Erasure Codes[1,2]

slides by Gary Jackson

1 J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. **A Digital Fountain Approach to Reliable Distribution of Bulk Data**. Proc. ACM SIGCOMM'98, 28 (4):56--67, Oct. 1998.

2 H. Weatherspoon and J.D. Kubiatowicz, **Erasure Coding vs. Replication: A Quantitative Comparison**, Peer-to-Peer Systems: First International Workshop, IPTPS 2002, LNCS 2429, pp. 328--337, 2002.

What is an Erasure Code?

- Given a signal of m blocks, recode to n blocks where n > m
  - Optimal: reconstruct signal given any m unique blocks
  - Suboptimal: Reconstruct signal using (1+e) m unique blocks
- Rate r=m/n, and storage overhead is 1/r

What are they used for?

- Signals from deep space satellites
- Reliable multimedia multicasting (Digital Fountain)
- Reliable Storage

Digital Fountain

- Problem
  - Transmitting a fixed set of data to multiple clients over unreliable links
  - Previous solution: transmit original data interleaved with erasure coded blocks
    - But this has undesirable overhead

Ideal Solution

- Reliable - client always gets the whole file
- Efficient - extra work is minimized
- On demand - client gets the file at their discretion
- Tolerant - solution is tolerant of clients with different capabilities

Solution: Digital Fountain

- Server transmits constant stream of encoding packets
- Client succeeds when minimal number of packets are received
- Assumes fast encode/decode

Building a Digital Fountain

- Use Tornado erasure codes, because they are fast
  - However, they are suboptimal
  - Reconstruction requires $(1+\epsilon)m$ packets
  - (or $(1+\epsilon)k$ packets, in the paper's terminology)

Tornado Codes

- Reed-Solomon codes: over-specified system of polynomials over some finite field:

  $\mathbf{y}=P_{\mathbf{x}}(\alpha)$

- Tornado codes: system of equations like
  - $y_n=x_i \oplus x_j \oplus x_k \oplus x_l$
  - $y_m=y_o \oplus y_p \oplus y_q$

Comparison

	Tornado	Reed-Solomon
Decoding inefficiency	$1+\epsilon$ required	1
Encoding times	$(k+\ell)\ln(1/\epsilon)P$	$k\ell P$
Decoding times	$(k+\ell)\ln(1/\epsilon)P$	kxP
Basic operation	XOR	Field operations

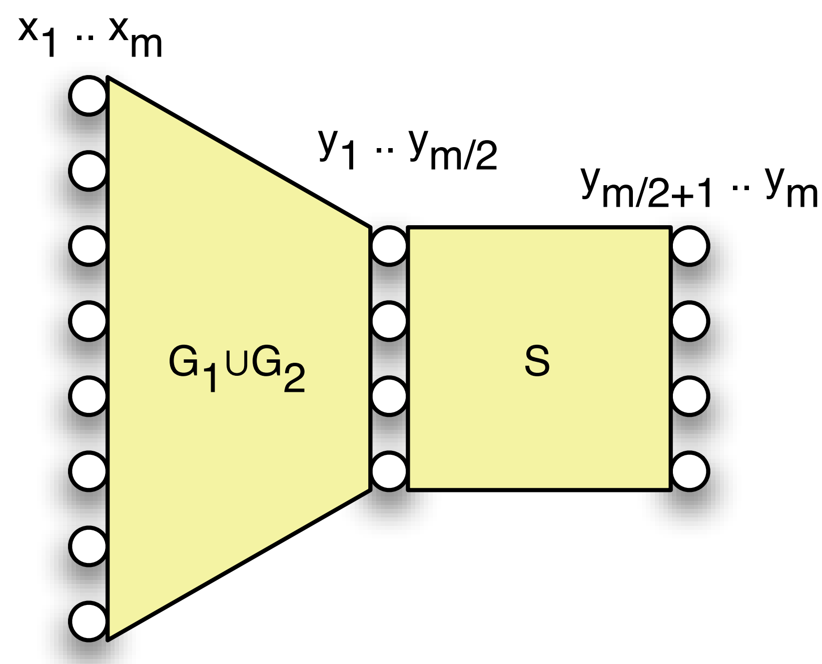
TABLE I

PROPERTIES OF TORNADO VS. REED-SOLOMON CODES

- Tornado codes appear to be much more efficient asymptotically, and use a much faster basic operation (XOR)
- But they have overhead

Example: Tornado Z

- G1: truncated heavy tail distribution to greater portion of right side
- G2: every node on left has out-degree of two connecting to remainder of right side
- S: Specially selected graph to connect second layer of redundancy



Runtime Comparison

Encoding Benchmarks		
	Reed-Solomon Codes	Tornado Codes
SIZE	Cauchy	Tornado Z
250 KB	4.6 seconds	0.11 seconds
500 KB	19 seconds	0.18 seconds
1 MB	93 seconds	0.29 seconds
2 MB	442 seconds	0.57 seconds
4 MB	1717 seconds	1.01 seconds
8 MB	6994 seconds	1.99 seconds
16M Bytes	30802 seconds	3.93 seconds

TABLE II

COMPARISON OF ENCODING TIMES.

Decoding Benchmarks		
	Reed-Solomon Codes	Tornado Codes
SIZE	Cauchy	Tornado Z
250 KB	2.06 seconds	0.18 seconds
500 KB	8.4 seconds	0.24 seconds
1 MB	40.5 seconds	0.31 seconds
2 MB	199 seconds	0.44 seconds
4 MB	800 seconds	0.74 seconds
8 MB	3166 seconds	1.28 seconds
16 MB	13629 seconds	2.27 seconds

TABLE III

COMPARISON OF DECODING TIMES.

Simulation: Set-up

- Compare two schemes:
  - Tornado
  - Interleaving
- Compare two variables
  - performance: encode/decode time
  - inefficiency: ratio of data needed to optimal

Interleaving Scheme

- Divided total message of size K in to B=K/*k* blocks, each of which is size *k*
- Erasure code each block
- Interleave encoded blocks with data in transmission

Choice of *k* is important

- Needs to be small for efficient encoding/ decoding
- But the smaller it is, the more overhead there will be when receiving: there is a greater likelihood that we will have to wait longer and receive more duplicate packets to reconstruct any given block

What happens when inefficiency is equal?

Speedup factor for Tornado Z					
	erasure probabilities				
SIZE	0.01	0.05	0.10	0.20	0.50
250 KB	1.37	2.05	5.55	11.1	11.1
500 KB	2.29	5.51	8.33	16.7	33.3
1 MB	4.12	10.3	17.1	25.8	51.6
2 MB	6.34	16.9	26.2	48.4	96.8
4 MB	7.87	22.3	34.6	62.7	115
8 MB	11.1	28.2	46.9	80	182
16 MB	14.2	34.9	56.4	100	212

TABLE IV

SPEEDUP OF TORNADO Z CODES OVER INTERLEAVED CODES WITH COMPARABLE EFFICIENCY.

- Decoding times are inferior for the interleaving scheme

... when decoding times are equal?

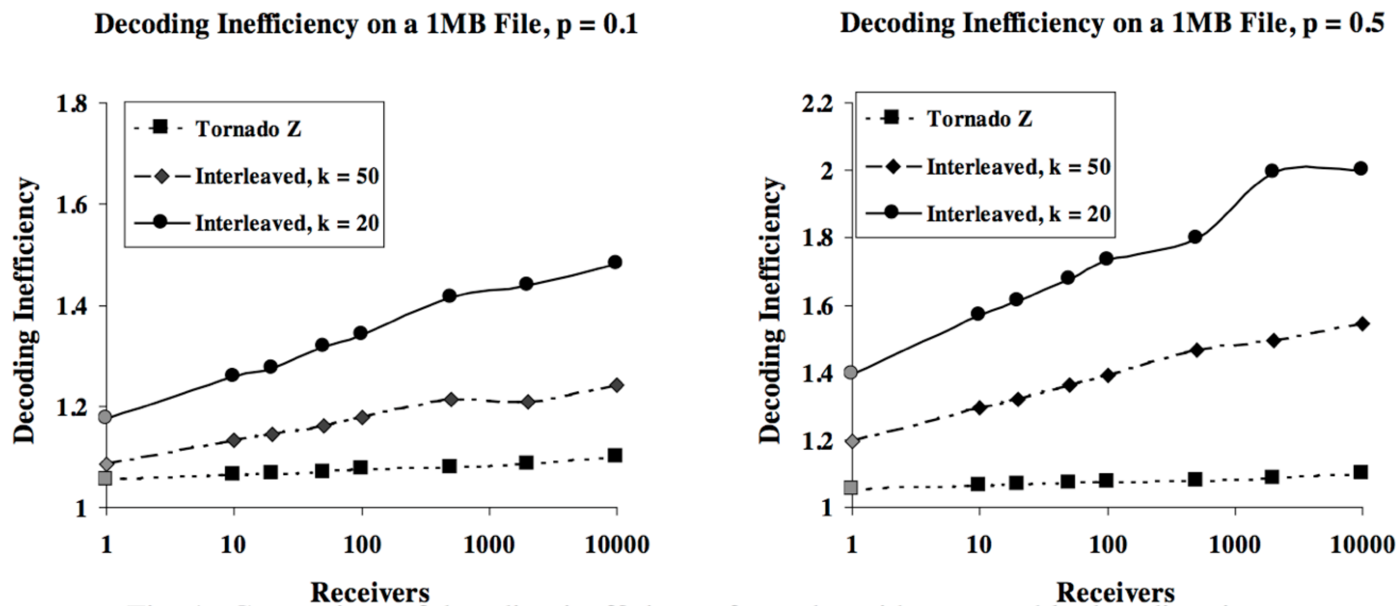


Fig. 4. Comparison of decoding inefficiency for codes with comparable decoding times.

- Inefficiency grows faster with the interleaved scheme
- These results scale with the size of the file, as well as with real trace data

Implementation: Idea

- Compare conventional multicast fountain versus layered multicast fountain
- Basis for comparison is the reception inefficiency η
  - η =total packets/used packets

Layered Multicast

- Clients can subscribe to layers of varying rate
- Clients at higher layers get everything at the lower layers
- Scheme for moving up and down in the layer hierarchy as conditions change

Scheduling across multiple layers

- One Layer Property
  - assuming fixed level and low packet loss
  - then signal can be reconstructed before duplicates are seen
- One Layer Property scheme exists for any set of layers

Result

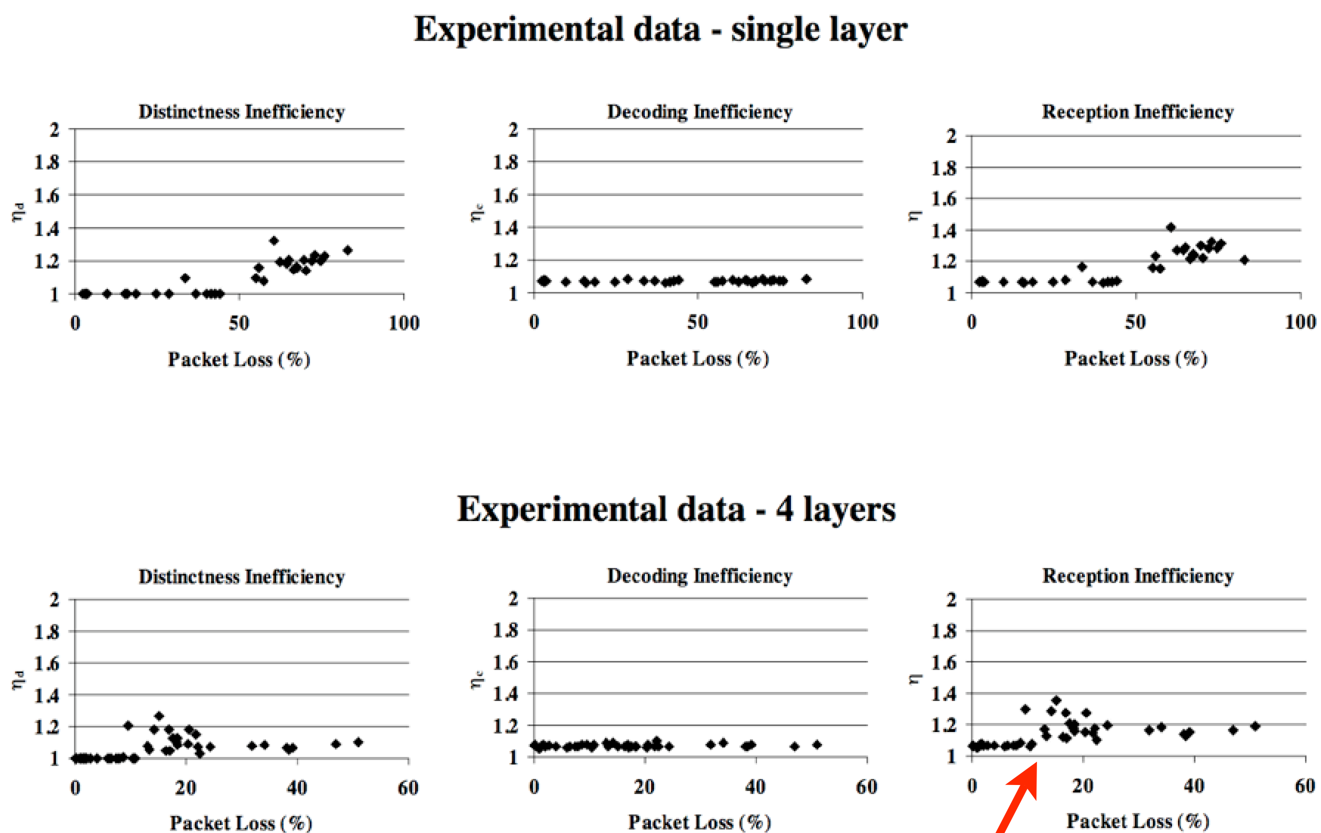


Fig. 7. Experimental Results of the Prototype

- Reception inefficiency is higher at lower packet loss levels on the 4-layer scheme than on the single layer scheme

Erasure Coding vs. Replication

- Premise
  - Erasure codes are good
  - Can the benefit over a replication scheme be quantified?

Assumptions

- Uniform environment
  - “independently, identically distributed failing disks”
- Repair is done on a polling basis, not on an interrupt basis
  - Have to check for problems

Availability

- For $N=10^6$, $M=10^5$

$$P_o = \sum_{i=0}^{n-m} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

  - $n=2$, $m=1$: $P_0=0.99$
  - $n=32$, $m=16$: $P_0=0.999999998$
  - Same overhead
- Conclusion: fragmentation increases availability

System Comparison

- Fix certain parameters and see what happens
- Important parameters
  - MTTF: average time before some component fails
  - B: number of blocks we care about
  - S: total size including overhead
  - BW: bandwidth
  - D: disk seeks
  - e: Repair Epoch - period of block verification

Fix MTTF and Repair Epoch

- Keep 100 petabytes of data for 1000 years
- Result: replicated system needs 11x size, bandwidth, and disk seeks that the erasure coded system needs

Fix Storage and Repair Epoch

- How long can we keep some block?
- Result: a given block has an MTTF of 74 years in a replicated system and 10^20 years in an erasure coded system

Fix MTTF and Storage Overhead

- Keep 1000 blocks for 1000 years
- Result: Erasure coded system needs a 28-month repair epoch. Replicated system needs near-instantaneous repair epoch.

Observation

- Not using the same sized problem for all comparisons

Discussion & Future Work

- Performance should be addressed separately from reliability
- Need to address:
  - failure independence
  - efficient repair - it takes a long time to sift through a petabyte of data

Conclusion

- Erasure coded fragments increase MTTF with lower storage, bandwidth, and disk seek requirements than the requirements for replicated systems

Questions

- Q: How do you decide what an optimal *r* would be?
- A: It depends on the other constraints of the application. If you expect a higher failure rate, you'll need a lower value for *r* to maintain the ability to reconstruct the signal.

Questions

- Q: Does erasure coding offer the flexibility to change the rate after it is chosen?
- A: There is something called a fountain code or a rateless erasure code:

  http://en.wikipedia.org/wiki/Rateless_erasure_codes

  These can produce as many encoded blocks as one needs, but generally require $(1+\epsilon)m$ blocks to recover the signal, just like Tornado codes.

Questions

- Q: Are Tornado codes still used today?
- A: Digital Fountain was commercialized. They now use a rateless proprietary erasure code called "Raptor".

Questions

- Q: Is there a version of the algorithm that allows for flawed packets but still able to reconstruct the original content?
- A: I’m not sure I understand the question. Generally, broken packets are considered lost.

Questions

- Q: What are unicast, multicast, and broadcast?
- A:
  - Unicast: transmitting to a single client
  - Multicast: transmitting to subscribed clients
  - Broadcast: transmitting to everyone

Questions

- Q: What are redundant codewords?
- A: Redundant codewords are just the erasure coded blocks, as opposed to the data itself. This refers to the difference between message packets in the interleaved scheme.

Questions

- Q: What is a lossy environment?
- A: A lossy environment is one where packets are lost frequently.

Questions

- Q: What are *wBlocks*, *s*, and *b*?
- A: *wBlocks* is the number of blocks written by a user and s is the time. So, *wBlocks/s* is the rate that a user produces data. The variable *b* is just the block size, where B is the number of blocks.

Questions

- Q: What is the Repair Epoch?
- A: The Repair Epoch e is the period of time that blocks are revisited and examined for repair.