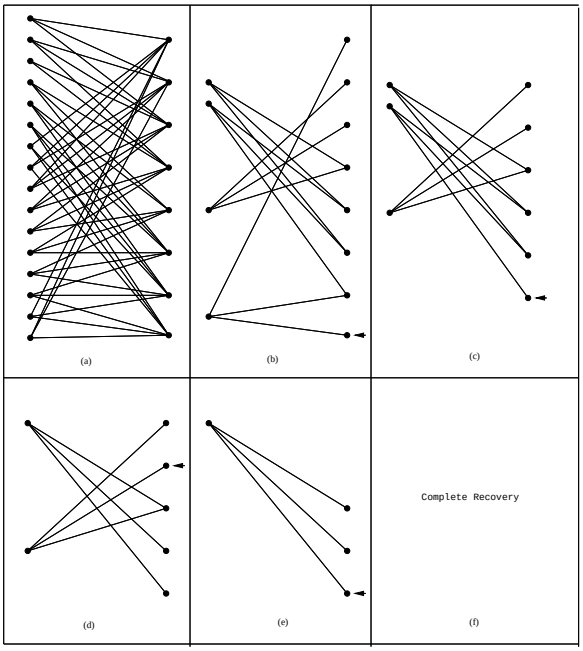
Low-Complexity Capacity-achieving Erasure Codes



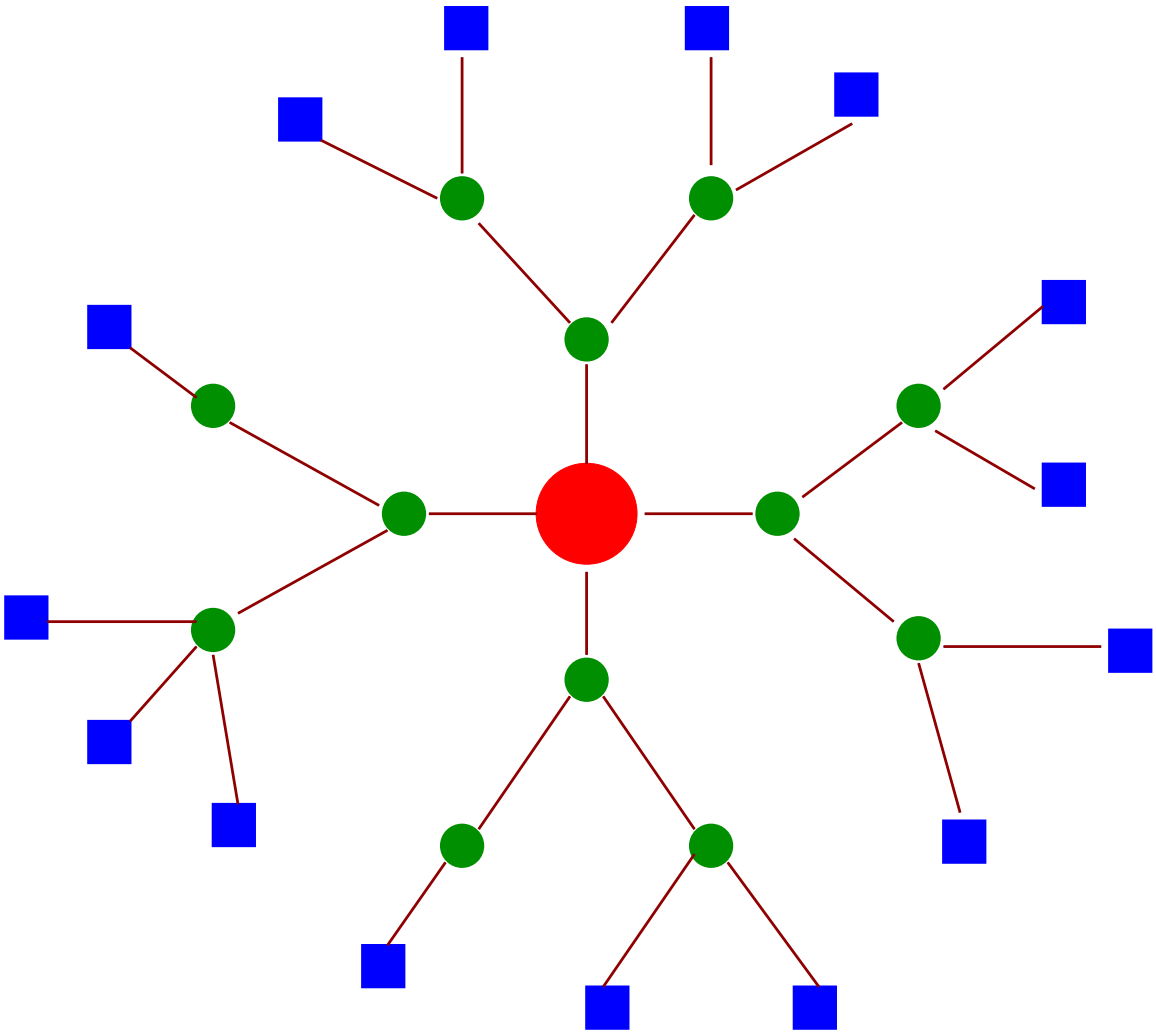
M. Amin Shokrollahi

Bell Laboratories

AAECC 13 - Honolulu

Motivation: 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

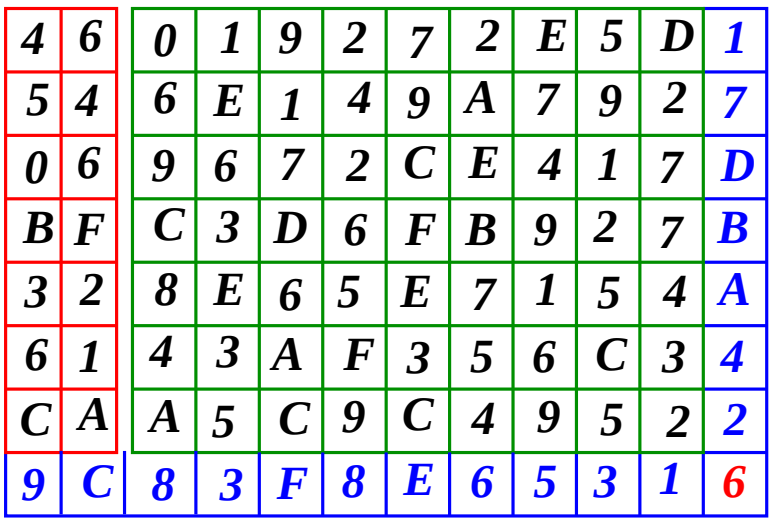
Each packet has an identifier.

Packets can get lost or corrupted.

Corruption is checked via checksums.

Corrupted packets are regarded as lost.

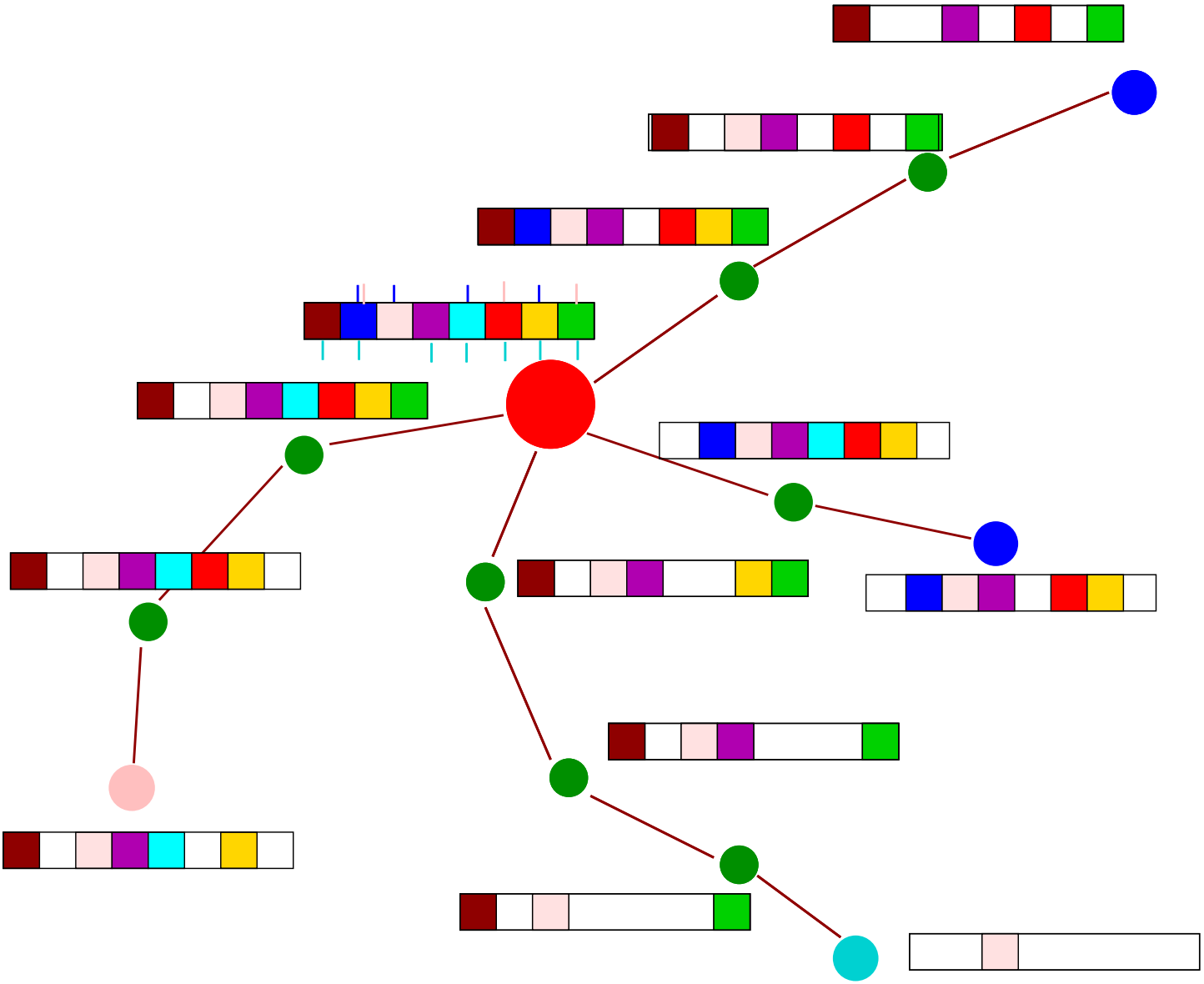
May without loss of generality only concentrate on losses.



Retransmission

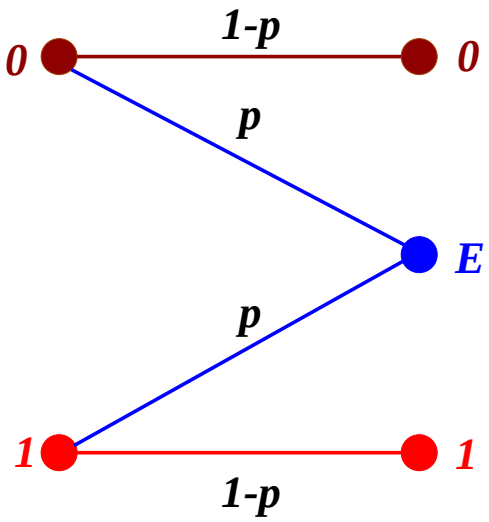
In many communication protocols lost packets are retransmitted.

Process is repeated until all packets are received.



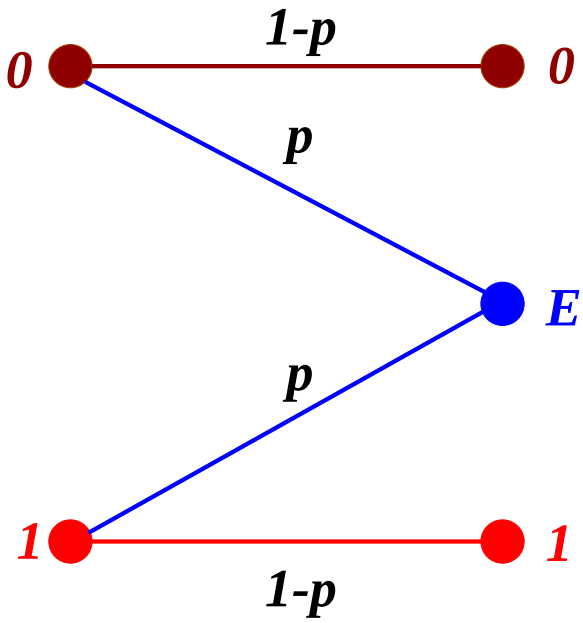
The Problem

Want to design good codes for the (binary) erasure channel with fast encoding and decoding.



What Can We Expect?

Elias'1955:



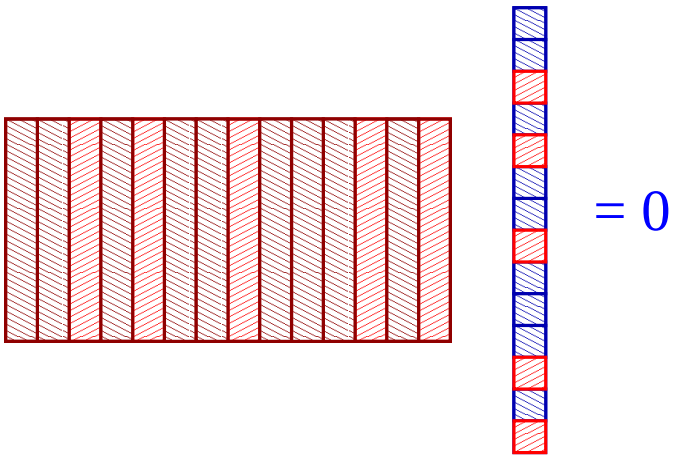
Capacity$=1-p$.

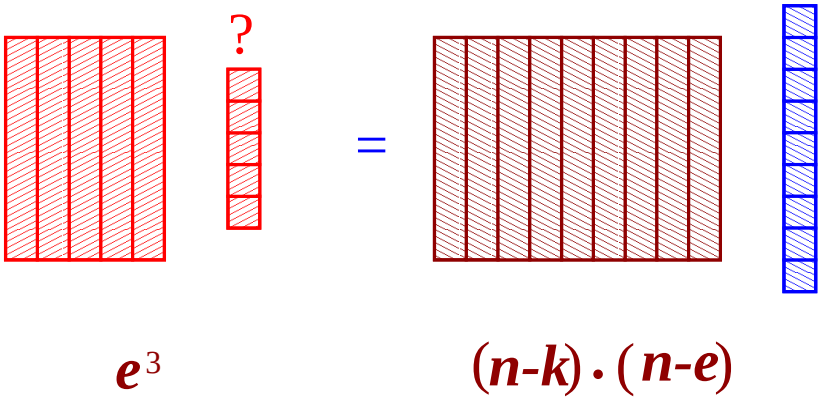
Random linear codes approach capacity, as block-length increases.

Linear Codes

Linear $[n, k, d]$-code is capable of correcting up to $d - 1$ erasures in arbitrary positions.

However: Solving systems of linear equations is too slow.



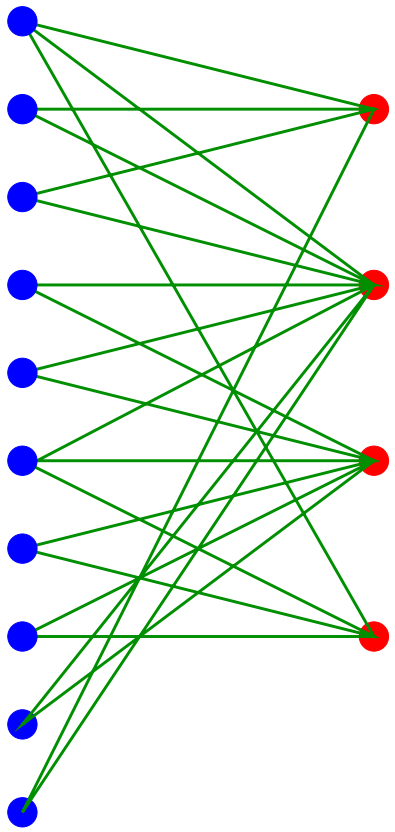


Low Density Parity Check Codes

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

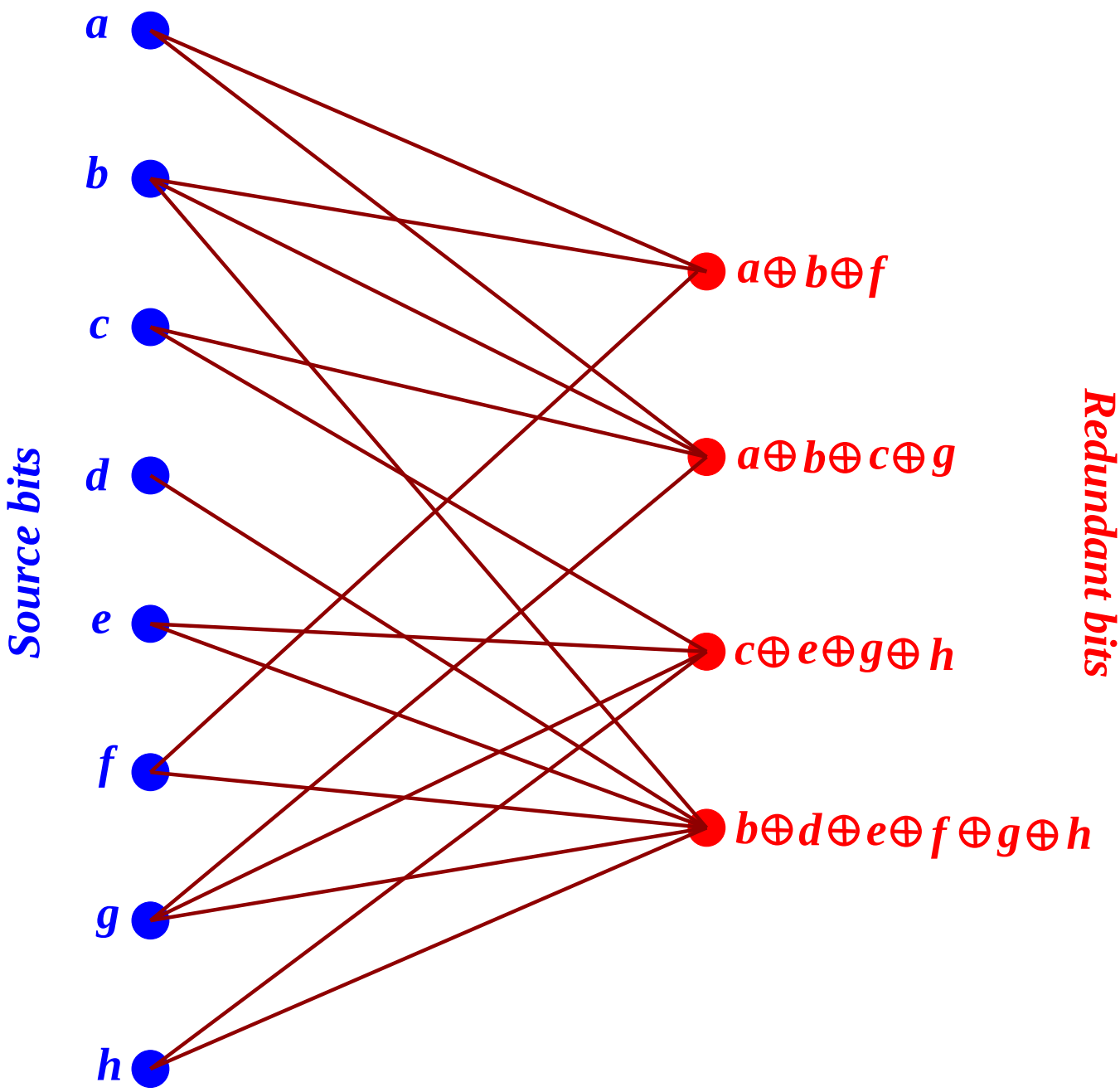
Codes are built from sparse bipartite graphs.

Encoding and Decoding are simple.



Encoding with Bipartite Graphs

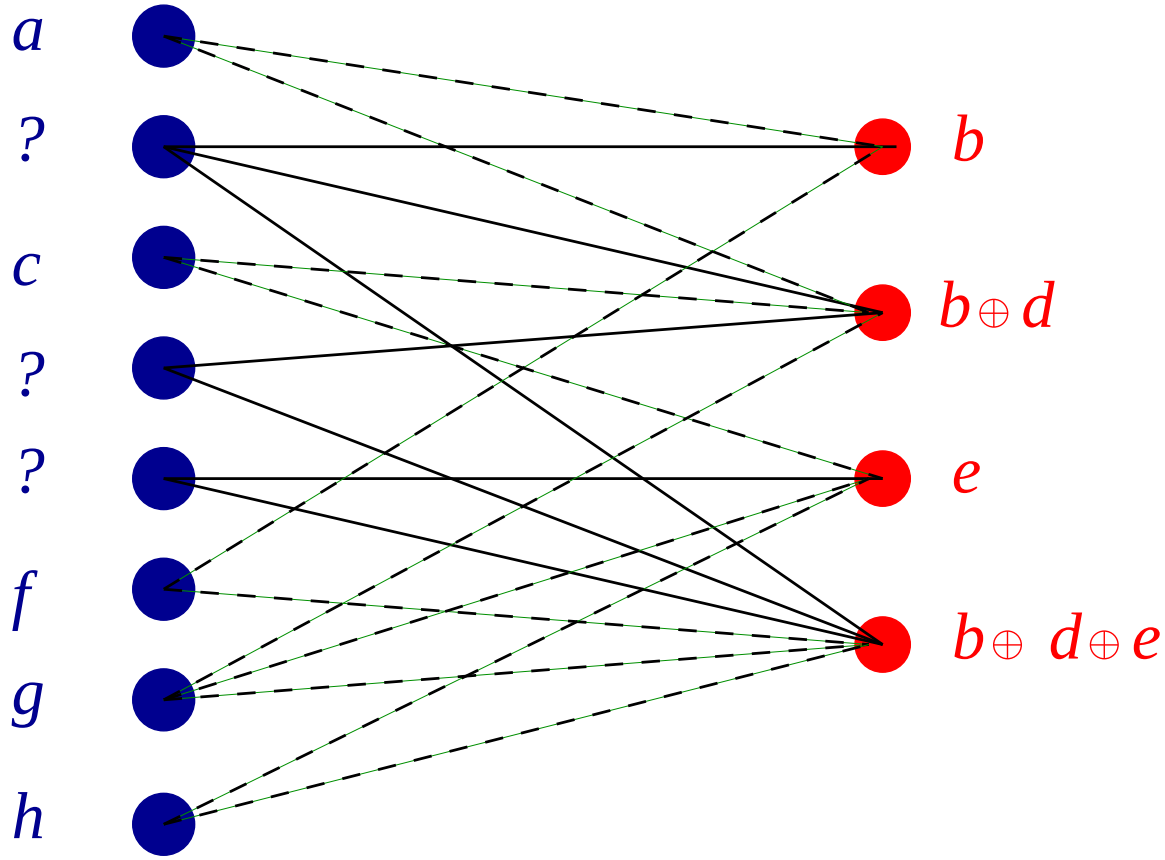
Take a bipartite graph between k nodes on the left and $n - k$ nodes on the right. Label left nodes with the k packets to be encoded. Label right nodes with the redundant packets. Compute value of each right node as XOR of values of adjacent left nodes.



Encoding time is proportional to number of edges in graph.

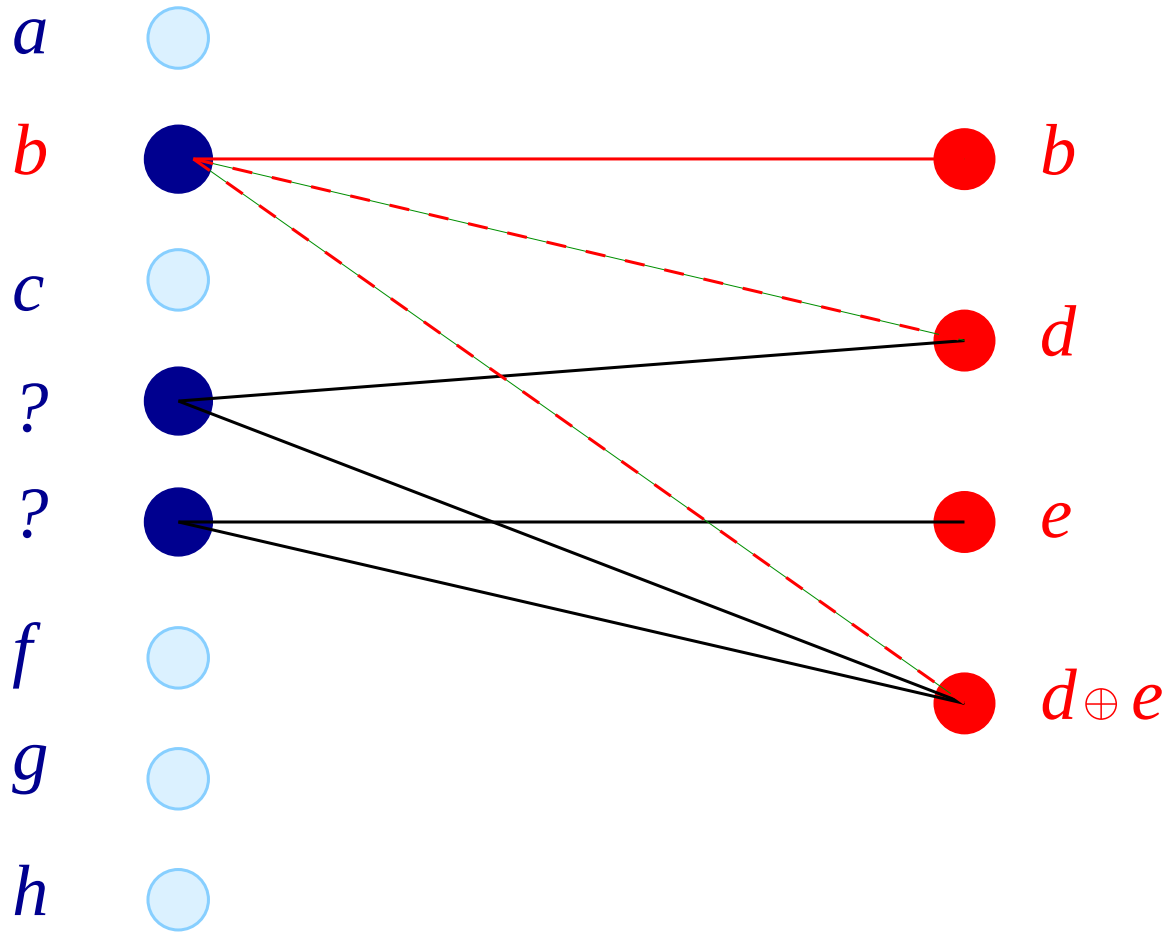
Decoding

Stage 1: Direct Recovery



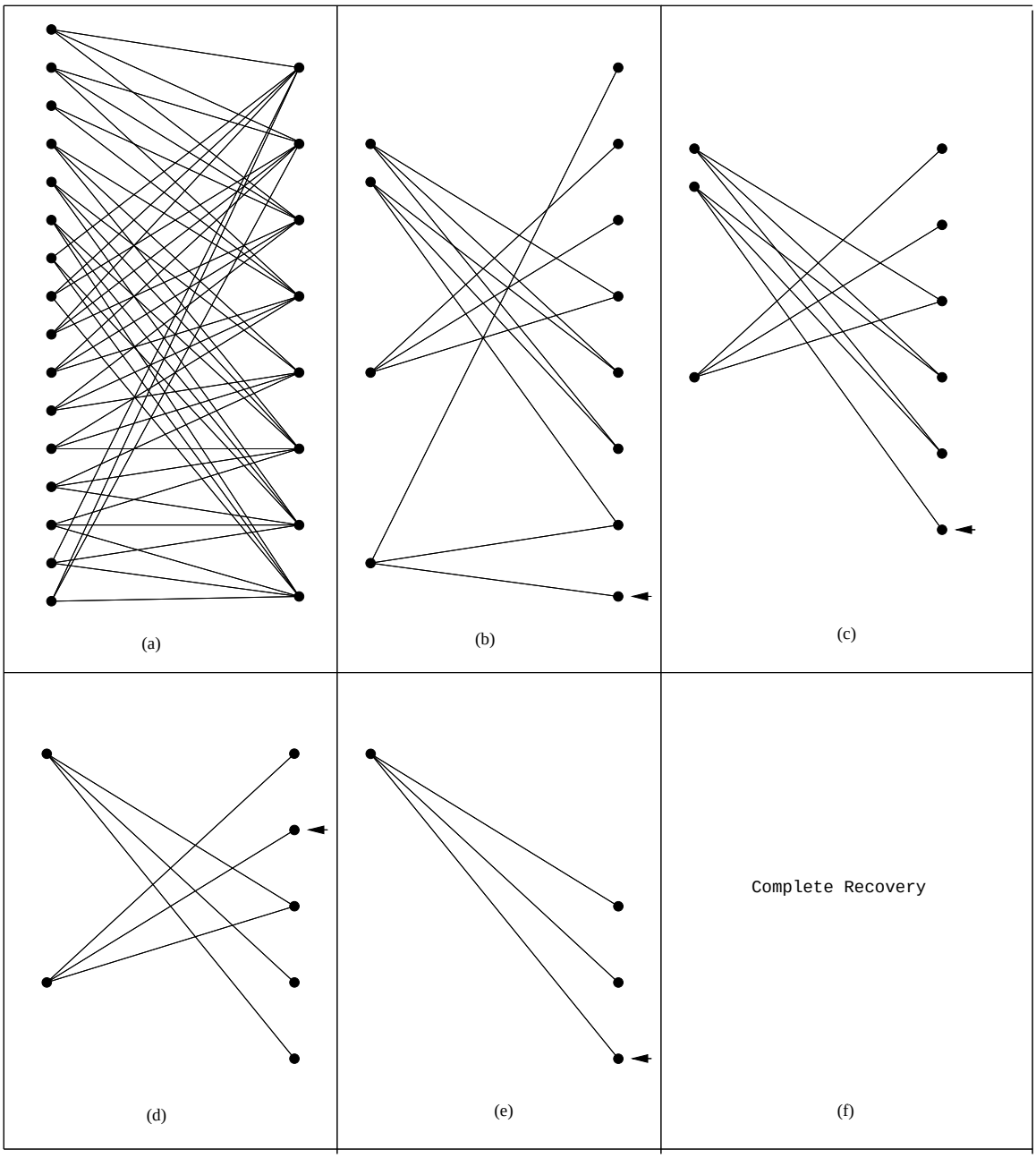
Decoding

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?

A Solution

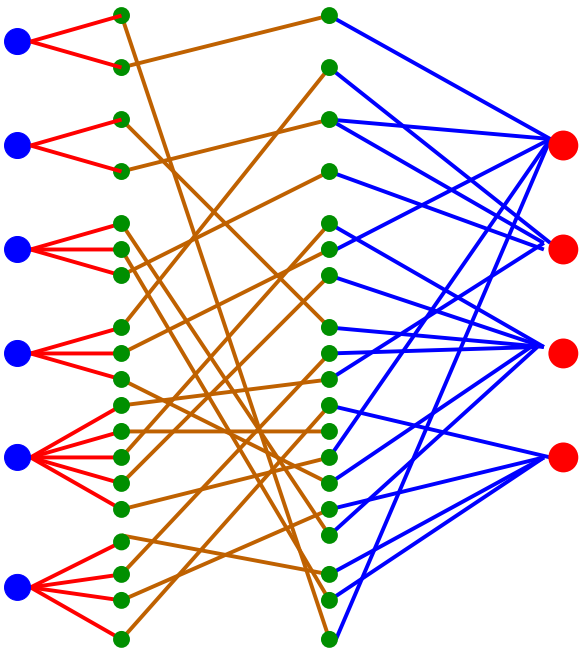
Let λ_i and ρ_i be the fraction of edges of degree i on the left and the right hand side, respectively.

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 p_0 is then

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

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



Capacity-Achieving Sequences

Sequence $(\lambda_m(x), \rho_m(x))$ is called capacity-achieving of rate R if

- Gives rise to codes of rate R,
- For any $\varepsilon > 0$ we have

  $$(1-R)(1-\varepsilon)\lambda_m(1-\rho_m(1-x)) < x$$

  on the interval $(0, (1-R)(1-\varepsilon)]$ for all sufficiently large m.

Can we Achieve Capacity?

Yes!

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 $1 + 1/2 + \cdots + 1/D$,

$$\mu = \frac{H(D)}{1-R}\left(1 - 1/(D+1)\right),$$

and R is the rate of the code.

Tornado codes

Luby-Mitzenmacher-Shokrollahi-Spielman-Stemann.

Trade-off

- Proximity to channel-capacity:

$$(1 - R)(1 - \varepsilon)\lambda(1 - \rho(1 - x)) < x$$

  for $x \in (0, (1 - R)(1 - \varepsilon)]$.

- Running time of the algorithm: proportional to the average right degree a_r.

Want: ε small and at the same time a_r small.

Tornado codes:

$$a_r \leq \mu \log\left(\frac{1}{\varepsilon}\right),$$

for some μ depending on the rate R.

Can we do better?

Trade-off

No!

This paper:

$$a_r \geq \frac{1}{1-R} \log\left(\frac{1}{\varepsilon}\right).$$

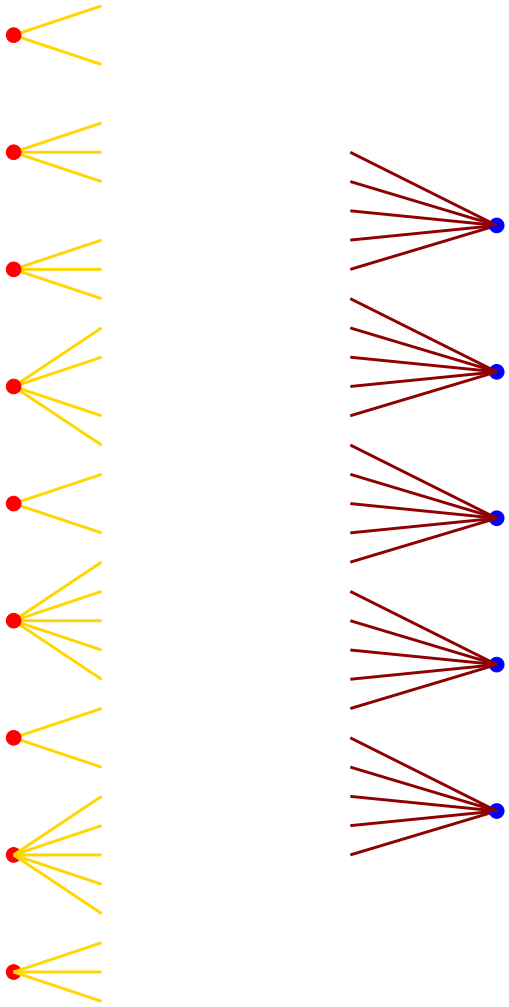
Tornado codes have essentially optimal trade-off!

Other Sequences?

Proof of trade-off theorem reveals: best trade-off if

Only one degree on the right-hand side.

Such graphs are called right-regular.



Fractional Binomial Coefficients

We have $\rho(x) = x^d$ for some d.

What is $\lambda(x)$?

$$\begin{aligned} p_0\lambda(1-\rho(x)) &< x \\ p_0\lambda(1-(1-x)^d) &< x \\ p_0\lambda(y) &< 1-\sqrt[d]{1-y} \end{aligned}$$

$$1-(1-x)^{1/d} = \sum_{k=1}^{\infty} \underbrace{\binom{1/d}{k}(-1)^{k+1}}_{>0} x^k.$$

$$\binom{\alpha}{k} = \frac{\alpha(\alpha-1)\cdots(\alpha-k+1)}{k!}.$$

The Ansatz

$$\Lambda_{N,d}(x) := \sum_{k=1}^{N} \binom{1/d}{k} (-1)^{k+1} x^k,$$

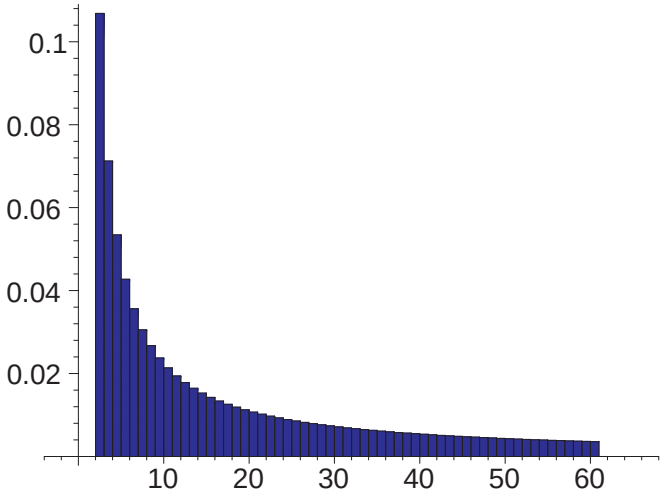
$$\lambda_{N,d}(x) = \frac{\Lambda_{N,d}(x)}{\Lambda_{N,d}(1)}$$

$$\Lambda_{N,d}(1) = 1 - Nd \binom{1/d}{N} (-1)^{N+1}.$$

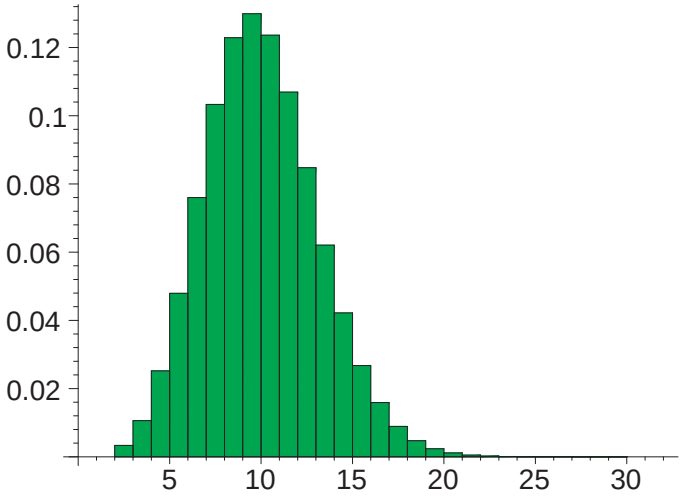
Proximity to Channel Capacity

$$a_r \leq c \log\left(\frac{1}{\varepsilon}\right)$$

Degree Profile: Tornado Codes

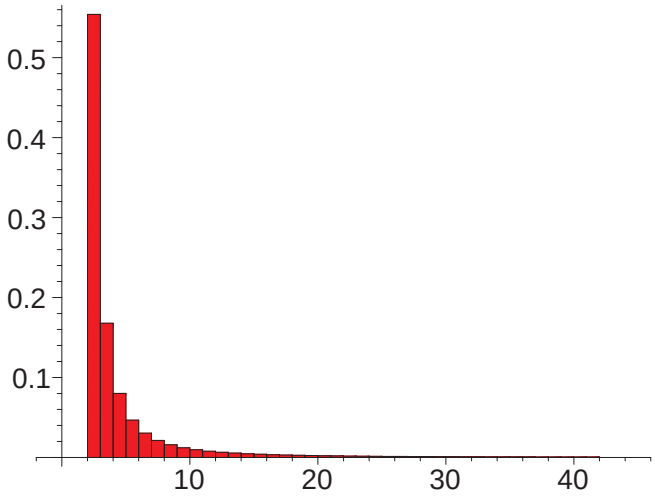


Heavy tail

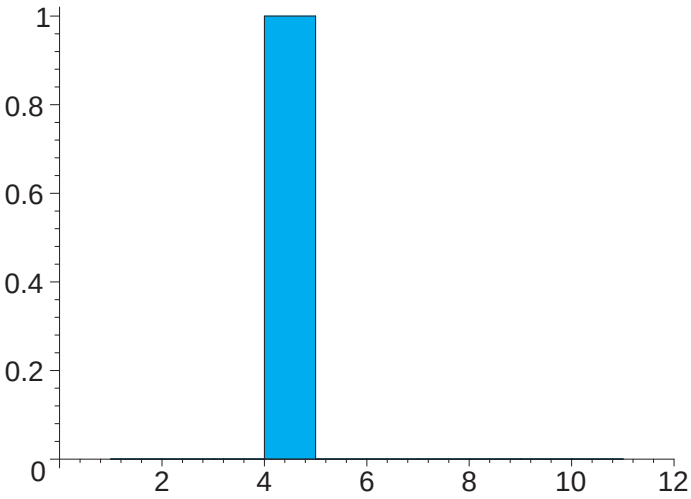


Poisson

Degree Profile: Right-Regular



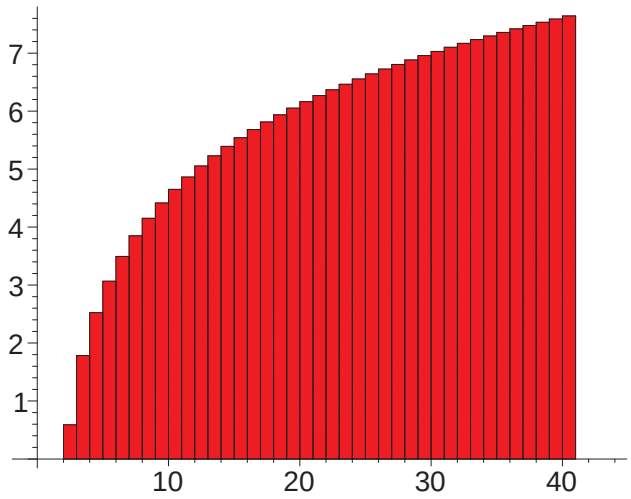
Right-regular



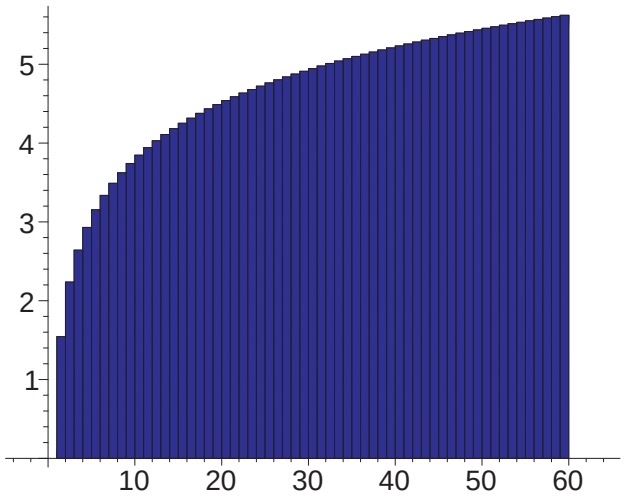
Constant

Log-Representation

–Logarithm of fraction of nodes of each degree on the left:



Right-regular



Tornado

Right-Regular vs. Tornado

Pro's:

- Less deviation from predicted performance since neighborhood of right node is exact.
- Better trade-off performance (see paper).
- Loop-unrolling.

Con's:

- Do not have a continuous range of average degrees.

Common Points

- δ largest fraction of tolerable erasures for given $\lambda(x)$ and $\rho(x)$.

$$\delta \leq \frac{\rho'(1)}{\lambda'(0)}.$$

Stability Condition.

Both degree sequences satisfy the condition with equality.

- For both sequences the function

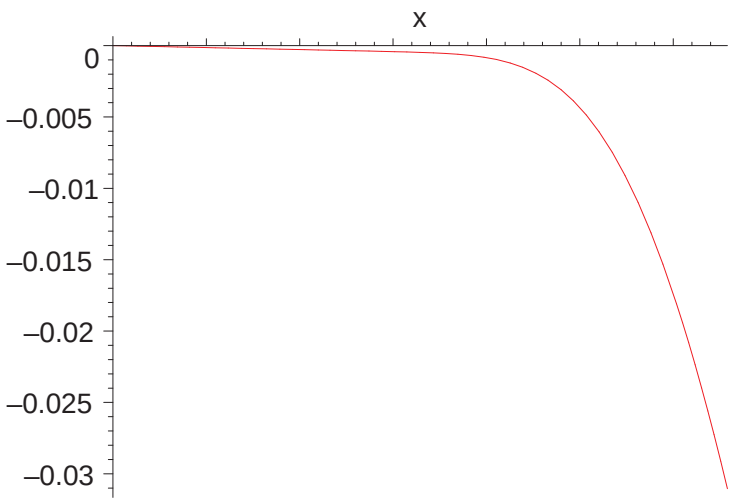
$$\delta\lambda(1-\rho(1-x))-x$$

converges to zero uniformly on the interval $[0,1]$, as $\lambda(x)$ and $\rho(x)$ run over the sequence.

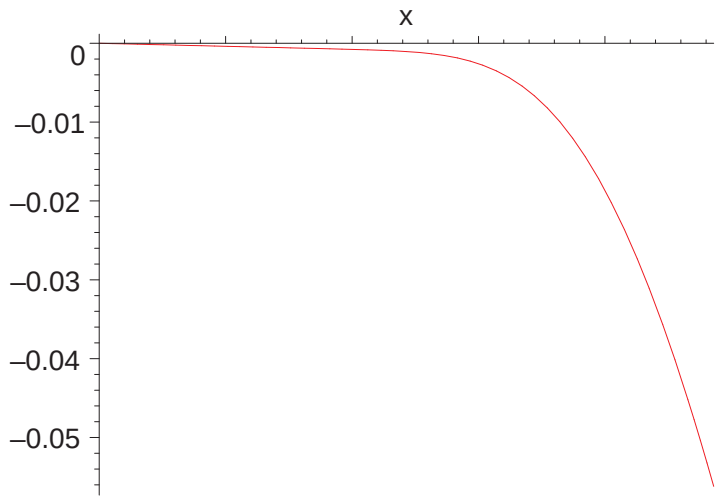
This condition is necessary for capacity-achieving sequences. (Shokrollahi'1999.)

Plots

Plot of $\delta\lambda(1-\rho(1-x))-x$ for $x \in (0,\delta]$.



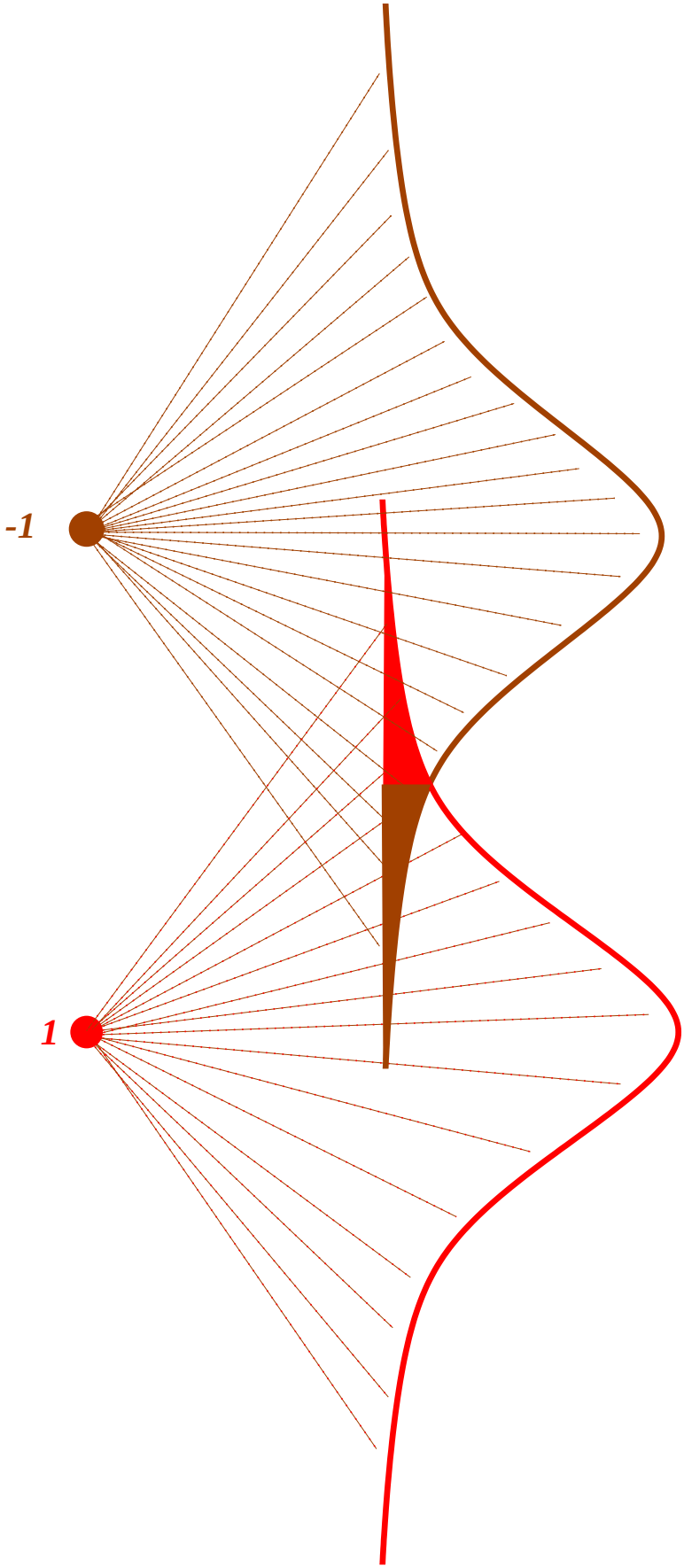
Right-regular

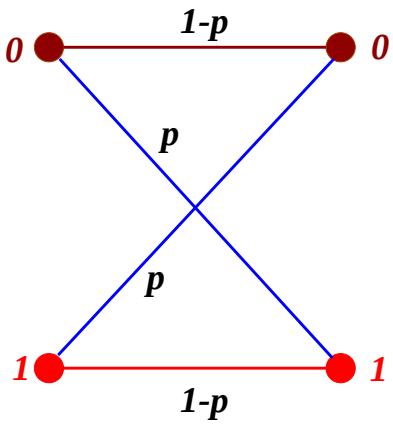


Tornado

Questions

- Other sequences of capacity-achieving codes?
- Capacity-achieving sequences without nodes of degree two?
- Capacity-achieving sequences for other channels? (Richardson-Shokrollahi-Urbanke)


-1
1


0
1-p
0
p
p
1
1
1-p