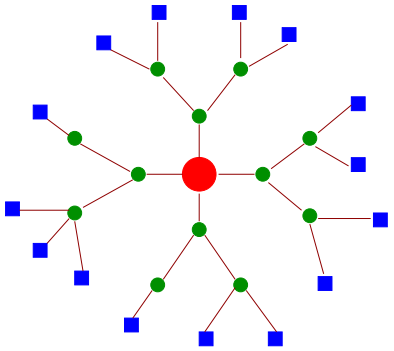
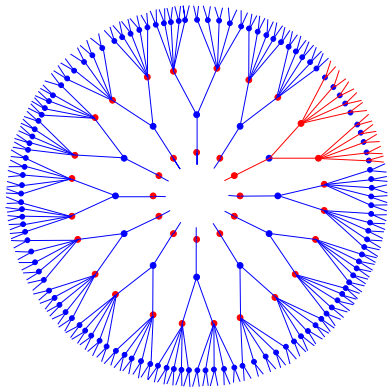
Codes, Graphs, and Algorithms





Amin Shokrollahi
Digital Fountain Research

digitalfountain

Outline

- Codes and information transmission
- Algorithmic issues
- Codes on Graphs

Goal

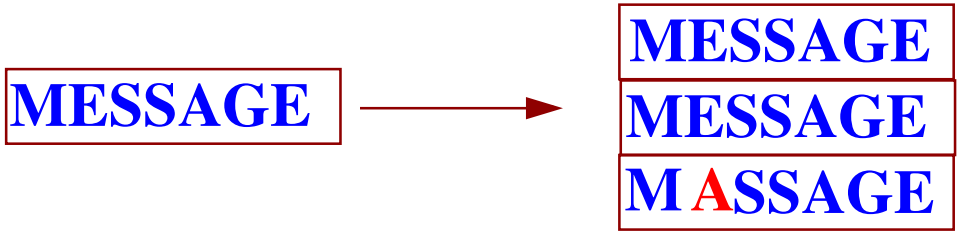
Transmit information reliably over an unreliable communication channel.

Examples:

- Transmission of data between deep space probes and earth station where data may be corrupted.
- Transmission of data on computer networks where data can be lost.
- Storage of information on magnetic disks, where transmission is preserving integrity over time and data may be corrupted.

Basic Idea of Coding

Adding redundancy to be able to correct from errors.



Objectives: Add as little redundancy as possible, correct as many errors as possible.

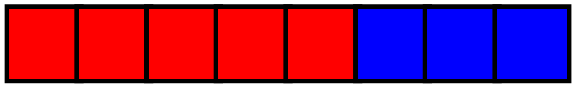
Basic problem: How many errors can we maximally correct for a given amount of redundancy? (Fundamental trade-offs, algorithmic issues)

Basic Parameters

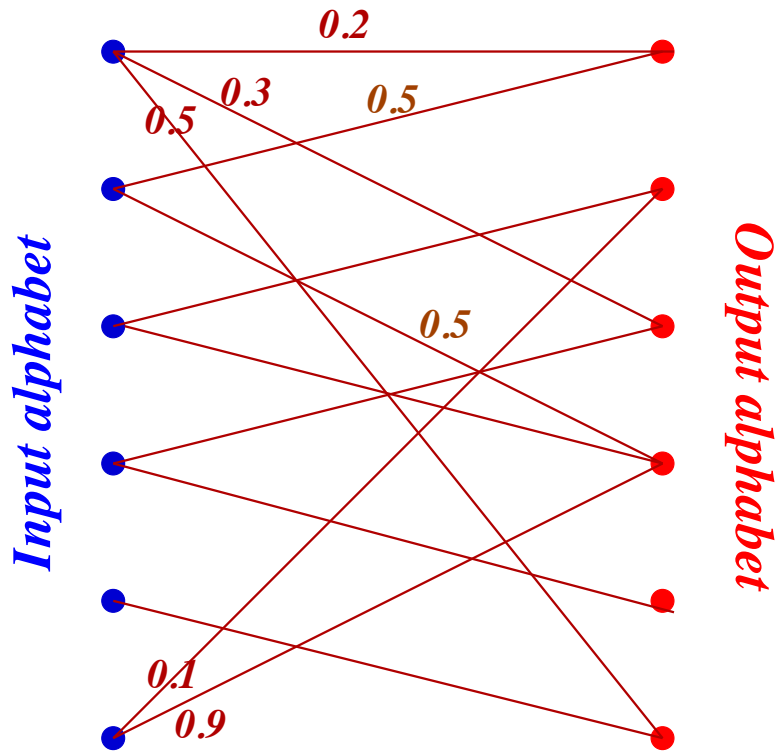
Encoding k bits of information to n bits.

n is the block-length of the code.

The fraction k/n is the rate of the code.



Communication Channels



Transition probabilities

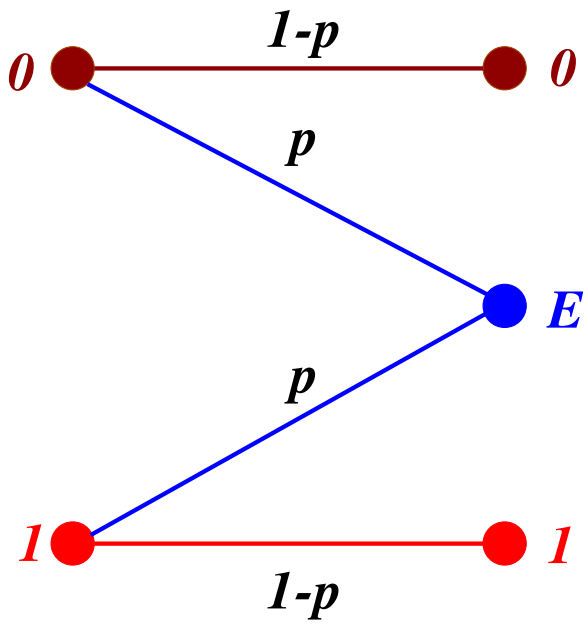
Shannon's Theorem

To every communication channel there is associated a number C, called the capacity of the channel such that for any rate $R \leq C$ there exists a sequence of codes of rate R such that the probability of error of the Maximum Likelihood Decoding for these codes approaches zero as the block-length approaches infinity.

The condition $R \leq C$ is necessary and sufficient.

Examples of Capacity: BEC

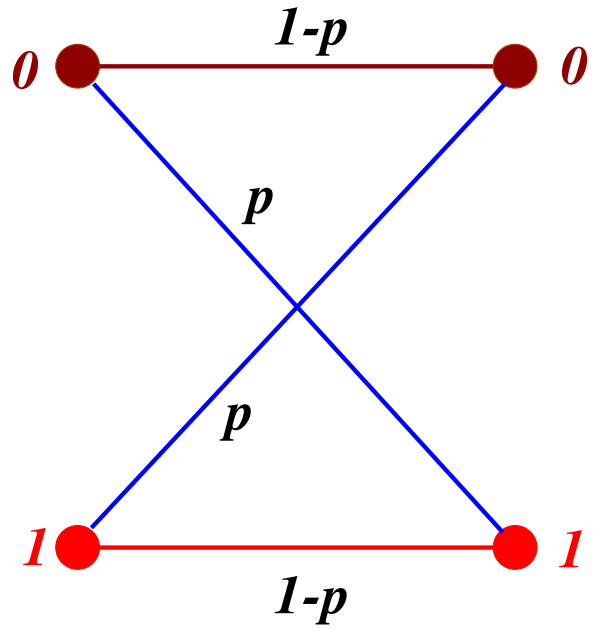
Binary Erasure Channel:



Capacity $= 1 - p$

Examples of Capacity: BSC

Binary Symmetric Channel:



Capacity $= 1 + p\log_2(p) + (1-p)\log_2(1-p)$

Problems

- Shannon's theorem is not constructive (how do we find the codes promised).

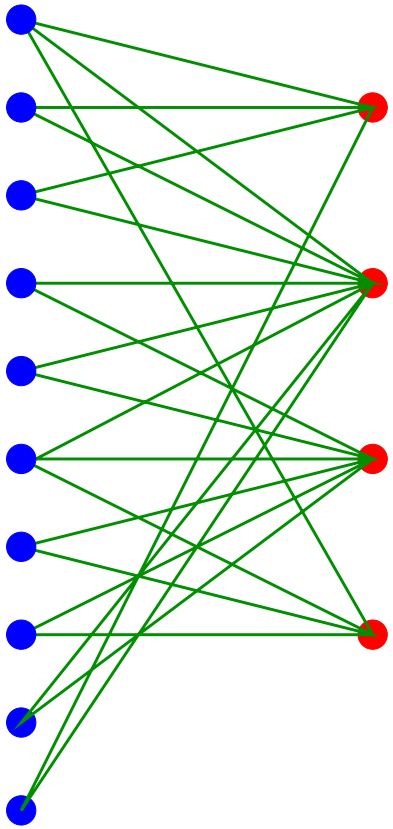
- Even if we can find the codes, the algorithm underlying the theorem (Maximum likelihood decoding) is exponential time, hence impractical.

- Algebraic codes (Reed-Solomon, Algebraic-Geometric, etc.) are very far from reaching the capacity.

Problem has been open for almost 50 years. Groundbreaking progress came from an unexpected turf: Theoretical Computer Science!

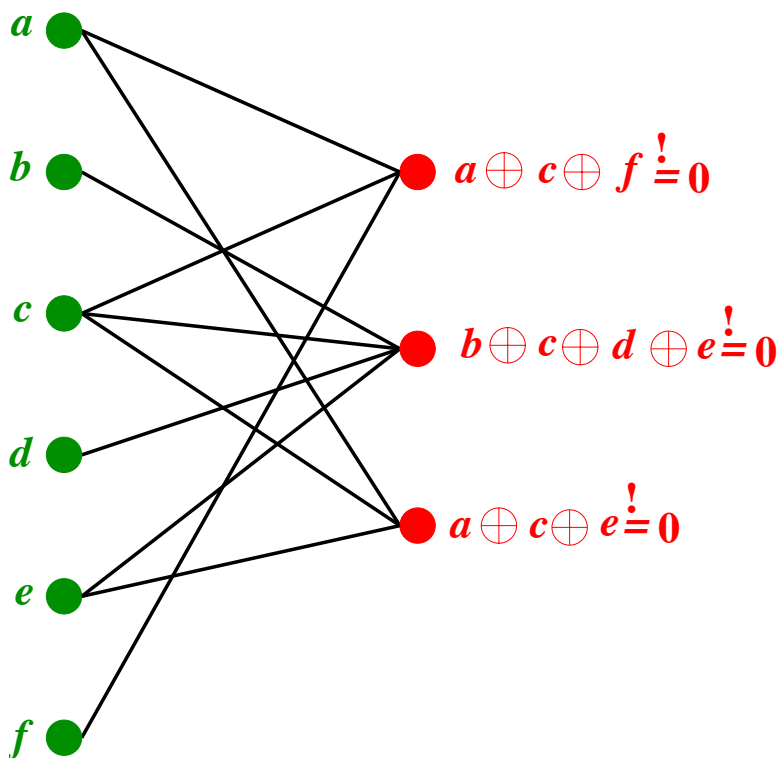
Low-Density Parity-Check Codes

Constructed from sparse bipartite graphs.



Left nodes are called message nodes, right nodes are called check nodes.

Construction



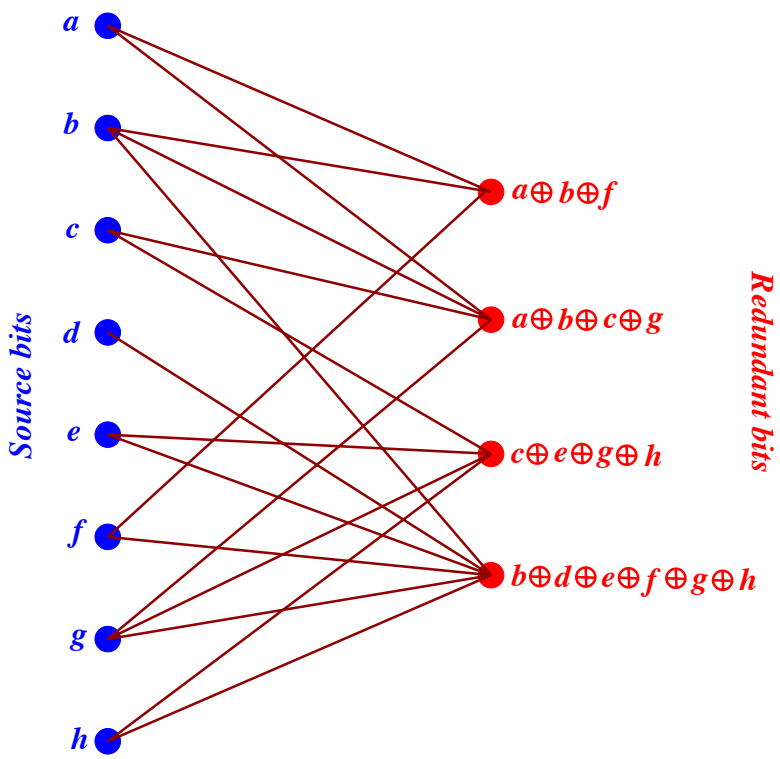
Every binary linear code has such a representation, but not every code can be represented by a sparse graph.

Encoding/decoding times

Encoding is quadratic time using a naive algorithm, but close to linear using a more sophisticated algorithm.

Decoding depends on the communication channel. Concentrate on the erasure channel.

Dual Construction



Encoding time is proportional to number of edges.

Algorithmic Issues

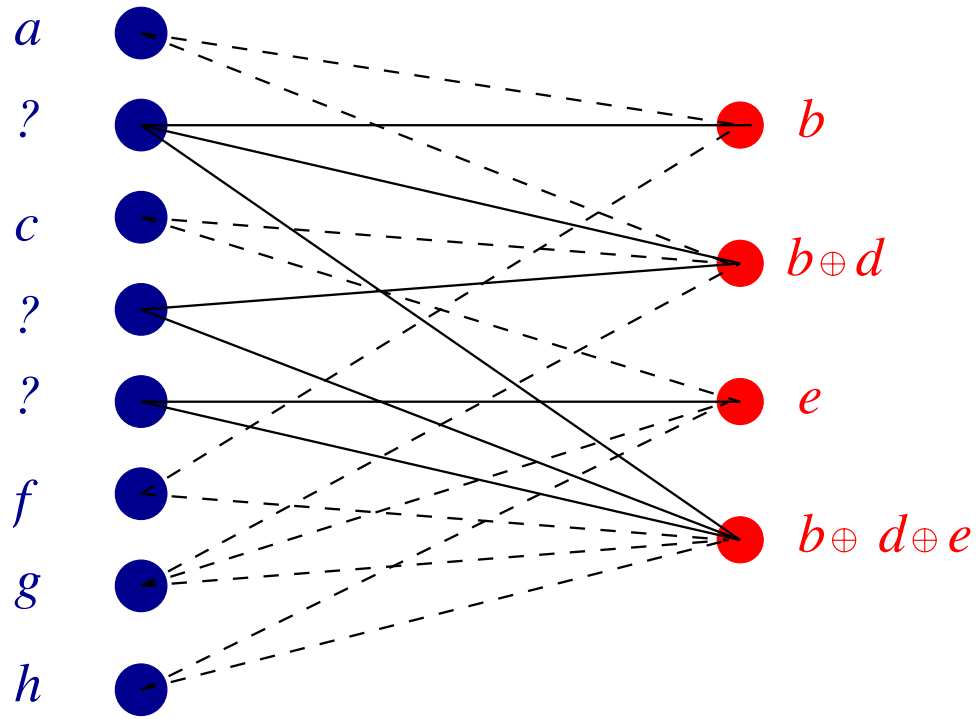
- Encoding?
  - Is linear time for the dual construction
  - Is quadratic time (after preprocessing) for the Gallager construction.
- Decoding?
  - Depends on the channel,
  - Depends on the fraction of errors.

Will concentrate on the erasure channel to clarify the concepts.

Decoding

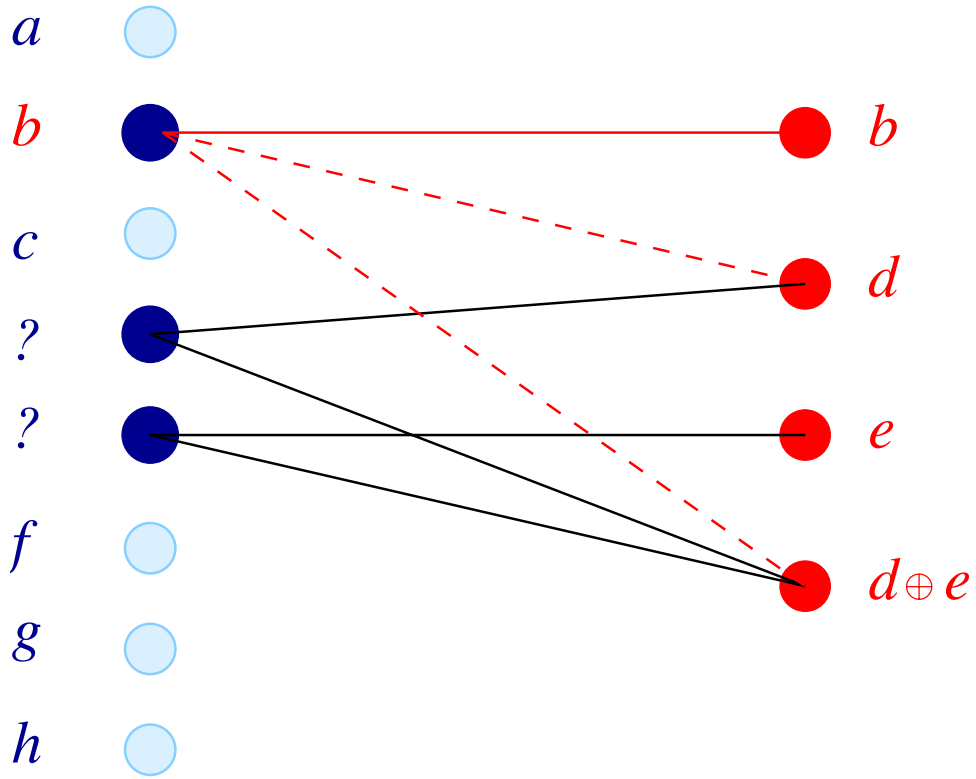
Luby-Mitzenmacher-Shokrollahi-Spielman-Stemann, 1997:

Phase 1: Direct recovery



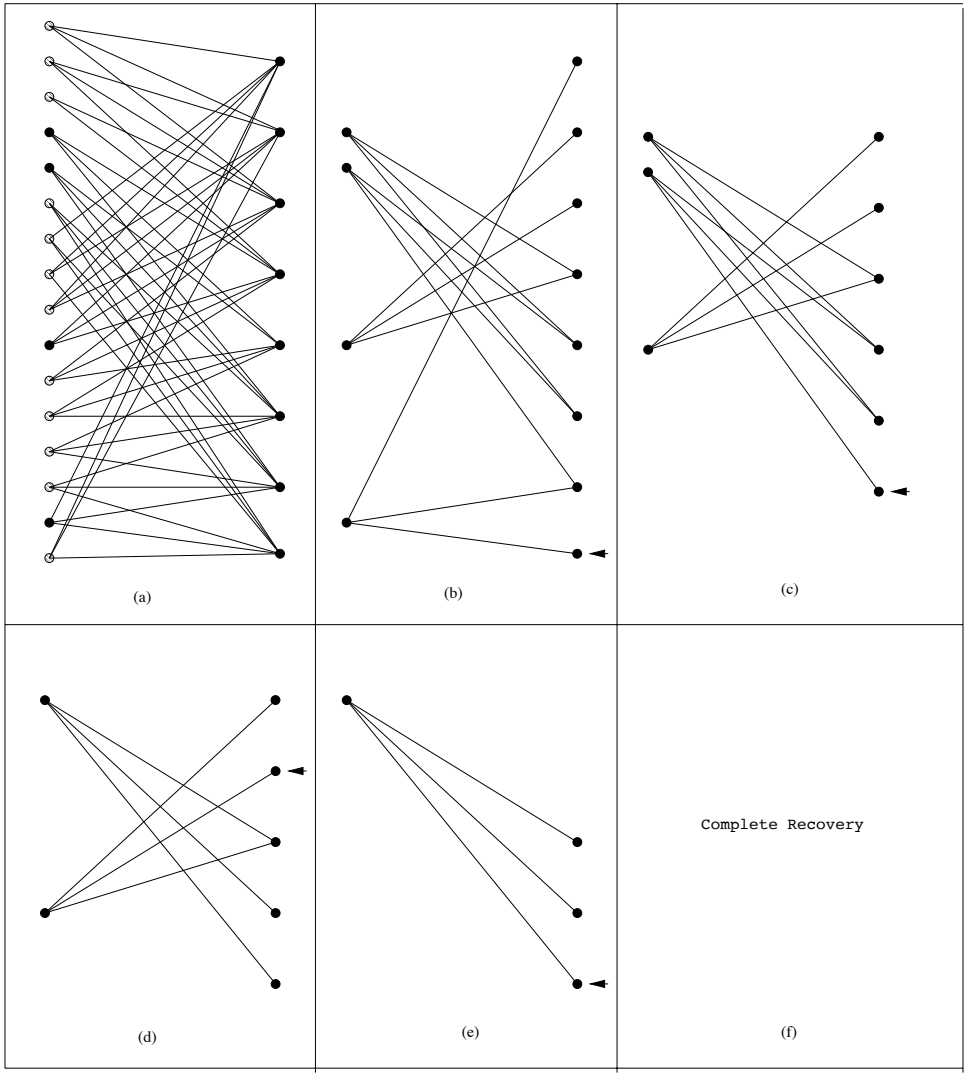
Decoding

Phase 2: Substitution



Decoding time is proportional to number of edges in the graph.

Example



The (inverse) problem

Have: fast decoding algorithms.

Want: design codes that can correct many errors using these algorithms.

Focus on the BEC in the following.

Experiments

Choose regular graphs.

An (d, k)-regular graph has rate at least $1 - d/k$. Can correct at most an d/k-fraction of erasures.

Choose a random (d, k)-graph.

p_0 := maximum fraction of erasures the algorithm can correct.

d	k	d/k	p_0
3	6	0.5	0.429
4	8	0.5	0.383
5	10	0.5	0.341
3	9	0.33	0.282
4	12	0.33	0.2572

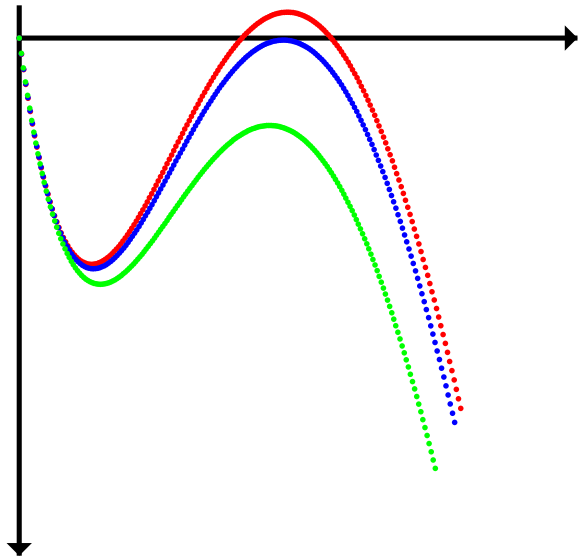
What are these numbers?

Theorem

(Luby, Mitzenmacher, Shokrollahi, Spielman, Stemann 1997) A random (d, k)-graph allows correction of a p_0-fraction of erasures (with high probability) if and only if

$$p_0 \cdot (1 - (1 - x)^{k-1})^{d-1} < x \qquad \text{für } x \in (0, p_0).$$

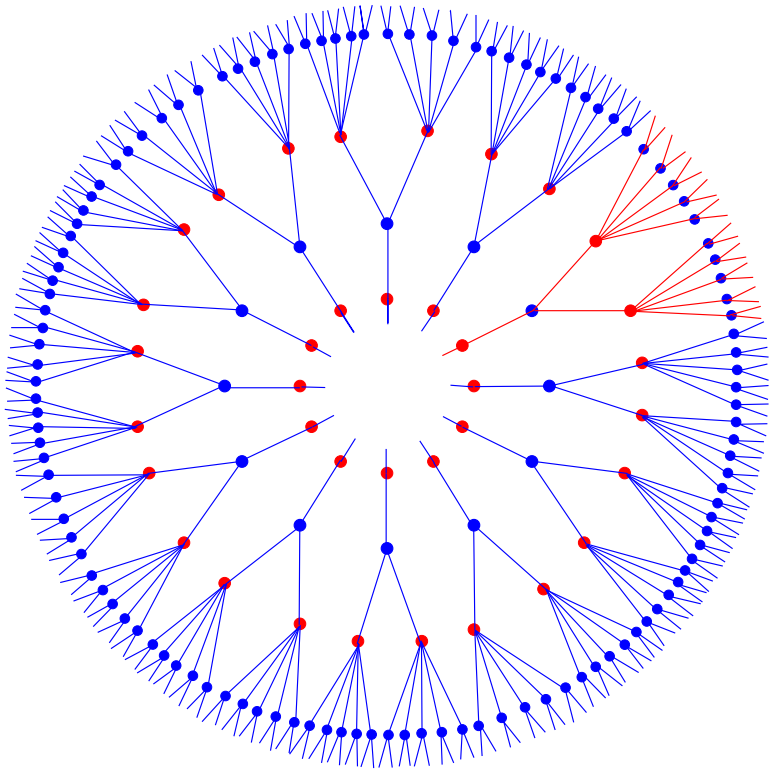
$d = 3, k = 6$: $\quad f(x) = p_0(1 - (1 - x)^5)^2 - x$



$p_0 = 0.435$
$p_0 = 0.429$
$p_0 = 0.41$

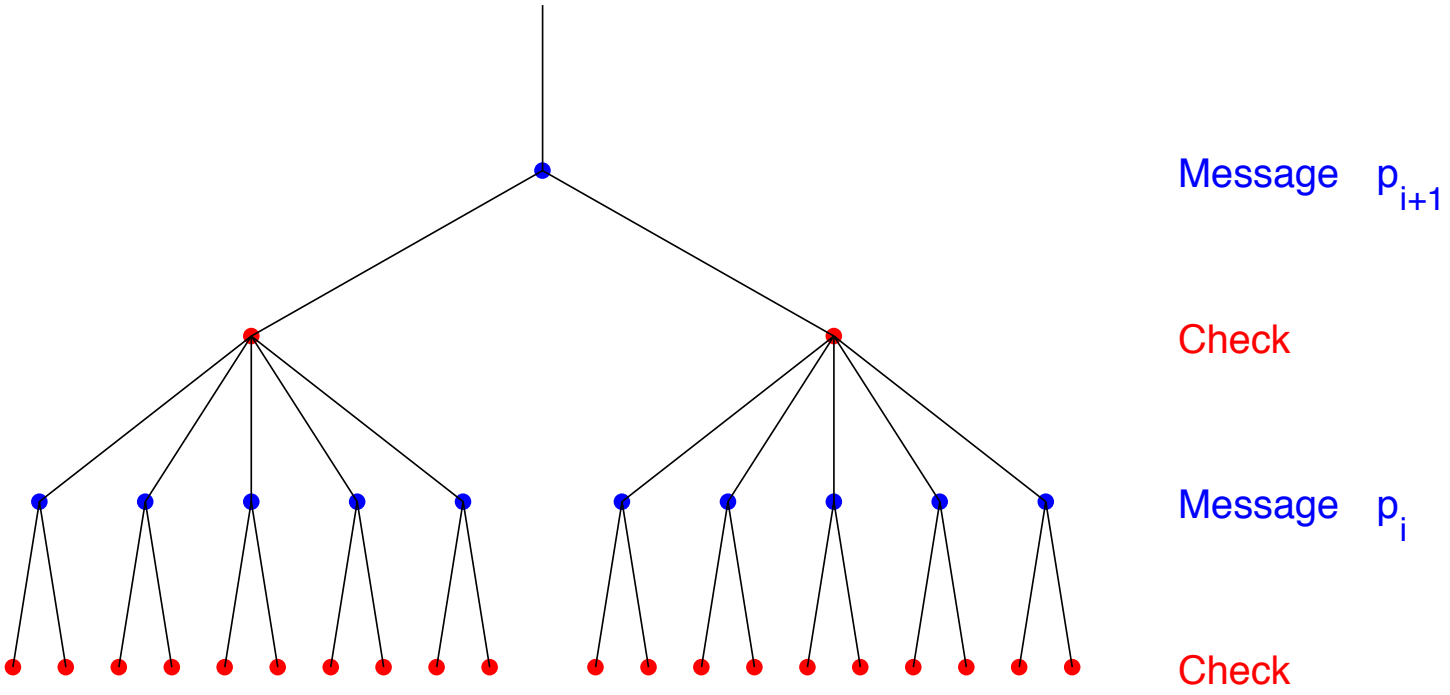
Analysis: $(3, 6)$-**Graph**

Expand neighborhood of message node



Analysis: $(3,6)$-Graph

p_i = Probability that a message node is not corrected after the i-th iteration.



$$p_{i+1} = p_0(1-(1-p_i)^5)^2 < p_i.$$

Analysis: $(3, 6)$-Graph

Rigorous argument:

- Neighborhood is a tree with high probability.

- Above argument works fine for the expected fraction of erasures after the i-th iteration.

  Actual value is concentrated around the expectation p_ℓ: Edge exposure martingale, Azuma's Inequality.

The General Case

λ_i and ρ_i fraction of edges of degree i on the left and the right hand side of the graph.

$\lambda(x) := \sum_i \lambda_i x^{i-1}$, $\rho(x) := \sum_i \rho_i x^{i-1}$.

Condition for successful decoding given a loss fraction p_0:

$$p_0 \lambda\left(1 - \rho(1-x)\right) < x$$

for all $x \in (0, p_0)$.

Achieving Capacity: Tornado Codes

Want to design codes which can asymptotically correct an optimal fraction of erasures, i.e., achieve capacity of the erasure channel.

Design λ and ρ such that

$$p_0\lambda(1-\rho(1-x)) < x$$

for all $x \in (0, p_0)$, and p_0 arbitrarily close to

$$1 - R = \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x}.$$

Tornado Codes

Choose design parameter D:

$$\lambda(x) := \frac{1}{H(D)}\left(x + \frac{x^2}{2} + \cdots + \frac{x^D}{D}\right)$$

$$\rho(x) := \exp\left(\mu(x-1)\right),$$

$H(D) = 1 + 1/2 + \cdots + 1/D$, $\mu = H(D)(1-R)/\left(1 - 1/(D+1)\right)$.

$$p_0\lambda(1-\rho(1-x)) = p_0\lambda(1-\exp(-\mu x)) < -\frac{p_0}{H(D)}\ln(\exp(-\mu x))$$

$$= \mu\frac{p_0}{H(D)}x < x.$$

This is true for $p_0 < H(D)/\mu = (1-R)(1-1/(D+1))$.

Tornado Codes: Efficiency

Need $k \cdot (1 + \varepsilon)$ entries of a codeword to recover the codeword.

Per-bit running time of encoding is $O(\log(1/\epsilon))$.

Per-bit running time of the decoder is $O(\log(1/\epsilon)/R)$.

It can be shown that this is essentially optimal for the class of codes considered.

Theoretical Applications

- Capacity achieving sequences on the erasure channel
- Relationship to random graphs
- Codes on other channels
- Algebraic constructions
- Analysis of finite length codes
- Cryptography
- ...

Practical Applications: The Internet

Want to transport data from a transmitter to one or more receivers over IP reliably.

Current solutions have limitations when amount of data is large and

- Number of receivers is large (point-multipoint transmission)
  - Video on Demand, new versions of computer games
- Or, network suffers from unpredictable and transient losses
  - Satellite, wireless
- Or, connection from transmitter to receiver goes over many hops
  - Software company with development sites around the globe

Cost Measures and Scalability

Cost measures:

- Number of servers
- Outgoing server bandwidth
- Bandwidth utilization

A solution is called *scalable* if its cost does not increase with the number of recipients. (Server-scalable, bandwidth-scalable.)

Are interested in scalable and reliable solutions which maximize bandwidth utilization.

Current Solutions

Current solutions are either not scalable (e.g., TCP/IP), or not reliable (UDP Unicast, UDP Multicast).

Want best of both worlds!

Channel Model

On a computer network data is sent as packets.

Each packet has an identifier which identifies the entity it is coming from and its position within that entity.

Each packet has a CRC checksum to check its integrity.

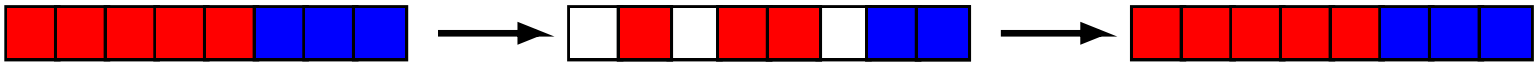
Corrupted packets can be regarded as lost.

Can concentrate on the erasure channel as a model for transmitting packets.

Solution: Codes

Want to have the advantages of Multicast and TCP/IP, but not their disadvantages.

Encode the original data and send encoded version across the network.

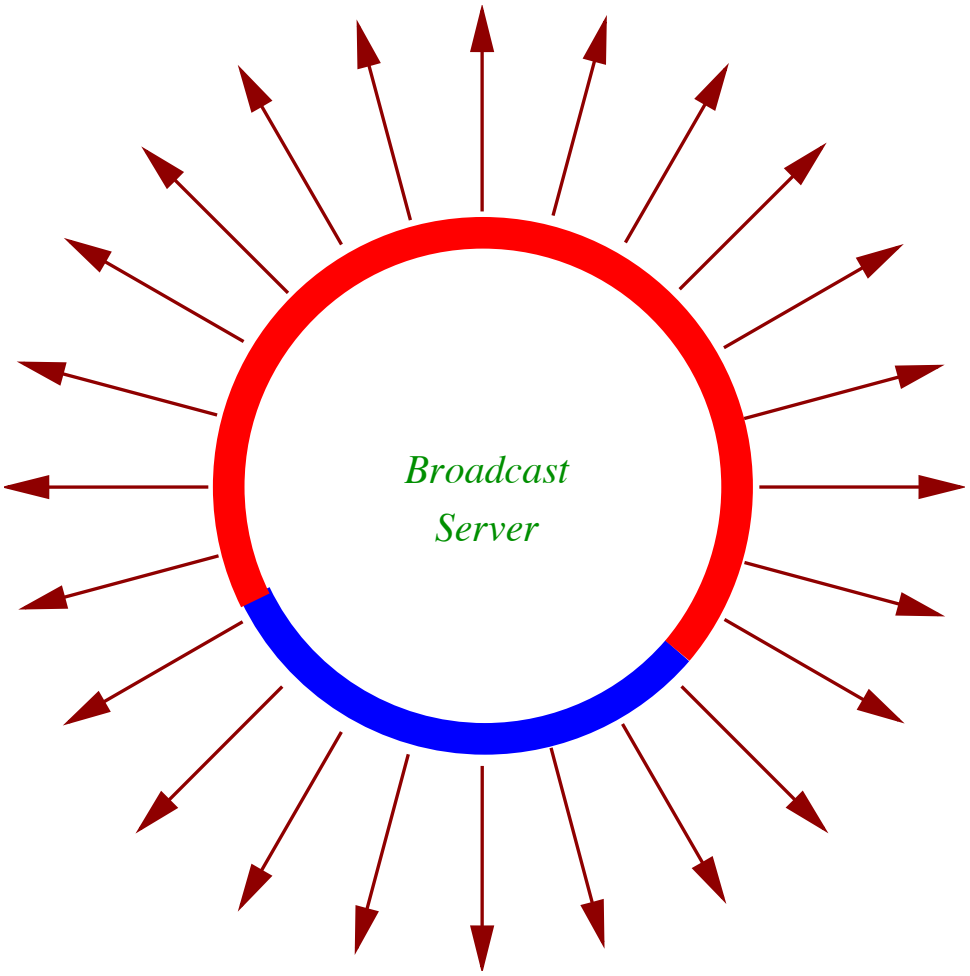


Reconstruction is possible if not too many packets were lost.

Reliability $\rightarrow$ Coding.

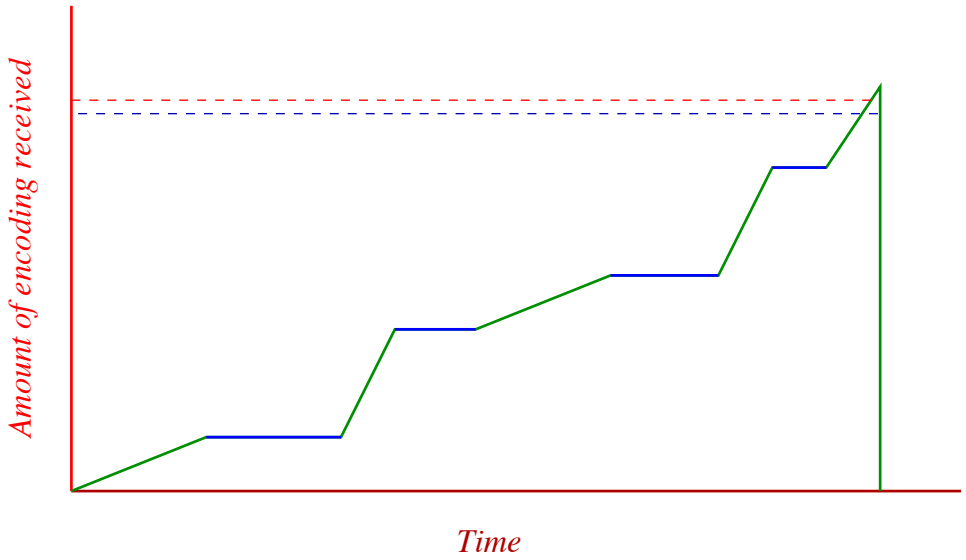
Scalability $\rightarrow$ Multicast (or unicast).

A Solution



A Solution

Client joins multicast group until enough of the encoding has been received, and then decodes to obtain original data.



Shortcomings of Traditional Codes

Protocols involving erasure correcting codes are excellent candidates for replacing TCP/IP in certain applications. However, traditional codes have many disadvantages:

- One has to have a good guess on the loss rate of the clients; this is very difficult in scenarios like mobile wireless.

- Many applications require coordination between senders and receivers to avoid reception of duplicate packets.

- Many applications require codes of very small rate. But, the running time of fast codes like Tornado codes is proportional to the block-length, rather than the length of the original content.

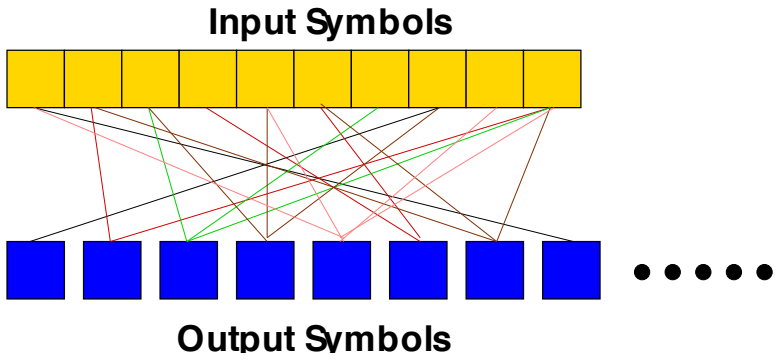
What we Really Want

To design a completely receiver driven and scalable system one needs codes

- That adapt themselves to the individual loss rates of the clients; clients with more loss need longer to recover the content;
- For which the decoding time depends only on the length of the content;
- That achieve capacity of the erasure channel between server and any of the clients;
- That have fast encoding and decoding algorithms.

Beyond Tornado: LT Codes

Michael Luby has invented a class of codes that achieves all these goals.

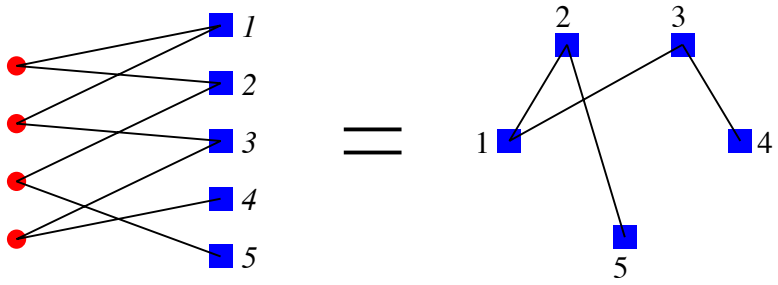


Their per-bit complexity of encoding/decoding is $\log(k)$. (This is optimal.)

If a receiver has loss rate p, then the code corresponding to that receiver has rate $1 - p - c/\sqrt{k}$.

Relationship to Random Graphs

Consider codes in which all message nodes have degree 2. These are graphs on the set of check nodes.



Choose neighbors of message nodes randomly (Poisson distribution). This yields a random graph on the set of check nodes.

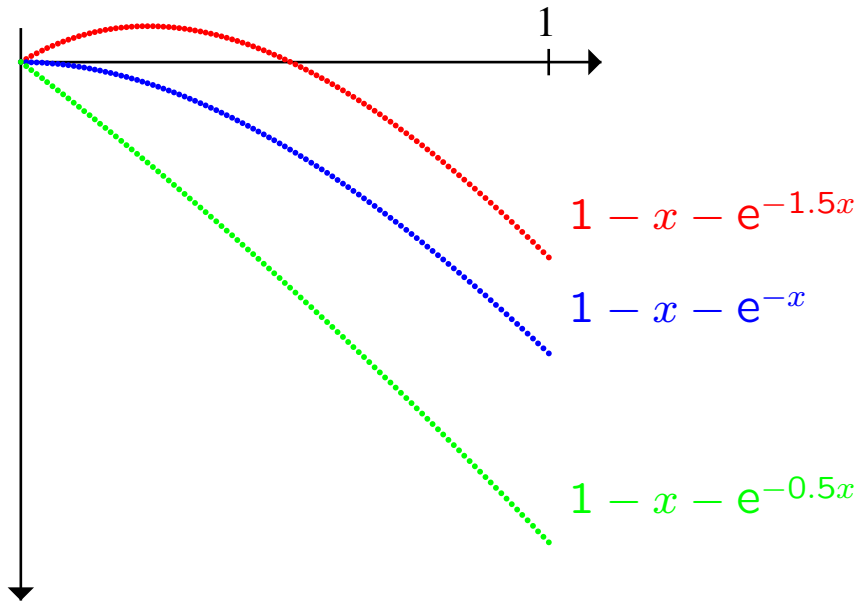
$\lambda(x) = x$, $\rho(x) = \mathrm{e}^{a(x-1)}$, a is average degree.

$\lambda(1 - \rho(1 - x)) < x$, i.e., $1 - x - \mathrm{e}^{-ax} < 0$.

Relationship to Random Graphs

Largest solution of $1 - x - \mathrm{e}^{-ax} = 0$ in the interval $(0, 1)$ gives fraction of the largest component of the graph.

There is a giant component if $a > 1$. The decoder will not be able to correct all erasures iff $a > 1$, i.e., iff there is a giant component.



Relationship to Random Graphs

Above relationship can be put into a general framework, which works even when not all the message nodes are of degree 2.

The Tornado distribution can be obtained from this observation using a "self-similarity" assumption.

This connection also yields linear time encoders.