with new parity
High SNR	30-90% gain over Prox-1	***Low***: LDGM multiplication	*Approach 1*: ***High*** *Approach 2*: ***Moderate***

We also provide preliminary results on a deep learning-based neural BP decoder, which can further enhance the throughput of the proposed new design. For future direction in terms of the CCSDS standards path or technology demonstration payload, issues that need to be addressed are:

- Rateless Protocol: Integrating the Raptor encoder and decoder designs into the layered protocol stack of the Prox-1 standard and addressing the necessary changes in the DLL or higher layers.
- Practical feedback link issues such as “factoring in long feedback delay”, “errors on the feedback link” and solutions for them. To fix any large RTD in proximity links, one solution is to multiplex L codewords.

Acknowledgments

We thank J. Leigh Torgerson for sharing his expertise through the valuable review comments. Amogh thanks the Communications Architectures and Research Section of NASA JPL for hosting him during the course of the project. We express our gratitude to the Engineering and Physical Sciences Research Council (EPSRC), UK and University of Bristol, UK for the grant supporting Amogh's work. The authors would also like to thank Peter Shames and Greg Kazz for providing important corrections to this paper.

References

[1] J. L. Torgerson, G. J. Miles, N. J. Richard, and H. Xie, "Adaptive Control of a Variable Coded Modulation Radio Using Delay Tolerant Networking Licklider Transmission Protocol Statistics," *The Interplanetary Network Progress Report*, vol. 42-217, Jet Propulsion Laboratory, Pasadena, California, May 15, 2019.
http://ipnpr.jpl.nasa.gov/progress_report/42-217/217A.pdf

[2] CCSDS 211.0-B-5, "Proximity-1 Space Link Protocol- Data Link Layer," Blue Book, Issue 5, December 2013. https://public.ccsds.org/Pubs/211x0b5s.pdf

[3] CCSDS 211.2-B-3, "Proximity-1 Space Link Protocol- Coding and Synchronization Sublayer," Blue Book, Issue 3, October 2019.

[4] U. Madhow, *Fundamentals of Digital Communication*, Cambridge University Press, First edition, 2008.

[5] D. Tse and P. Vishawanth, *Fundamentals of Wireless Communication*, Cambridge University Press, First edition, 2005.

[6] CCSDS 131.0-B-3, "TM Synchronization and Channel Coding," Blue Book, Issue 3, September 2017. https://public.ccsds.org/Pubs/131x0b3e1.pdf

[7] E. Nachmani, E. Marciano, L. Lugosch, W. Gross, D. Burshtein and Y. Be'ery, "Deep Learning Methods for Improved Decoding of Linear Codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, February 2018.

[8] A. Kharel and L. Cao, "Analysis and Design of Physical Layer Raptor Codes," *IEEE Communications Letters*, vol. 22, no. 3, pp. 450–453, March 2018.

[9] W. Yu, K. Narayanan, J. Cheng and J. Wu, "Raptor Codes with Descending Order Degrees for AWGN Channels," *IEEE Communications Letters*, vol. 24, no. 1, pp. 29–33, January 2020.

[10] R. Abbas, M. Shirvanimoghaddam, T. Huang, Y. Li and B. Vucetic, "Novel Design for Short Analog Fountain Codes," *IEEE Communications Letters*, vol. 23, no. 8, pp. 1306–1309, August 2019.

[11] K. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. 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, November 2007.

[12] J. Hoydis "Neural network BP decoder,"
https://colab.research.google.com/drive/1T8qP0fdXQjm7wH7qaBLuSFwHXmtT4VkE.