A Practical Implementation of Fountain Codes over WiMAX Networks with an Optimised Probabilistic Degree Distribution

Jaco du Toit and Riaan Wolhuter
Department of Electrical and Electronic Engineering
Stellenbosch University
Stellenbosch, South Africa
e-mail: 14409607@sun.ac.za, wolhuter@sun.ac.za

Abstract—**Recently, rateless codes have attracted much attention in the communications research community. The most well known being Luby transform codes, were the first practical realisation of record-breaking sparse-graph codes for binary erasure channels. These codes have the advantage of not requiring a priori knowledge of specific channel conditions and lends itself to application in nondeterministic wireless networks. This paper revisits the Luby transform fountain code, predecessor of the well known Raptor codes, and proposes a novel parameterised probabilistic degree distribution, which is used in the encoding process, along with the belief propagation decoding algorithm. By combining piecewise-defined convex functions and running a non-symmetric Kullback-Leibler divergence measure between the expected and actual degree distributions, we optimise our degree distribution and substantiate a significant reduction in reception overhead and symbol operations. This will support such forward error correction codes in efficient multimedia communication systems. Our proposition was implemented over a WiMAX network and the practical results obtained indicate that a few conditions are sufficient to define an optimal encoding process.**

Keywords-Rateless Codes; Universal Codes; Belief Propagation; Parameterised Degree Distribution.

I. INTRODUCTION

Binary linear rateless coding is an encoding method that can generate potentially infinite parity check bits for any given fixed-length binary sequence as they do not have a fixed rate as the case for conventional codes. Fountain codes constitute a class of rateless codes, which were first discovered in by Luby. [1] Luby Transform (LT) codes are linear rateless codes that transform k information symbols into infinite coded symbols. Regardless of the statistics of the erasure events on the channel, we can send as many encoded packets as needed in order for full recovery of the source data. Typically $N = k(1 + \varepsilon)$ packets are needed to successfully decode the original input message with a certain degree of probability where ε is the overhead. Each encoded symbol is generated independently and randomly, where the randomness is governed by the so-called Robust Soliton distribution. Luby's main theorem proved that there exists bounds around the belief propagation decoding failure probability as a function of reception overhead, that for a value c given N received packets, the decoding algorithm will recover the k source packets with probability $1 - \delta$. [1] [8] For large k (thousands), the Robust Soliton distributions have shown good performance. For smaller k Markov chain approaches have been implemented, which also showed good results. One conclusion to this study was that in a well-chosen parametric form of the degree distribution, just a few parameters need to be tuned in order to get maximal performance. [3] Given the work already done, optimal forms of parameterised degree distributions for different message lengths continue to provide an interesting problem. In this paper we will investigate a new parameterised degree distribution shaped by convex functions and test its performance on a WiMAX network in real world scenarios, where random channel noise introduce packet loss.

The rest of this paper is organised as follows: In Section 2, we review the theory of rateless encoding and believe propagation (BP) decoding, in particular the LT process and probabilistic degree distributions (PDD). In Section 3, we present our proposed optimised degree distribution, utilising a set of piecewise convex functions shaping the ideal degree distribution to an improved solution as presented in literature, after reviewing related performance enhancing methods. We analyse the computational cost, and performance of our proposition in Section 4 and show results of emulation and practical implementation of our suggested solution. We finally state our conclusion and future work in Section 5.

II. PRELIMINARIES

A. LT codes

LT codes proposed by Luby in 1998 are the first codes fully realising the digital fountain concept. [1][4] They are rateless, i.e., the number of generated encoded packets are potentially limitless, and encoded symbols are generated on the fly. [8]

1) Encoding of LT code: Randomly choose the degree $\boldsymbol{d}$ of the packet from a key element in the process, the so-called *degree distribution*. The encoded symbol is then generated by choosing $\boldsymbol{d_n}$ blocks from the original file uniformly at random. The value of the encoded symbol is the bitwise exclusive-or of the $\boldsymbol{d_n}$ neighbours. The encoding operation defines a irregular sparse graph connecting encoded symbols to source symbols.

2) Decoding of LT codes: Decoding is done iteratively by using the Belief Propagation decoding algorithm. First we *release* a encoded symbol of degree-one, with *complete certainty*, and subtract the connected symbols from each received packet by taking an exclusive-or between the packet and the known symbols. This procedure removes all edges connected to the source packets and is repeated until all source symbols are recovered. The set of covered input symbols that have not yet been processed is called the ripple. This process is well illustrated in most fountain code

 ISBN: 978-1-61208-166-3

literature. [5][6][8] Algorithm 1 and 2 demonstrates the encoding and decoding procedures respectively.

Algorithm 1: LT Encoding

repeat
  choose a degree d from degree distribution *p(d)*
  choose uniformly at random d input symbols $n(i_1),..,n(i_d)$.
  calculate value $n(i_1)$ xor $n(i_2)$ xor ... xor $n(i_d)$
until stop bit received

Algorithm 2: LT Decoding

repeat
  if d = 1 packet in buffer
   n(j) ← recover j
  for all n(j) in buffer : v includes n(j) **do**
   d ← d - 1   (reduce degree)
   v ← v xor n(j) (update value)
  end for
until all input symbols recovered

The complexity of BP, prominent in the decoding of LT codes, is essentially the same as the complexity of the encoding algorithm [1] i.e., there is exactly one symbol operation performed for each edge in the bipartite graph between the source symbols and the encoded symbols during both encoding and decoding. Therefore, the computational complexity of this algorithm is linear in the average degree of the degree distribution multiplied by the size of the source block. [6] BP will, however, fail when output nodes of degree-one exhaust and various algorithms i.e., Gaussian Elimination (GE) have been suggested [5][8][11] to counter this failure. However, this adds undesirable running time where fast decoding is required, especially for large matrices. For small code block lengths GE could be used efficiently, since BP requires a larger overhead for small block sizes. For this reason it is extremely important to find a degree distribution to effectively reduce reception overhead and the number of symbol operation for any block size.

B. Degree Distributions

The LT process described in [1] helps explain the design and analysis of a good degree distribution for the LT codes by comparing the process to the well known balls in bins problem, where encoded symbols are analogous to balls and input symbols are analogous to bins. The analysis of this problem shows that $\mathbf{N} = \boldsymbol{k}\mathbf{ln}(\boldsymbol{k}/\boldsymbol{\delta})$ balls are needed on average to ensure that each of the $\boldsymbol{k}$ bins is covered by at least one ball, with probability at least $\mathbf{1-\delta}$. This classic process can be viewed as a special case of the LT process, where all encoded symbols have degree-one and released simultaneously. It is shown in [1] that the Ideal Soliton distribution in (1), ensures that just over $\boldsymbol{k}$ encoding symbols with the sum of their degrees being $\boldsymbol{O(k\mathbf{ln}(k/d))}$ will suffice to cover all $\boldsymbol{k}$ input symbols and produces the least number of symbol operations.

Luby further explained that the goal of the degree distribution design is to slowly release encoding symbols as the process evolves and to keep the ripple size small to prevent redundant coverage. The ripple should also be large enough to prevent it from disappearing prematurely. An ideal property required by a good distribution is that input symbols are added to the ripple at the same rate as they are processed. The Ideal Soliton in Fig. 1 displays this desired behaviour.

$$\rho(d) = \begin{cases} \frac{1}{k}, & d = 1 \\ \frac{1}{d(d-1)}, & d = 2,3,\ldots,k \end{cases} \qquad (1)$$

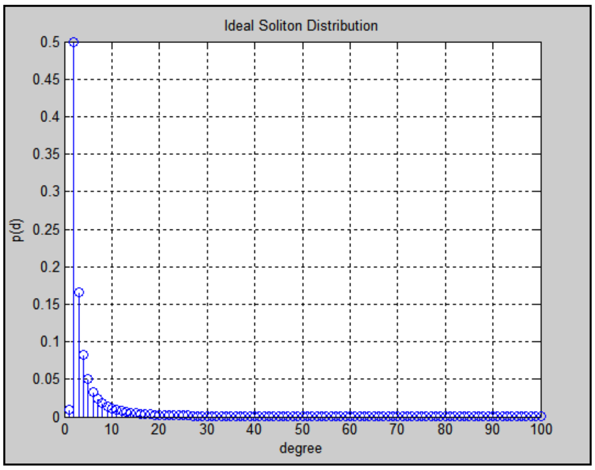

Figure 1: Ideal Soliton degree distribution for k = 100 input symbols.

The expected degree of an encoding symbol for this distribution is the harmonic sum up to $\boldsymbol{k}$:

$$\sum_{d=1}^{k} \rho(d) \approx \ln(k) \qquad (2)$$

This means that in order to cover all the input symbols the degrees of all the encoding symbols needs to be around $\boldsymbol{k}\mathbf{ln}(\boldsymbol{k})$ and the Ideal Soliton compresses this into the least number of encoding symbols possible. This distribution, however ideal in theory, turned out to be quite fragile in practice, since the slightest variation in its expected behaviour can cause the ripple to disappear prematurely.

The Robust Soliton distribution from [1] ensures the ripple size stays large enough at each decoding step so that it never disappears completely and that few released encoding symbols are redundantly covered by input symbols already in the ripple. The Robust Soliton distribution (3) was designed so that the expected ripple size is roughly $\mathbf{ln}\,(\frac{\mathbf{k}}{\boldsymbol{\delta}})\sqrt{\mathbf{k}}$ throughout this process. Let $\mathbf{R} = \mathbf{c}\sqrt{\mathbf{k}}\mathbf{ln}(\frac{\mathbf{k}}{\boldsymbol{\delta}})$, where $\boldsymbol{c}$ is some suitable constant of order one.

$$\tau(d) = \begin{cases} \frac{R}{dk}, & d = 1,\ldots,\frac{k}{R}-1 \\ \left(\frac{R}{k}\right) ln\left(\frac{R}{\delta}\right), & d = k/R \\ 0, & d = \frac{k}{R}+1,\ldots,k \end{cases} \qquad (3)$$

The small-$\boldsymbol{d}$ end of $\boldsymbol{\tau}$ ensures that the decoding process starts with a reasonable ripple size and the larger spike at $\mathbf{d} = \mathbf{k/R}$ ensures all source packets are connected, keeping the ripple large enough. The expected number of encoded

 ISBN: 978-1-61208-166-3

packets required at the receiver to ensure that the decoding can run to completion, with probability 1 - $\boldsymbol{\delta}$ has now increased to $\mathbf{N} = \boldsymbol{k}\mathbf{Z}$. Where the normalising factor becomes $\mathbf{Z} = \sum_{\mathbf{d}} \mathbf{p}(\mathbf{d}) + \boldsymbol{\tau}(\mathbf{d})$. The Robust Soliton distribution is shown in Fig. 2.

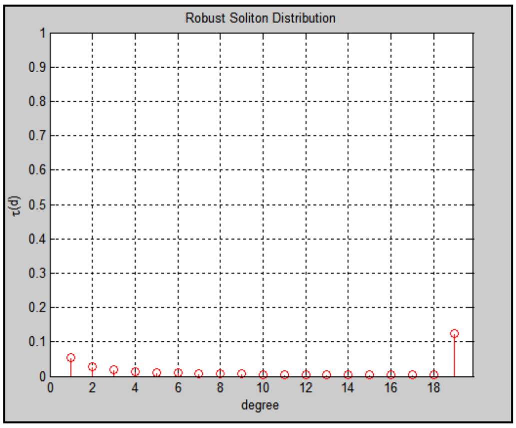

Figure 2: Robust Soliton degree distribution for k = 100, c = 0.1 and δ = 0.5.

Theoretical analysis of the properties of the Robust Soliton distribution is given in [1] where pessimistic estimates was used to prove the amount of encoding symbols necessary for full recovery of an input message. This was simplified to be $\mathbf{N} = \mathbf{k} + O(\sqrt{\mathbf{k}}\,\mathbf{ln}\,(\frac{\mathbf{k}}{\boldsymbol{\delta}})^{2})$ and the average degree of an encoding symbol was shown to be $\mathbf{D} = O(\mathbf{ln}(\frac{\mathbf{k}}{\boldsymbol{\delta}}))$. A typical Robust Soliton distribution, normalised using (4), is illustrated below in Fig. 3.

$$\mu(d) = \frac{(\rho(d)+\tau(d))}{Z} \tag{4}$$

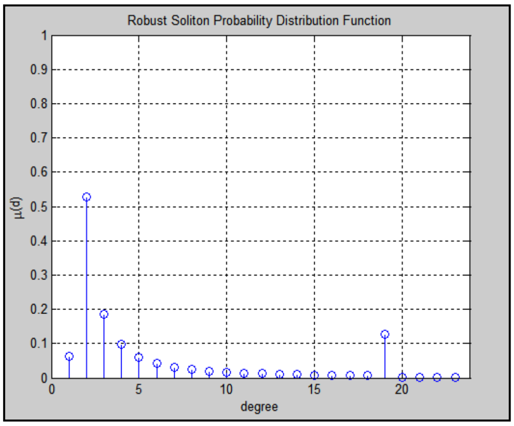

Figure 3: Robust Soliton degree distribution for k = 100, c = 0.1 and δ = 0.5.

A lot of previous work studying the various performance aspects of LT codes and their applications [7][9][10] have implicitly accepted the Robust Soliton degree distribution as sufficient and optimal. This is a sound assumption from the theoretical proofs presented in [1]. However, many of these studies present limited effort in deriving a optimal parameterised form of the degree distribution or even an practical implementation of a general LT code over an actual network. Our work is centred around the potential use of LT codes as an AL-FEC for media distribution, we have chosen not to test $\boldsymbol{k}$ values larger than 1000. Too much latency is introduced while waiting for the large amounts of encoded symbols, and in various other works we have seen that very small values introduce high reception overhead. Therefore, we have chosen to test both $\boldsymbol{k}$ = 100 and $\boldsymbol{k}$ = 1000 block sizes. The analysis of the Robust Soliton distribution based on probability and statistics is sound only if $\boldsymbol{k}$ is infinite. In practice however, the behaviour of the LT code will not match the mathematical analysis exactly, especially for small $\boldsymbol{k}$. Typical results for the Robust Soliton degree distribution is illustrated below in Table I. The constant $\boldsymbol{c}$ = 0.1 were chosen as it produced an acceptably low standard deviation and overhead mean.

TABLE I. TYPICAL RESULTS FOR THE ROBUST SOLITON DEGREE DISTRIBUTION

Input Symbols (k)	δ	Z	Mean (N)	Std (N)	Mean (Symbol Operations)	Std (Symbol Operations)
100	0.01	1.89	172.49	17.64	1007	166
	0.1	1.51	149.26	14.41	858	153
	0.9	1.24	135.69	13.21	704	139
1000	0.01	1.43	1373.65	39.92	14364	1232
	0.1	1.28	1256.70	33.37	12521	1113
	0.9	1.16	1171.99	33.11	10488	1128

Interestingly enough we see that by increasing $\boldsymbol{\delta}$ beyond 1 the efficiency increases even more. In the original case where it is used to predict failure of decoding, this parameter becomes more accurate only when a linear congruential generator is used for random number generation. [10]

The focus of our work is on finding a more efficient parameterised degree distribution to reduce the number of symbol operations and amount of overhead with small deviation.

III. PROPOSED OPTIMISED DEGREE DISTRIBUTION

By combining convex functions and the expected ripple size $\mathbf{R} = \mathbf{c}\sqrt{\mathbf{k}}\,\mathbf{ln}(\frac{\mathbf{k}}{\boldsymbol{\delta}})$ from the Luby transform a new set of equations can be derived to shape the Ideal Soliton distribution to optimise the amount of symbol operations and overhead $\boldsymbol{N}$. The expected ripple size determining the position of the spike somewhere on $\boldsymbol{d}$, ensures that all unprocessed input symbols are covered. [1] However, instead of keeping the weight at d = k/R a constant, (6) and (7) distributes the expected area exponentially over k, which maintains a good ripple size throughout the decoding steps by ensuring ample symbol connections. If Z is close to 1, (where Z ≥ 1) we expect the optimal amount of symbol operations. Parameters c_1, c_2 and c_3 determine the curvature and area supplementary to the Ideal Soliton PDD, which is proportional to the average degree of an encoded symbol. Tweaking these parameters leads to an optimal solution if

 ISBN: 978-1-61208-166-3

the correct distributed area is added to the correct location on the degree distribution.

A. Piecewise functions used to shape the PDD

Fig. 4 illustrates the shape of each exponential function given by (5), (6) and (7). The parameters c_1, c_2 and c_3 are used to alter the amplitudes and curvatures of each set. By changing these parameters, the total area under the graph (affecting Z) can be modified to reduce N by keeping $\mathbf{D} \geq O(\ln(k))$.

$$y_1(d) = \left(\frac{c_1}{\sqrt{k}}\right)(c_2)^{-d} \qquad , d = 1, \ldots, k \qquad (5)$$

$$y_2(d) = \left(\frac{c_1}{\sqrt{k}}\right)(4c_2)^{-d+\frac{k}{R}-1}, d = \frac{k}{R}+1, \ldots, k \qquad (6)$$

$$y_3(d) = \left(\frac{c_3}{k}\right)(3)^{d-\frac{K}{R}} \qquad , d = 1, \ldots, \frac{k}{R} \qquad (7)$$

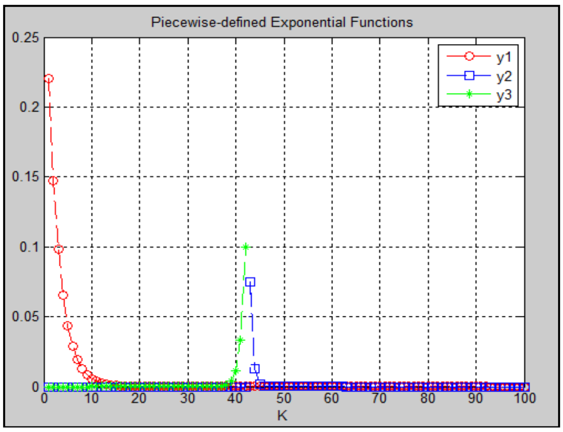

Figure 4: Scaled illustration of piecewise-defined Exponential functions used to shape the new PDD

B. Discrete Kullback-Leibler optimisation approach

The Kullback Leibler distance in (8) can be interpreted as a natural distance function from a "true" probability distribution $\boldsymbol{p}$ to a "target" probability distribution $\boldsymbol{q}$. In each set of decoded samples of $\boldsymbol{N}$, the average of the best degree distributions becomes our target degree distribution. The PDD is shaped accordingly and the process continues recursively until the Kullback Leibler distance converges to zero.

$$D(P \parallel Q) = \sum_i p_i \log_2 \frac{p_i}{q_i} \qquad (8)$$

C. Practical Implementation over WiMAX

Our test setup consisted of a WiMAX micro base station and Si indoor CPE 2.5. Consecutive tests were run to determine the effect of SNR and packet loss on the LT code as an application layer implementation. The simple network management protocol (SNMP) was used to retrieve channel information from the base station's client burst profiles. The WiMAX system slots in this receiver to transmitter feedback for adaptive physical layer modulation purposes. The WiMAX network setup and AL-FEC screenshots are illustrated in Figs. 5 - 7.

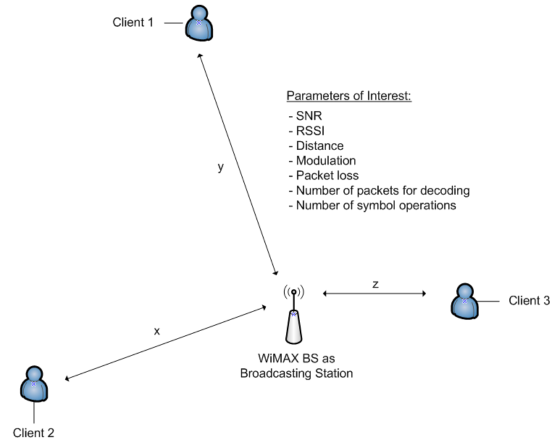

Figure 5:Illustration of the WiMAX Test Setup

In almost all deployed IPTV linear media broadcasting services, audio and video streams are multiplexed into some codec transport stream. Our AL-FEC was implemented over the UDP stream shown in Figs. 6 - 7.

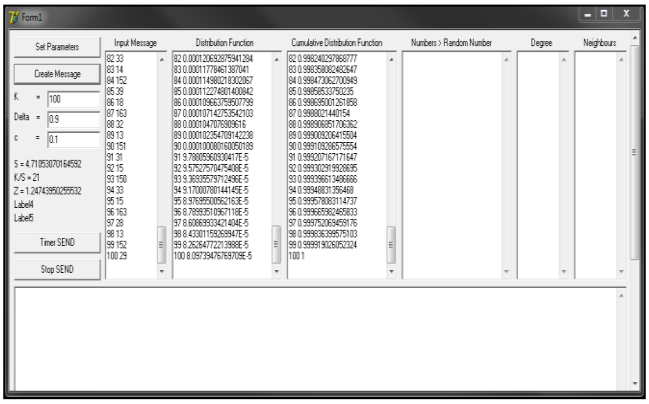

Figure 6: Application Layer UDP encapsulated LT Fountain Encoder

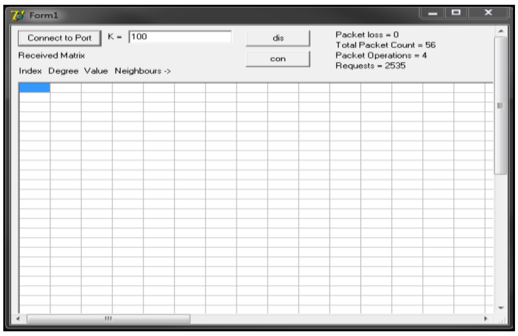

Figure 7: Application Layer UDP encapsulated LT Fountain BP Decoder

The radio link is a quickly varying link, often suffering from great interference. Physical channel conditions such as pathloss, fading and shadowing etc. place constraints on wireless signal transmissions. WiMAX inherently utilises advance FEC techniques such as the concatenated Reed-Solomon Convolutional codes to overcome such destructive effects. For the purpose of our tests the application layer measured packet loss is an indication of the system suffering from packet loss after the inherent FEC layers built in WiMAX.

IV. RESULTS

Figs. 8 - 12 and Figs. 19 - 23 shows simulated and practical results of the improved degree distribution ***y(d)*** for ***k*** = 100 and ***k*** = 1000. Figs. 11 - 18 and Figs. 22 - 23 illustrates practical results over the WiMAX network.

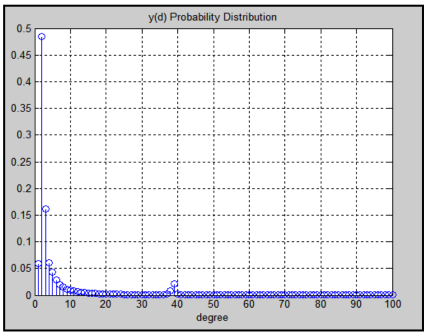

Figure 8: k=100, c_1=1.08, c_2=2.316, c_3=1, δ=4, c=0.08, Z=1.12

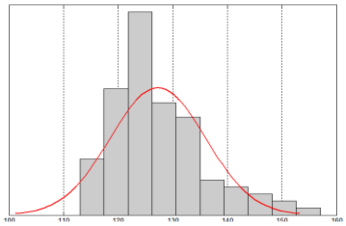

Figure 9: Simulated number of packets N (mean=127.2, std=8.6)

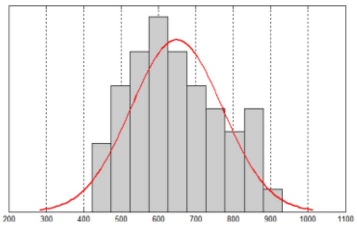

Figure 10: Simulated number of symbol operations (mean=648, std=121.8)

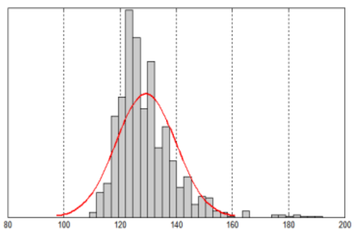

Figure 11: Number of packets N (mean=129.2, std=10.6)

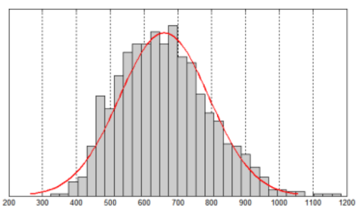

Figure 12: Number of symbol operations (mean=660, std=132.1)

Figs. 13 - 18 indicate practical result obtained over WiMAX (CPE 800m from BS) for ***k*** = 1000**,** ***c*** = 0.1 and **δ** = 0.9, using the Robust Soliton degree distribution. From these measurements it is clear that the fountain code did not suffer significantly when introduced to a drastic reduction in SNR.

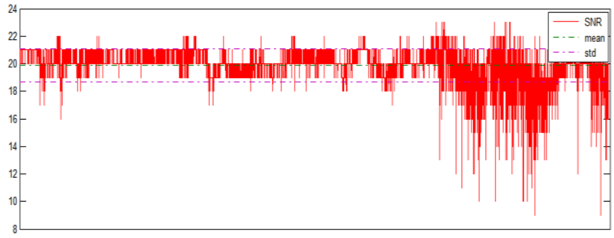

Figure 13: DL Signal to Noise Ratio

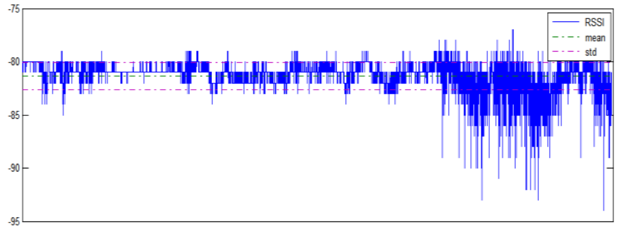

Figure 14: DL Received Signal Strength Indication

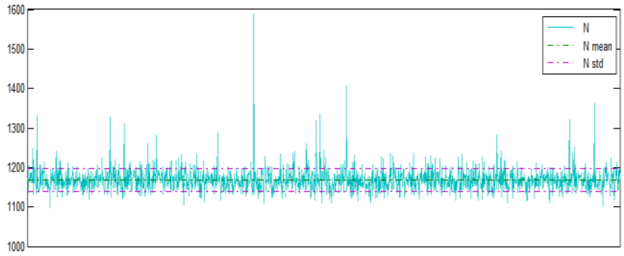

Figure 15: Number of Packets N

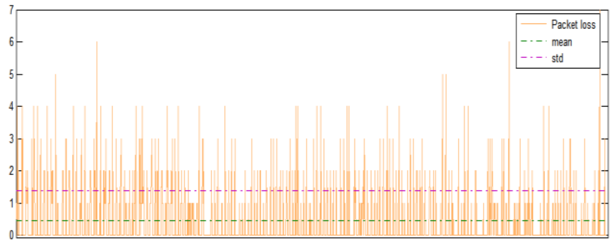

Figure 16: Packet loss

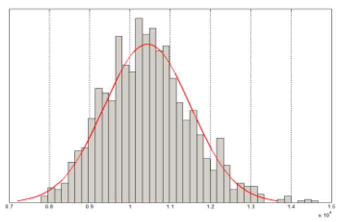

Figure 17: Number of symbol operations (mean=10434 , std=1076)

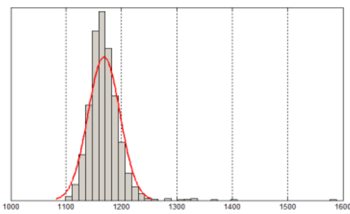

Figure 18: Number of packets N (mean=1168 , std=28.8)

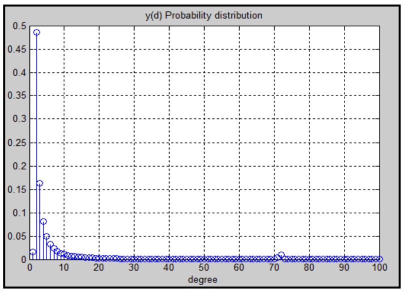

Figure 19: k=1000, c_1=1, c_2=2, c_3=9.5, δ=4, c=0.08, Z=1.04

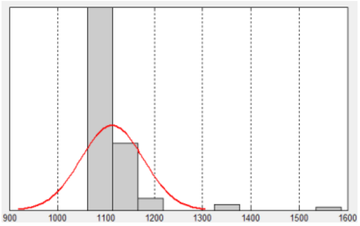

Figure 20: Simulated number of packets N (mean=1112.7, std=64.6)

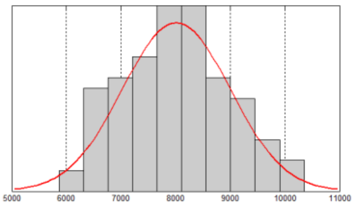

Figure 21: Simulated number of symbol operations (mean=8012.5, std=987.2)

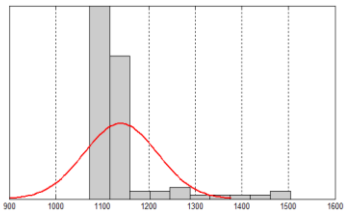

Figure 22: Number of packets N (mean=1139, std=76)

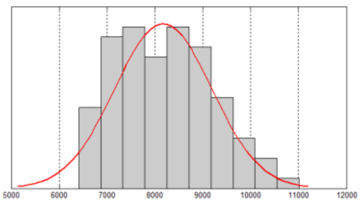

Figure 23: Number of symbol operations (mean=8174, std=1011)

TABLE II. COMPARISON BETWEEN ROBUST SOLITON AND OPTIMISED PDD

Input Symbols (k)	PDD	Mean	Std	Mean	Std
		N		Symbol Operations	
100	y(d)	127.20	8.60	648	121
	μ(d)	135.69	13.21	704	139
1000	y(d)	1112.70	64.60	8012	987
	μ(d)	1373.65	39.92	14364	1232

V. CONCLUSION AND FUTURE WORKS

In this paper, we presented an improved degree distribution by shaping the theoretically optimal distribution with convex functions until optimal results were obtained. Only five parameters were sufficient to define an optimal encoding process to reduce decoding cost and overhead. The practical and simulated results shown is a significant improvement over LT codes using the popular Robust Soliton as degree distribution. To the best of our knowledge we also introduced the first practical implementation of fountain codes over a WiMAX network, and presented useful data regarding the transmission thereof. Regarding LT codes, it turns out that BP alone is not efficient enough to get very tight bounds on decoding failure probability as a function of reception overhead. This was the rationale behind the Raptor codes [6], which combines a weak LT code with a traditional block code and decodes with both GE and BP. Future investigations include the analysis of Raptor codes and the design of alternative degree distributions with desirable properties in terms of both overhead and decoding complexity.

VI. REFERENCES

[1] M. Luby, “LT Codes,” in Proceedings of The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002, pp. 271–282, doi: 10.1109/ICICIC.2008.250.

[2] M. Rossi, G. Zanca, L. Stabellini, R. Crepaldi, A. F. Harris, and M. Zorzi, "SYNAPSE: A Network Reprogramming Protocol for Wireless Sensor Networks using Fountain Codes, July 2008, doi: 10.1109/SAHCN.2008.32.

[3] E. Hyytia, T. Tirronen, and J. Virtamo, "Optimal Degree Distribution for LT Codes with Small Message Length", in IEEE INFOCOM mini-symposium, pp. 2576-2580, doi: 10.1109/INFCOM.2007.324.

[4] J. W. Byers, M. Luby, and M. Mitzenmacher, “A Digital Fountain Approach to Asynchronous Reliable Multicast” in IEEE Journal 2002, doi: 10.1109/JSAC.2002.803996.

[5] D.J.C. MacKay, "Capacity Approaching Codes Design and Implementation", Fountain Codes, IEE Proc-Commun, Vol. 152. No. 6, December 2005, doi: 10.1049/ip-com:20050237.

[6] A. Shokrollahi, "Raptor Codes", Foundation and Trends in Communications and Information Theory, IEEE Information Theory Society, doi: 10.1109/TIT.2006.874390.

[7] H. Tarus, J. Bush, J. Irvine, and J. Dunlop, "Exploiting Redundancies to Improve Performance of LT Decoding", 2008 IEEE, doi: 10.1109/CNSR.2008.81.

[8] D.J.C. MacKay, "Information Theory, Inference, and Learning Algorithms", Cambridge University Press 2003, August 2004.

[9] D. H. Wang, “Hardware Designs for LT Coding”, Masters Dissertation, Delft University of Technology, 2006.

[10] C. Harrelson, L. Ip, and W. Wang, “Limited Randomness LT Codes,” Proceedings of the 41st Annual Allerton Conference on Communication Control and Computing, 2003.

[11] V. Bioglio, M. Grangetto, R. Gaeta, and M. Sereno, "On the fly Gaussian Elimination for LT Codes", doi: 10.1109/LCOMM.2009.12.091824.