Rabin's Information Dispersal Algorithm: A Prescient Look at Coding on Networks

Michael Mitzenmacher

Coding in Networks, Backwards

Network Coding

Fountain Codes

Information Dispersal

JACM, April 1989

Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance

MICHAEL O. RABIN

Harvard University, Cambridge, Massachusetts

Abstract. An Information Dispersal Algorithm (IDA) is developed that breaks a file F of length $L = |F|$ into n pieces F_i, $1 \leq i \leq n$, each of length $|F_i| = L/m$, so that every m pieces suffice for reconstructing F. Dispersal and reconstruction are computationally efficient. The sum of the lengths $|F_i|$ is $(n/m) \cdot L$. Since n/m can be chosen to be close to 1, the IDA is space efficient. IDA has numerous applications to secure and reliable storage of information in computer networks and even on single disks, to fault-tolerant and efficient transmission of information in networks, and to communications between processors in parallel computers. For the latter problem provably time-efficient and highly fault-tolerant routing on the n-cube is achieved, using just constant size buffers.

The IDA Approach

- A file consists of *N* symbols = numbers mod p for large prime *p*.
- Split the file up into *N*/*m* pieces of *m* symbols
- For each piece, derive *n* encoded symbols, each by a linear combination of the *m* symbols.
  - Use the same coefficients for the linear combinations for each piece
- Derive *n* packets, with the *i*th packet containing the *i*th derived symbol for each of the *N*/*m* pieces.
- Given *m* packets, with *N* total symbols, can invert a matrix to solve for the original message.

Coding

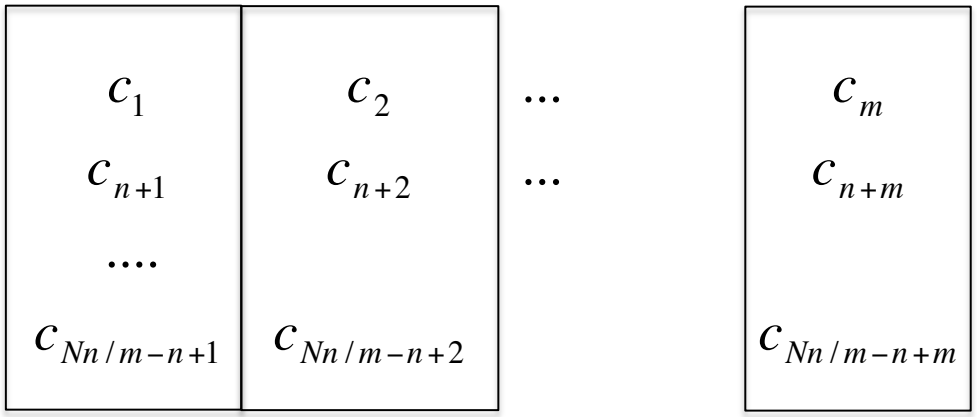
$$\begin{array}{cccc} b_1 & b_2 & b_3 & \dots & b_m \\ b_{m+1} & b_{m+2} & b_{m+3} & \dots & b_{2m} \\ \dots & & & & \\ b_{N-m+1} & b_{N-m+2} & b_{N-m+3} & & b_N \end{array}$$

$$c_{in+j} = (a_{j,1}, a_{j,2}, \dots a_{j,m}) \bullet (b_{im+1}, b_{im+2}, \dots b_{i(m+1)})$$

c_1	c_2	…	c_n
c_{n+1}	c_{n+2}	…	c_{2n}
….			
$c_{Nn/m-n+1}$	$c_{Nn/m-n+2}$		$c_{Nn/m}$

Packets

Decoding



For convenience, assume you get first *m* packets

$$\text{Let } A = (a_{i,j}), 1 \le i, j \le m$$

$$A \bullet \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} = A^{-1} \bullet \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix}$$

Need to be able to invert coefficient matrix, for any set of packets that is received.

Coefficients

- Need the corresponding matrix to be invertible.
- Solution 1: Coefficient chosen according to a Cauchy matrix.
  - Resulting m by m matrix of received coefficients can be inverted in $\Theta(m^2)$ time.
- Solution 2: Vandermonde matrix.
- Solution 3: Choose the coefficients $a_{i,j}$ randomly.
  - Linearly independent with high probability.
  - But $\Theta(m^3)$ decoding by standard means.

Path to Digital Fountains

- Quadratic decoding + field operations too slow for big files.
  - "Obvious" solution – break file up into smaller subfiles and encode those – has various problems in many settings.
- Inspired Michael Luby (and others) to search for improvements.

Cauchy Codes

Let $x_1, \ldots, x_n, y_1, \ldots, y_m \in Z_p$ satisfy the conditions: For all i and j

$$x_i + y_j \neq 0; \qquad i \neq j \rightarrow x_i \neq x_j \quad \text{and} \quad y_i \neq y_j, \tag{3}$$

(this requires $n + m < p$). Define

$$a_i = \left(\frac{1}{x_i + y_1}, \ldots, \frac{1}{x_i + y_m} \right), \qquad 1 \leq i \leq n.$$

Let A be the matrix with rows $a_1, \ldots, a_m$ then [5, p. 35]

$$|A| = \frac{\prod_{i<j} (x_i - x_j)(y_i - y_j)}{\prod_{i,j} (x_i + y_j)}. \tag{4}$$

It follows from (3) and (4) that $|A| \neq 0$. Thus any m vectors in $\{a_1, \ldots, a_n\}$ are linearly independent.

Cauchy Codes

Furthermore, if $A^{-1} = [b_{ij}]$, then $b_{ij} = (-1)^{i+j}|A(j, i)|/|A|$, where $A(j, i)$ is the matrix obtained from A by deleting the jth row and ith column. Denote, for $1 \le k \le m$,

$$c_k = \prod_{\substack{i<k \\ k<j}} (x_i - x_k)(x_k - x_j),$$

$$d_k = \prod_{\substack{i<k \\ k<j}} (y_i - y_k)(y_k - x_j),$$

$$e_k = \prod_{j} (x_k + y_j),$$

$$f_k = \prod_{i} (x_i + y_k).$$

These quantities can be calculated in $O(m^2)$ operations. Now (4) implies

$$|A(j, i)| = \frac{|A|}{c_j \cdot d_i \cdot e_j \cdot f_i} \cdot (x_j + y_j).$$

Thus A^{-1} can be computed by $O(m^2)$ operations.

XOR-Based Cauchy Codes

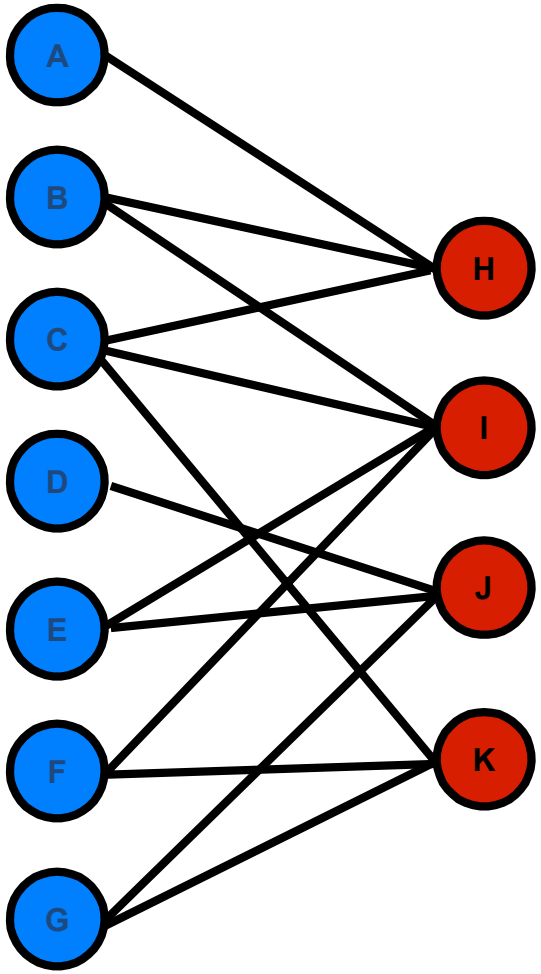
- Paper: An XOR-Based Erasure-Resilient Coding Scheme (Blomer, Kalfane, Karp, Karpinski, Luby, Zuckerman)
- Design code that allows field operations to be replaced by XOR operations.
- Directly utilizes the Cauchy code framework.
- Still $\Theta(m^2)$ time, but much faster in practice.

Tornado Codes

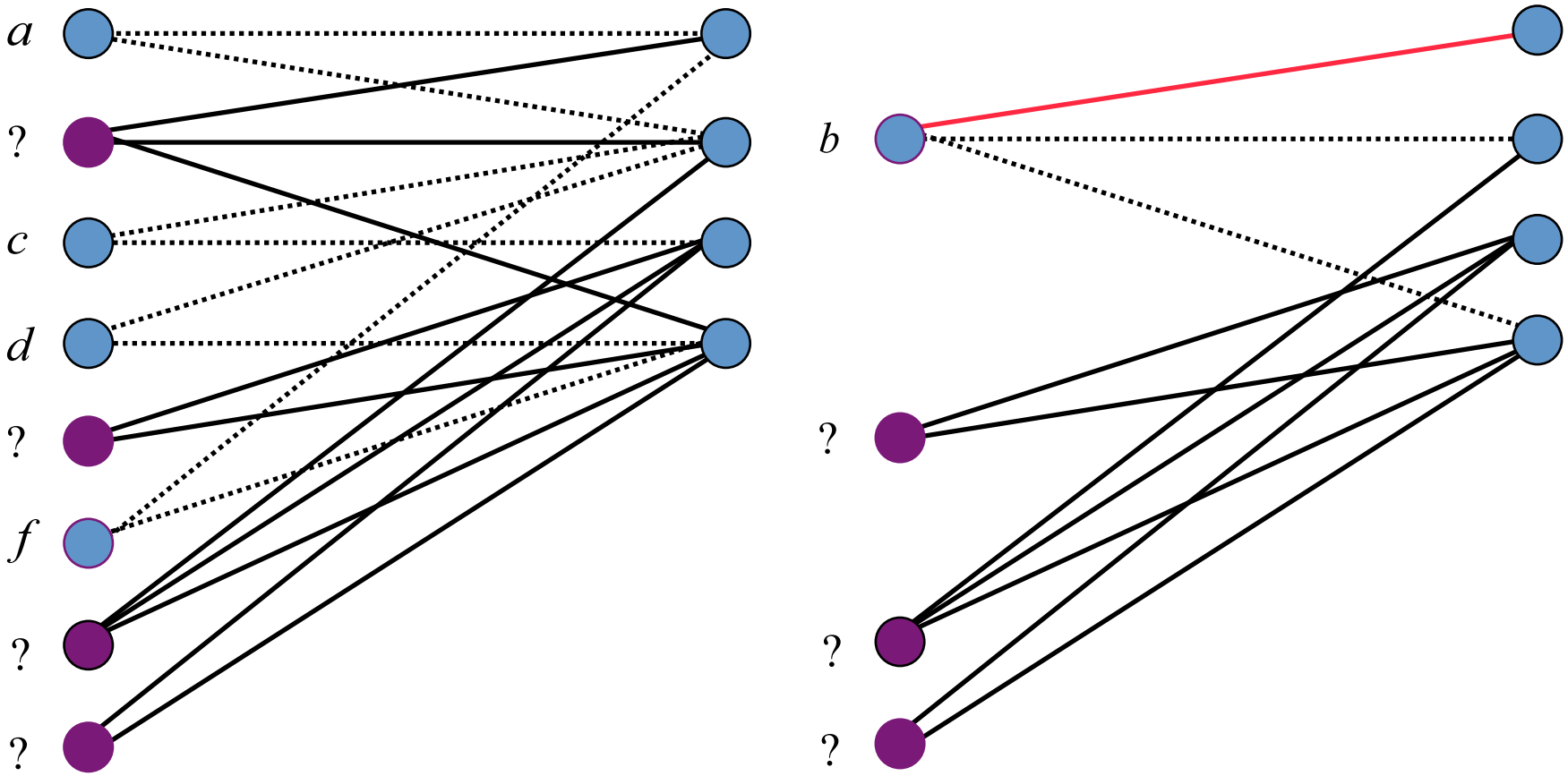
- Paper: Efficient Erasure Correcting Codes (Luby, Mitzenmacher, Shokrollahi, Spielman)
- Replace field operations with XORS
- Packets are derived by *sparse random* combinations of symbols.
  - Prescience of Rabin's use of random matrices.
  - Re-deriving ideas by Gallager : low-density parity-check codes.
- Technical challenge : designing the right sparsity.
  - Irregular combinations of symbols.

Low Density Parity Check Codes Setup

- View code as a bipartite graph.
- Variable nodes on left.
- Check nodes on right.
- Codewords satisfy: sum of all neighbors of a check node is 0.
- Note: sum allows general q. For bits/ packets, sum is XOR.
- Regular degree distributions (all nodes on left have same degree, all nodes on right have same degree NOT optimal.)
- Sparsity leads to faster decoding.



Decoding Process: Recovery



Digital Fountain Paradigm

- Paper : A Digital Fountain Approach to Reliable Distribution of Bulk Data (Byers, Luby, Mitzenmacher, Rege)
  - See also Digital Fountains: A Survey and Look Forward (Mitzenmacher)

What is a Digital Fountain?

- A *digital fountain* is an idealized paradigm for data transmission.
  - Vs. the standard (TCP) paradigm: data is an ordered finite sequence of bytes.
- Instead, with a digital fountain, a *m* symbol file yields an infinite data stream; once you have received any *m* symbols from this stream, you can quickly reconstruct the original file.

Ideals for Digital Fountain

- O(1) time to generate new encoded packet
- "Infinite" supply of packets that can be generated online.
- Information-theoretically optimal – receive only *m* packets to decode *m* packet message.
- Linear decoding time.
- …

Tornado to LT to Raptor…

- Continuing improvements in codes to get closer and closer to Digital Fountain paradigm.
- Rabin's scheme is information-theoretically optimal.
  - Variation of Reed-Solomon codes.
- Others are not – require slightly more than *m* packets received to recover *m* packets of info.
  - To get the large decoding speed gain.

Raptor Codes

- Developed by Amin Shokrollahi.
- Closest to Digital Fountain paradigm.
- Key ideas:
  - Precode data with a fixed-rate erasure code.
    - So you only need to recover 99% of the data.
  - Then encode by taking a random XOR of a random number of symbols.
    - According to right distribution on number of symbols.
  - Near-infinite supply of packets, generated online in constant time, very close to information theoretically optimal

Prescience : Applications

- The IDA paper suggests many applications.
  - Fault-tolerant distributed storage.
    - Node failures.
  - Fault-tolerant transmission.
    - Link failures.
    - Multipath routing ; load distribution.
  - Routing in parallel computers.
    - Studies the hypercube; Valiant-style routing for large messages.

Recurring Themes

- Same issues re-examined with new codes.
  - Parallel downloading
    - Accessing Multiple Mirror Sites in Parallel
  - Dissemination in overlay networks
    - Bullet : High Bandwidth Data Dissemination Using an Overlay Mesh
    - Informed Content Delivery Across Adaptive Overlay Networks
  - Storage
    - Distributed Fountain Codes for Network Storage
    - On the Practical Use of LDPC Codes for Distributed Storage
- And many, many more papers.

Network Coding

- What we missed : coding beyond the endpoints, inside the network.
- Main idea of network coding : can “combine” encoded packets inside the network to derive new encoded packets.

Network Coding : A Standard Approach

- E.g., A Random Linear Network Coding Approach to Multicast (Ho, Medard, Koetter, Karger, Effros, Shi, Leong)
- A packet contains (one or more) linear combinations (over a suitable field) of symbols.
- If two packets meet in the network (say at a buffer), can derive a new linear combination by multiplying each packet by a random multiplier.
- If old packets had random coefficients, so do the new packets!
  - Implies likelihood of invertibility still high.

Prescience : Random Coding

If we want every a_i to depend on more than one parameter α_i, we can simply choose $a_i = (a_{i1}, \ldots, a_{im})$ *randomly*, by randomly and independently selecting the residues $a_{ij} \in Z_p$. Every $m \times$ m matrix A, obtained by selecting m different vectors out of $\{a_1, \ldots, a_n\}$ as rows, is again a randomly chosen matrix. It is readily seen that

$$1 - \frac{1}{p} \cdot \frac{p}{p-1} \leq \Pr(A : |A| \not\equiv 0) \leq 1 - \frac{1}{p},$$

so that for $100 < p$ with probability nearly $1 - (1/p)$, the matrix is nonsingular.

.....

In implementing the file dispersal scheme, it is useful to include the coding vector $a_i = (a_{i1}, \ldots, a_{im})$ as a header of the piece F_i, so that $F_i = d_{i1}\, d_{i2} \cdots d_{iM}$, where $M = N/m + m$, $d_{i1} = a_{i1}$, $d_{i2} = a_{i2}$, $d_{i,m+1} = c_{i1}$, etc. In this way there is no need to store the vectors $a_1, \ldots, a_n$ separately, they also are protected by dispersal.

Keys to Random Linear Network Coding

- Random linear network coding depends significantly on random coefficients, keeping coefficients in the header.
  - Both ideas in the IDA paper…
- Neither really *needed* to be in the IDA paper.
  - Had alternate schemes, including more efficient Cauchy schemes.
- But Michael has *always* understood the importance and utility of random methods.
  - Even if the application was not immediate.

Network Coding + TCP

Original message : p_1, p_2, p_3…

Coded Packets

$4p_1 + 2p_2 + 5p_3$

c_1	4	2	5	0	0	0	0
c_2	3	1	2	5	0	0	0
c_3	1	2	3	4	1	0	0
c_4	3	3	1	2	1	0	0
c_5	1	2	5	4	5	0	0

$$\begin{pmatrix} 4 & 2 & 5 & 0 & 0 & 0 & 0 \\ 3 & 1 & 2 & 5 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 1 & 0 & 0 \\ 3 & 3 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 5 & 4 & 5 & 0 & 0 \end{pmatrix}$$

When c_1 comes in, you've "seen" packet 1; eventually you'll be able to decode it. And so on…

Network Coding + TCP

Original message : p_1, p_2, p_3…

Coded Packets

$4p_1 + 2p_2 + 5p_3$

c_1	4	2	5	0	0	0	0
c_2	3	1	2	5	0	0	0
c_3	1	2	3	4	1	0	0
c_4	3	3	1	2	1	0	0
c_5	1	2	5	4	5	0	0

$$\begin{pmatrix} 1 & 4 & 5 & 3 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 6 & 0 & 0 \\ 0 & 0 & 1 & 6 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Use partial Gaussian elimination as packets arrive to check for a packet that can be removed from the TCP buffer at the sender and send a corresponding ACK.

Prescience : Security

- Revelation of information (IDA is not a “secret sharing scheme”).
  - Lightweight encryption schemes.
- Attack : insertion of fake packets.
  - Digital signature schemes.

Repeated Themes

- XORs
  - XORs in the Air: Practical Wireless Network Coding
- Applications
  - Network Coding for Large Scale Content Distribution
  - A Random Linear Network Coding Approach to Multicast
- Security
  - Secure Network Coding
  - An Efficient Signature-Based Scheme for Securing Network Coding Against Pollution Attacks

Conclusion

- Coding schemes for networks on the rise in the last twenty ears.
  - In research, and increasingly in practice.
- Rabin's IDA paper laid foundations for much of what has followed.
  - Covers many ideas and issues that re-appear throughout subsequent work.
- A paper ahead of its time.
  - An inspiration to those working on the boundary of coding and theoretical computer science.