Sejdinovic, D., Piechocki, R. J., & Doufexi, A. (2009). AND-OR tree analysis of distributed LT codes. In IEEE Information Theory Workshop on Networking and Information Theory, 2009 (ITW 2009), Volos, Greece. (pp. 261 - 265). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/ITWNIT.2009.5158583

AND-OR Tree Analysis of Distributed LT Codes

Dino Sejdinović, Robert J. Piechocki and Angela Doufexi
Centre for Communications Research
Department of Electrical & Electronic Engineering
University of Bristol, Bristol, UK
Email: {d.sejdinovic, r.j.piechocki, a.doufexi}@bristol.ac.uk

***Abstract*—In this contribution, we consider design of distributed LT codes, i.e., independent rateless encodings of multiple sources which communicate to a common relay, where relay is able to combine incoming packets from the sources and forwards them to receivers. We provide density evolution formulae for distributed LT codes, which allow us to formulate distributed LT code design problem and prove the equivalence of performance of distributed LT codes and LT codes with related parameters in the asymptotic regime. Furthermore, we demonstrate that allowing LT coding apparatus at both the sources and the relay may prove advantageous to coding only at the sources and coding only at the relay.**

I. Introduction

Fountain codes [1] [2], [3] are an efficient and robust application layer forward error correction solution for data transmission over packet erasure networks. Unlike traditional coding schemes, fountain codes are able to adapt their rate on-the-fly - they are *rateless* in the sense that a potentially limitless number of encoded packets can be generated from the source data. This makes fountain codes particularly suitable for multicasting and broadcasting applications where users may experience different channel characteristics. LT (Luby Transform) codes [2] are the first practical realization of fountain codes and offer capacity approaching performance at any packet erasure rate at the encoding and decoding computational cost of $O(k \log k)$, where k is the number of packets in an original source block. Raptor codes [3] are a compound coding structure, usually including a high-rate outer LDPC code and an inner LT code, which is able to nearly optimally solve the transmission problem over an unknown erasure channel at the reduced computational cost of $O(k)$. A systematic version of Raptor codes has been standardized by 3GPP [4].

Distributed LT codes were introduced in [5] for independent rateless erasure encodings of multiple sources. Code design at the sources aims to result in a decoding behaviour at the sink which resembles the Soliton-like LT decoding behaviour. Soliton-like distributions arise from AND-OR tree analysis, where they appear as solutions to the problem of constructing an LT code which complies with graph pruning belief propagation algorithm for erasure recovery. Here, we describe and study a more general version of distributed LT codes, applicable to any number of sources, where relay is allowed to selectively combine incoming packets independently of their degrees in a randomized fashion, naturally extending distributed LT coding scenario of [5]. We then provide a version of the AND-OR tree lemma for such selective distributed LT codes, which allows us to formulate the preliminary optimization framework for code parameters. We also prove the equivalence of distributed LT codes and certain LT codes, thereby answering the question posed in [5] of whether distributed LT code design problem should target the same code parameters as classical LT code design problem, namely Soliton-like distributions.

II. LT codes and asymptotic analysis

Fountain code $\mathcal{F}_k$ on the information sequence of length k is generally described by a random variable R on $\mathbb{F}_2^k$ (column vector). The generator matrix of the code can be formed row by row as

$$G = [R_1 \;\; R_2 \;\; \ldots \;\; R_n \;\; \ldots]^{\mathrm{T}}, \tag{1}$$

where R_i are independent realizations of R, and the number of rows can be determined on-the-fly.

Although LT codes fully realize the digital fountain paradigm [1], their simplicity makes them amenable to theoretical analysis and various design considerations. The only two parameters of an LT code are the length of information sequence k and the probability distribution $(\Omega_1, \ldots, \Omega_k)$ on the set $N_k = \{1, 2, \ldots, k\}$, where Ω_i is the probability that the value i is chosen. The distribution $(\Omega_1, \ldots, \Omega_k)$ is called the output degree distribution, since it determines the weights of rows in a generator matrix, which translates to the degrees of the output nodes in the decoding graph. The output degree distribution is usually written in the generating polynomial notation as $\Omega(x) = \sum_{d=1}^{k} \Omega_d x^d$, and we refer to an LT code with these parameters as $\mathrm{LT}(\Omega(x), k)$.

The generation of LT encoded packets consists of two simple steps: (1) Sample an output degree d with probability Ω_d, and (2) Sample d data packets uniformly at random from the information sequence and XOR them. These steps can be performed as many times as necessary in order to produce enough encoded packets for successful decoding. Distribution on $\mathbb{F}_2^k$ is determined by Ω as $Pr(R = v) = \Omega_{w(v)} / \binom{k}{w(v)}$, for any $v \in \mathbb{F}_2^k$, where $w(v)$ is the Hamming weight of v.

The decoding of an LT code utilizes a belief propagation decoding algorithm on the factor graph of the linear encoder $\mathbb{F}_2^k \to \mathbb{F}_2^N$ obtained by restriction of the fountain code mapping to exactly those N coordinates in the fountain encoded stream observed at the receiver. In other words, receiver sees LDGM

978-1-4244-4536-3/09/$25.00 ©2009 IEEE

(Low Density Generator Matrix) encoded packets. Several parameters describe the decoding graph - its output node degree distribution $\Omega(x)$, its input node degree distribution $\Lambda(x)$, and edge perspective [6] output and input degree distributions[1] $\omega(x) = \frac{\Omega'(x)}{\Omega'(1)}$ and $\lambda(x) = \frac{\Lambda'(x)}{\Lambda'(1)}$. The decoding algorithm is particularly simple in the case of erasure channels and is equivalent to greedy graph pruning (cf. [6]). The performance of a particular code ensemble can be measured as a function of code overhead $\varepsilon = \frac{N}{k} - 1$, as N varies.

Good LT codes require the average output degree $\mu = \Omega'(1)$ to grow at least as $\log k$ - otherwise large error floors occur in the waterfall region of the decoder. Robust soliton distribution [2] is the example of degree distribution that complies with greedy graph pruning very well in practice and its average degree is about $\log k$. Raptor codes utilize degree distributions independent of k (with constant average degree) and error floor is removed by an outer LDPC code. These "light" degree distributions are capped at some maximum output degree d_{max}, i.e., $\Omega(x) = \sum_{d=1}^{d_{max}} \Omega_d x^d$.

The asymptotic behaviour of an LT code with constant average degree distribution is captured by a standard AND-OR tree analysis argument [7]:

Lemma 1. *The erasure rate of an* $\mathrm{LT}(k, \Omega(x))$ *at overhead* ε*, as* $k \to \infty$*, is given by* $y = \lim_{l\to\infty} y_l$*, where* y_l *is given by:*

$$\begin{aligned} y_0 &= 1, \\ y_l &= \exp\left(-\alpha\omega(1 - y_{l-1})\right), \end{aligned} \tag{2}$$

where $\alpha = \Lambda'(1)$ *is the average input degree.*

Note that α can be expressed via the output average degree $\mu = \Omega'(1)$ and the code overhead ε, as $\alpha k = \mu k(1 + \varepsilon)$ since both sides are the number of edges on the decoding graph. This allows us to express the AND-OR formulae (2) in terms of code overhead ε:

$$\begin{aligned} y_0 &= 1, \\ y_l &= \exp\left(-(1 + \varepsilon)\Omega'(1 - y_{l-1})\right). \end{aligned} \tag{3}$$

Lemma 1 can be easily transformed into an optimization procedure for asymptotically good output degree distributions $\Omega(x)$. The linear programs used for optimization are given in the Appendix. In [9], related linear programming techniques are used to optimize the asymptotic degree distributions for partial recovery. Note that these linear programs can be extended to a heuristic finite length design problem, as discussed in [3], [8]. The degree distribution obtained this way generally resemble Soliton distributions described in [2] and we thus refer to them as to Soliton-like distributions.

III. Distributed LT codes

In [5], the authors introduce techniques of decomposing LT codes into distributed LT (DLT) codes. DLT codes can be used to encode data from multiple sources independently and after that a common relay combines encoded packets from multiple sources to produce a bit-stream approximating that of an LT code and transmit it over an erasure channel. The deconvolution of Robust soliton distribution was used to formulate design of good DLT codes in the cases of two and four sources, and substantial performance benefits have been noted in comparison to a strategy where each source uses independent LT encoder and the relay simply forwards the encoded packets.

In a general scenario, one may consider a network with t sources, such that each source contains independent and disjoint set of k data packets. Let us assume that each source i encodes its k packets via $\mathrm{LT}(\Phi_i(x), k)$ code, i.e., a DLT code with properly modified output degree distribution, and relay simply XOR's all the incoming packets. We will refer to this scenario as to $\mathrm{DLT}(\{\Phi_i(x)\}_{i=1}^t, t, k)$ and $\mathrm{DLT}(\Phi(x), t, k)$ for $\Phi_i(x) = \Phi(x)$, $i = 1, \ldots, t$. The receiver that has successfully obtained $N = tk(1 + \varepsilon)$ encoded packets from the relay needs to invert a generator matrix $G = [G_1 \; G_2 \; \cdots \; G_t]$ formed as the horizontal concatenation of generator matrices G_i, $i = 1, 2, \ldots, t$ of encoding operation at the i-th source. The decoding graph $\mathcal{G}$ has tk input nodes and N output nodes, and can be thought of as the union of all factor graphs $\mathcal{G}_i$, corresponding to matrices G_i, $i = 1, 2, \ldots, t$, assuming they share the same output nodes. The resulting output degree distribution in graph $\mathcal{G}$ is $\Omega(x) = \prod_{i=1}^t \Phi_i(x)$ but it is not immediately clear if the decoding operation on graph $\mathcal{G}$ is equivalent to decoding operation of an $\mathrm{LT}(\Omega(x) = \prod_{i=1}^t \Phi_i(x), tk)$ code whose decoding graph bears the same output degree distribution. Namely, the random variable R on $\mathbb{F}_2^{t\cdot k}$ induced by $\mathrm{LT}(\Omega(x) = \prod_{i=1}^t \Phi_i(x), tk)$ is distributed as $Pr(R = v) = \Omega_{w(v)} / \binom{t\cdot k}{w(v)}$. On the other side, $\mathrm{DLT}(\{\Phi_i(x)\}_{i=1}^t, t, k)$ forms a different distribution on $\mathbb{F}_2^{t\cdot k}$, as $Pr(R = v) = \prod_{i=1}^t \frac{\Phi_{i,w(v^i)}}{\binom{k}{w(v^i)}}$, where v^i are the values of v within i-th group of k bits. In the following, we will prove that in the limit of large k, two decoding problems when $\Phi_i(x) = \Phi(x)$, $i = 1, \ldots, t$ are equivalent. If not all output degree distributions are the same, this is generally not true, as output degree distributions of different average degree may induce different average input degrees in different sources and such scenario may yield an Unequal Error Protection (UEP) property across sources. Fountain codes for UEP are an active area of research [10],[11], and this may be another interesting application of distributed LT coding scenario.

Next, we formulate a version of AND-OR lemma for density evolution of DLT codes.

Lemma 2. *The erasure rate of an* $\mathrm{DLT}(\{\Phi_i(x)\}_{i=1}^t, t, k)$*, as* $k \to \infty$*, is given by* $y = \lim_{l\to\infty} y_l$*, where* y_l *is given by:*

$$\begin{aligned} y_{0,i} &= 1 \\ y_{l,i} &= \exp\left(-\alpha_i \phi_i(1 - y_{l-1,i}) \prod_{j\neq i} \Phi_j(1 - y_{l-1,j})\right). \end{aligned} \tag{4}$$

In particular, for $\Phi_i(x) = \Phi(x)$, $i = 1, \ldots, t$*, all input average*

[1] Throughout the paper we will reserve symbols Ω, Λ, Γ and Φ to denote node perspective degree distributions and symbols ω, λ, γ and ϕ to denote their respective edge perspective degree distributions.

degrees are the same, i.e., $\alpha_i = \alpha$, $i = 1, \ldots, t$, and

$$\begin{aligned} y_0 &= 1 \\ y_l &= \exp\left(-\alpha\phi(1-y_{l-1})(\Phi(1-y_{l-1}))^{t-1}\right). \end{aligned} \tag{5}$$

Proof: The input node degrees in the i-th source are distributed as Binomial($\frac{1}{k}, \alpha_i k$), which converges pointwise to Poisson(α_i) as $k \to \infty$. Pick a random source i and a random input node a within that source. At the zeroth iteration, no information is available about this node and thus the probability that a is erased is $y_{0,i} = 1$. Now consider the l-th iteration. Node a stays erased if and only if it receives erasure-message from each of its neighbours, and it has d of them with probability $\Lambda_{i,d}$. Consider a random output node $f \in \mathcal{N}(a)$ - which is equivalent to choosing a random edge connected to a in $\mathcal{G}_i$, but to choosing a random output node in $\mathcal{G}_j$ for $j \neq i$. Node f has degree d_i in graph $\mathcal{G}_i$ with probability ϕ_{i,d_i} and degree d_j in graph $\mathcal{G}_j$ with probability Φ_{j,d_j}, $j \neq i$. For the moment fix degrees of node f within each of the sources to some values $d_1, d_2, \ldots, d_t$. Then, the probability that f sends an erasure to input node a in source i at the l-th iteration is given by

$$1 - (1 - y_{l-1,i})^{d_i - 1} \prod_{j \neq i} (1 - y_{l-1,j})^{d_j}, \tag{6}$$

since f needs to receive an erasure from any of its $d_i - 1$ neighbours in source i or any of its d_j neighbours in any other source j. Averaging over exponents for each of the sources together with Poisson approximation $\Lambda_i(x) = \exp(-\alpha_i(x-1))$ of the input degree distribution gives the lemma. ■

Above lemma asserts the equivalence of BP decoding of DLT($\Phi(x), t, k$) and LT($\Phi(x)^t, tk$) and this can be directly checked as well. However, we will prove a more general result in the next section.

IV. Selective combining at the relay

The obvious problem that arises in the scenario considered above is that BP decoder requires an output node of degree 1 in the decoding graph to begin decoding. On the other side, when $t \geq 2$, if no distribution $\Phi_i(x)$, $i = 1, \ldots, t$ allows encoded packets of degree 0, no degree 1 packets will be transmitted from the relay. But allowing degree 0 packets is clearly wasteful of resources. The way around this problem is to consider allowing relay to *selectively* combine incoming packets. In [5], selective combining was demonstrated for two and four sources and it was observed that these naturally extend to 2^m sources for any $m \in \mathbb{N}$. We will extend this approach for any number of sources t, and fundamental difference of our approach is that, as opposed to [5], selective combining can be performed independently of the degrees of the incoming packets at the relay. Consider allowing that for each set of incoming packets at a single iteration, relay may choose a value from the set $N_t = \{1, \ldots, t\}$ according to distribution $(\Gamma_1, \Gamma_2, \ldots, \Gamma_t)$, where Γ_i is the probability that the value i was chosen. As usual, let us denote this probability distribution in its generating polynomial notation by $\Gamma(x) = \sum_{d=1}^{t} \Gamma_d x^d$. After choosing "degree" d, relay may choose d incoming packets uniformly at random, XOR them and forward the new packet. We will refer to the scenario where relay is allowed to selectively combine incoming packets as to the code ensemble SDLT($\Gamma(x), \{\Phi_i(x)\}_{i=1}^t, t, k$) and SDLT($\Gamma(x), \Phi(x), t, k$) when $\Phi_i(x) = \Phi(x)$, $i = 1, \ldots, t$. Distribution $\Gamma(x)$ forms another factor graph $\mathcal{H}$ attached to this coding scenario, which effectively describes coding operation at the relay. This graph has N output nodes and t source nodes and an edge between output node f and source node i signifies that packet f is a linear combination of packets from source i, amongst others. We note that this factor graph has output degree distribution $\Gamma(x)$ and edge perspective output degree distribution $\gamma(x) = \frac{\Gamma'(x)}{\Gamma'(1)}$.

However, the issue of wastage of resources arises again, as some packets that reach relay are never going to be used. Namely, any kind of centralized coordination which would guarantee that only S_i, $i = 1, 2, \ldots, N$ sources transmit at each time slot, where S_i are independent realizations of random variable S on N_t with distribution $\Gamma(x)$, would preclude the need for distributed LT codes, as it could as well be used to contruct an exact Soliton-distributed LT code across all the sources. Nonetheless, we believe our setting to be justified as it is consistent with digital fountain paradigm, which penalizes reception rather than transmission of data [13]. The following example illustrates a decentralized and asynchronous data dissemination scenario taking advantage of selective combining.

Example 1. Consider an SDLT($\Gamma(x), \{\Phi_i(x)\}_{i=1}^t, t, k$) scenario, where t sources are continually broadcasting data to a large number $r >> t$ of relays via lossy links. As all incoming packets are equally important descriptions of its source, a relay can tune into a desired number of ongoing broadcasts at any time, and can combine incoming packets from different time slots, when a packet loss has occurred.

The next example compares two extreme cases of our setting, which assume encoding either at the sources or at the relay, but not both.

Example 2. Consider a network of t sources, each containing an independent data set of k packets. Assume that relay is able to receive packets from different sources at a single time slot, but may transmit only single packet per time slot. Consider two following coding scenarios:

- *Coding at sources* - SDLT($x, \Phi(x), t, k$): At each time slot, randomly selected source creates one coded packet using LT($\Phi(x), k$) and transmits it to the receiver, whereas relay simply forwards the coded packet, i.e., $\Gamma(x) = x$.
- *Coding at relay* - SDLT($\Gamma(x), x, t, k$): Each source transmits single, randomly chosen uncoded packet, i.e., $\Phi(x) = x$, and relay encodes the incoming sequence of t packets with an LT($\Gamma(x), t$) to create a coded packet which is being transmitted to the receiver.

Not surprisingly, for values $k = 1000$ and $t = 10$, coding at relay can exhibit better performance than coding at sources, as shown in Fig. 1, even though distribution $\Gamma(x)$ has a very

low maximum degree, i.e., $t = 10$. In simulation we used robust soliton distribution $\Phi_\rho(x)$ with parameters $k = 1000$, $c = 0.03$ and $\delta = 0.05$ [2] for coding at the sources, and an optimized distribution $\Gamma_R(x)$ described in Section V. In coding at relay scenario, decoding is performed on a matrix of size $t \cdot k$, instead of t separate decodings on matrices of size k for coding at sources, since in this case data from different sources is not being combined. Since k is fairly small, this brings large difference in performance. However, this example also motivates us to study the case where both distributions $\Phi(x)$ and $\Gamma(x)$ are non-trivial, which is the general case we are interested in. Namely, when t is also small, available degree distributions $\Gamma(x)$ have a very low allowed maximum degree and thus suffer from high error floors. Allowing coding at both source nodes and relay may help produce higher output degrees and thus alleviate the error floor but also benefit from combining data from different sources to produce decoding problem of larger size.

We may capture asymptotic decoding performance of selective distributed LT codes at the receivers by AND-OR formulae given in the following lemma.

Lemma 3. *The erasure rate of an* $\mathrm{SDLT}(\Gamma(x), \Phi(x), t, k)$*, as* $k \to \infty$*, is given by* $y = \lim_{l\to\infty} y_l$*, where* y_l *is given by:*

$$\begin{aligned} y_0 &= 1 \\ y_l &= \exp\left(-\bar{\alpha}\phi(1 - y_{l-1})\gamma\left(\Phi(1 - y_{l-1})\right)\right), \end{aligned} \tag{7}$$

where $\bar{\alpha} = \Gamma'(1)\Phi'(1)(1 + \varepsilon)$ is the average input degree on the decoding graph.

Proof: The proof follows closely from Lemma 2. On decoding graph $\mathcal{G}$ choosing a neighbour f to a random input node a in a random source i is equivalent to choosing a neighbour to a random source node i on factor graph $\mathcal{H}$. This node has degree s in $\mathcal{H}$ with probability γ_s, and averaging over $\gamma(x)$ gives (7), as long as we can prove that input degrees on the decoding graph are Poisson distributed with mean $\bar{\alpha}$. Now, in each graph $\mathcal{G}_i$, degree of an input node is $D \sim \mathrm{Poisson}(\alpha)$, where $\alpha = \Phi'(1)t(1 + \epsilon)$. Selectively combining incoming packets at the relay is actually *thinning* [12] of D since each edge connected to an input node is going to be transferred to graph $\mathcal{G}$ with probability $\frac{\beta}{N}$, where $\beta = \Gamma'(1)k(1 + \epsilon)$ is the average source node degree in $\mathcal{H}$. Thus, in $\mathcal{G}$, degree of an input node is

$$\bar{D} \sim \sum_{i=1}^{D} X_i, \tag{8}$$

where $X_1, X_2, \ldots, X_D$ are i.i.d. Bernoulli($\frac{\beta}{N}$) variables. Since thinning conserves the Poisson law, $\bar{D} \sim \mathrm{Poisson}(\bar{\alpha})$, where $\bar{\alpha} = \frac{\alpha\beta}{N} = \Gamma'(1)\Phi'(1)(1 + \varepsilon)$, which proves the claim. ■

We note that this lemma allows simple linear programming optimization of distribution $\Gamma(x)$ in case when $\Phi(x)$ is apriori known, and these linear programs are given in Appendix.

Let us now consider the code ensemble $\mathcal{F}_k = \mathrm{LT}(\Gamma(\Phi(x)), tk)$ over entire set of tk packets from all the sources. Its edge perspective output degree distribution is

$$\begin{aligned} \omega(x) &= \frac{\Gamma'(\Phi(x))\Phi'(x)}{\Gamma'(1)\Phi'(1)} \\ &= \phi(x)\gamma(\Phi(x)). \end{aligned} \tag{9}$$

The average input degree is $\bar{\alpha} = \Gamma'(1)\Phi'(1)(1 + \varepsilon)$ and thus its AND-OR formula reads

$$\begin{aligned} y_0 &= 1 \\ y_l &= \exp\left(-\bar{\alpha}\phi(1 - y_{l-1})\gamma\left(\Phi(1 - y_{l-1})\right)\right), \end{aligned} \tag{10}$$

which is exactly the same as (7). Thus, we have proven the following theorem.

Theorem 4. *The performance of a selective distributed LT code ensemble* $\mathrm{SDLT}(\Gamma(x), \Phi(x), t, k)$ *is identical to the performance of an LT code ensemble* $\mathrm{LT}(\Gamma(\Phi(x)), tk)$*, as* $k \to \infty$.

This theorem answers one of the questions posed in the discussion of [5] of whether distributed LT code design should target Soliton-like output degree distributions in the resulting bitstream from the relay. The answer is, at least in the asymptotic regime, yes.

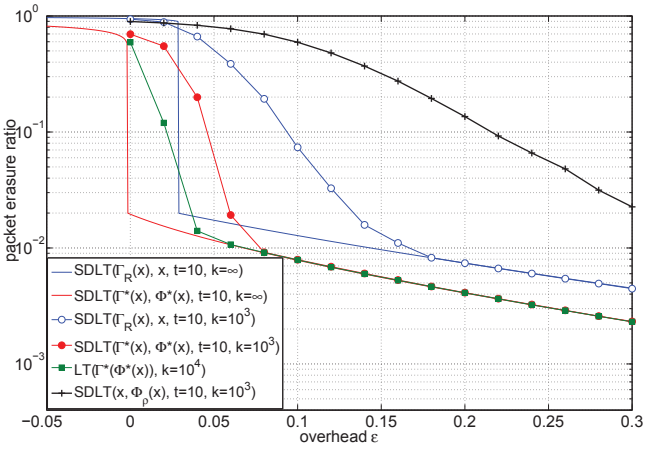

Fig. 1. Simulation results for various selective distributed LT scenarios.

V. Asymptotically Good Pairs of Distributions

We used linear program LP B' in Appendix to obtain good degree distribution pairs $(\Gamma(x), \Phi(x))$ for selective distributed LT coding scenario with $t = 10$ sources. In all linear programs, the value of the desired erasure ratio was fixed to $\delta = 0.02$, and we ran linear programs for various values of α in order to find optimum in terms of code overhead. Two examples of such distributions are given in Table I. For trivial case $\Phi(x) = x$, i.e., coding only at the relay, not surprisingly, obtained distribution $\Gamma_R(x)$ resembles a Soliton-like distribution. On other hand, through extensive search we obtained a good value of distribution $\Phi(x)$ for non-trivial case and optimized $\Gamma(x)$ accordingly. The pair of distributions $(\Gamma^*(x), \Phi^*(x))$ we obtained is given in the table and is superior to coding only at the relay both asymptotically, which is illustrated by the value

of objective function $1+\varepsilon^*$ given in the Table I and in finite length $k=1000$ as demonstrated on Figure 1.

We have also simulated LT($\Gamma^*(\Phi^*(x)), 10000$) performance and as demonstrated on Figure 1, its packet erasure ratio is close to that of SDLT($\Gamma^*(x), \Phi^*(x), 10, 1000$) as predicted by Theorem 4. Note that the asymptotic packet erasure ratio of LT($\Gamma^*(\Phi^*(x)), k$) is identical to that of SDLT($\Gamma^*(x), \Phi^*(x), 10, k$).

TABLE I
PAIRS ($\Gamma(x), \Phi(x)$)

$\Gamma_{R,1}$	$\Gamma_{R,2}$	$\Gamma_{R,3}$	$\Gamma_{R,4}$	$\Gamma_{R,10}$	α
0.0048	0.4422	0.3075	0	0.2455	4.39
Φ_1					$1+\varepsilon^*$
1					1.0290
Γ^*_1	Γ^*_2	Γ^*_3	Γ^*_4	Γ^*_{10}	α
0.7741	0.0025	0.1490	0.0026	0.0718	4.78
Φ^*_1	Φ^*_2	Φ^*_3	Φ^*_4		$1+\varepsilon^*$
0.05	0.5	0.4	0.05		0.9983

ACKNOWLEDGEMENTS

D. Sejdinović and R. Piechocki would like to thank Toshiba Telecommunications Research Laboratory and its directors for supporting this work. The authors would like to thank Mohamed Ismail for his advice with this research and the anonymous reviewers for their insightful comments.

REFERENCES

[1] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *Proc. of the ACM SIGCOMM 98*, pp. 56–67, Vancouver, Canada, Sept. 1998.

[2] M. Luby, "LT Codes," *Proc. of the 43rd Annual IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, Canada, Nov. 2002.

[3] A. Shokrollahi, "Raptor Codes," *IEEE Trans. Info. Theory*, vol. 52, No. 6, pp. 2551–2567, June 2006.

[4] 3GPP, "3GPP TS 26.346 V7.0.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs", Sept. 2007.

[5] S. Puducheri, J. Kliewer, T. Fuja, "The Design and Performance of Distributed LT Codes", in *IEEE Trans. Info. Theory*, vol. 53, no. 10, pp. 3740–3754, Oct. 2007.

[6] T. Richardson, R. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008.

[7] M. Luby, M. Mitzenmacher and A. Shokrollahi, "Analysis of Random Processes via And-Or Tree Evaluation," *Proc. of the 9th Annual SIAM Symp. on Discrete Algorithms (SODA)*, pp. 364–373, San Francisco, USA, Jan. 1998.

[8] R. Karp, M. Luby, A. Shokrollahi, "Finite Length Analysis of LT Codes", *Proc. of IEEE International Symposium on Information Theory 2004*, Chicago, USA, June 2004.

[9] S. Sanghavi, "Intermediate performance of rateless codes", *IEEE Information Theory Workshop 2007*, Tahoe, USA, Sept. 2007.

[10] N. Rahnavard, B. N. Vellambi, F. Fekri, "Rateless Codes With Unequal Error Protection Property" *IEEE Trans. Info. Theory*, vol. 53, No. 4, pp. 1521–1532, Apr. 2007.

[11] D. Sejdinović, D. Vukobratović, A. Doufexi, V. Šenk, R. Piechocki, "Expanding window fountain codes for unequal error protection", in *Proc. of the 41st Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 4-7, 2007.

[12] A. Renyi, "A characterization of Poisson processes," Magyar Tud. Akad. Mat. Kutaló Int. Közl., vol. 1, pp. 519-527, 1956.

[13] S. Shamai, E. Telatar, S. Verdú, "Fountain Capacity", *IEEE Trans. Info. Theory*, vol. 53, no. 11, pp. 4372–4377, Nov. 2007

APPENDIX

When optimizing degree distributions for LT codes, there are two different approaches and both include the transformation of AND-OR formulae 1 into an appropriate linear program by discretizing intervals of erasure rates. In all linear programs below, $0=x_1<x_2<\cdots<x_m=1-\delta$ are m equidistant points on $[0,1-\delta]$, δ is the desired erasure ratio, and d_{max} is the maximum degree of the degree distribution which is being optimized. Moreover, all the variables (to be determined) are non-negative and sum to 1.

The first linear program (LP A) minimizes average degree of distribution, which is required to reach the desired performance at a fixed overhead ε:

$$\text{LP A}: \quad \min \sum_{d=1}^{d_{max}} d\Omega_d \tag{11}$$
$$\sum_{d=1}^{d_{max}} d\Omega_d x_i^{d-1} \geq \frac{-\ln(1-x_i)}{1+\varepsilon},\ i\in 1,2,\ldots,m$$

The second linear program (LP B) fixes the average input node degree $\alpha=\Omega'(1)(1+\varepsilon)$ and minimizes the code overhead such that the desired erasure ratio δ is achieved. The code overhead can be expressed in terms of edge perspective distribution $\omega(x)$ as $1+\varepsilon=\alpha\sum_d \frac{\omega_d}{d}$. By appropriately transforming constraints in LP A, new linear program becomes:

$$\text{LP B}: \quad \min \alpha \sum_{d}^{d_{max}} \frac{\omega_d}{d} \tag{12}$$
$$\alpha\sum_{d=1}^{d_{max}} \omega_d x_i^{d-1} \geq -\ln(1-x_i),\ i\in 1,2,\ldots,m$$

Note that $\Omega(x)$ used in encoding of packets can be determined from $\omega(x)$ as

$$\Omega(x)=\frac{\int_0^x \omega(z)dz}{\int_0^1 \omega(z)dz}. \tag{13}$$

In an SDLT scenario, when we optimize degree distribution $\Gamma(x)$ at the relay, when distribution $\Phi(x)$ at the sources is fixed, we can use one of the following two linear programs LP A' and LP B', analogous to LP A and LP B.

$$\text{LP A}': \quad \min \sum_{d=1}^{d_{max}} d\Gamma_d \tag{14}$$
$$\sum_{d=1}^{d_{max}} d\Gamma_d \Phi(x_i)^{d-1} \geq \frac{-\ln(1-x_i)}{(1+\varepsilon)\Phi'(x_i)},\ i\in 1,2,\ldots,m$$

$$\text{LP B}': \quad \min \frac{\bar{\alpha}}{\Phi'(1)} \sum_{d=1}^{d_{max}} \frac{\gamma_d}{d} \tag{15}$$
$$\sum_{d=1}^{d_{max}} \gamma_d \Phi(x_i)^{d-1} \geq -\frac{\ln(1-x_i)}{\bar{\alpha}\phi(x)},\ i\in 1,2,\ldots,m$$