Relaying a Fountain Code Across Multiple Nodes

Ramakrishna Gummadi, Ramavarapu Sreenivas
Coordinated Science Lab, University of Illinois at Urbana-Champaign
gummadi2@uiuc.edu, rsree@uiuc.edu

ABSTRACT

Fountain codes are designed for erasure channels, and are well suited for broadcast applications from a single source to its one hop receivers. The problem of designing a rateless erasure coding scheme in the network case for on-the-fly recoding is important, as relaying the data over multiple nodes is fundamentally useful in a network. Fountain codes, however, are unsuited for on-line recoding, and simply forwarding packets over subsequent hops is provably suboptimal for throughput. Random linear codes are throughput optimal, but they do not have the low complexity that is a prime feature of fountain codes. Can we get the low complexity of the LT codes while maintaining on-the-fly recoding and being throughput optimal in a network? With this motivation, we consider packet level coding on a line network (or a tree) of discrete memoryless erasure channels (with potentially unlimited nodes), and propose a rateless coding scheme with logarithmic per-symbol coding complexity , but avoids the delay of having to decode and then re-encode entire block lengths at intermediate nodes while achieving close to min-cut capacity throughput.

Categories and Subject Descriptors: C.2.1 [Communication Networks]

General Terms: Algorithms

1. INTRODUCTION

Fountain codes are erasure coding schemes that are rateless, meaning that they can adapt to erasure channels with unknown parameters. LT codes([1]) operate by encoding a block of k input packets together. Each coded packet is a binary addition of a subset of input packets. The number of packets added is called the degree of the output packet, and is chosen according to a cleverly designed probability distribution that leads to a simple iterative decoding. It was shown (as $k \to \infty$) in [1] that a set of $k + O(\sqrt{k} \ln^2 \frac{k}{\delta})$ coded packets is sufficient to recover the k packets with a probability of at least $1 - \delta$ through the simple belief propagation decoder, where $\delta > 0$. A key feature is that encoding and decoding have a very low per-symbol complexity of $O(\log \frac{k}{\delta})$. However, they are designed for a single erasure channel and are not applicable to a network - the only obvious way to use them would be to completely decode and reencode at each intermediate node. This is neither throughput rate optimal nor efficient in terms of the high delay and overhead involved in waiting for an entire block to be decoded before further encoding to the next node. For erasure coding on a network, Random Linear Coding (RLC) involves combining packets according to a uniformly random choice of each packet taken over a finite field([3]). Although it is rate optimal, the average packet degree is linear in k, implying an $O(k)$ complexity of packet operations.

Can we achieve the low complexity of LT codes and low *delay* while maintaining *ratelessness* and achieving a throughput close to capacity for anything beyond the single erasure link to a network? This is the motivation for this work, where we assume (1) a *tree* network of discrete memoryless erasure channels (2) we have an estimate for a universal upper bound(α) on erasure probabilities.

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

2. THE CODING SCHEME

The aim is to make the set of all coded packets received to be *equivalent* to an LT code. The constraints we have do lead to a loss of strict independence between coded packets, but if the erasures themselves are iid, the order in which the coded packets are sent becomes immaterial.

Assume slotted time and a line network of DMC erasure channels with i^{th} erasure probability ϵ_i. Consider a sequence of $n + 1$ nodes with source node labeled as node 0 that encodes a block of k message packets. Node 0 performs standard LT coding. An intermediate node re-encodes coded packets received from its parent, treating them as information packets themselves in an online manner. Decoding at node t thus involves t successive instances of the LT decoding. Fix a sequence of "block lengths", $k_0 \dots k_n$, at the $n+1$ nodes defined by $k_0 = k$, and for $1 \leq i \leq n$, k_i is set to ensure that collecting a total of k_i LT coded packets at node i is enough to recover the k_{i-1} packets that were recoded by node $i - 1$ w.h.p. Recall that for LT coding k packets, a total of at least $k(1 + \frac{\log^2 \frac{k}{\delta}}{\sqrt{k}})$ coded packets gives an error probability of at most δ. On our line network of $n + 1$ nodes for a union bound error probability of δ, fix a $\frac{\delta}{n}$ error at each intermediate node and set $k_{i+1} = k_i(1 + \frac{\log^2 \frac{k_i n}{\delta}}{\sqrt{k_i}})$. By setting $k = \Omega(n^3)$, we can verify that $k_n = O(k)$. This would mean an encoding complexity same as for LT coding ($O(\log k)$ per symbol) while the decoding complexity is $O(n \log k) = O(k^{1/3} \log k)$.

DEFINITION 1. *A* ***Code Symbol*** *is a set of indices, that correspond to packet indices which will be summed(exor-ed) to form a coded packet.*

DEFINITION 2. *An **Online Code Copy** at node i is an* ordered sequence of k_{i+1} **code symbols** that can be generated in an "online fashion", described below.

A set of n coded packets generated from $k(< n)$ information packets taken in some sequence can be encoded in an "online fashion", if for $1 \leq i \leq k$, the i^{th} coded packet is a combination of subset of solely the first i information packets. If we consider $k + o(k)$ LT coded packets, we can construct an online code copy using the LT *decoding* process itself. Systematic versions of raptor codes involve similar ideas([2]): *SEQCODE*

(1) Generate a random set of $k + o(k)$ *symbols* according to the LT encoding process.

(2) Obtain a permutation π from the encoded symbols as follows:

   (a) Run an LT decoder on the encoded symbols.

   (b) If the index decoded at i^{th} iteration of the decoding process has a label j, then set $\pi(i) = j$.

PROPOSITION 1. *The set of $k + o(k)$ code symbols generated by SEQCODE , ordered according to the permutation π it generates, renders itself to an online encoding.*

DEFINITION 3. *A **Code Matrix** at node i is a $T(k) \times k_{i+1}$ random matrix of* Code Symbols *in which each row corresponds to an independently generated* Online Code Copy. *The number of rows, $T(k) = k^{1+\delta}$ where $\delta > 0$.*

DEFINITION 4. ***Online phase** at node i is the period until the time slot at which node i collects a total of k_{i+1} coded packets. We further partition the online phase into $k_{i+1} + 1$ states, indexed through $0, 1, \ldots, k_{i+1}$ where the node is in **state** j when it has collected a total of j packets. Let s_j be the number of erasures seen during state j.*

We now describe the operations performed at node i. Procedure $LT - RELAY$ (node i)

1. First generate:

   (i) A random *code matrix*, $\mathcal{M}_i = [c_{ij}]_{1 \leq i \leq T, 1 \leq j \leq k_{i+1}}$

   (ii) Another independent random *online code copy*, $\mathcal{R}_i = \{\theta_j\}_{1 \leq j \leq k_{i+1}}$

2. *Online phase (i.e. while in state $j, 0 \leq j \leq k_{i+1}$):*

   (i) In the first time slot upon entering state j, send a packet coded according to the code symbol θ_j.

   (ii) After the first time slot (i.e. for the remaining s_j slots): Choose a code symbol uniformly at random from $\hat{e}_j$, the j^{th} column of $\mathcal{M}_i$. Use it for encoding ($\hat{e}_j$ contains only indices of at most j by construction), if it was not previously picked. Else, it becomes an *idle slot.*

3. Beyond the online phase, generate independent coded packets at each time slot using the standard LT coding procedure for a block length of k_i.

2.1 Analysis

THEOREM 2. *(w.h.p.) For any node the number of idle slots during encoding, N_i, satisfies $N_i \leq \log k$*

DEFINITION 5. *Let $\Lambda = [\lambda_{ab}]_{1 \leq a \leq T(k), 1 \leq b \leq k}$ be a random matrix with elements taking $\{0, 1\}$ values be defined as follows. Each element λ_{ab} is a Bernoulli random variable defined as:*

$$\lambda_{ab} = \begin{cases} 1 & \text{, if } c_{ab} \text{ was used for encoding} \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

THEOREM 3. *Given any integer c, and arbitrary $S \subset \{1, \ldots, T(k)\} \times \{1, \ldots k\}$ such that*

$$|\{x : (x, j) \in S\}| \leq c \qquad (1 \leq j \leq k) \ ,$$

the Random Variables, $\{\lambda_{ab}\}_{(a,b) \in S}$ become independent, and identically distributed as $k \to \infty$.

The above theorem implies the following corollary, which says that the code symbols chosen by the algorithm are scattered "uniformly random" in the following sense:

COROLLARY 4. *Let $\chi = \{0, 1\}^{T(k) \times k}$ denote the ensemble of all possible realizations of the random matrix, Λ. For $\Psi = [\psi_{ij}] \in \chi$ and for any $S \subset \{1, \ldots, T(k)\} \times \{1, \ldots, k\}$, denote*

$$W_S(\Psi) = \sum_{(i,j) \in S} \psi_{ij}$$

Consider any given integer r, and $E \subset \{1, \ldots, T(k)\} \times \{1, \ldots, k\}$, with $|E| = r$ denoted as $E = \{e_1, \ldots, e_r\}$. For any $\phi = (\phi_1, \ldots, \phi_r) \in \{0, 1\}^r$ let $\Theta_\phi = \{\Psi \in \chi : \psi_{e_j} = \phi_j$ for $1 \leq j \leq r\}$. Then, as $k \to \infty$, in the probability space generated by "LT-Relay", $P(\Theta_\phi)$ depends solely on $\sum_{i=0}^{r} \phi_i = W_E(\Psi) \ \forall \ \Psi \in \Theta_\phi$.

LEMMA 5. *Given that (i) the subset of* code symbols *chosen from $\mathcal{M}_i$ is uniformly random and (ii) t denotes a time slot past the online phase, the* set *of all coded packets generated by node i till time slot t has the same degree distribution as an LT code.*

We need above that erasure probabilities are increasing down any path from the source. This is not a severe restriction, and can be enforced artificially while maintaining the same min cut capacity. The scheme described could then be shown to have a capacity achieving throughput because packets are being generated at every time slot except for a vanishing fraction of idle slots and LT codes themselves are capacity achieving.

3. REFERENCES

[1] M.Luby, "LT Codes", Proc. FOCS 2002.

[2] A. Shokrollahi, "Raptor Codes", IEEE Transactions on Information Theory, vol. 52, num. 6 (2006), p. 2551-2567 2006

[3] D. S. Lun, M. Mt'edard, and M. Effros, "On coding for reliable communication over packet networks", Proc. Allerton 2004.

[4] R. Gummadi, R. Sreenivas, " Relaying a Fountain Code across multiple nodes ", Proc. ITW 2008, Portugal.