Performance-Complexity Tradeoffs of Raptor Codes over Gaussian Channels

Ketai Hu, Jeff Castura, and Yongyi Mao

Abstract— **We investigate the performance-complexity tradeoffs of Raptor codes over Gaussian channels. Two different implementations of the Belief-Propagation (BP) decoding algorithm are considered, which we respectively refer to as "message-reset decoding" and "incremental decoding". We show that incremental decoding offers great advantages over message-reset decoding in terms of this tradeoff.**

Index Terms— **Raptor codes, performance-complexity tradeoff, belief propagation, incremental decoding.**

I. Introduction

THE invention of fountain codes [1], [2] has resulted in reviving interest in incremental redundancy schemes for communication under channel uncertainty. Remarkably, fountain codes — consisting of LT codes [1] and Raptor codes [2] — are shown to be capacity-achieving over unknown erasure channels. Recently these codes are also shown to be nearly capacity-achieving under other channel models such as binary symmetric channels (BSC), AWGN and fading channels [3]–[5]. In this paper, we investigate — under two different decoding strategies — the tradeoffs between the performance and decoding complexity of fountain codes, particularly Raptor codes, over Gaussian channels. To date, no results are available concerning performance-complexity tradeoffs of fountain codes decoded over noisy channels (such as BSC, AWGN, or fading channels). The two decoding strategies we investigate are referred to as "message-reset decoding" and "incremental decoding" respectively and we show that incremental decoding offers significantly improved performance-complexity tradeoff.

II. System Model and Two Decoding Strategies

We consider communication over a standard AWGN channel using Raptor codes under BPSK modulation. A Raptor code is a linear code constructed by the serial concatenation of a high-rate LDPC code with an LT code. The LDPC code encodes a $k-$bit message $(a_1, a_2, \ldots, a_k)$ to a k'-bit vector $(b_1, b_2, \ldots, b_{k'})$, and the LT code encodes the vector $(b_1, b_2, \ldots, b_{k'})$ to an infinite binary sequence $(c_1, c_2, \ldots)$ by multiplying $(b_1, b_2, \ldots, b_{k'})$ with a sparse randomly-constructed generator matrix of k rows and infinite number of columns, and the sequence $(c_1, c_2, \ldots)$ is transmitted sequentially through the channel. $k' = 10,000$ and

Manuscript received November 27, 2006. The associate editor coordinating the review of this letter and approving it for publication was Dr. Mischa Dohler.

The authors are with SITE, University of Ottawa, 800 King Edward Ave., Ottawa, Canada, K1N 6N5 (email: {ketaihu, jcastura, yymao}@site.uottawa.ca).

Digital Object Identifier 10.1109/LCOMM.2007.061939.

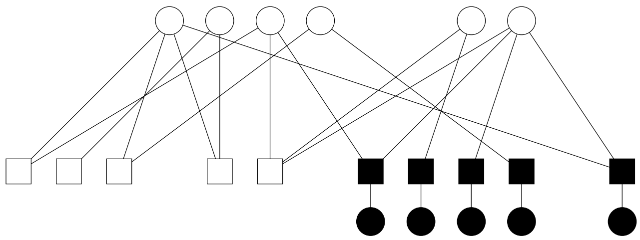

Fig. 1. The factor graph of a Raptor code. White circles represent LDPC codeword bits, white boxes represent the parity checks of the LDPC code, shaded circles represent the Raptor codeword bits, and shaded boxes represent LT code parity checks. The graph shows a Raptor code truncated to length n.

degree sequences for the LDPC code and LT given in [4]. The canonical representation of Raptor codes are factor graphs [6]. An example of such a factor graph, truncated to length n and to be referred to as $\mathcal{G}_n$, is shown in Fig. 1.

The decoder attempts decoding at a prescribed set of times $\{n_1, n_2, \ldots\}$, with $n_1 < n_2 < \ldots$. At the l^{th} decoding attempt, it performs BP decoding on factor graph $\mathcal{G}_{n_l}$ by iteratively passing the LLR (log-likelihood ratio) messages, where in each iteration all variables pass messages followed by all checks passing messages. In the l^{th} decoding attempt, if the decoder is confident that the transmitted message is decoded, it sends an ACK through a noiseless feedback channel to terminate the transmission of the current codeword; otherwise it waits until the next $((l+1)^{\text{th}})$ decoding attempt to decode again. We assume that a small number of CRC bits are embedded in message $(a_1, \ldots, a_k)$, and that the decoder knows whether it has decoded correctly at any decoding attempt (the resulted slight rate loss is ignored in the forth-coming discussion). For simplicity, we only consider that all decoding attempts are uniformly spaced in time by T channel uses, starting from some n_1 (such that k/n_1 is slightly higher than the channel capacity).

Under the BP rule, in the initialization step (namely, when messages are passed from variables to checks in the first iteration) of the l^{th} decoding attempt, every message passed from a codeword variable c_i to its connected check on $\mathcal{G}_{n_l}$ is computed as $\log(p(y_i|c_i = 0)/p(y_i|c_i = 1))$, where y_i is the noisy observation of c_i. Two different schemes may exist for initializing messages passed from variables $(b_1, \ldots, b_{k'})$ to their connected checks: in a usual and default strategy which we refer to as "message-reset" decoding, these messages are initialized to 0; in an alternative initialization strategy which we call "incremental decoding", at the l^{th} decoding attempt, a message passed from a variable in $(b_1, \ldots, b_{k'})$ is set to the

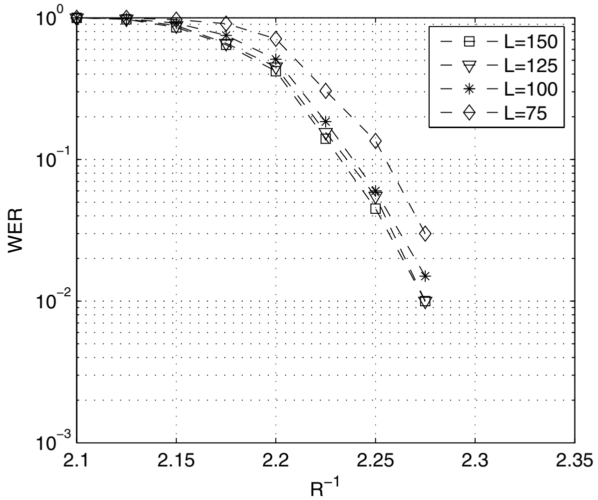

Fig. 2. Word error rate for message-reset decoding over capacity-0.5 channel with various choices of L.

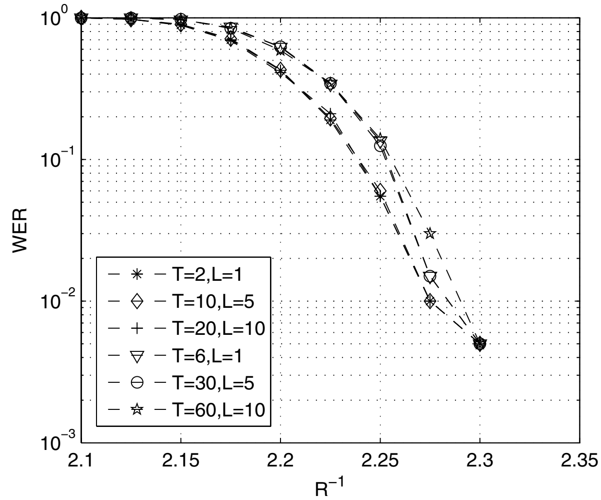

Fig. 3. Word error rate for incremental decoding over capacity-0.5 channel with various settings of (T, L).

corresponding message computed at the end of the $(l-1)^{\text{th}}$ decoding attempt if the edge is contained in graph $\mathcal{G}_{n_{l-1}}$, and set to 0 otherwise. Simply speaking, message-reset decoding at every decoding attempt *starts "from scratch"* without making use of the soft information produced in the previous decoding attempt, whereas incremental decoding *continues* from the decoding results of the last decoding attempt. The idea of incremental decoding being under the guise of various literature, this is however the first time it is presented in the context of decoding fountain codes.

Details of the message-passing rules follow the standard BP algorithm (see, e.g., [6]). We assume that at every decoding attempt the decoder will perform L BP iterations for some predetermined L, with only one exception with the first decoding attempt of incremental decoding, where the number of BP iterations is fixed to 100.

For each codeword transmission, we define the realized rate by k/n, where n is the time of the final decoding attempt at which the codeword is successfully decoded. We then evaluate the performance of two decoding strategies using average realized rate (over all codeword transmissions). The complexity of decoding schemes is evaluated by the total number of BP iterations needed to decode a word (across all decoding attempts) on average.

Intuitively, one may see that as we decrease T and L simultaneously in incremental decoding, there is an opportunity for it to out-perform message-reset decoding for the same complexity. This is because re-calculation of the same messages is avoided with incremental decoding, and in addition, there is a possibility for channel information to propagate much farther in the graph even with very small value of L, as long as T is also made sufficiently small.

III. Simulation Results

We simulated both message-reset decoding and incremental decoding with various respective settings of (T, L) and over channels with capacity $0.25, 0.5$, and 0.75 bits/channel use respectively. Following [4], we use a Raptor code construction

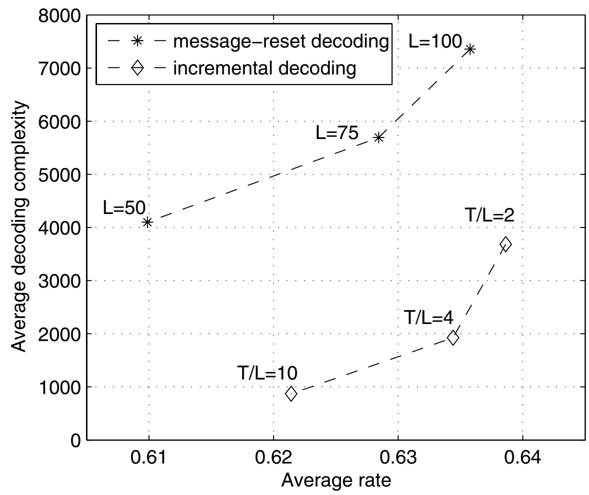

Fig. 4. Complexity vs average realized rate for incremental decoding and message-reset decoding over channel with capacity 0.75.

with $k = 9,500$, $k' = 10,000$ where the outer LDPC code has 4-regular left-degree distribution and Poisson right-degree distribution, and the LT code has degree distribution given by

$$\begin{aligned}\Omega(x) \;\; = \;\; & 0.007969x + 0.493572x^2 + 0.166220x^3 \\ & +0.072464x^4 + 0.082558x^5 \\ & +0.056058x^8 + 0.037229x^9 \\ & +0.055590x^{19} + 0.025023x^{65} \\ & +0.003135x^{66}.\end{aligned}$$

For message-reset decoding, as the choice of decoding interval T only affects the "resolution" of realized rates, we fix T as 100, following [5]. We then investigate the performance of the code, under this choice of T, for various values of L. It appears that $L = 100$ gives practically the "optimal" performance, and that the performance gain becomes negligible as we further increase L. This may be seen in Fig. 2, where the word error rate (WER) is plotted against R^{-1} for channel with capacity

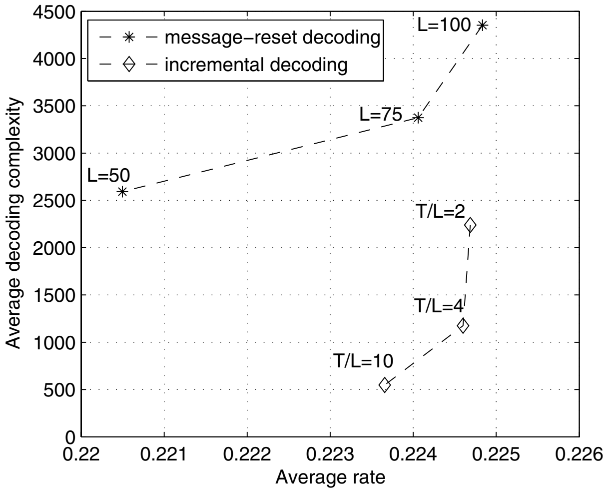

Fig. 5. Complexity vs average realized rate for incremental decoding and message-reset decoding over channel with capacity 0.25.

0.5 bits/channel use — Here for a given value of R, WER is defined as the probability that the successful decoding does not occur before k/R channel uses.

For incremental decoding, we observe that for all channels and all parameter settings of (T, L), the performance is primarily governed by the ratio T/L and that as long as the ratio T/L is kept fixed, the performance of incremental decoder is virtually independent of T or L alone. For example, over the channel with capacity 0.5 bits/channel use, this is demonstrated in Fig. 3. We thus choose $L = 1$ for incremental decoder to minimize the number of BP iterations at each decoding attempt.

Figures 4 and 5 plot the tradeoffs between decoding complexity and average realized rate achieved respectively by message-reset decoding and incremental decoding over channels with capacity 0.75 and 0.25. It is evident that significantly improved performance-complexity tradeoff is seen with incremental decoding. Depending on the targeted performance, $50-80\%$ reduction of decoding complexity is achievable with incremental decoding compared with message-reset decoding. Similar phenomenon is observed for channel with capacity 0.5. We note that this improvement of performance-complexity tradeoff appears more pronounced for poorer channels.

IV. Conclusion

In this paper, we show that over Gaussian channels, incremental decoders provide significantly improved performance-complexity tradeoff over message-reset decoders. We believe that this result to a large extent holds true for fading channels, promising Raptor codes with incremental decoding as an appealing solution for practical communication under channel uncertainty.

References

[1] M. Luby, "LT codes," in *Proc. 43rd Annual IEEE Symp. on the Found. of Comp. Sci. 2002*, pp. 271–280.

[2] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[3] O. Etesami, M. Molkaraie, and A. Shokrollahi, "Raptor codes on symmetric channels," in *Proc. IEEE Int. Symp. on Inform. Theory 2004*, p. 38.

[4] R. Palanki and J. S. Yedidia, "Rateless codes on noisy channels," in *Proc. IEEE Int. Symp. on Inform. Theory 2004*, p. 37.

[5] J. Castura and Y. Mao, "Rateless coding over fading channels," *IEEE Commun. Lett.*, vol. 10, no. 1, pp. 46–48, Jan. 2006.

[6] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.