MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Rateless Codes on Noisy Channels

Ravi Palanki and Jonathan S. Yedidia

TR2003-124 October 2003

Abstract

We study the performance of the newly invented rateless codes (LT- and Raptor codes) on noisy channels such as the BSC and the AWGNC. We find that Raptor codes outperform LT codes, and have good performance on a wide variety of noisy channels.

1. First printing, October 2003. 2. Minor corrections, November 2003. 3. More minor corrections, April 2004. 4. See also MERL TR2004-037 for a shorter version of this technical report published in the Proceedings of the Conference on Information Sciences and Systems, 2004.

Rateless codes on noisy channels

Ravi Palanki
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, USA
email: ravi@systems.caltech.edu

Jonathan S. Yedidia
Mitsubishi Electric Research Laboratories
Cambridge, MA 02139, USA
email: yedidia@merl.com

Abstract

We study the performance of the newly invented rateless codes (LT and Raptor codes) on noisy channels such as the BSC and the AWGNC. We find that Raptor codes outperform LT codes, and have good performance on a wide variety of noisy channels.

1 Introduction

Recent advances in coding theory, especially the invention of regular [1] and irregular [2] low density parity check (LDPC) codes, have shown that very efficient error correction schemes are possible. LDPC codes, decoded using the belief propagation algorithm, can achieve capacity on the binary erasure channel (BEC) [2, 3] and achieve rates very close to capacity on other channels such as the binary symmetric channel (BSC) and the additive white Gaussian noise channel (AWGNC)[4]. Because of this, one could say that the problem of reliable communication over many practical channels has been solved. However, such a statement comes with a caveat: both the transmitter and the receiver must know the exact channel statistics a-priori. While this assumption is valid in many important cases, it is clearly not true in many other equally important cases. For example, on the internet (which is modeled as a BEC), the probability p that a given packet is dropped varies with time, depending on traffic conditions in the network. A code designed for a good channel (low p) would result in decoding failure when used over a bad channel (high p). Conversely, a code designed for a bad channel would result in unnecessary packet transmissions when used over a good channel.

We can solve this problem by using *rateless codes*. Instead of encoding the k information bits to a pre-determined number of bits using a block code, the transmitter encodes them into a potentially infinite stream of bits and then starts transmitting them. Once the receiver gets a sufficient number of symbols from the output stream, it decodes the original k bits. The number of symbols required for successful decoding depends on the quality of the channel. If decoding fails, the receiver can pick up a few more output symbols and attempt decoding again. This process can be repeated until successful decoding. The receiver can then tell the transmitter over a feedback channel to stop any further transmission.

The use of such an "incremental redundancy" scheme is not new to coding theorists. In 1974, Mandelbaum [5] proposed puncturing a low rate block code to build such a system. First the information bits are encoded using a low rate block code. The resulting codeword is then punctured suitably and transmitted over the channel. At the receiver the punctured bits are treated as erasures. If the receiver fails to decode using just the received bits, then some of the punctured bits are transmitted. This process is repeated till every bit of the low rate codeword has been transmitted. If the decoder still fails, the transmitter begins to retransmit bits till successful decoding. It is easy to see such a system is indeed a rateless code, since the encoder ends up transmitting a different number of bits depending on the quality of the channel. Moreover, if the block code is a random (or random linear) block code, then the rateless code approaches the Shannon limit on every binary input symmetric channel (BISC) as the rate of the block code approaches zero. Thus such a scheme is optimal in the information theoretic sense.

Unfortunately, it does not work as well with practical codes. Mandelbaum originally used RS codes for this purpose and other authors have investigated the use of punctured low rate convolutional [6] and turbo [7] codes. In addition to many code-dependent problems, all these schemes share a few common problems. Firstly, the performance of the rateless code is highly sensitive to the performance of the low rate block code i.e., a slightly sub-optimal block code can result in a highly sub-optimal rateless code. Secondly, the rateless code has very high decoding complexity, even on a good channel. This is because on any channel, the decoder is decoding the same low-rate code, but with varying channel information. The complexity of such a decoding grows at least as $O(k/R)$ where R is the rate of the low rate code.

In a recent landmark paper, Luby [8] circumvented these problems by designing rateless codes which are not obtained by puncturing standard block codes. These codes, known as Luby Transform (LT) codes, are low density generator matrix codes which are decoded using the same message passing decoding algorithm (belief propagation) that is used to decode LDPC codes. Also, just like LDPC codes, LT codes achieve capacity on every BEC. Unfortunately, LT codes also share the error floor problem endemic to capacity achieving LDPC codes. Shokrollahi [9, 10] showed that this problem can be solved using raptor codes, which are LT codes combined with outer LDPC codes. These codes have no noticeable error floors on the BEC though their rate is slightly bounded away from capacity.

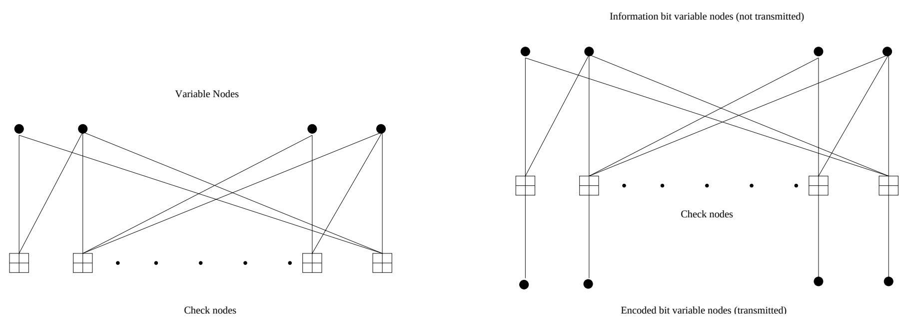
The aim of this paper is to study the performance of LT and raptor codes on channels other than the BEC. Since LDPC codes designed for the BEC perform fairly well on other channels, one might conjecture that such a result holds for LT codes as well. We test this conjecture using simulation studies and density evolution [11].

2 LT codes

The operation of an LT encoder is very easy to describe. From k given information bits, it generates an infinite stream of encoded bits, with each such encoded bit generated as follows:

1. Pick a degree d at random according to a distribution $\mu(d)$.
2. Choose uniformly at random d distinct input bits.
3. The encoded bit's value is the XOR-sum of these d bit values.

The encoded bit is then transmitted over a noisy channel, and the decoder receives a corrupted version of this bit. Here we make the non-trivial assumption that the encoder and decoder are completely synchronized and share a common random number generator i.e., the decoder knows which d bits are used to generate any given encoded bit, but not their values. On the internet, this sort of synchronization is easily achieved because every packet has an uncorrupted packet number. More complicated schemes are required on other channels; here we shall just assume that some such scheme exists and works perfectly in the system we're studying. In other words, the decoder



(a) (b)

Figure 1: Tanner graph of (a) LDPC code (b) LT code.

can reconstruct the LT code's Tanner graph without error.

Having done that, the decoder runs a belief propagation algorithm on this Tanner graph. The message passing rules are straightforward and resemble those of an LDPC decoder. However there is one important difference: the Tanner graph of an LDPC code contains only one kind of variable node (figure 1(a)), while that of an LT code contains two kinds of variable nodes (figure 1(b)) These are the information bit variable nodes which are not transmitted (and hence have no channel evidence) and the encoded bit variable nodes which are transmitted over the channel. Clearly, for large block lengths, the performance of such a system depends mostly on the degree distribution μ. Luby uses the Robust Soliton (RS) distribution, which in turn is based on the ideal soliton distribution, defined as follows:

$$\begin{aligned} \rho(1) &= 1/k \\ \rho(i) &= 1/i(i-1) \ \forall i \in \{2,3,...,k\} \end{aligned} \tag{1}$$

While the ideal soliton distribution is optimal in some ways (see [8]), it performs rather poorly in practice. However, it can modified slightly to yield the robust soliton distribution $RS(k,c,\delta)$. Let $R \triangleq c \cdot \ln(k/\delta)\sqrt{k}$ for some suitable constant $c > 0$. Define

$$\tau(i) = \begin{cases} R/ik & \text{for } i = 1,\dots,k/R-1 \\ R\ln(R/\delta)/k & \text{for } i = k/R \\ 0 & \text{for } i = k/R+1,\dots,k \end{cases} \tag{2}$$

Now add $\tau(.)$ to the ideal soliton distribution $\rho(.)$ and normalize to obtain the robust soliton distribution:

$$\mu(i) = (\rho(i) + \tau(i))/\beta \tag{3}$$

where β is the normalization constant chosen to ensure that μ is a probability distribution.

Luby's analysis and simulation studies show that this distribution performs very well on the erasure channel. The only disadvantage is the decoding complexity grows as $O(k \ln k)$, but it turns out that such a growth in complexity is in fact necessary to achieve capacity [9]. However, slightly sub-optimal codes called raptor codes, can be designed with decoding complexity $O(k)$ [9]. On the BEC, theoretical analysis of the performance of LT codes and raptor codes is feasible, and both codes have been shown to have excellent performance. In fact, raptor codes are currently being used by Digital Fountain, a Silicon Valley based company, to provide fast and reliable transfer of large files over the Internet.

On other channels such as the BSC and the AWGNC, there have been no studies in the literature on the use of LT and raptor codes, despite the existence of many potential applications (Eg: Transfer of large files over a wireless link, multicast over a wireless channel). We hope to fill this void by presenting some simulation results and some theoretical analysis (density evolution). In this paper, we focus on the BSC and the AWGNC, but we hope to extend our studies to time varying and fading channels.

3 LT codes on noisy channels

When the receiver tries decoding after picking up a finite number n of symbols from the infinite stream sent out by the transmitter, it is in effect trying to decode an (n, k) code, with a non-zero rate $R = k/n$. As R decreases, the decoding complexity goes up and the probability of decoding error goes down. In this paper, we have studied the variation of bit error rate (BER) and word error rate (WER) with the rate of the code on a given channel.

In figure 2, we show some results for LT codes on a BSC with 11% bit flip probability. We mention that the results are similar in nature on other BSCs and other AWGNCs as well. In this figure, we plot R^{-1} on the x-axis and BER/WER on the y-axis. The receiver buffers up kR^{-1} bits before it starts decoding the LT code using belief propagation. On a BSC with 11% bit flip

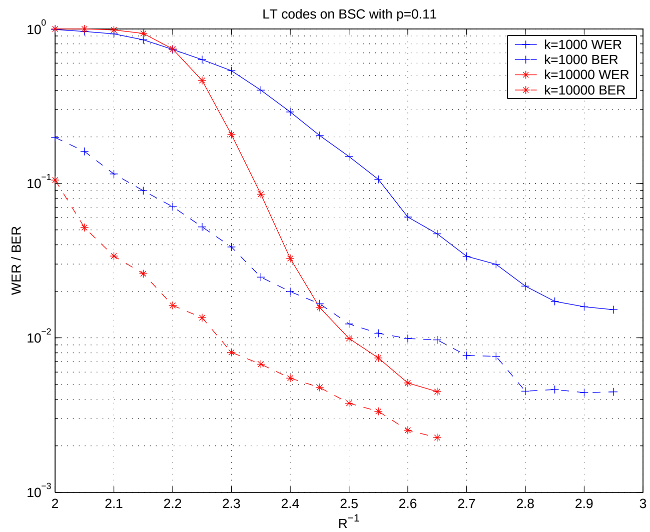


Figure 2: LT codes generated using the RS(k,0.01,0.5) distribution on a BSC with $p = 0.11$

probability, the Shannon limit is $R^{-1} = 2$ i.e., a little over $2k$ bits should suffice for reliable decoding in the large k limit. We see from the figure that an LT code with $k = 10000$ drawn using the RS(10000,0.1,0.5) distribution can achieve a WER of 10^{-2} at $R^{-1} = 2.5$ (or $n = 25000$). While this may suffice for certain applications, neither a 25% overhead nor a WER of 10^{-2} is particularly impressive. Moreover, the WER and BER curves bottom out into an error floor, and achieving very small WERs without huge overheads in nearly impossible. Going to higher block lengths is also not practical because of the $O(k \ln k)$ complexity.

The error floor problem is not confined to LT codes generated using a robust soliton distribution. Codes generated using distributions optimized by Shokrollahi for the BEC[1][9] also exhibit similar behaviour. The main advantage of these distributions is that the average number of edges per node remains constant with increasing k, which means the decoding complexity grows only as $O(k)$. On the minus side, there will be a small fraction of information bit nodes that are not connected to any check node. This means that even as k goes to infinity, the bit error rate does not go to zero and consequently, the word error rate is always one.

[1]Note that these distributions were not designed to be used directly in LT codes.

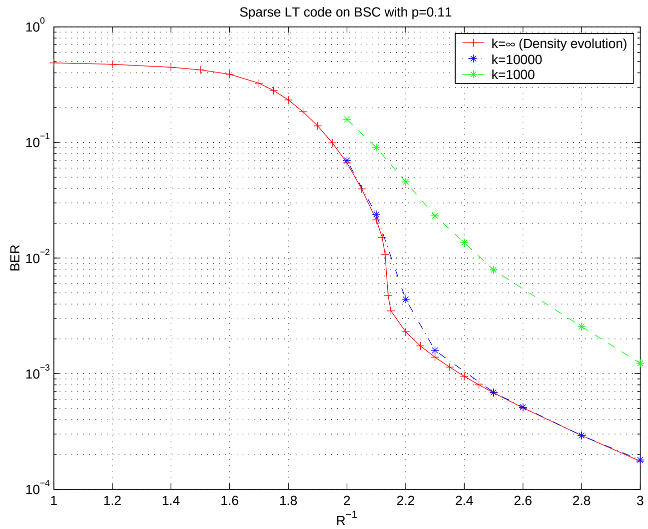


Figure 3: LT codes generated using right distribution in Eqn. 4 on a BSC with $p = 0.11$

In this paper, we discuss the performance of one particular distribution from [9]:

$$\begin{aligned}\mu(x) &= 0.007969x + 0.493570x^2 + 0.166220x^3 \\ &\quad +0.072646x^4 + 0.082558x^5 + 0.056058x^8 + 0.037229x^9 \\ &\quad +0.055590x^{19} + 0.025023x^{65} + 0.0003135x^{66} \qquad (4)\end{aligned}$$

Shown in figure 3 is the performance of codes generated using the distribution in Eqn 4 at lengths 1000, 10000 and infinity. The performance at length infinity is computed using density evolution [11]. Again, we observe fairly bad error floors, even in the infinite blocklength limit.

We must mention that these error floors are not just due to the presence of information bit variable nodes not connected to any check nodes. For example, when $R^{-1} = 3.00$, only a very small fraction of variable nodes (2.25×10^{-8}) are unconnected, while density evolution predicts a much larger bit error rate (1.75×10^{-4}). This can be attributed in part to the fact that there are variable nodes which are connected to a relatively small number of output nodes and hence are always unreliable.

4 Raptor Codes

The error floors exhibited by LT codes suggest the use of an outer code. Indeed this is what Shokrollahi does in the case of the BEC [9, 10] where he introduces[2] the idea of raptor codes, which are LT codes combined with outer codes. Typically these outer codes are high rate LDPC codes. In this paper, we use the distribution in Eqn 4 for the inner LT code. For the outer LDPC code, we follow Shokrollahi [9] and use a left regular distribution (node degree 4 for all nodes) and right Poisson (check nodes chosen randomly with a uniform distribution).

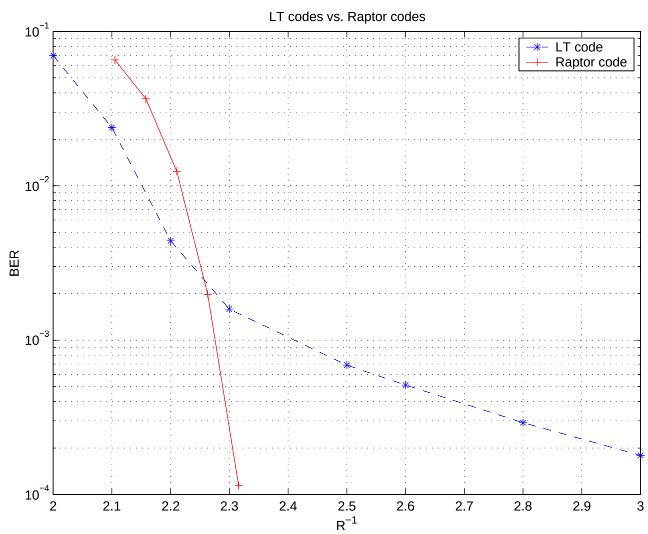


Figure 4: Comparing LT codes with raptor codes on a BSC with p=0.11. The LT code has $k = 10000$ and is generated using the distribution in Eqn. 4. The raptor code has $k = 9500$ and has two components: an outer rate-0.95 LDPC code and an inner rateless LT code generated using the distribution in Eqn. 4.

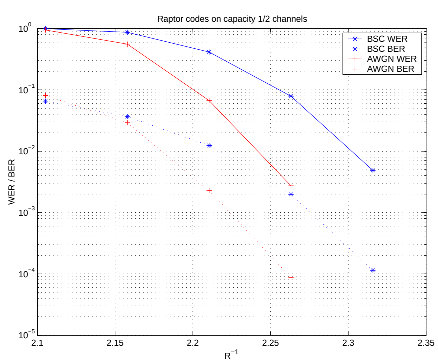
The encoder for such a raptor code works as follows: the k input bits are first encoded into k' bits to form a codeword of the outer LDPC code. These k' bits are then encoded into an infinite stream of bits using the rateless LT code. The decoder picks up a sufficient number (n) of output symbols,

[2]We must mention here that Maymounkov [12] independently proposed the idea of using an outer code.

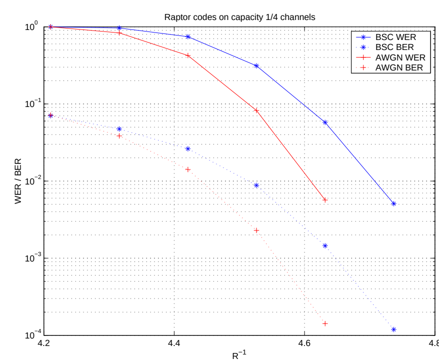
constructs a Tanner graph that incorporates both the outer LDPC code and the inner LT code, and decodes using belief propagation on this Tanner graph.

Simulation studies, such as the one shown in figure 4 clearly indicate the superiority of raptor codes. Figure 4 shows a comparison between LT codes and raptor codes on a BSC with bit flip probability 0.11. The LT code has $k = 10000$ and is generated using the distribution in Eqn. 4. The raptor code has $k = 9500$ and uses an outer LDPC code of rate 0.95 to get $k' = 10000$ encoded bits. These bits are then encoded using an inner LT code, again generated using the distribution in Eqn 4. Figure 4 clearly shows the advantage of using the outer high-rate code.

Raptor codes not only beat LT codes comprehensively, but also have near-optimal performance on a wide variety of channels as shown in figure 5, which shows the performance of the aforementioned raptor code on four different channels. On each of these channels, the raptor code has a waterfall region close to the Shannon capacity, with no noticelable error floors. Of course, this does not rule out error floors at lower WERs.



(a)



(b)

Figure 5: Performance of raptor code with k=9500 and k'=10000 on different channels: (a) BSC with $p = 0.11$ and AWGNC with $E_s/N_0 = -2.83dB$. Both channels have capacity 0.5. (b) on BSC with $p = 0.2145$ and AWGNC with $E_s/N_0 = -6.81dB$. Both channels have capacity 0.25

Another indicator of the performance of a rateless code on any given channel is the number of bits required for successful decoding. We must note here this indicator not only depends on the code, but also on the number of decoding attempts made by the receiver. For example, the decoder could

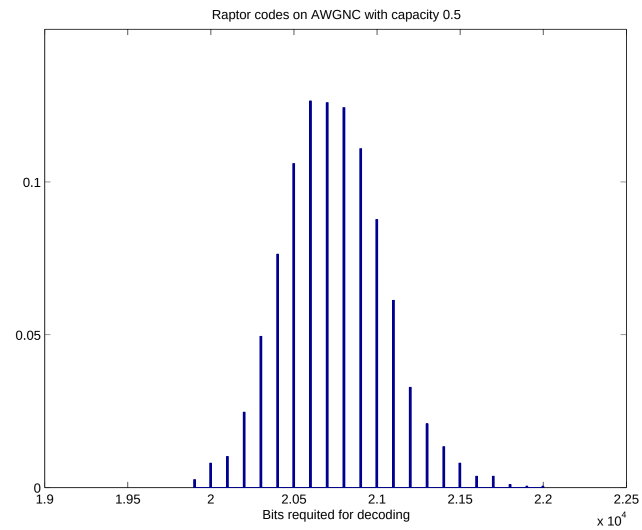


Figure 6: Histogram of number of bits required for successful decoding of raptor code with $k = 9500$ on an AWGNC with $E_s/N_0 = -2.83dB$. The capacity of this channel is 0.5. The receiver first attempts decoding after receiving 19000 noisy bits (Shannon limit). Whenever decoding fails, the receiver waits for another 100 bits before attempting to decode again.

attempt decoding each time it receives a new noisy bit. While such a decoder would be optimal in terms of number of bits required for successful decoding, it would have prohibitively high decoding complexity. A more practical decoder would wait for more bits to come in before decoding. Such a decoder would have a vastly lower complexity at the expense of slightly larger number of bits received. Note that such a tradeoff is absent in the case of the BEC. This is because the decoder fails when the Tanner graph is reduced to a stopping set. After new bits are received, further decoding can be done on the stopping set instead of the original Tanner graph. Such a scheme is not appplicable to noisy channels where the decoder must start over every time new bits are received.

Figure 6 shows a histogram of the number of noisy bits needed for decoding the previously described raptor code with $k = 9500$. We observe that the expected number of noisy bits required for successful decoding (20737) is fairly close to the Shannon limit (19000).

5 Conclusion

We have conducted simulation studies of rateless codes on channels such as the BSC and the AWGNC. We found that raptor codes have excellent performance on such channels, while the performance of LT codes is not as good. These results suggest that raptor codes are ideal for use in data transfer protocols on noisy channels. A similar observation has already been made on the BEC [9] and consequently, commercial applications that use raptor codes on the Internet are already in the market.

We must point out here that we have not explicitly designed any forward error correction based data transfer scheme for noisy channels. We have only shown that raptor codes are likely to outperform any other known class of rateless codes in such a scheme. Therefore, a natural direction for future work is the design of a raptor code based protocol and a study of its performance on relevant noisy channels, such as fading channels.

6 Acknowledgements

We would like to thank Amin Shokrollahi for his comments and for reference [13], which also studies raptor codes on noisy channels.

References

[1] R. G. Gallager, *Low Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi and D. Spielman and V. Stemann, "Practical loss-resilient codes," Proceedings of the 29th annual ACM Symposium on Theory of Computing, pp. 150-169, May 1997.

[3] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erausre channel," *IEEE Trans. Inform. Theory,* vol. 48, pp. 3017-3028, Dec 2002.

[4] S.-Y. Chung, G. Forney, T. Richardson, and R. Urbanke, "On the Design of low-density parity check codes within 0.0045 db of the Shannon Limit," *IEEE Comm. Letters,* vol. 5, pp. 58-60, Feb 2001.

[5] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Trans. Inform. Theory*, vol. 20, pp. 388-389, May 1974.

[6] J. Hagenauer, "Rate-compatible punctured convolutional codes and their applications," *IEEE Transactions on Communications,* vol. 36, pp. 389-400, Apr 1988.

[7] C. F. Leanderson, G. Caire and O. Edfors, "On the Performance of Incremental Redundancy Schemes with Turbo Codes," *Proc. Radiovetenskap och Kommunikation 2002,* pp. 57-61, Jun 2002.

[8] M. Luby, "LT- codes," *Proceedings of the 43rd Annual IEEE Symposium on the Foundations of Computer Science*, pp. 271-280, 2002.

[9] A. Shokrollahi, "Raptor codes," preprint 2002. Available online at `http://algo.epfl.ch/index.php?p=output_pubs_XX&db=pubs/pubs_fountain.txt`

[10] A. Shokrollahi, S. Lassen and M. Luby, "Multi-stage code generator and decoder for communication systems," U.S. Patent application number 20030058958, Dec 2001.

[11] T. Richardson and R. Urbanke, "The capacity of LDPC codes under message passing decoding," *IEEE Trans. Inform. Theory,* vol. 47, pp. 595-618, Feb 2001.

[12] P. Maymounkov, "Online codes," NYU Technical Report TR2003-883, Nov 2002. Available online at `http://www.scs.cs.nyu.edu/~petar`

[13] O. Etesami, M. Molkaraie and A. Shokrollahi, "Raptor codes on symmetric channels," preprint 2003. Available online at `http://algo.epfl.ch/index.php?p=output_pubs_XX&db=pubs/pubs_fountain.txt`