Performance Analysis of Fountain codes with Robust Soliton Distribution for Erasure Channels

Geetha Prakash[1], Mahesh Nayak[4]
Dept of ECE
PES Institute of Technology
Bangalore, India
geetha.prakash@pes.edu

Murlidhar Kulkarni[2], U Sripati[3]
Dept of ECE
National Institute of Technology
Suratkal, India
mkuldce@gmail.com

***Abstract*—Fountain Codes are essentially Forward Error Correction (FEC) codes, which were developed for applications involving multicasting or for delivering large amounts of content to multiple recipients simultaneously. FEC increases the reliability of a system in a noisy environment. Luby Transform codes (LT codes) which come from a new coding family of the Fountain codes are the first realization of rateless codes which can generate potentially limitless code words from data. In this paper, a performance analysis of Fountain codes for erasure channels, which use the Robust Soliton distribution for encoding the data packets and belief propagation to decode the received packets has been carried out Results show that the overhead during decoding is around 11.2%.**

Keywords: Fountain codes, rateless codes, LT codes, erasure channels, soliton distribution, belief propagation

I. INTRODUCTION

Fountain codes are a new family of codes designed for protection of data in noisy environments where the noise level is not known apriori. The Fountain code produces limitless streams of output symbols, which are a function of the input symbols, and the output symbols are generated independently and randomly according to a distribution, which is chosen by the designer. The overhead of the decoding algorithm for a fountain code over a communication channel is measured as the fraction of the additional symbols needed to achieve the desired error rate [1]. Fountain codes work by sending packets as long as needed for decoding, with the ability to generate a practically infinite number of packets. This is called the rateless property of the codes. The transmitter sprays packets at the receiver without any knowledge of which packets are received. Once the receiver has received any N packets, where N is just slightly greater than the original file size K, the whole file can be recovered [2]. An appropriately designed Fountain code eliminates the need for a sender know details about the quality of the channel to a particular receiver [3].

II. ENCODING

A binary erasure channel is such that each transmitted packet is either correctly received without errors or entirely lost with probability 'p' which identifies the channel erasure probability [4]. Fig 1 shows the source packets $x_1,x_2,x_3\ldots x_5$ which are the five source packets that need to be transmitted between a sender and multiple receivers and consider a zero erasure probability, p=0. Each transmitted packet is obtained as a linear combination of the subset of the input packets x_1, $x_2\ldots x_5$. The number of x_i's to be summed together is picked according to a "degree distribution", whose design is to obtain good performance [5]. Tanner graph representation for the message and the encoded words is indicated in Fig 1

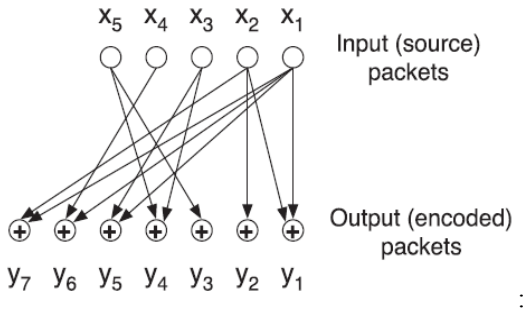

Fig.1 Encoding the input packets to output packets and representation of Tanner graph

The relationship between x_i and y_i is obtained by using a transformation matrix. Each encoded packet is associated with an encoding vector which is the i^{th} row of G. Encoding vectors are sent along with the encoded packets and used at the receivers to invert the transformation to retrieve x_1 through x_5

A. Encoding procedure

The degree of distribution is a probability distribution, which determines the number of input packets to combine to form any given encoded packet. After a degree of distribution has been chosen, as many packets are picked among the K given as input. The encoded packet is obtained is obtained through bit wise modulo 2 sum of these packets. This is done until the desired numbers of packets are encoded. [4]. Fountain codes are rateless since the number of encoded packets that can be generated from the source message is potentially limitless. The number of packets generated can be decided on the fly. The encoding process defines a graph connecting encoded packets to source packets.

978-1-4244-9477-4/11/$26.00 ©2011 IEEE

III. DECODER

The decoding is done by message passing using Tanner graphs. Tanner graphs are bipartite graphs. This means that the nodes of the graph are separated into two distinctive sets and the edges connect nodes of two different types. The two types of nodes in a Tanner graph are called variable nodes (v-nodes) and check nodes (c-nodes) [7]. Decoding is done by passing messages between the nodes. The source packets are called s_k and encoded packets are called check nodes, denoted as t_n.

The decoding is done as follows:

- A check code that is connected to only one source packet s_k is found. If such a node does not exist, the decoding process halts at this point.
- Set $s_k=t_n$
- Add s_k to all check nodes that are connected to s_k.
- Remove the edges connected to the source packet s_k
- Repeat step 1 until all s_k are determined.

IV. DESIGNING THE DEGREE OF DISTRIBUTION

The degree of the probability distribution ρ (d) is a critical part of the fountain codes. Some encoded packets must have high degree to ensure that there are not some source packets that are connected to no one. Many packets must have low degree; so that the decoding can start and keep going and so that the total number of addition operations in encoding and decoding is kept small. The encoding and decoding complexity vary linearly with the number of edges in the graph. The encoding and decoding complexity are both dependent on the number of edges in the graph. In order for the decoding to be successful, every source packet must have at least one edge in it. The encoder sends edges into source packets at random, so the number of edges must be at least of order $K\log_e K$. If the number of packets received are close to Shannon's optimal K and decoding is possible, the average degree of each packet must be at least $\log_e K$ and the encoding and decoding complexity of an LT code will definitely be at least $K\log_e K$.It has been shown by Luby that this bound on complexity can be achieved by a careful choice of degree distribution [2].

Ideally the decoding process should run in such a way that in every iteration, only one codeword with degree 1 appears after a message symbol has been recovered.

This goal can be achieved by the ideal Soliton distribution

$$\rho(d) = \begin{cases} 1 & for \quad d = 1 \\ \dfrac{1}{d(d-1)} & for \quad d = 2,3,...K \end{cases} \tag{1}$$

However this algorithm works poorly in practice. The degree one codeword may disappear at some point of time of the decoding process. This can be solved by increasing the number of code words of degree one and slightly modify the ideal distribution. A positive function is defined as

$$\tau(d) = \begin{cases} \dfrac{S}{K}\dfrac{1}{d} & for\ d = 1,2,...(K/S)-1 \\ \dfrac{S}{K}\ln(S/\delta) & for\ d = K/S \\ 0 & for\ d > K/S \end{cases} \tag{2}$$

where S is the new amount of code words with degree 1. This is added to the ideal Soliton distribution μ

$$\mu(d) = \frac{\rho(d)+\tau(d)}{Z} \quad where\ Z = \sum_d \rho(d)+\tau(d) \tag{3}$$

Equation (3) describes the robust Soliton distribution[2].

The number of encoded packets required to at the receiving end to ensure that the decoding can run to completion, with probability at least (1-δ) is K'= KZ .

V. IMPLEMENTATION

A. *Encoding the data into Fountain codes*

The Soliton distribution [2,5,6] is used to decide the degree checks and also in turn the generator matrix. We used the generator matrix to encode the data. The LT code is a kind of Low Density Parity Check (LDPC) code, which in turn is produced from sparse matrices. It has been found that the smaller the size of the sparse matrices, the performance of the system deteriorates. As a result large matrices are required for the generation of the sparse matrices. Here we use the Soliton distribution to decide the sparse matrices depending on the number of degree one checks S. the data is obtained by the simple matrix multiplication and modulo 2 additions of bits.

B. *Decoding of Fountain Codes:*

As the data is transmitted across the channel there are chances of packets getting erased or lost and there are chances of packets getting corrupted. But in the case of erasure codes all the messages are either completely certain or completely uncertain. Decoding is done by message passing. Uncertain messages assert that a message packet s_k could have any value, with equal probability; certain messages assert that s_k has a particular value with probability one [8, 9, 10, 11].

In the case of the decoder also, the same generator matrix is used as in the encoder. Similar to encoding we start to decode the data using the generator matrix. The generator matrix is represented using Tanner graphs [6]. Initially there is a search for a single edge in the generator matrix. After the single edge is identified all edges that are connected to this source packet are removed before which, s_k, the source packet is added to the check nodes. If a single edge does not exist, the decoding algorithm halts and fails to recover all the packets.

VI. SIMULATION RESULTS

The Robust Soliton distribution has been used to determine the degree of distribution for the encoding of the message. The Robust Soliton distribution and its performance as obtained is given in Fig 2.

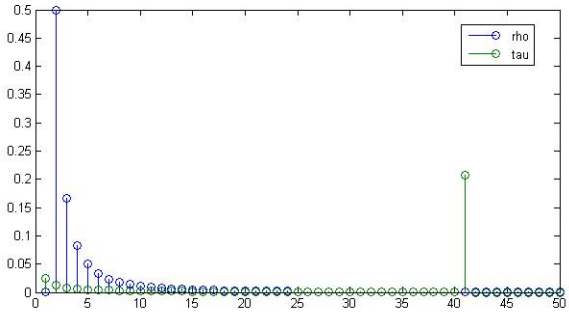

Figure 2: Plot of ρ(d) and τ(d) as given by Equations (2) and (3)

Fig 2 shows the plot of the distributions ρ(d) and τ(d) for the case of K=10000, c=0.2, δ = 0.05 which gives S=244 and K/S = 41 and Z ≅ 1.3. The distribution τ is largest at d=1 and d/K =S.

The small-d end of τ has the role of ensuring that the decoding process gets started, and the spike in τ at d=K/S is included to ensure that every source packet is likely to be connected to a check node at least once.

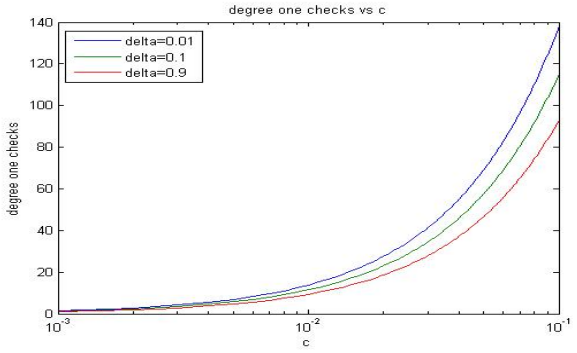

Figure 3: Number of degree one checks against the parameters c and δ

Fig 3 shows the number of degree one checks for various values of decoder failure probability δ.

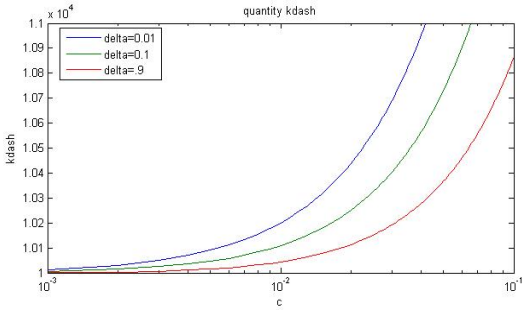

Figure 4: Plot of c versus K' for constant δ , K' is the excess packets needed to decode

Fig 4 shows the plot of the constant 'c' versus the quantity 'k'' for a constant δ where k' is the excess packets required to decode the data and δ is the probability that the receiver will not be able to decode the file. The allowable decoder failure probability δ has been set quite large, because the actual failure probability is much smaller than is suggested by Luby's conservative analysis. From the graph it is evident that as the probability of decoding failure δ increases the number of degree one checks also decreases. This is because in case any of the degree one check bits failure the data will be decoded wrong.

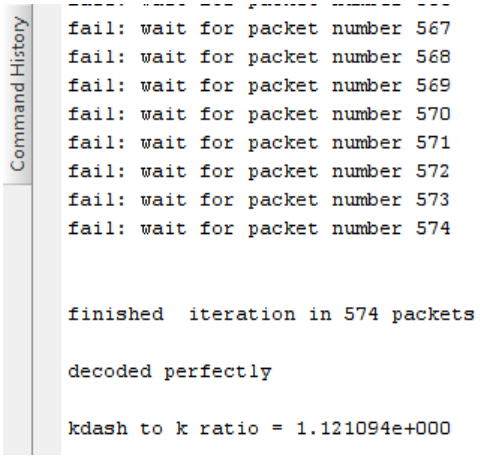

Figure 5: Snapshot to show the decoder can successfully decode with excess packets

In order to show that all of the encoded packets are not necessary and only little above K packets, are required for decoding, data of K packets is encoded into n packets. Decoding the data starts when K+1 packets are initially received and find whether the receiver is able to decode the packet. If it fails another packet is sent and repeated until the data is completely decoded as shown in Fig 5. As per Luby, the overhead should be about 10%. From our implementation as shown for one of the cases, a ratio of 11.2% has been obtained which is very close to what has been specified by Luby.

Decoding has been successfully implemented for transmission of 512 packets, for about 50 iterations, and the histogram shown in Fig 6 was obtained. The ratio is repeated maximum at around 560 to 580 of received packets, which are in excess of the 512 encoded packets.

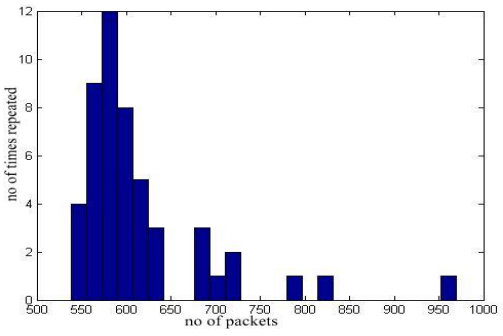

Figure:6 Histogram to show the number of excess packets required for successful decoding and the number of repetitions of the successful decoding has been completed

The results from the histogram show that a maximum of 560 to 580 packets were required to decode when 512 packets

were sent, irrespective of how the encoding was done as long as the encoding scheme satisfied the rules for degrees of distribution as specified by Luby.

VII. CONCLUSIONS

In this paper, a performance analysis of Luby Transform codes from the family of Fountain codes has been implemented for erasure channels using Robust Soliton distribution for encoding and belief propagation for decoding. The scheme employs an LT code with low complexity of decoding. LT matrices have been generated using the sparse properties and also those, which satisfy the Robust Soliton distribution and hence allows the use of Belief propagation for decoding. The simulation results show overheads very close to those proposed by Luby.

REFERENCES

[1].Giuseppe Caire, Shlomo Shamai, Amin Shokrollahi, And Sergio VerdU, "Fountain Codes For Lossless Data Compression", DIMACS Series In Discrete Mathematics And Theoretical Computer Science, 2005

[2]. D.J.C. Mackay, "Fountain Codes ", IEE Proc.-Commun., Vol. 152, No. 6, Dec 2005 , 1062-1068

[3]. Amin Shokrollahi, "Raptor Codes", IEEE Transactions on Information Theory,vol 52,no.6,June 2006, pp 2551- 2567

[4]. Michele Rossi, Nicola Bui, Giovanni Zanca, Luca Stabellini,Riccardo Crepaldi, Michele Zorzi, "SYNAPSE++: Code Dissemination In Wireless Sensor Networks Using Fountain Codes", IEEE Transactions On Mobile Computing, Vol. 9, No. 12, December 2010, 1749-1765

[5]. Michael Luby , "LT Codes" , The 43rd Annual IEEE symposium on Foundations of Computer Science , 2002, FOCS'02, pp 271-282

[6] .Trienko Lups Grobler, "Fountain Codes and their typical application in wireless standards like EDGE",ME Thesis, July 2008,University of Pretoria, Dept of Electrical, Electronic and Computer Engineering.

[7]. Todd K Moon, "Error correction Coding, Mathematical methods and Algorithms ",Wiley, 2005

[8].David J.C MacKay, "Information Theory, Inference and Learning Algorithms",Cambridge University Press, 2003.

[9]. Tuomas Tirronen, "Fountain Codes: Performance Analysis And Optimization", Phd Dissertation March 2010, Helsinki University of Technology Department of Telecommunications and Networking.

[10]. Han Wang, "Hardware Designs For LT Coding ", MSc Thesis, 2006 Circuit And Systems Department Of Electrical Engineering , Mathematics And Computer Science Delft University Of Technology

[11]. Anya Apavatjrut, Claire Goursaud, Katia Jaffrès-Runser, Cristina Comaniciu, and Jean-Marie Gorce, "Toward Increasing Packet Diversity for Relaying LT Fountain Codes in Wireless Sensor Networks", IEEE Comm Letters, Vol. 15, No. 1, January 2011, Pages 52-54.