Raptor Codes

Vansh Kapoor & Pranava Singhal

EE605 Course Project
Guide: Prof. Nikhil Karamchandani

November 24, 2022

Summary

1. Motivation
    - Shortcomings of TCP/IP
    - UDP
    - Conventional Error Coding
2. Fountain Codes
    - Design Requirements
    - LT Codes
3. Decoding Algorithms
4. Raptor Codes
5. Systematic Raptor Codes

Motivation

The Internet is a Binary Erasure Channel
Protocols like TCP/IP & UDP are used for communication

TCP/IP Shortcomings
Idle Times

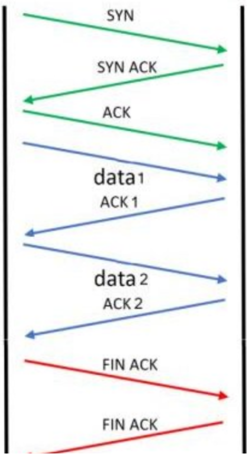

Figure: sender waits for ACK (acknowledgement)

TCP/IP Shortcomings
Point to Multi-Point

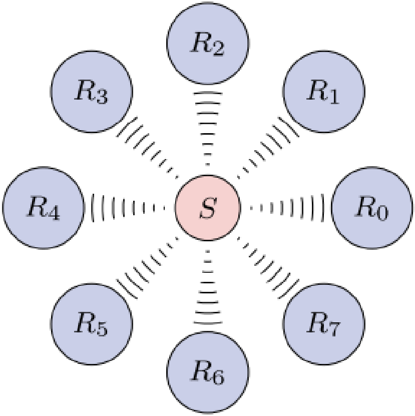

Figure: Channel Usage $\sim$ O(# receivers)

TCP/IP Shortcomings
Multi-Point to Point

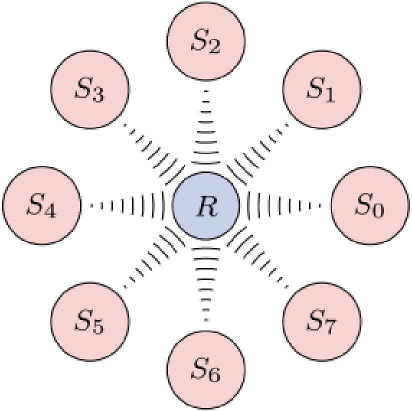

Figure: Unsynchronised transmitters create redundancy

TCP/IP Shortcomings
Multi-Point to Multi-Point

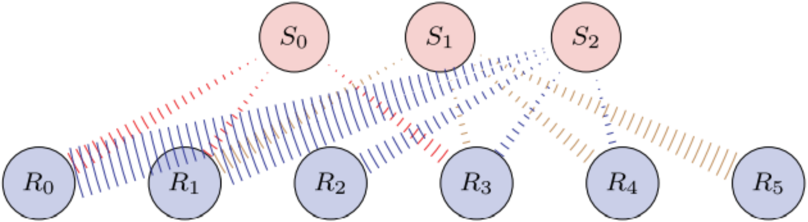

Figure: Transient nodes - Peer to Peer Networks

UDP

Advantage: Maximum transmission rate
Disadvantage: Unreliable

UDP with Conventional Error Coding

Can handle erasures but leads to increased overhead

Designing a New Class of Codes

Design Objectives

1. Unlimited transmission rate - limited only by encoding rate
2. An infinite stream of output symbols must be generated
3. All symbols must be independently generated
4. Any $k(1+\epsilon)$ received symbols should be enough to recover the original k message symbols
5. Encoding and decoding cost $\sim O(1) =$ constant number of operations per input message symbol

Fountain Codes

- k message symbols $x_1, x_2, ..., x_k$
- Infinite stream of output symbols $z_1, z_2, z_3, ...$
- Each z_i is the XOR of some message symbols
- The subset of message symbols to be XOR-ed is sampled from a distribution D
- For now, assume that this combination is known to the receiver for each output symbol it receives

LT Codes

LT codes are a particular class of Fountain codes designed as follows

Let Ω_d denote the probability of choosing a given value $d \in \{1, 2, ..k\}$. The distribution generator polynomial is given by $\Omega\left(x\right) = \sum_{d=0}^{k} \Omega_d x^d$

$\Omega\left(x\right)$ induces a distribution on F_2^k such that for any v $\in F_2^k$ of weight d the probability of v is $\Omega_d / \binom{k}{d}$

Ex: Uniform distribution on F_2^k is given by $\Omega\left(x\right) = \frac{1}{2^k}\left(1 + x\right)^k$

Code Performance Metrics

Definitions

- Encoding Cost: Expected number of operations needed to generate a single output symbol
- Decoding Cost: Expected number of operations needed to recover a single input symbol
- Overhead: $\epsilon = \frac{n-k}{k} \Rightarrow n = k(1+\epsilon)$
- Reliable Decoding: the error probability is at most $1/k^u$ for some positive constant u

Encoding Cost (1)

Proposition 1

If an LT-code with k-input symbols possesses a reliable decoding algorithm, then there is a constant c such that the graph associated to the decoder has at least $ck \log(k)$ edges

Proof

Let d denote the degree of a particular output node whose distribution is given by Ω_d. The probability that a given input node is not its neighbour is $P(k) = \sum_{d=1}^{k} \Omega_d \cdot (1 - d/k) = 1 - a/k$, where a is the expected degree of the output node and is given by $\Omega^{'}(1)$. Thus the probability that the given input node is connected to none of the n output symbols is $(1 - a/k)^n$. We know that $-ln(1-x) = x + \frac{x^2}{2} + \frac{x^3}{3} ... \leq x/(1-x)$ (for $x < 1$)

Encoding Cost (2)

Proof Continued

Thus we get -$ln\,(1 - a/k) \leq (a/k)(1 - a/k)$
$\Rightarrow (1 - a/k)^n \geq e^{-\alpha/(1-\alpha/n)}$
Since the decoder is reliable $\Rightarrow (1 - a/k)^n \leq 1/k^u$
$\Rightarrow e^{-\alpha/(1-\alpha/n)} \leq 1/k^u$, where $\alpha = an/k$ (the expected number of edges per input symbol)

$\Rightarrow \alpha \geq \ln\,(k)\,\frac{u}{(1+u\ \ ln(k)/n)}$
$\geq ln\,(k)\,\frac{u}{(1+u\ \ ln(k)/k)}$
$\geq ln\,(k)\,\frac{u}{(1+u\ \ ln(3)/3)}$
$\Rightarrow \alpha = c\log\,(k)$

This implies the encoding cost is O($\log k$)

Decoding Algorithms (1)

Maximum Likelihood Decoding

- In the case of an erasure channel ML decoding is nothing but Gaussian Elimination
- Each output symbol is a linear combination of a certain number of input symbols
- View it as solving n linear equations in k unknowns
- Thus the decoding cost of of this algorithm is $O(nk)$ (because Gaussian elimination can be performed in $O(nk^2)$ operations)

Fact

A Random LT-code with k input symbols has encoding cost k/2, and ML decoding is a reliable decoding algorithm for this code with overhead $O(\log k/k)$

Decoding Algorithms (2)

ML Decoding has $O(k^2)$ complexity. We want a more efficient decoder
Belief Propagation Decoding

Imagine the graph associated to the decoder. It performs the following steps until either no output symbols of degree 1 are present or all input symbols have been recovered.

- Step 1: Identify all output symbols of degree 1
- Step 2: If no output symbols of degree 1 are present are not all input symbols have been recovered then report a decoding failure else the value of the output symbol(of degree 1) gives the value of the input symbol
- Step 3: After decoding the input symbol add its value to all the neighbouring output symbols(Basically we are eliminating all the edges from the recovered input node)
- Step 4: Repeat the process until all the input symbols are recovered or a decoding failure is received

Decoding Cost of BP decoder is $O(k)$

Decoding Algorithms (3)

Random LT-codes fail miserably with BP decoder. Can you guess why?
We need to change the design of $\Omega(x)$
The Soliton Distribution is given by
$\Omega\left(x\right) = \frac{x}{k} + \sum_{k \geq d \geq 2}^{\infty} \frac{x^d}{d(d-1)}$
A slight variation of Soliton Distribution by Luby is an excellent distribution for BP decoding and reliable overhead of $\mathrm{O}(\log^2\left(x\right)/\sqrt{k})$

Raptor Codes

- LT-codes needed an order of $k \log k$ edges for reliable recovery of all input symbols
- The idea of raptor codes is to relax this condition so that only a constant fraction of input symbols must be recoverable

Notation

A Raptor code with parameters (k, C, $\Omega(x)$), is an LT-code with distribution $\Omega(x)$ on the n symbols received after a message with k symbols is encoded with precode C

These n symbols receives from C are called intermediate symbols

PCO Raptor Codes (1)

- Pre-Code Only Raptor codes(PCO) are simplest possible Raptor Codes
- Probability distribution is given by $\Omega(x) = x$
- Output symbol is generated by randomly choosing an input symbol
- The performance of a PCO raptor code depends on its pre-code C

PCO Raptor Codes (2)

Proposition 2

Let C be a linear code of dimension and block length with encoding and decoding algorithms that have the following properties.

- An arbitrary input vector of length k can be encoded with $k \cdot \eta$ arithmetic operations for some $\eta > 0$
- There is an $\epsilon > 0$ such that the decoding algorithm can decode over a BEC with erasure probability 1-R(1+ϵ) with high probability using $k \cdot \gamma$ arithmetic operations for some $\gamma > 0$.

Then the PCO code with the pre code C has space consumption 1/R, overhead -$ln(1 - R(1 + \epsilon))/R - 1$, encoding cost η and decoding cost γ with respect to the decoding algorithm for C, where R=k/n is the rate of C.

PCO Raptor Codes (3)

Proof

The proof for space consumption, encoding and decoding cost is obtained directly from their definition, and since the overhead is -$ln(1 - R(1+\epsilon))/R - 1$, and if is the number of symbols the decoder collects then

$\Rightarrow$ m= $-k\ ln(1 - R(1+\epsilon))/R =$ -n $ln(1 - R(1+\epsilon))$

Note that the probability that an intermediate symbol is not covered by any of the m output symbols is

(1-1/n)$^m \leq e^{-m/n} =$ 1-R(1+ϵ)

and since the code C can handle an erasure probability of R(1+ϵ) efficiently over a BEC, the PCO code thus formed does provide reliable decoding.

Two extremes on the spectrum of Raptor Codes

	LT Codes	PCO Raptor Codes
	no precode	precode
space	1	large
overhead	small	grows with k for fixed space
encoding cost	O($\log k$)	O(1)
decoding cost	O(k)	O(1)

Systematic Raptor Codes

Disadvantages of Raptor codes: they are not systematic $\Rightarrow$ input symbols are not necessarily reproduced by the encoder.
Why should a code be systematic?
For example, suppose that the deployment of a Raptor code is done in phases during which some receivers are equipped with a decoder, and others are not. Suppose further that a broadcast network is used to send data to the receivers. If a non-systematic Raptor code is used for this application, then the application needs to transmit a stream of source symbols to be used by receivers without a decoder and another stream of encoded symbols to be used by receivers equipped with a decoder.
This strategy wastes network resources, i.e., the network resource usage can be essentially double of what it would be if a systematic Raptor code was used instead. There are a variety of other applications for systematic Raptor codes, and thus systematic Raptor codes are preferable to non-systematic Raptor codes.

Systematic Raptor Codes

- Note: For systematic Raptor codes, the symbols among the encoded symbols that are not source symbols are called repair symbols.

So why not simply use the encoded symbols generated from a non-systematic Raptor code as the repair symbols and then just designate the source symbols to also be encoded symbols?
This trivial construction works very poorly with respect to the systematic decoding property i.e., the overhead-failure curve depends strongly on the mix of received source symbols and repair symbols, and is particularly bad when among the received encoded symbols a small fraction are source symbols and a large fraction are repair symbols.
Systematic Raptor Codes solve this problem, they accept k input symbols x_1,..,x_k and produce a set i_1,.., i_k of k distinct indices between 1 and K(1+ϵ) and an unbounded string z_1,.. of output symbols such that $z_{i1} = x_1$,.., $z_{i_k} = x_k$, and such that the output symbols can be computed efficiently. Moreover, we will also design a reliable decoding algorithm of overhead ϵ for this code.

Encoding And Decoding Systematic Raptor Codes

Let z^T denote the column corresponding to the n output symbols of the Raptor Code then

$S \cdot G^T \cdot x^T = z^T$

where S is a $N \times n$ Matrix weight matrix, where each row gives the weight of the input symbol selected. This system is solvable if and only if the rank of $S \cdot G^T$ is k. Consider its sub-matrix R$= A \cdot G$

Algorithm For Constructing R

- Step 1: Generate k$(1+\epsilon)$ output symbols using the Distribution $\Omega(x)$ to obtain v_1, v_2 .. $v_{k(1+\epsilon)}$
- Step 2: Generate S using v_1, v_2 ...$v_{k(1+\epsilon)}$ as rows of the matrix and find $S \cdot G^T$
- Step 3: Using Gaussian elimination, calculate rows i_1, i_2 .. i_k such that the submatrix of consisting of these rows is invertible, and calculate R^{-1}. If the rank of $S \cdot G$ is less than k, output an error flag

Encoding And Decoding Systematic Raptor Codes

Encoding Algorithm

- Step 1: Calculate $y^T = R^{-1} \cdot x^T$ and $u^T = G^T \cdot y^T$
- Step 2: Calculate $y^T = R^{-1} \cdot x^T$ and $u^T = G^T \cdot y^T$
- Step 3: Calculate $z_i = v_i \cdot u^T$ for $1 \leq i \leq k(1+\epsilon)$
- Step 4: Generate the other output symbols $\mathrm{z}_{k(1+\epsilon)+1}$, $\mathrm{z}_{k(1+\epsilon)+2}$,.. by applying LT-Code with Parameters (k,Ω(x))to the vector u.

Proposition 3

The output symbols z_j coincide with the input symbols x_j for 1≤j≤k

Proof

Projection of z on the first k coordinates be $\tilde{z}$

$\Rightarrow \tilde{z} = \mathrm{A} \cdot \mathrm{u}^T$

$\Rightarrow \tilde{z} = \mathrm{A} \cdot \mathrm{G}^T \cdot \mathrm{y}^T$

$\Rightarrow \tilde{z} = \mathrm{A} \cdot \mathrm{G}^T \cdot \mathrm{R}^{-1} \cdot \mathrm{x}^T$

Encoding And Decoding Systematic Raptor Codes

Proof continuation

$\Rightarrow \tilde{z} = R \cdot \mathrm{R}^{-1} \cdot \mathrm{x}^T$

$\Rightarrow \tilde{z} = \mathrm{x}^T$

Thus this encoder is indeed a systematic encoder

Decoding Algorithm

- Step 1: Decode the output symbols using the decoding algorithm for the original Raptor code(Either BP decoding or ML decoding) to obtain the intermediate symbols. Flag an error if decoding is not successful.
- Step 2: Calculate $\mathrm{x}^T = \mathrm{R} \cdot \mathrm{y}^T$