Protograph-Based Raptor-Like LDPC Codes for Rate Compatibility with Short Blocklengths

Tsung-Yi Chen
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90024
Email: tychen@ee.ucla.edu

Dariush Divsalar
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109
Email: Dariush.Divsalar@jpl.nasa.gov

Jiadong Wang and Richard D. Wesel
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90024
Email: {wjd, wesel}@ee.ucla.edu

***Abstract*—This paper presents a new class of rate-compatible LDPC codes, protograph-based Raptor-like (PBRL) codes. The proposed PBRL codes are jointly decodable with an iterative belief propagation decoder. As with Raptor codes, additional parity bits can be easily produced by exclusive-or operations on the precoded bits, providing extensive rate compatibility. This paper provides a design procedure that optimizes this class of rate-compatible LDPC codes. The new PBRL codes outperform 3GPP rate-compatible turbo codes with the same short blocklength at high SNR and show no sign of an error floor at the FER region of 10^{-7}.**

I. Introduction

Low-density parity-check (LDPC) codes are a prominent class of error correcting codes. They were proposed by Gallager [1] in the early 1960s but did not receive much attention until decades later [2]. LDPC codes have a sparse parity check matrix and are decoded efficiently by the iterative belief propagation (BP) algorithm. (See, for example, [3].) In recent years, a new class of LDPC codes was introduced by Thorpe [4] and studied extensively in [5] and [6]. These protograph-based LDPC codes (protograph codes) use a relatively small graph (the protograph) that is replicated many times. This structure allows efficient decoder implementation in hardware.

Introduced by Luby [7] and Shokrollahi [8], LT codes and Raptor codes share many similarities with LDPC codes and are shown to achieve the capacity of the binary erasure channel (BEC) universally. Etesami et al. [9] explore the application of Raptor codes to binary memoryless symmetric channels and derive various results on the output degree distribution of LT codes. Results on Raptor codes such as [8] and [9] rely heavily on the assumption of large information blocks.

Rate-compatible punctured codes are widely used in incremental redundancy (IR) schemes, including rate-compatible punctured convolutional (RCPC) codes and rate-compatible turbo (RCPT) codes. These code families use a good "mother" code at the lowest rate and obtain the higher-rate codes by puncturing. One must carefully choose the puncturing patterns to avoid undue performance degradation as the rate increases. One drawback of rate-compatible puncturing is the difficulty of optimization over a large number of possible puncturing patterns.

Recent work [10]–[12] using a sphere-packing analysis to investigate the potential performance of feedback with incremental redundancy indicates that feedback with IR can allow short blocklength codes to achieve high throughput (approaching capacity) with surprisingly low latency. RCPC codes perform close to the sphere-packing analysis until the latency exceeds the effective traceback of the convolutional code. In general, the strength of convolutional codes scales with the number of states in the trellis rather than the code length. Hence it is of interest to find rate-compatible LDPC codes whose performance improves with blocklength in the short blocklength regime.

This paper proposes two schemes, protograph-based Raptor-like (PBRL) codes that perform well in the short-blocklength regime and punctured-node protograph-based Raptor-like (PN-PBRL) codes that achieve improved thresholds by introducing punctured nodes. Similar to Raptor codes, the PBRL and PN-PBRL codes are rate-compatible and lend themselves to the IR application. Unlike rate-compatible punctured LDPC codes, PBRL and PN-PBRL codes do not puncture a mother code. Rather, they encode a precode and generate additional parity bits by exclusive-or operations on the precoded symbols.

Some similar construction techniques for finding rate-compatible LDPC codes with fixed information blocks are available in the literature. See, for example, [13]–[15], for the technique called extending. The main differences between extending and the construction presented in this paper are as follows: First, the structural design is based on a protograph and can hence be easily optimized in two stages, the protograph design and the design of the permutations used during the lifting process. Second, an additional degree-one variable node attached to each new check node allows efficient encoding of the incremental redundancy.

The paper is organized as follows: Section II reviews the preliminaries of LT codes and Raptor codes. Section III reviews the structure of protograph-based LDPC codes and introduces the construction of PBRL codes. Section IV gives the optimization method of PBRL codes, and Section V provides the construction of PN-PBRL codes. Simulation results comparing PBRL and PN-PBRL codes to 3GPP turbo codes

This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. This research was supported by a gift from the Broadcom Foundation. Dr. Wesel has consulted for the Broadcom Corporation on matters unrelated to this research.

are provided in Section VI. Finally, Section VII concludes the paper.

II. Raptor Codes

This section reviews the preliminaries of Luby transform (LT) codes [7] and Raptor codes [8]. An LT code is described by the output degree distribution $\mathbf{\Omega}$ on its output symbols. Let n be a positive integer that denotes the number of the input symbols. Let $\mathbf{\Omega} = [\Omega_1, \Omega_2, \ldots, \Omega_n]$ be a distribution on a set of integers $\{1, 2, \ldots, n\}$ such that Ω_j denotes the probability of the value j being chosen. For the ith output bit, the encoder first chooses an integer d_i randomly according to the distribution $\mathbf{\Omega}$. It then chooses d_i input symbols uniformly (without replacement) from $\{1, 2, \ldots, n\}$, and taking exclusive-or of these chosen input bits yields the output bit. This encoding process continues indefinitely for $i = 1, 2, \ldots$, often concluding only when all interested receivers have been able to decode the message.

Let C be an (n, k) linear block code. A Raptor code is a serial concatenation of a code C, which is also called the "precode," and an LT code. A Raptor code is described by the parameters $(k, C, \Omega(x))$, where $\Omega(x) = \sum_{i=k}^{n} \Omega_i x^i$ is the generator polynomial of the output degree distribution of the LT code.

The decoding of the Raptor code is performed in two-stages: the decoder first decodes the LT code and recovers a fraction of the precoded symbols (or provides the soft information of the precoded symbols in the case of AWGN channel). The decoder then attempts to recover the remaining symbols by decoding the precoded symbols with the precode C.

III. Protograph-Based Raptor-Like LDPC Code

This section reviews the structure of a protograph-based LDPC code and introduces the protograph-based Raptor-like (PBRL) LDPC codes. We will refer this family of codes as PBRL codes for the rest of the paper.

A protograph-based LDPC code is constructed by a "copy-and-permute" operation (also called "lifting") from a Tanner graph with a relatively small number of nodes. The lifting operation first makes N copies of the protograph and then the edges of the same type among the protograph replicas are permuted.

Fig. 1 shows the protograph of a PBRL code. This protograph consists of two parts: (1) a relatively simple protograph code (on the left) representing the protograph of the precode and (2) a number of check nodes (on the right) that are each connected to several variable nodes of the first part and an additional degree-one variable node. The second part represents the protograph of an LT code.

After the lifting operation, the first part can be seen as an LDPC precode, and the degree-one variable nodes of the second part can be efficiently encoded with the precoded symbols in a manner similar to the LT code. The structure of this protograph code resembles a Raptor code, but with a deterministic (rather than random) encoding rule for combining the precoded symbols. The rate of the precode in this example is $3/4$. As we increase the number of transmitted degree-one variable nodes in the LT part, the code rate is reduced gradually.

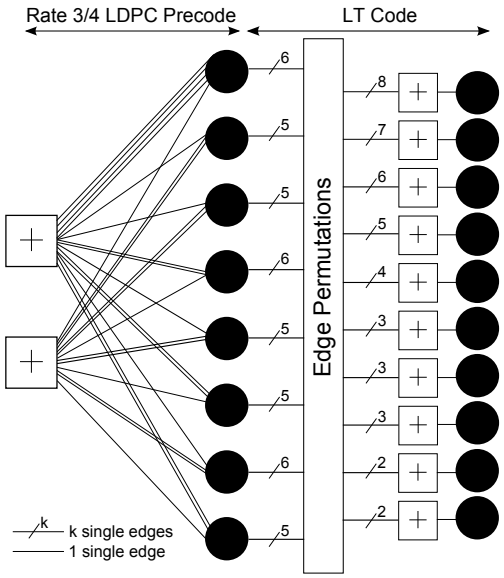

Fig. 1. Protograph for a PBRL code with a rate-$3/4$ precode. Subsequent lower-rate codes are obtained by transmitting the variable nodes in the LT code protograph starting from the top node. The matrices shown in (1) and (2) give the details of the edge connections as described in Section VI .

Consider the decoding of a traditional Raptor code that collects the precoded symbols and encodes them with an LT code. In the case of an LDPC precode used with an LT code, decoding proceeds as follows: The decoder first performs BP decoding on the LT code. Then the decoder performs BP decoding on the precode. The two-stage decoding implies the use of two different BP decoders, each exchanging their extrinsic information after the iterative decoding.

In [16], the authors comment that the complexity of the Raptor codes is higher than rate-compatible LDPC codes. In view of reducing system complexity, it is natural to consider a joint decoding of the Raptor code. The PBRL code family always transmits the output symbols of the precode, allowing joint decoding of the LT code and the LDPC precode. This property also guarantees that the BP algorithm will always work for the initial transmission as well as the lower-rate codewords comprised of the original transmission and additional incremental redundancy. For traditional Raptor codes that use randomized encoding, the initial transmission may not contain enough information for BP decoding to succeed even in a noiseless setting.

IV. Optimization of Protograph-Based Raptor-Like LDPC Codes

This section proposes a design technique for finding good PBRL LDPC codes. Belief propagation (BP) decoding is assumed and we begin by designing the protograph. Given a fixed initial code rate, the design begins by finding a good protograph code to serve as the precode and then optimize the protograph of the LT code part. Optimization of the precode

$$H_p = \begin{bmatrix} \sigma^0+\sigma^1+\sigma^3+\sigma^7 & \sigma^{24} & \sigma^{14} & \sigma^{17}+\sigma^0 & \sigma^7 & \sigma^1+\sigma^6 & \sigma^{21} & \sigma^{21}+\sigma^0 \\ \sigma^4 & \sigma^4+\sigma^9 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^2 & \sigma^0 & \sigma^0+\sigma^3 & \sigma^2 \end{bmatrix}. \quad (1)$$

$$H_{LT} = \begin{bmatrix} \sigma^{29} & \sigma^0 & \sigma^0 & \sigma^1 & \sigma^5 & \sigma^6 & \sigma^{10} & \sigma^4 \\ \sigma^{12} & \sigma^0 & \sigma^1 & \sigma^3 & \sigma^4 & \sigma^{16} & \sigma^{13} & 0 \\ \sigma^{16} & \sigma^0 & \sigma^2 & \sigma^6 & \sigma^0 & 0 & 0 & \sigma^1 \\ \sigma^{26} & 0 & \sigma^0 & 0 & 0 & \sigma^1 & \sigma^6 & \sigma^9 \\ 0 & \sigma^1 & 0 & 0 & \sigma^0 & 0 & \sigma^2 & \sigma^3 \\ \sigma^1 & 0 & 0 & \sigma^2 & 0 & \sigma^9 & 0 & 0 \\ 0 & 0 & \sigma^{16} & 0 & \sigma^0 & 0 & \sigma^4 & 0 \\ 0 & \sigma^{21} & 0 & \sigma^0 & 0 & \sigma^2 & 0 & 0 \\ \sigma^0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^1 \\ 0 & 0 & 0 & \sigma^{12} & 0 & 0 & \sigma^0 & 0 \end{bmatrix}. \quad (2)$$

protograph is omitted due to space limitations. See [6] for an extensive study on finding good protograph codes. The selection of the precode protograph is based on a greedy search of low-threshold high-rate protographs with computer simulations used to make a final selection from the best candidates.

To construct the protograph of the LT code part, first add a new check node and a new degree-1 variable node to the protograph. Connect the new check node and the new degree-1 variable node with an edge. Additional edges are added between the new check node and the precoded variable nodes according to the degree that will optimize the density evolution threshold. This process continues until the underlying protograph reaches the lowest rate desired.

For a given precode protograph, the optimization procedure of the LT code part is summarized as follows:

1) Add a new check and a new variable node that are connected to the protograph.
2) Perform density evolution on the new protograph to determine the optimal degree distribution and the connections of the new check node to the precoded symbols.
3) Start over with step 1) if the lowest rate desired is not yet reached.
4) Select circulant permutations so that small cycles are avoided when the protograph is lifted. (Based on the circulant progressive edge growth algorithm [17]).
5) Lift the resulting protograph with selected circulant permutations to match the desired initial blocklength.

Note that in the optimization process parallel edges in the *LT code* part of the protograph are kept to a minimum (at most one pair of parallel edges in our examples). This prevents short-cycles in the lifting process. Fig. 1 is an example of an optimized PBRL code. This code *does not* have any parallel edges in the LT code part. Experimental results indicate that for PBRL codes with short blocklengths, direct lifting of the protograph with parallel edges yields better codes than a two-stage lifting such as the one described in [6].

The initial code rate, or the precode code rate, is $3/4$. The threshold of the precode is 2.196 dB (E_b/N_0). Suppose that the code is lifted 32 times, the initial block length is then 256. With a step size of 32, subsequent code rates $6/9, 6/10, \ldots, 6/18$ are obtained by transmitting the output symbols of the LT code from each successive group of variable nodes starting from the top.

The corresponding thresholds of each code rate are summarized in Table I. We observe an increase in the gap between the threshold and the capacity as the code rate decreases. This is due to the structural restrictions imposed on the protograph of the LT code part. Each subsequent protograph inherits the connections of the next-higher-rate protograph; the new protograph can only optimize over the connections emanating from the one additional check node. Also, the new check node must connect with a new degree-one variable node.

TABLE I
THRESHOLDS OF THE PBRL CODES (E_b/N_0 IN DECIBELS).

Rate	Threshold	Capacity	Gap
$6/8$	2.196	1.626	0.570
$6/9$	1.804	1.059	0.745
$6/10$	1.600	0.679	0.921
$6/11$	1.464	0.401	1.063
$6/12$	1.358	0.187	1.171
$6/13$	1.250	0.018	1.232
$6/14$	1.136	-0.122	1.258
$6/15$	1.016	-0.238	1.254
$6/16$	0.922	-0.337	1.259
$6/17$	0.816	-0.422	1.238
$6/18$	0.720	-0.495	1.215

V. PUNCTURED-NODE PROTOGRAPH-BASED RAPTOR-LIKE LDPC CODE

This section introduces Punctured-Node Protograph-Based Raptor-Like (PN-PBRL) LDPC codes that have structure similar to PBRL LDPC code, but the protograph of the precode has at least one punctured (untransmitted) node. We will refer them as PN-PBRL codes for the rest of the paper.

Fig. 2 shows an example of an optimized PN-PBRL code. Note that the first variable node of the precode protograph is punctured, giving a rate-$6/7$ precode. To obtain an initial code rate of $3/4$, the first variable node of the LT code protograph is transmitted. The optimization procedure is the same as in Section IV.

$$H_p = \begin{bmatrix} \sigma^0+\sigma^1 & \sigma^{24} & \sigma^{14} & \sigma^{17}+\sigma^5 & \sigma^7 & \sigma^1+\sigma^3 & \sigma^{21} & \sigma^{21}+\sigma^0 \\ \sigma^4 & \sigma^0+\sigma^2 & \sigma^0+\sigma^3 & \sigma^{31} & \sigma^6+\sigma^0 & \sigma^1 & \sigma^0+\sigma^1 & \sigma^2 \end{bmatrix}. \quad (3)$$

$$H_{LT} = \begin{bmatrix} \sigma^2+\sigma^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma^{29} & \sigma^0 & \sigma^2 & \sigma^0 & \sigma^9 & \sigma^6 & \sigma^7 & \sigma^6 \\ \sigma^{12} & \sigma^0 & \sigma^4 & \sigma^1 & \sigma^5 & \sigma^4 & \sigma^{10} & 0 \\ \sigma^{16} & \sigma^0 & \sigma^5 & \sigma^6 & \sigma^1 & 0 & 0 & \sigma^{11} \\ \sigma^{26} & 0 & \sigma^0 & 0 & 0 & \sigma^2 & \sigma^9 & \sigma^0 \\ 0 & \sigma^1 & 0 & 0 & \sigma^0 & 0 & \sigma^3 & \sigma^0 \\ \sigma^1 & 0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & 0 \\ 0 & 0 & \sigma^{16} & 0 & \sigma^0 & 0 & \sigma^2 & \sigma^0 \\ 0 & \sigma^{21} & 0 & \sigma^0 & 0 & \sigma^1 & 0 & 0 \\ \sigma^0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^1 \\ 0 & 0 & 0 & \sigma^{12} & 0 & 0 & \sigma^0 & 0 \end{bmatrix}. \quad (4)$$

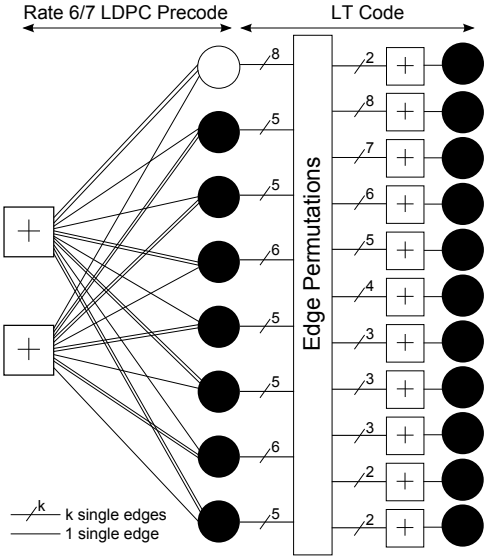

Fig. 2. Protograph of a PN-PBRL code with a rate-$6/7$ precode. The first node in the precode is always punctured. Lower-rate codes are obtained by transmitting the variable nodes in the LT code protograph starting from the top node. The matrices shown in (3) and (4) give the details of the edge connections as described in Section VI .

The subsequent code rates of $6/9, 6/10, \ldots, 6/18$ are obtained by transmitting the variable nodes of the LT code protograph from top to bottom. Regardless of the operating rate, the first variable node of the precode protograph is always punctured. The PN-PBRL codes yield better thresholds as shown in Table II.

Adding more parallel edges connected between the punctured variable node in the precode protograph and the check nodes in the LT code protograph reduces the threshold significantly, as shown in Table III. The gap between the threshold and the capacity are all less than 0.34 dB. The lifted codes with blocklength 256, however, do not manifest the gain obtained in threshold. This is because the large number of extra parallel edges is likely to cause undesirable trapping sets or absorbing sets in the decoding graph, especially when a short blocklength such as 256 is used.

TABLE II
THRESHOLDS OF THE PN-PBRL LDPC CODES (E_b/N_0 IN DECIBELS).

Rate	Threshold	Capacity	Gap
$6/8$	2.020	1.626	0.394
$6/9$	1.638	1.059	0.579
$6/10$	1.468	0.679	0.789
$6/11$	1.352	0.401	0.951
$6/12$	1.248	0.187	1.061
$6/13$	1.186	0.018	1.168
$6/14$	1.018	-0.122	1.140
$6/15$	0.930	-0.238	1.168
$6/16$	0.848	-0.337	1.185
$6/17$	0.692	-0.422	1.114
$6/18$	0.602	-0.495	1.097

TABLE III
THRESHOLDS OF THE PN-PBRL LDPC CODES WITH PARALLEL EDGES(E_b/N_0 IN DECIBELS).

Rate	Threshold	Capacity	Gap
$6/8$	1.965	1.626	0.339
$6/9$	1.314	1.059	0.255
$6/10$	0.948	0.679	0.269
$6/11$	0.678	0.401	0.277
$6/12$	0.422	0.187	0.235
$6/13$	0.270	0.0179	0.252
$6/14$	0.118	-0.122	0.240
$6/15$	0.005	-0.238	0.243
$6/16$	-0.102	-0.337	0.235
$6/17$	-0.172	-0.422	0.250
$6/18$	-0.266	-0.495	0.229

VI. SIMULATIONS

This section presents the frame error rate (FER) and bit error rate (BER) simulations of the PBRL and PN-PBRL codes. Lifting of the protograph is accomplished by circulant permutation of each edge, which allows efficient implementation of the decoder. The design of the circulant permutation uses a greedy algorithm to avoids all length-4 cycles and minimizes the number of length-6 cycles.

The PBRL and PN-PBRL codes that are considered in this section can be described as follows: let H_p be the parity check matrix of the precode and H_{LT} be the parity check matrix of the LT code excluding the degree-one variable nodes. Let σ

be a 32×32 identity matrix shifted to the left by 1. The full parity check matrix is given by

$$H = \begin{bmatrix} H_p & O \\ H_{LT} & I \end{bmatrix}$$

where H_p and H_{LT} are given in equations (1) and (2) for the PBRL code and equations (3) and (4) for the PN-PBRL code, respectively. I is the identity matrix and O is the all-zero matrix with proper dimensions. Entries with multiple terms of σ indicate parallel edges in the protograph.

Figs. 3 and 4 show the simulations of the PBRL and PN-PBRL code family with rates $6/8, 6/9, \ldots, 6/18$. Layered belief propagation is used for the decoder simulations shown in Figs. 3 and 4. We observe a saturation of the performance, as expected from Table I and Table II. Consistent with the threshold results, the PN-PBRL code family out-performs PBRL code family with a slight increase of encoding complexity.

Consider Raptor codes with the same precode as the PBRL code and the output distributions drawn from [9] and [16]. Simulation results show that these Raptor codes with information block of 192 bits have frame error rates much higher than both PBRL and PN-PBRL codes. This result is not surprising because a relatively short block of information is considered: since the degrees of each output node are drawn at random according to the optimal degree distribution, a few hundreds of samples might not be enough to exhibit the optimal degree distribution. The performance plots are omitted due to space limitations.

Fig. 5 shows the simulations of 3GPP RCPT codes with the same range of code rates and blocklengths. Different code rates of the RCPT codes are obtained by pseudo-random puncturing, or circular buffer rate matching (CBRM), described in [18]. Although the RCPT code family performs better at low SNR regime, it also suffers from an error floor as soon as the FER reaches 10^{-3} for rate $3/4$ and 10^{-5} for rate $1/3$, respectively.

For easier comparison, the FER and BER with rate $3/4$ and $1/3$ of the PBRL, PN-PBRL and RCPT codes are plotted separately in Fig. 6 and 7. Note that in Figs. 6 and 7, flooding is used for decoder simulations, which gives slightly worse performance than the layered belief propagation decoding used in Figs. 3 and 4. At rate $3/4$, the PN-PBRL code performs similarly to RCPT code and outperforms RCPT code when SNR is higher than 3 dB in terms of FER. At rate $1/3$, the PN-PBRL code starts to gain an advantage at SNR higher than 3.5 dB in terms of FER.

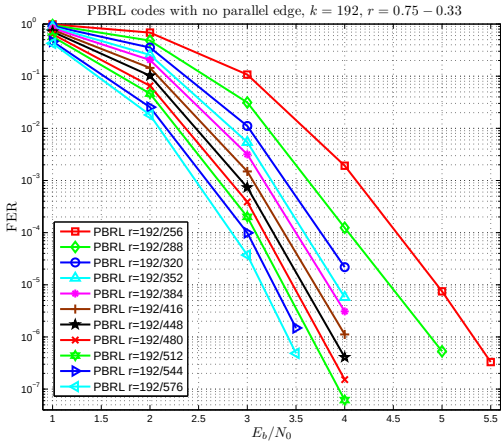

Fig. 3. Frame error rate of the rate-compatible PBRL code family. Layered belief propagation is used for the decoder simulations.

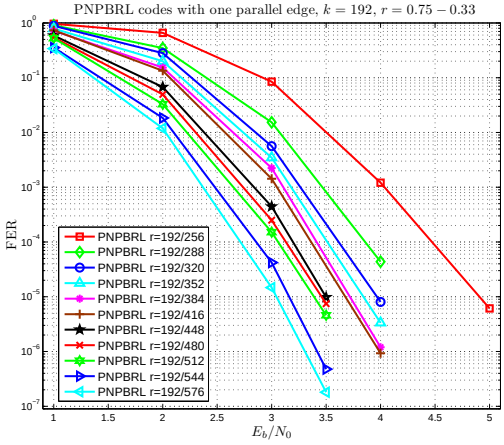

Fig. 4. Frame error rate of the rate-compatible PN-PBRL code family. Layered belief propagation is used for the decoder simulations.

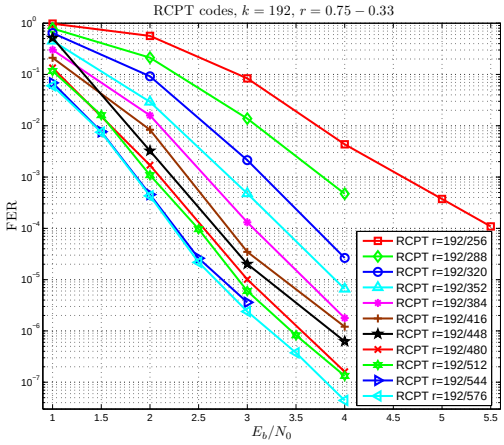

Fig. 5. Frame error rate of the rate-compatible RCPT code family. Iterative BCJR algorithm is used for decoding with maximum 12 iterations.

VII. Conclusion and Future Work

This paper proposes a class of Raptor-like rateless codes and provides a systematic procedure of constructing practical codes. Optimization of the code is based on asymptotic results of LDPC codes, i.e., density evolution. The simulation results show that although we are operating in a short blocklength regime, optimization using density evolution still enhances the performance of the PBRL code.

The proposed PBRL codes have several beneficial structures in terms of complexity. First, it is based on a protograph structure and hence allows simple encoder and decoder structures. Second, the systematic structure allows joint decoding

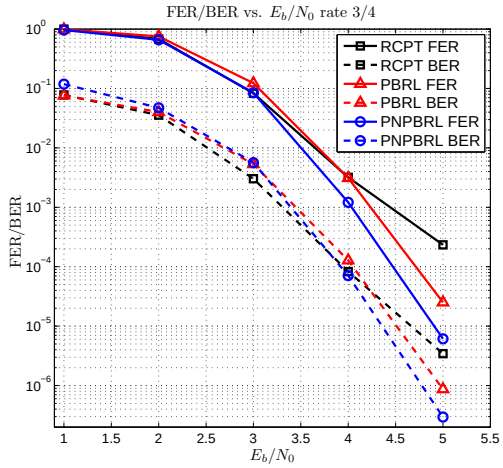

Fig. 6. Frame error rate and bit error rate of the PBRL code, PN-PBRL code and RCPT code at code rate $3/4$. Flooding is used for the decoder simulations. Both PBRL and PN-PBRL codes outperform the RCPT codes at high E_b/N_0 regime but perform slightly worse than the RCPT code in the low E_b/N_0 regime.

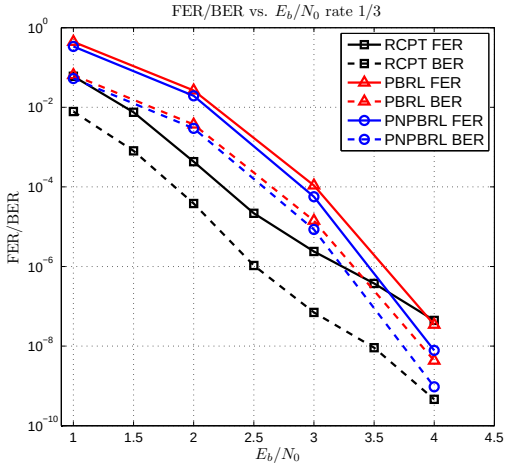

Fig. 7. Frame error rate and bit error rate of the PBRL code, PN-PBRL code and RCPT code at code rate $1/3$. Flooding is used for the decoder simulations. Here the RCPT code outperforms the PN-PBRL and PBRL code at low SNR range, but the PN-PBRL code starts to outperform the RCPT code at around SNR 3.5 dB. There is no sign of an error floor for the both PBRL and PN-PBRL code.

of the precode and LT code with the same decoder. Finally, the code is rate-compatible and can be readily applied to any incremental redundancy scheme that requires rate-compatible channel codes.

The PN-PBRL code has the same structure as the PBRL code, but one of the variable nodes in the precode protograph is punctured and one variable node in the LT code protograph is transmitted. The PN-PBRL code has better performance but a slightly more complicated encoder for the initial transmission.

The main contribution of this paper is to introduce a new class of rate-compatible LDPC codes with simple encoding and decoding structures. Motivated by [10]–[12], we focus on the short blocklength regime. These short blocklength codes perform well and do not have error floors up to the highest SNRs studied. In the short blocklength regime, the asymptotic analysis of the PBRL code might not be as accurate as for the long blocklength code. Finding a better criterion for designing a good short blocklength PBRL code is an interesting direction for future research.

A threshold saturation is observed as the rates decrease. Adding parallel edges in the LT code part of the PN-PBRL protograph alleviates the saturation issue. The lifted code with blocklength 256, however, does not have performance better than the codes considered in Section VI. These low thresholds indicate a promising research area for extending the PN-PBRL code structure to the design of good LDPC codes with long blocklengths. Indeed, we have found that longer-blocklength PBRL codes do perform very close to capacity, as we will show in a future publication.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, MIT, Cambridge, MA, 1963.

[2] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, Issue 18, p. 1645, Aug. 1996.

[3] M. R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, No. 5, pp. 533–547, 1981.

[4] J. Thorpe, "Low Density Parity Check (LDPC) codes constructed from protographs," *JPL IPN Progress Report*, vol. 42–154, Aug. 2003.

[5] D. Divsalar, C. Jones, S. Donlinar and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. GLOBECOM*, St. Louis, MO, USA, Nov. 2005.

[6] D. Divsalar, S. Donlinar, C. R. Jones and Kenneth Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, No. 6, pp. 876–888, Aug. 2009.

[7] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, BC, Canada, Nov. 2002.

[8] A. Shokorollahi, "Raptor Codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2551–2567, Jun. 2006.

[9] O. Etesami and A. Shokorollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2033–2051, May 2006.

[10] T-Y. Chen, N. Seshadri and R. D. Wesel, "A sphere-packing analysis of incremental redundancy scheme with feedback," in *Proc. IEEE Intl. Conf. Comm.*, Kyoto, Japan, 2011.

[11] ——, "Incremental redundancy: a comparison of a sphere-packing analysis and convolutional codes," in *Proc. Inform. Theory and Applicat. Workshop*, San Diego, CA, USA, Feb. 2011.

[12] T-Y. Chen, B-Z. Shen and N. Seshadri, "Is feedback a performance equalizer of classic and modern codes?" in *ITA Workshop*, San Diego, CA, USA, Feb. 2010.

[13] M. R. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," *IEEE Commun. Lett.*, vol. 8, No. 3, pp. 159–161, Mar. 2004.

[14] N. Jacobsen and R. Soni, "Deign of rate-compatible irregular LDPC codes based on edge growth and parity splitting," in *Proc. IEEE Vehicular Technology Conference (VTC)*, Baltimore, MD, USA, Oct. 2007.

[15] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "The deisgn of rate-compatible protograph LDPC codes," in *Proc. 48th Annual Allerton Conference*, Allerton, IL, USA, Sep. 2010.

[16] E. Soljanin, "Punctured vs. rateless codes for hybrid ARQ," in *Proc. IEEE Inform. Theory Workshop '06*, Punta del Este, Uruguay, Mar. 2006.

[17] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, No. 1, pp. 386–398, Jan. 2005.

[18] 3rd Generation Partnership Project (http://www.3gpp.org), "3GPP TS 36.212 Multiplexing and channel coding."