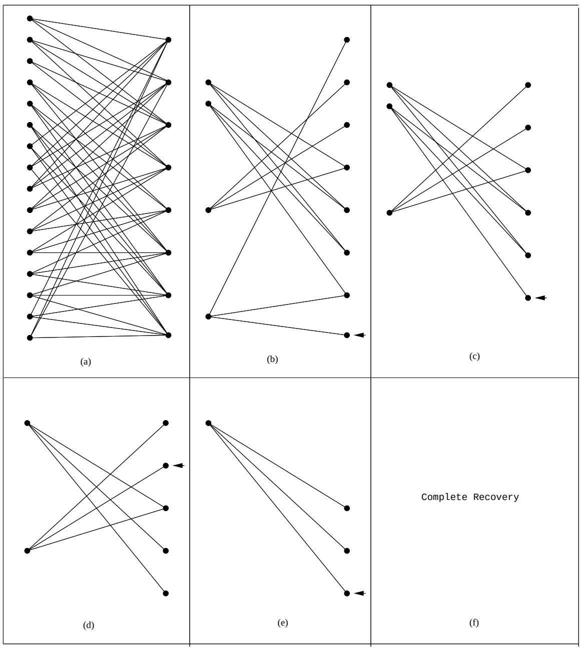
Codes and Graphs

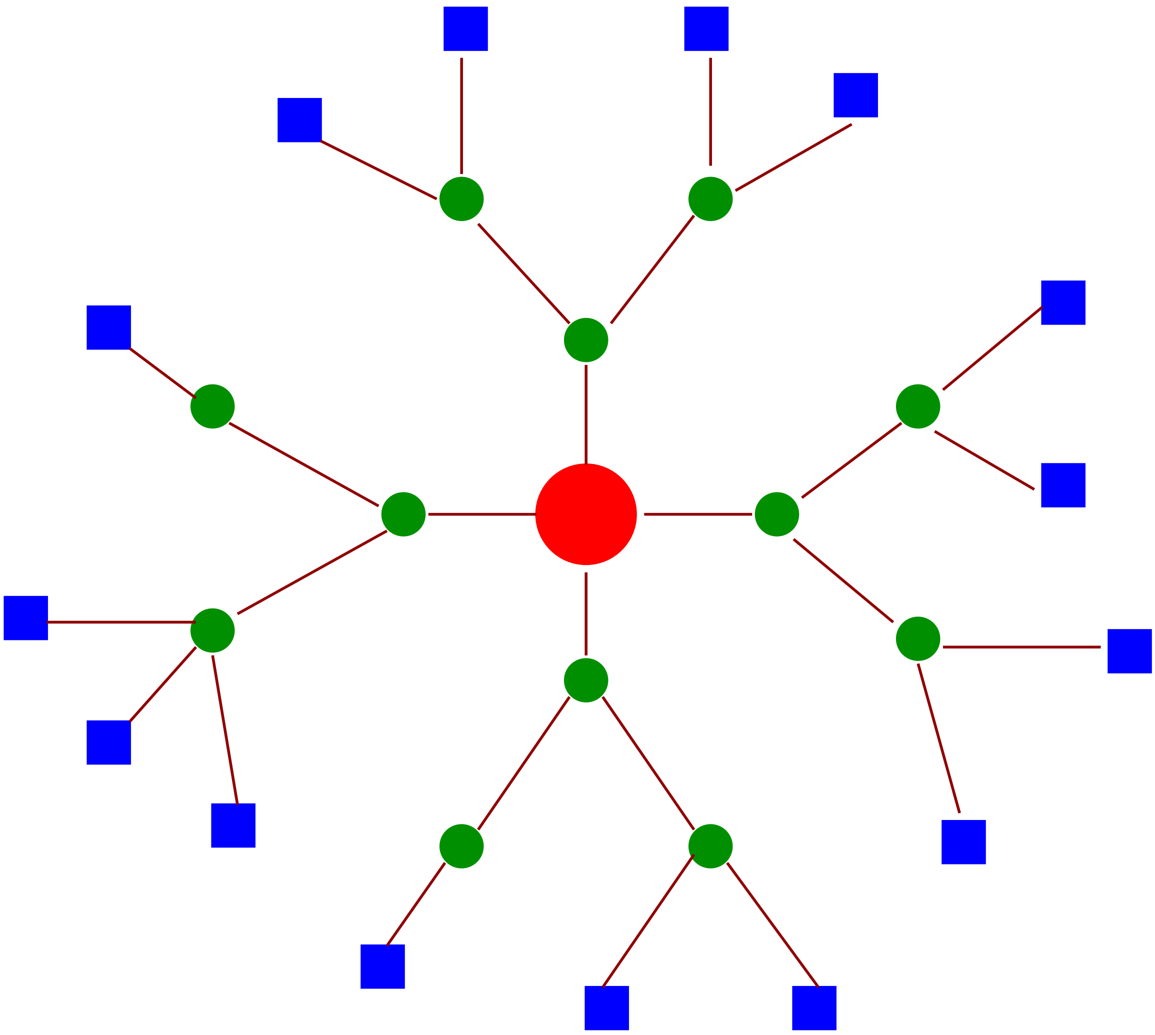


M. Amin Shokrollahi

The Problem

Packets in Networks

Data sent in a network is divided into packets which are routed through the network from a sender to a recipient.



Packet Loss

Each packet has an identifier.

Packets can get lost or corrupted.

Corruption is checked via checksums.

Corrupted packets are regarded as lost.

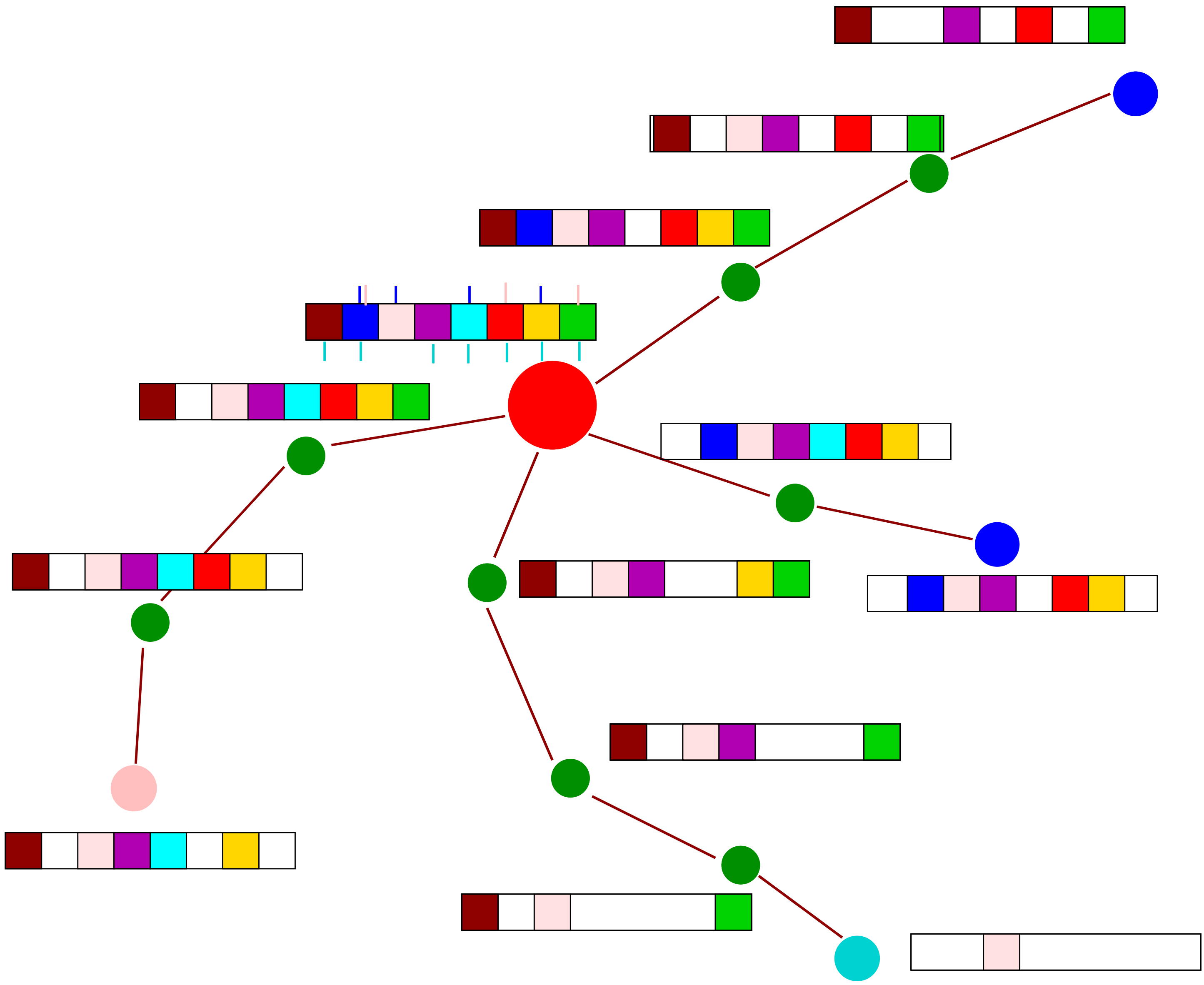
May without loss of generality only concentrate on losses.

4	6	0	1	9	2	7	2	E	5	D	1
5	4	6	E	1	4	9	A	7	9	2	7
0	6	9	6	7	2	C	E	4	1	7	D
B	F	C	3	D	6	F	B	9	2	7	B
3	2	8	E	6	5	E	7	1	5	4	A
6	1	4	3	A	F	3	5	6	C	3	4
C	A	A	5	C	9	C	4	9	5	2	2
9	C	8	3	F	8	E	6	5	3	1	6

Retransmission

In many communication protocols lost packets are re-transmitted.

Process is repeated until all packets are received.

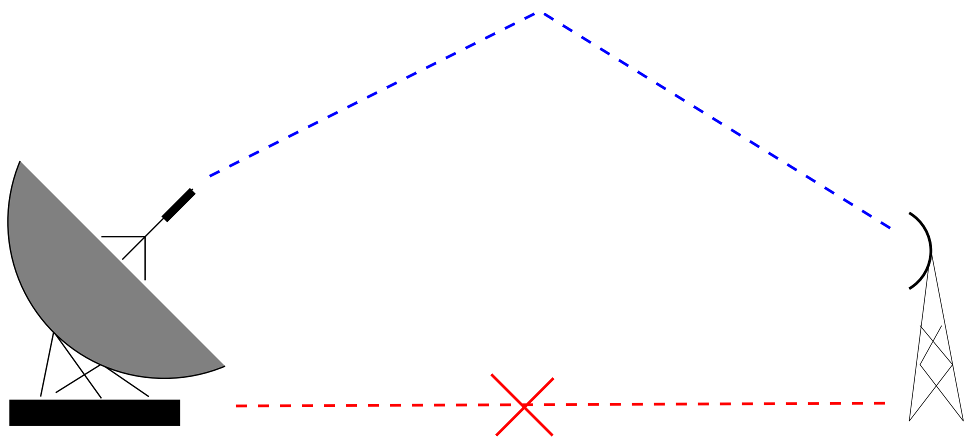


Retransmission Protocols

Requires existence of feedback channel.

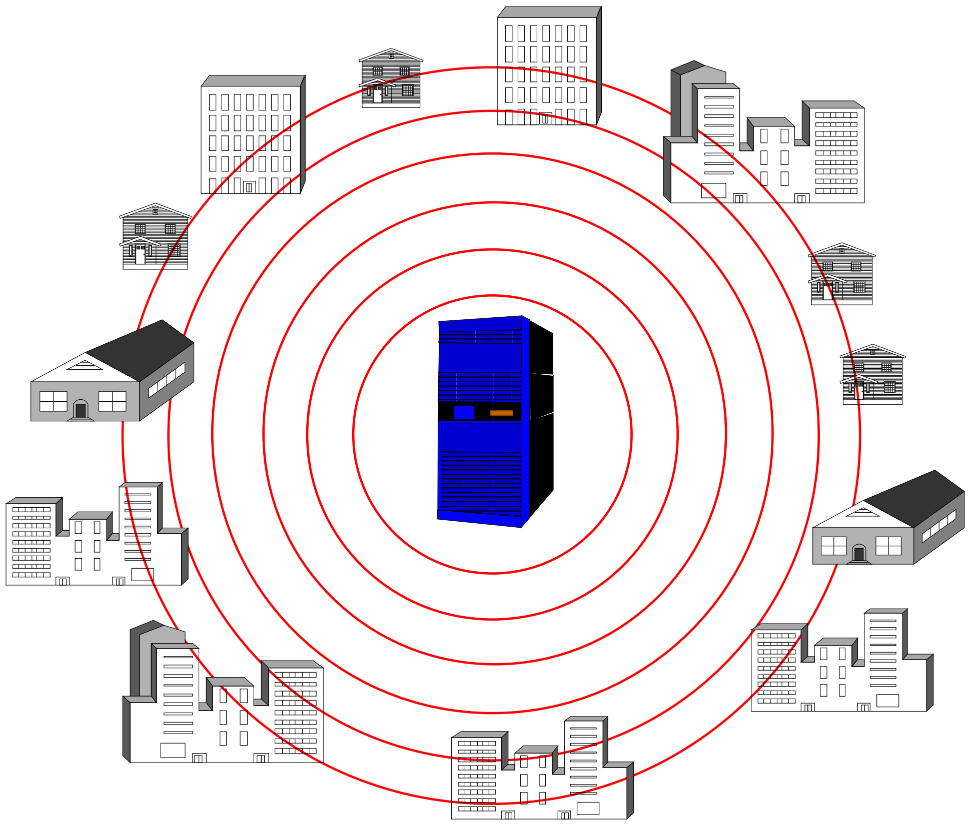
May not exist, or maybe too expensive.

Example: satellite links.



Retransmission Protocols

Not good enough in broadcast application: one server, many clients. Request for retransmission leads to huge server load.



Forward Error Correction

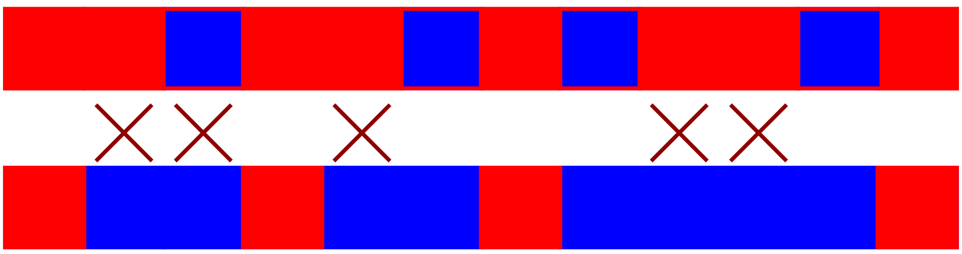
Want to make it in first attempt.

Idea: use linear codes!

block stream of packets into blocks each containing k packets.

Add $n - k$ redundant packets so that recovery possible after some fraction of packets lost.

If the minimum distance of the code is d, then we can always correct up to $d - 1$ erasures.



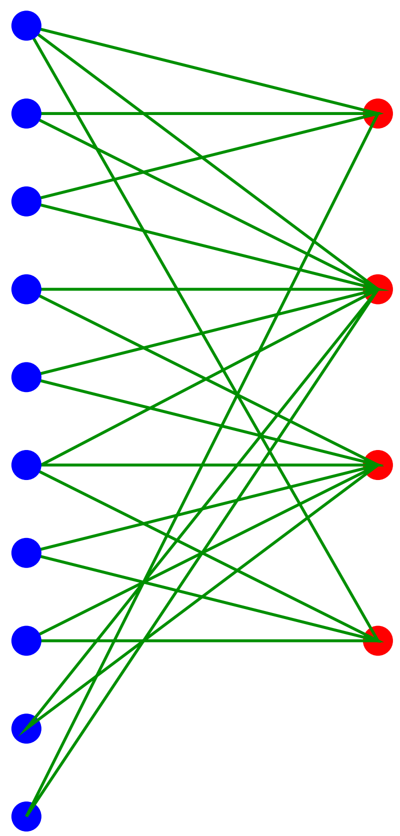
The Codes

Low Density Parity Check Codes

Rediscovered codes that were built almost 40 years ago by R. Gallager.

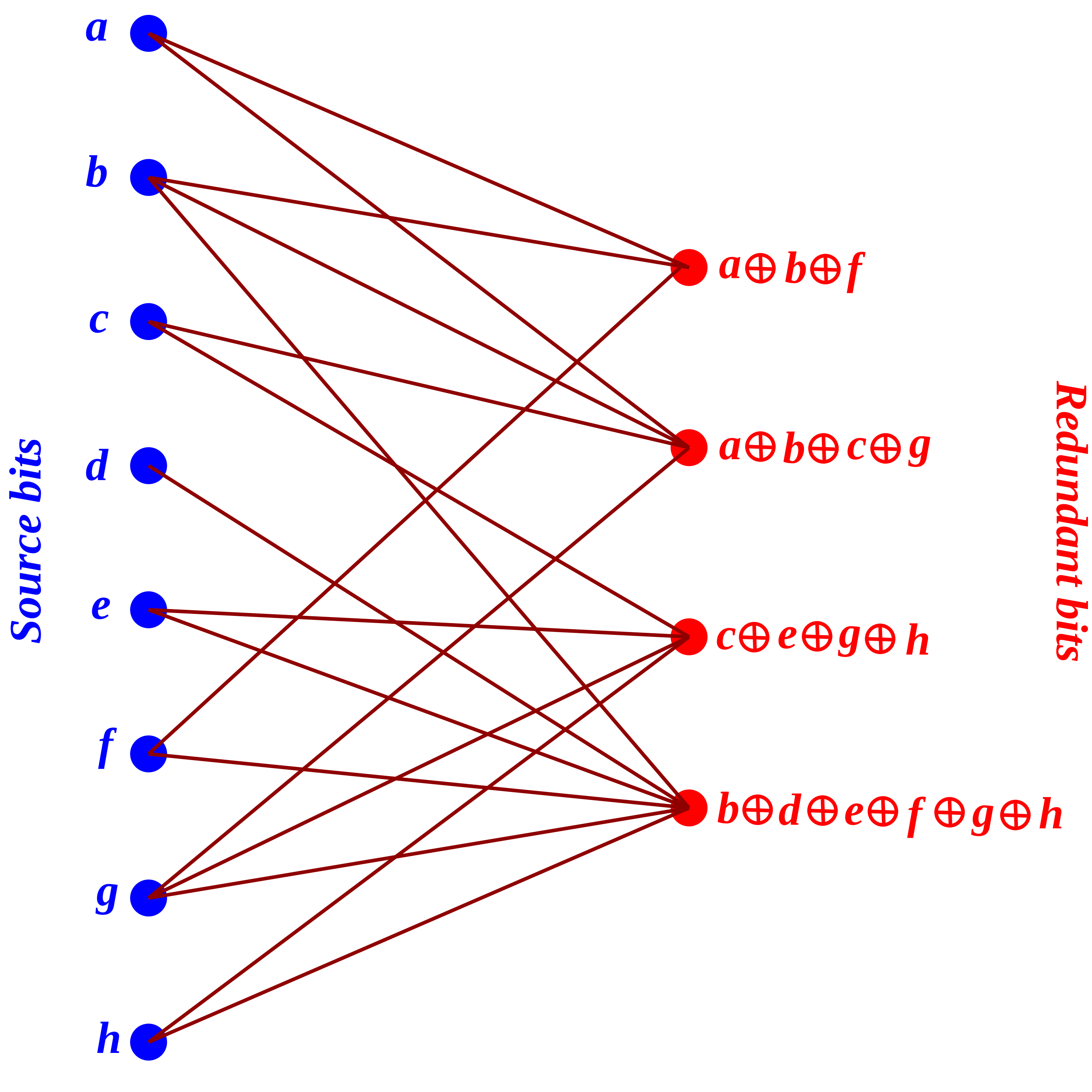
Codes are built from sparse bipartite graphs.

Encoding and Decoding are simple.



Encoding with Bipartite Graphs

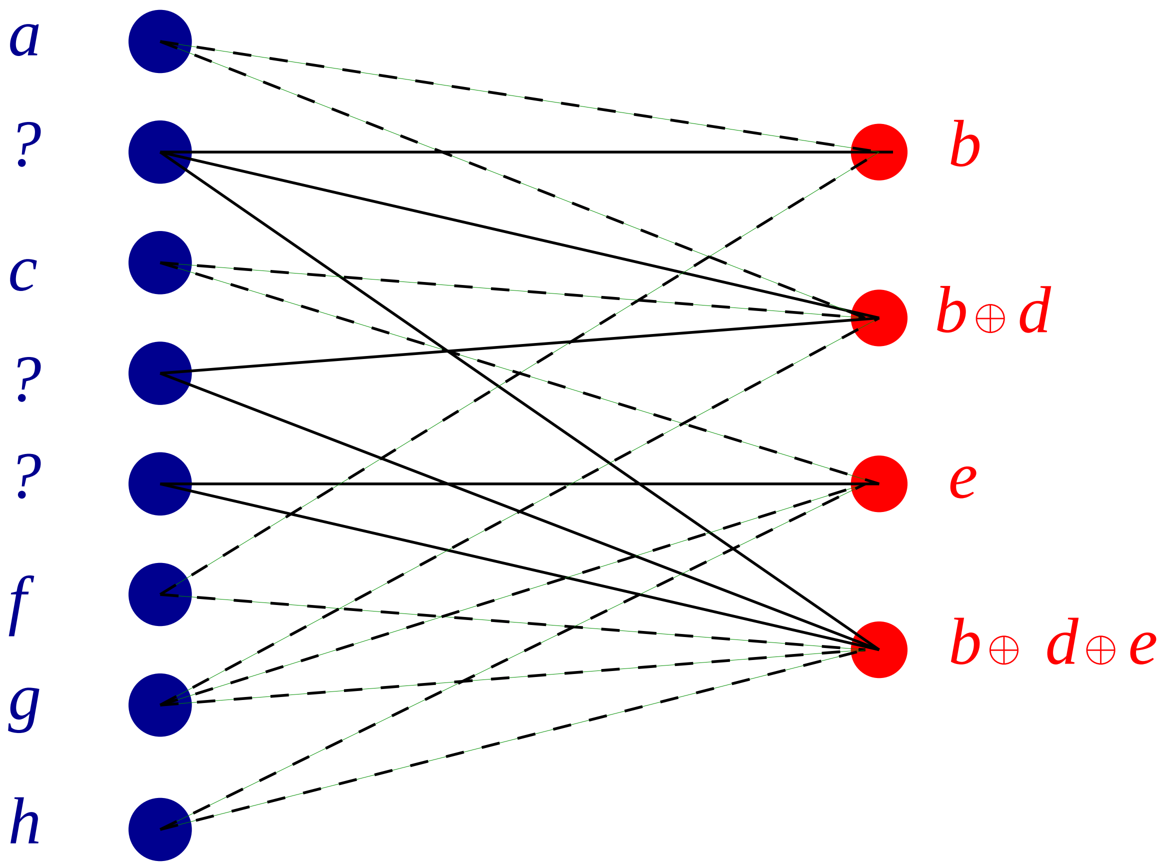
Take a bipartite graph between k nodes on the left and $n-k$ nodes on the right. Label left nodes with the k packets to be encoded. Label right nodes with the redundant packets. Compute value of each right node as XOR of values of adjacent left nodes.



Encoding time is proportional to number of edges in graph.

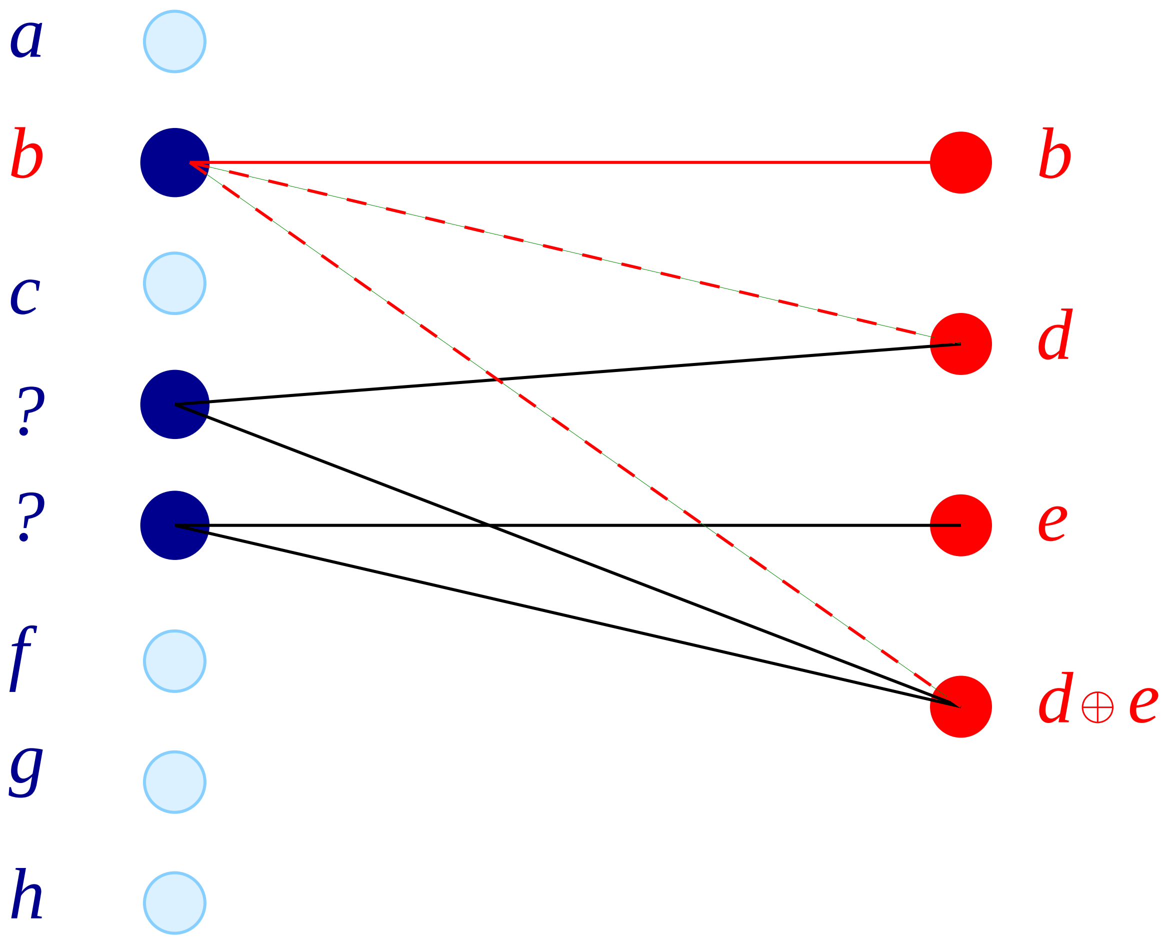
Decoding

Stage 1: Direct Recovery



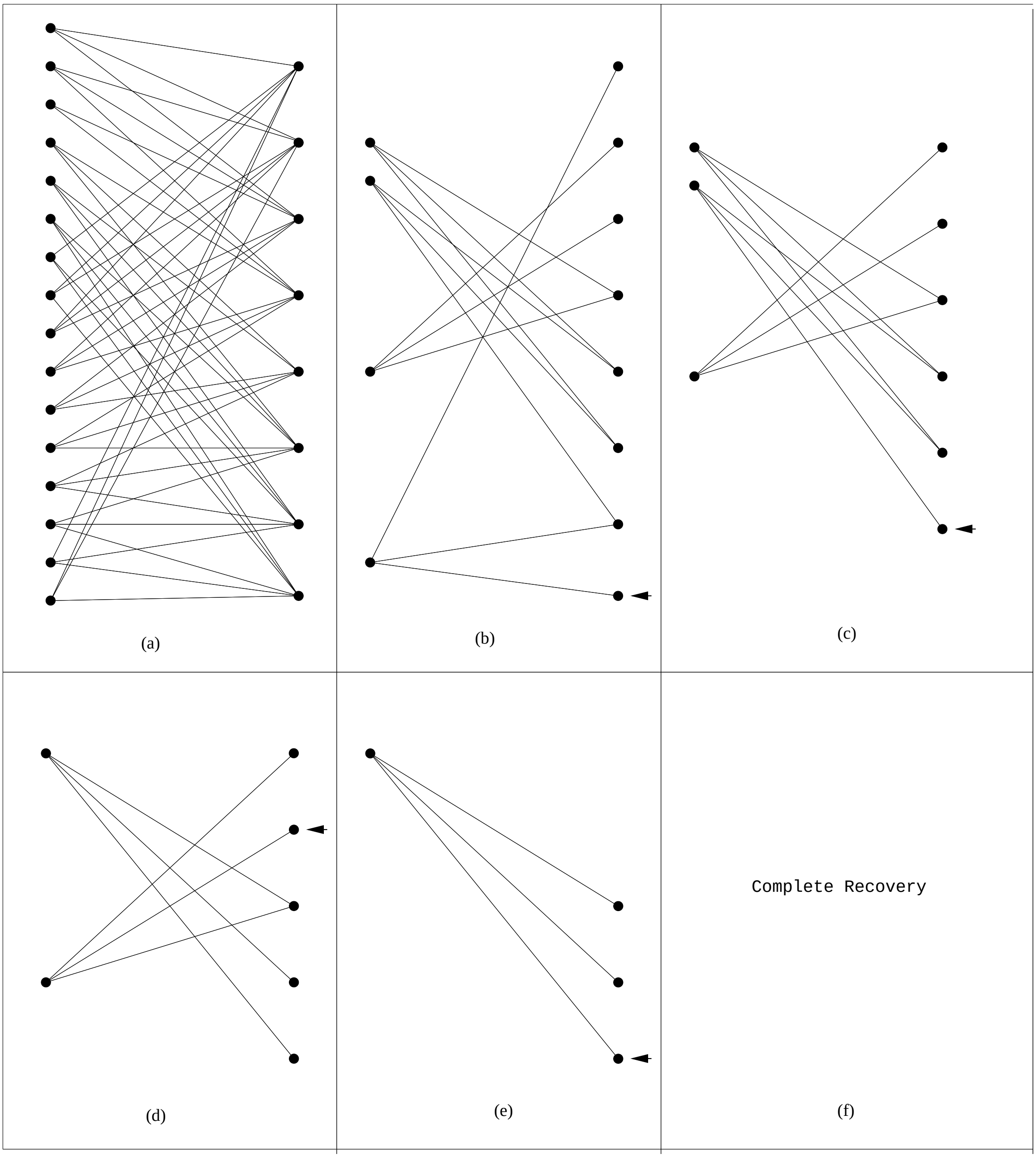
Decoding

Stage 2: Substitution Recovery



Decoding time is proportional to number of edges in graph.

Example



The Problem

We now have a fast encoding and decoding algorithm.

Want to design codes that perform good with respect to these algorithms.

How do we design the graphs?

Experiments

Choose regular graphs.

Experiments show that a $(3,6)$-graph recovers from 42.9% erasures.

A $(4,8)$-graph recovers from 38.3% erasures.

A $(5,10)$-graph recovers from 34.1% erasures.

What are these numbers?

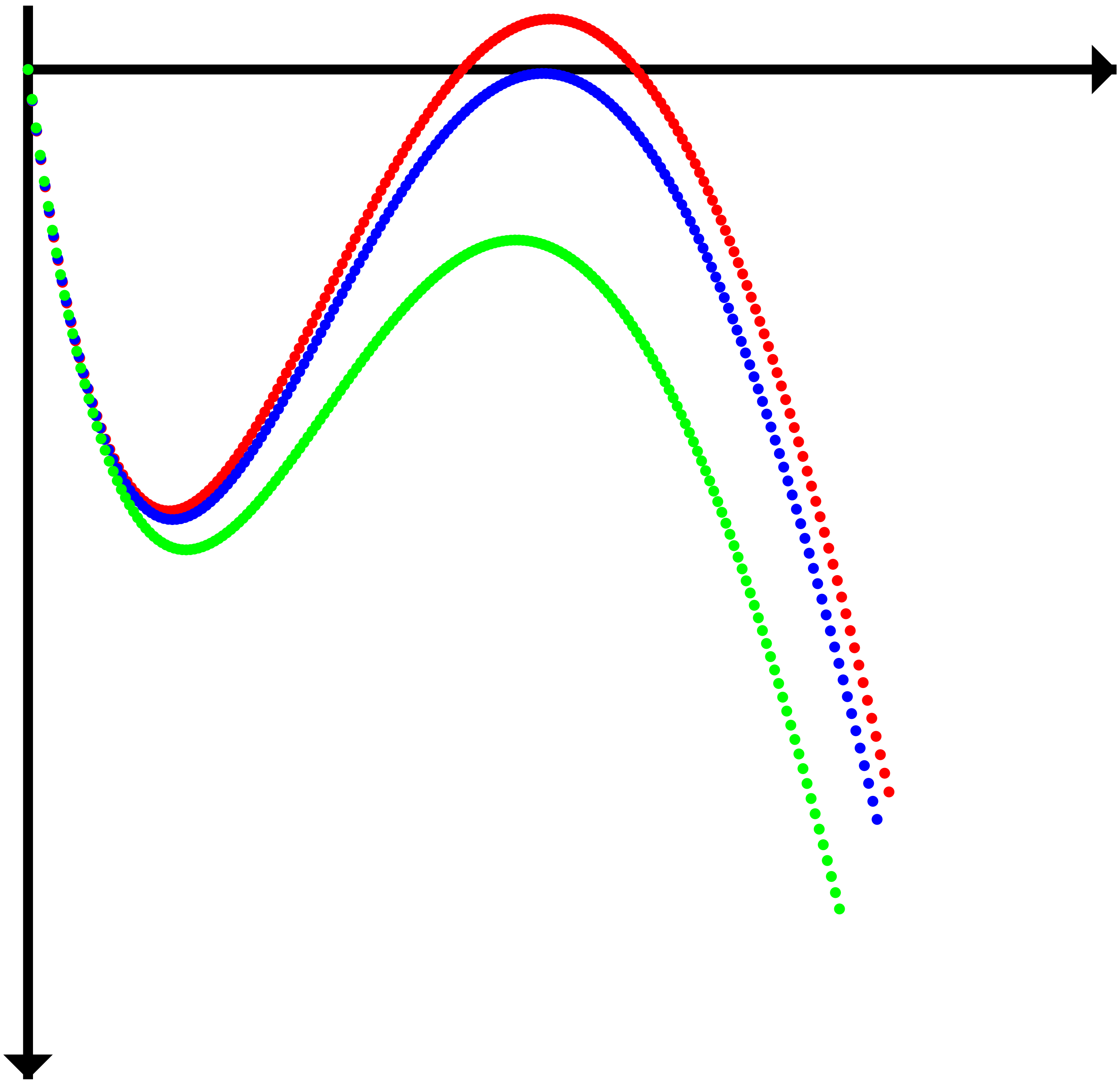
Revelation

Theorem. A random (k, d)-graph recovers from a p-fraction of erasures with high probability iff

$$p(1-(1-x)^{d-1})^{k-1} < x \quad \text{for } x \in (0, p).$$

(Luby, Mitzenmacher, Shokrollahi, Spielman, Stemann)

Proof: uses probability theory (martingales, tail inequalities, large deviation results).



General Case

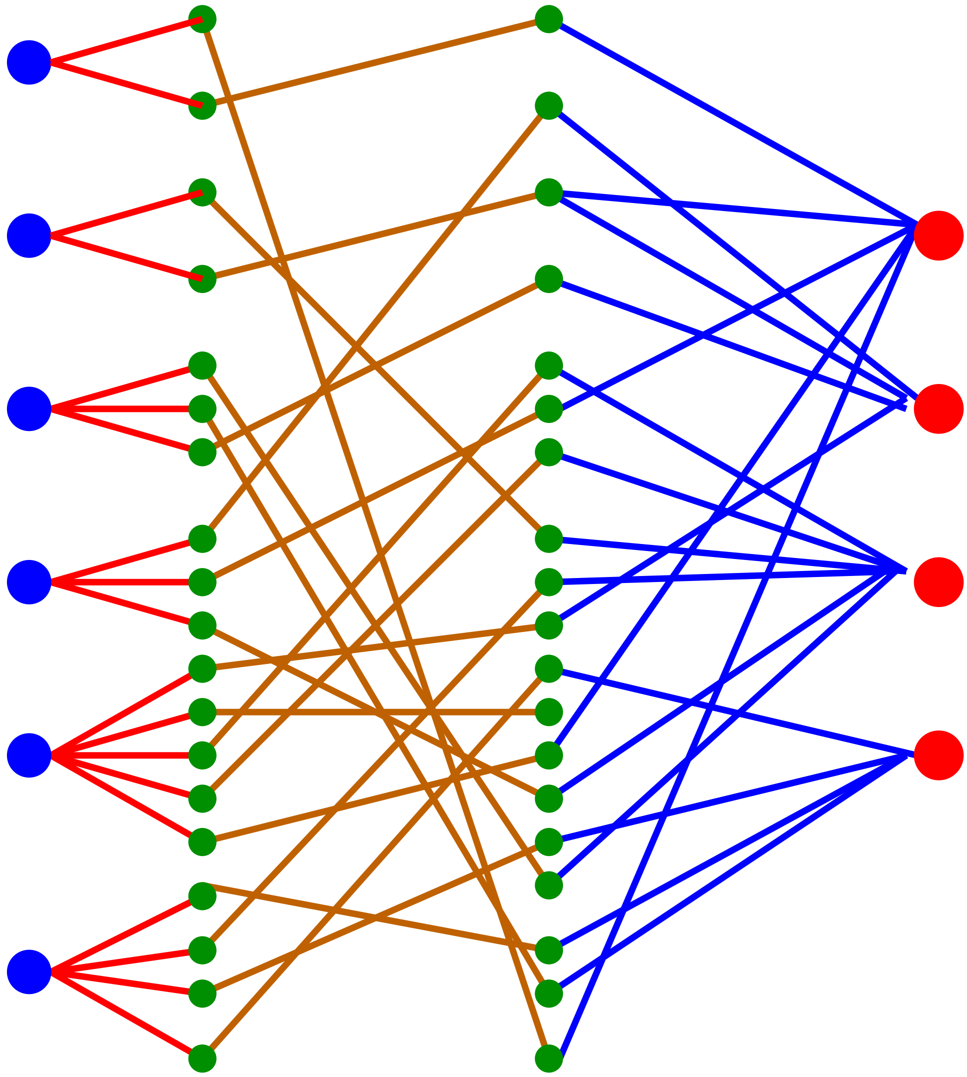
Let λ_i and ρ_i be the fraction of edges of degree i on the left and the right hand side, respectively.

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

Condition for successful decoding for erasure probability is then

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

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



Design of Graphs: Linear Programming

Fix right hand side $\rho(x)$, and find best left hand side $\lambda(x)$ using the condition

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

on $(0,1)$ using linear programming.

Once best left hand side found, fix left hand side and use dual condition

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

on $(0,1)$ with linear programming to find best right hand side.

Iterate!

Asymptotically Optimal Codes

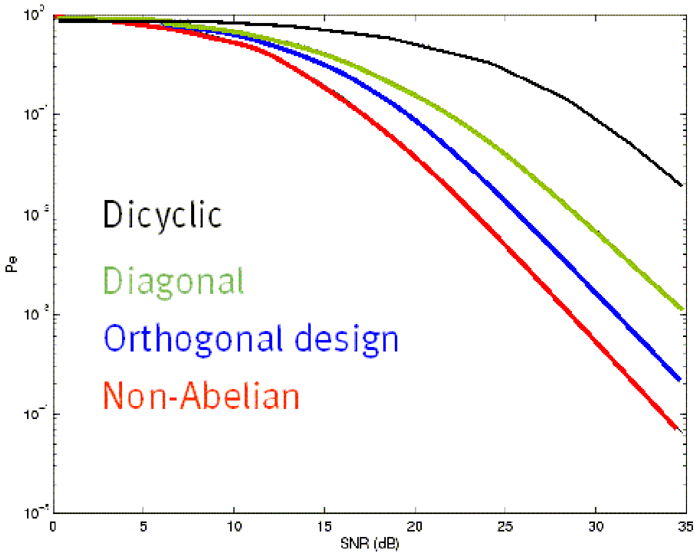
Using highly irregular graphs, one obtains, for any rate R sequences of codes that can get arbitrarily close to the capacity of the erasure channel.

Degree structure? Fix design parameter D.

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

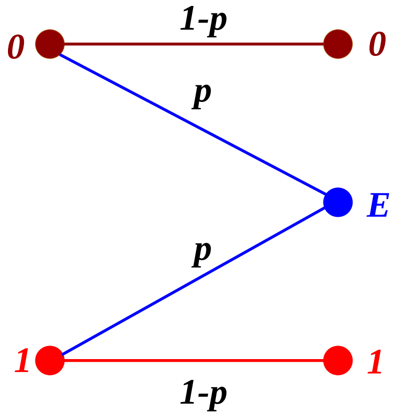
$$\rho(x) := \exp\left(\mu(x-1)\right)$$

where $H(D)$ is harmonic sum $1 + 1/2 + \cdots + 1/D$ and $\mu = H(D)/\left(1 - 1/(D+1)\right)$.


Dicyclic
Diagonal
Orthogonal design
Non-Abelian
Pe
SNR (dB)
0
5
10
15
20
25
30
35

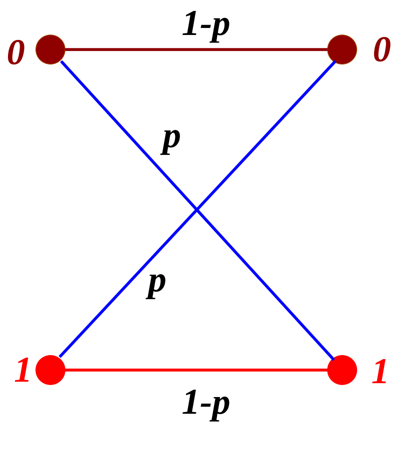
Other Channels

Erasure channel:



Capacity $= 1 - p$.

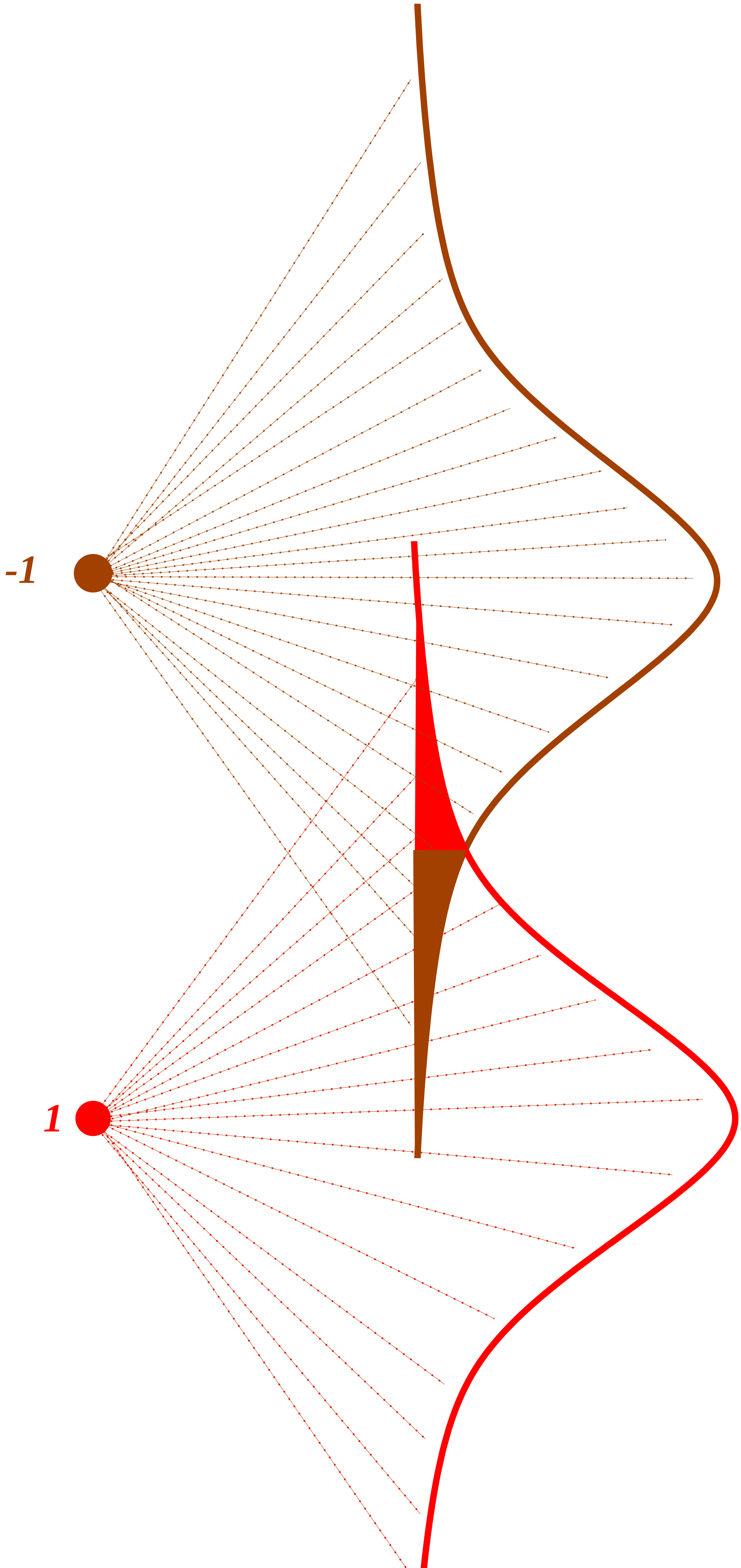
Binary symmetric channel:



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

Other Channels

Additive White Gaussian Noise Channel:

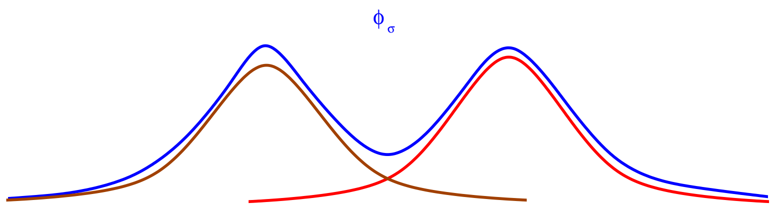

-1
1

Gaussian noise with variance σ^2.

Capacity=

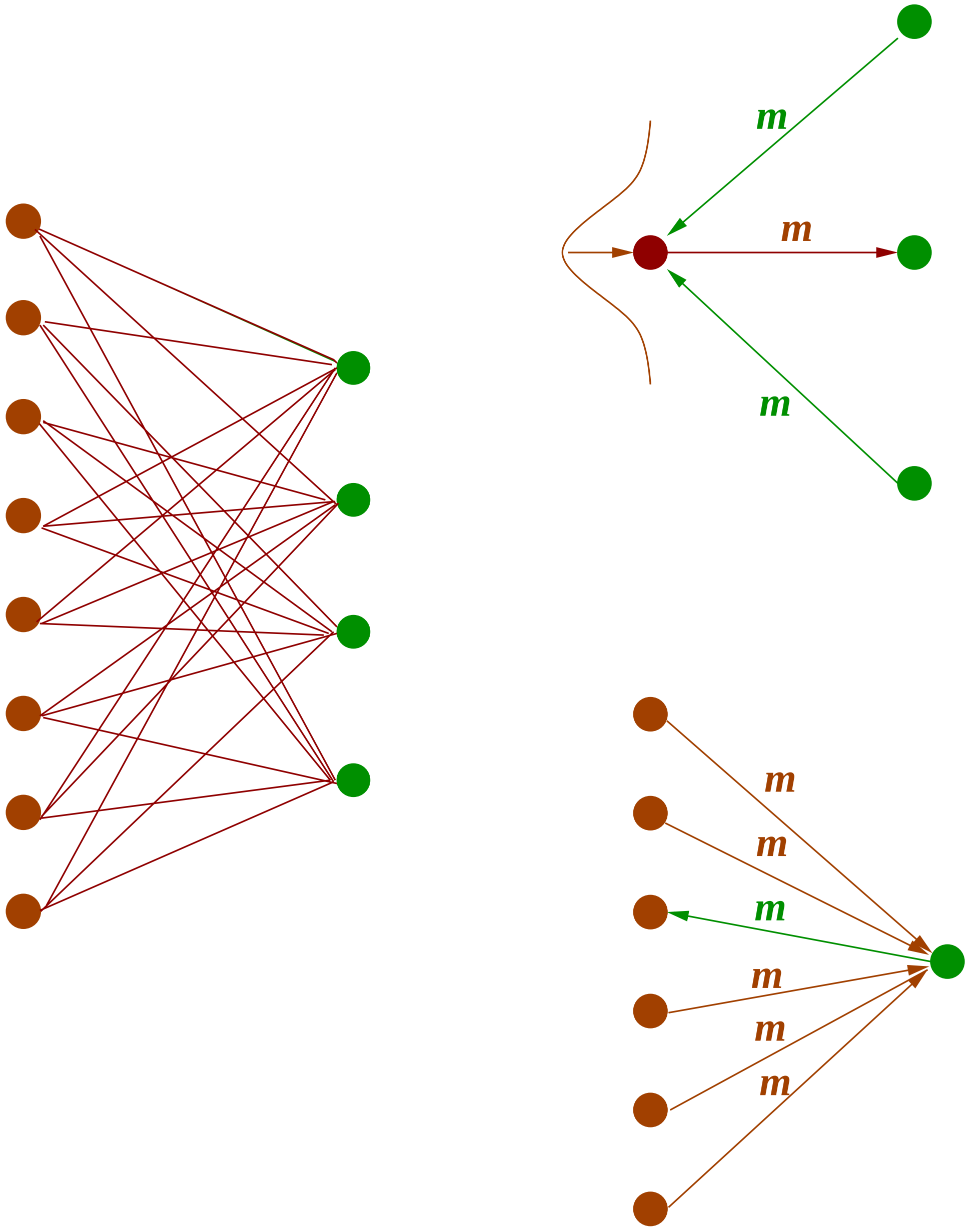
$$-\int_{-\infty}^{\infty} \phi_\sigma(x) \log_2 \phi_\sigma(x) \mathrm{d}x - \frac{1}{2} \log_2(2\pi e \sigma^2),$$

where



Other Channels

Decoder? Belief propagation.



Other Channels

- Certain embodiments of the belief propagation can be rigorously analysed using the same methods as the ones used for erasure channels.

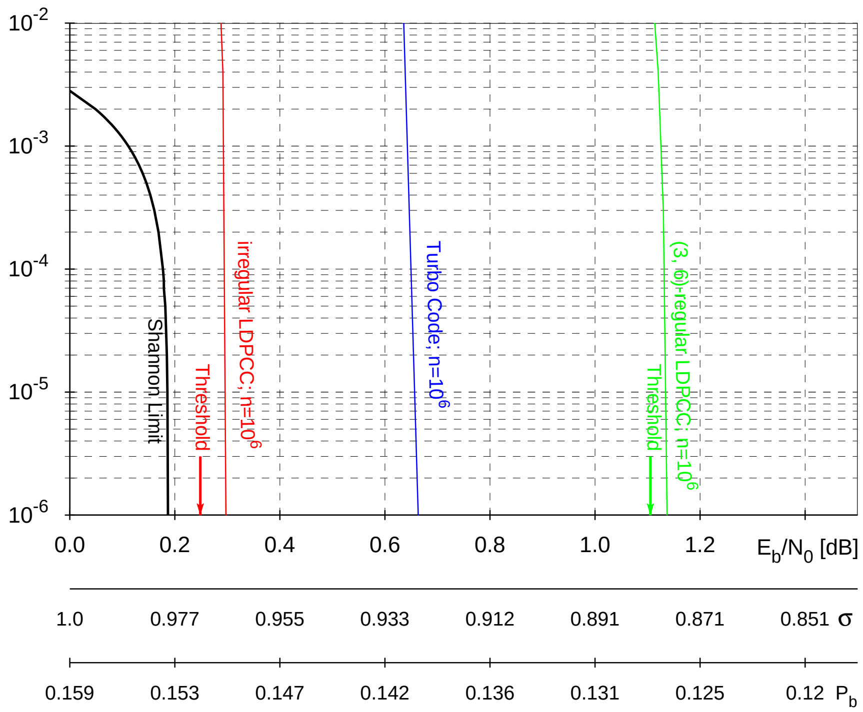
- Experiments show that some codes that are good for the erasure channel are also good for the binary symmetric or the AWGN-channel. But: can we prove it?

(Luby, Mitzenmacher, Shokrollahi, Spielman, 1998).

Analysis

Richardson and Urbanke (both Bell Labs) observed that the analysis of Luby et al. can be generalized to analyse the full belief propagation algorithm (1998).

Based on this analysis, Richardson, Shokrollahi, Urbanke construct low-density codes that are closer to the Shannon capacity than other types of codes, such as Turbo codes.



Race for Capacity

Explicit sequences of low-density codes which approach the Shannon-capacity when decoded with belief propagation!

Conjecture: They exist!!

Known only for the erasure channel (Luby et al.).