CIE6020/MAT3350
Selected Topics in Information Theory

Lecture 18: Fountain Codes

---

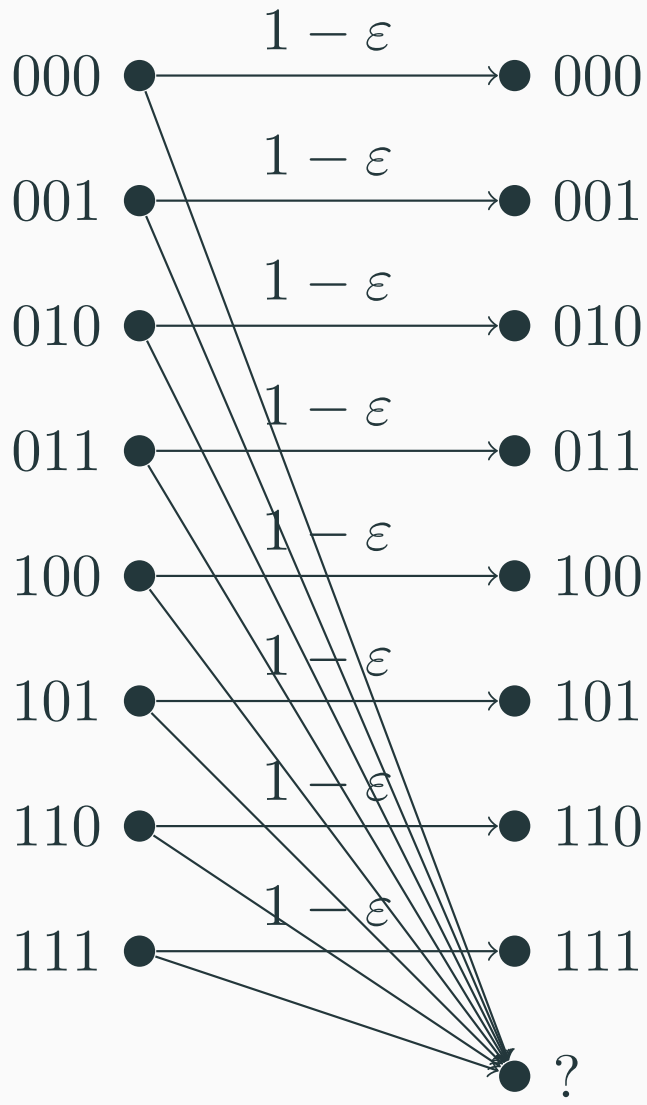
12 April 2019

The Chinese University of Hong Kong, Shenzhen

Coding for Erasure Channels

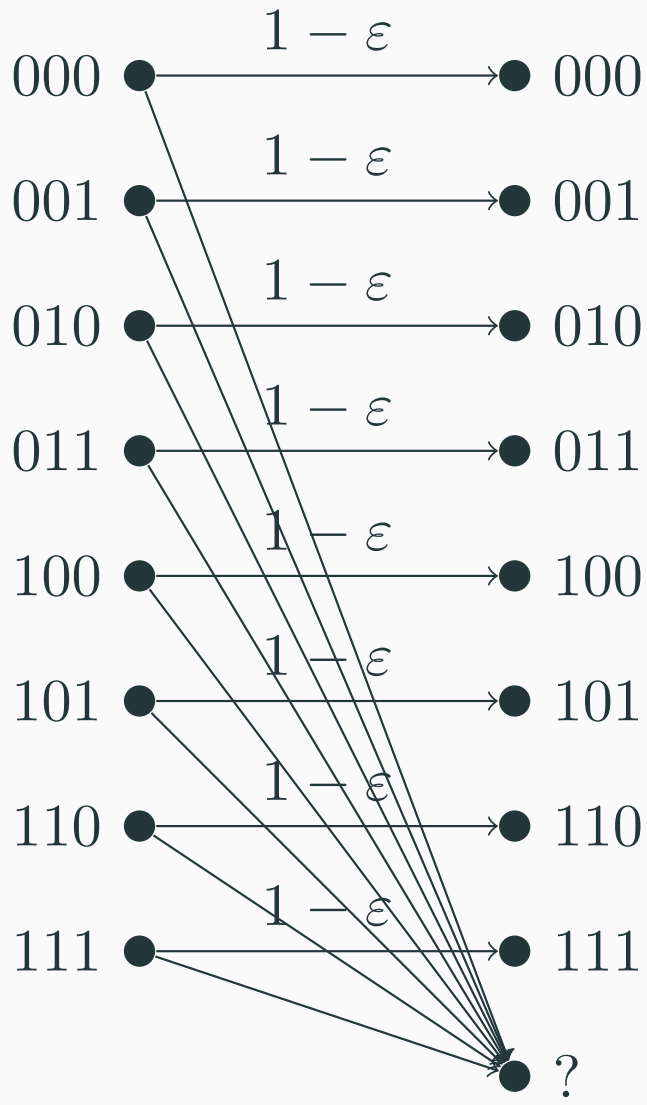
Erasure channel

- Model packet loss in networks (e.g. Internet, wireless networks)
- Capacity: $1 - \varepsilon$ symbol per use
- Solutions:



Erasure channel

- Model packet loss in networks (e.g. Internet, wireless networks)
- Capacity: $1 - \varepsilon$ symbol per use
- Solutions:
  - Retransmission
  - Forward error correction



Retransmission

- Example: TCP, 802.11 MAC, cellular networks
- Achieve capacity
- Require feedbacks

Retransmission

- Example: TCP, 802.11 MAC, cellular networks
- Achieve capacity
- Require feedbacks
- Not good for many scenarios
  1. wireless transmissions
  2. deep-space (satellite), underwater communications
  3. multicast transmissions

Forward error correction

- Capacity achieving without feedbacks
- Reed-Solomon code $(n, k, n-k+1)$
  - Encoding and decoding complexity: $O((n-k)\log n)$ per symbol.
- Can we have better solutions?
  - $O(1)$ complexity (per symbol).
  - Adaptive for different erasure rates/patterns.

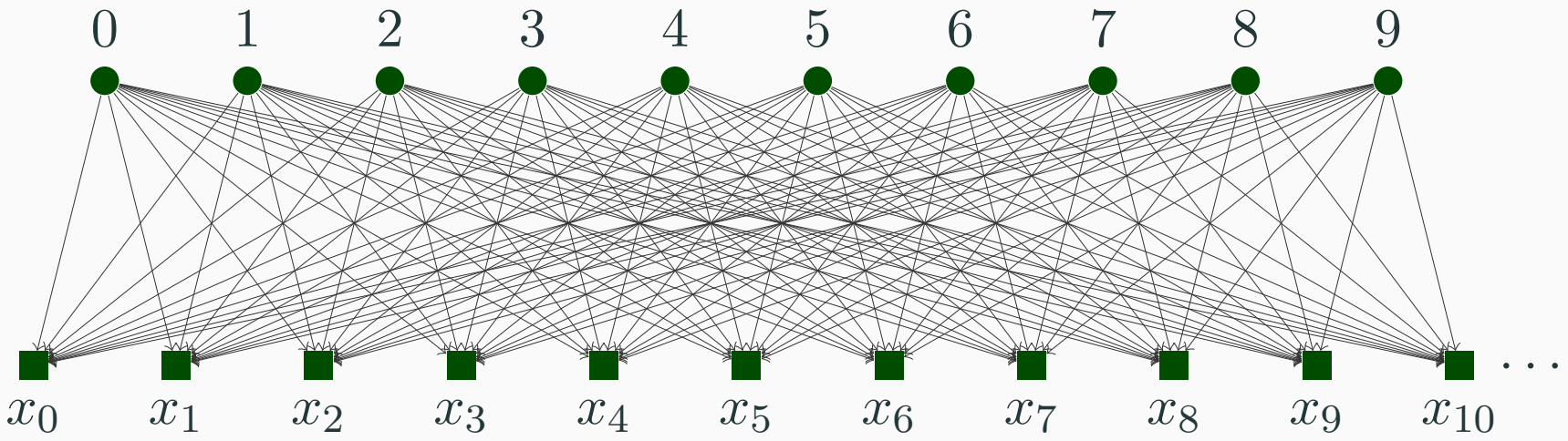
What are fountain codes?

- Transmit a file of k packets: $\mathbf{B} = [b_1, b_2, \ldots, b_k]$ where $b_i \in \mathbb{F}_q^T$
- Encoder generates potentially an infinite number of coded packets
- The file can be recovered from any set of n coded packets, where n is slightly larger than k.
- Also known as *rateless codes*

Rateless Random Linear Codes

Rateless random linear codes

- Encoding: $x_j = \sum_{i=1}^{k} \alpha_{j,i} b_i$.
- Decode from any k coded packets with linearly independent coding vectors.
- Work for any erasure patterns – universal.



Error Probability as a block code of length n

- When n packets are transmitted, the number of received packets N is $B(n, 1-\epsilon)$.
- The received packets $\mathbf{Y} = [y_1, y_2, \ldots y_N]$ is given by $\mathbf{Y} = \mathbf{BA}$.
- The decoding is correct iff $\mathrm{rank}(\mathbf{A}) = k$.
- Error probability

$$\begin{aligned} P_e &= 1 - \Pr\{\mathrm{rank}(\mathbf{A}) = k\} \\ &= 1 - \sum_{j=k}^{n} \binom{n}{j} \epsilon^{n-j} (1-\epsilon)^j \Pr\{\mathrm{rank}(\mathbf{A}) = k | N = j\} \\ &= 1 - \sum_{j=k}^{n} \binom{n}{j} \epsilon^{n-j} (1-\epsilon)^j \zeta_k^j, \end{aligned}$$

where ζ_k^j is the probability that a $k \times j$ totally random matrix has rank k.

Coding overhead (as a rateless code)

- Coding overhead is the number of packets received minus k when decoding.
- Expected coding overhead:

$$\mathrm{CO} = \sum_{i=1}^{\infty} i \Pr\{E_i \cap E'_{i-1}\} = \sum_{i=1}^{\infty} i \zeta_{k-1}^{k,k+i-1}(1 - q^{-1}),$$

  where

  - E_i: the first $k + i$ received packets have rank k.
  - $\zeta_r^{m,n}$: the probability that an $m \times n$ tatally random matrix has rank r.

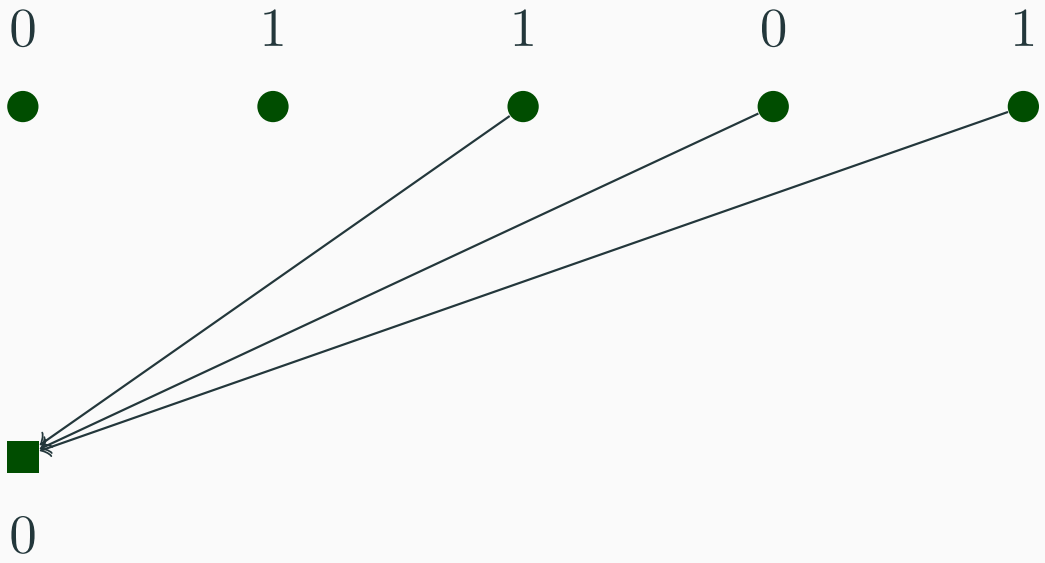
LT codes [Luby 98]

LT codes

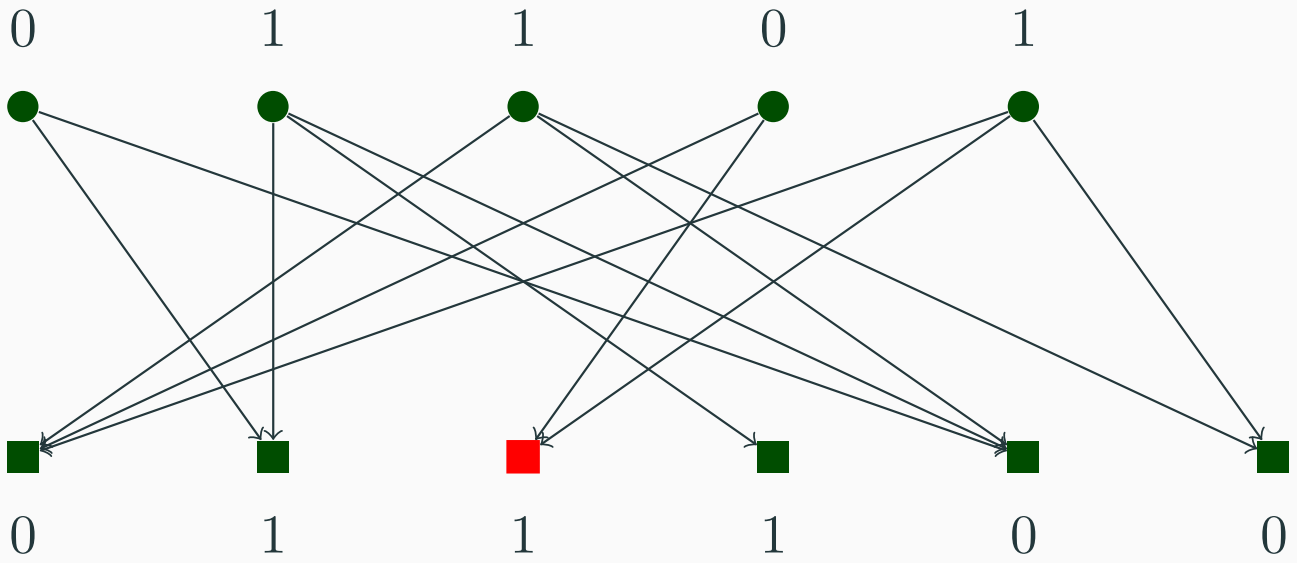
- Sparse encoding
  1. pick a degree d by sampling a degree distribution $\Psi = (\Psi_1, \Psi_2, \ldots, \Psi_K)$.
  2. uniformly at random pick d input packets.
  3. generate a coded packet by linearly combinate of the d input packets.
  4. repeat 1 - 3.
- Belief propogation decoding
  1. find a coded packet with degree one, which recovers the corresponding input packet.
  2. substitute the recovered input packet into the other coded packets that it involves.
  3. repeat 1 - 2 until there is no coded packets with degree one.
- Encoding/decoding complexity: $O(\log K)$ per packet, determined by the average degree $\mathbb{E}[\Psi]$.

Tanner graph of LT codes

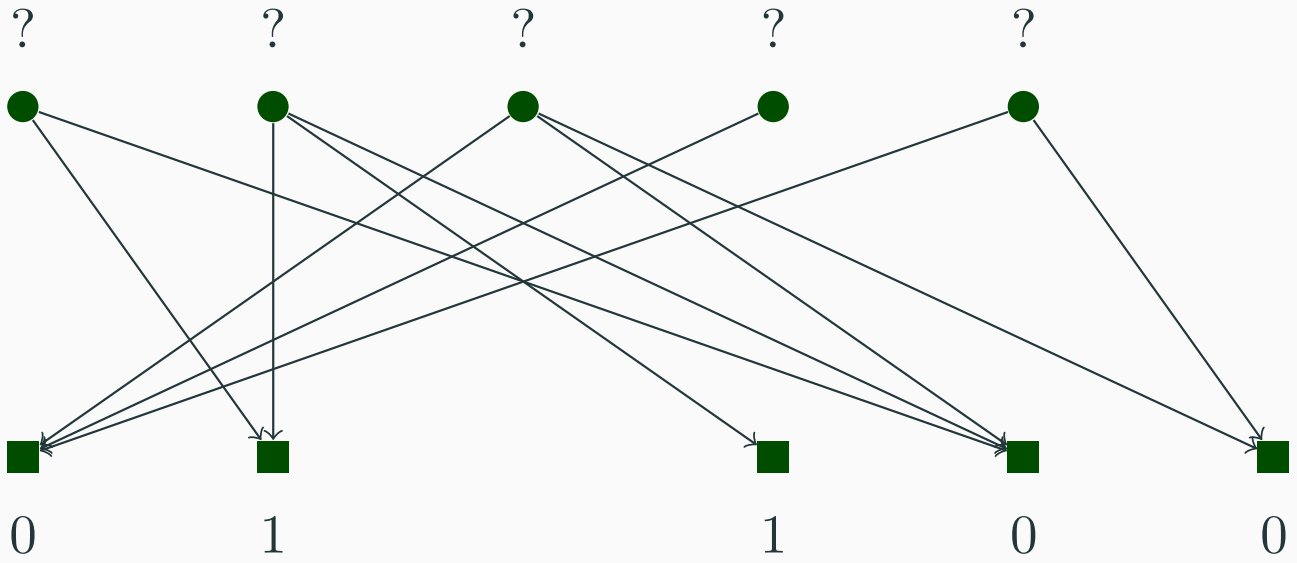
Tanner graph of LT codes



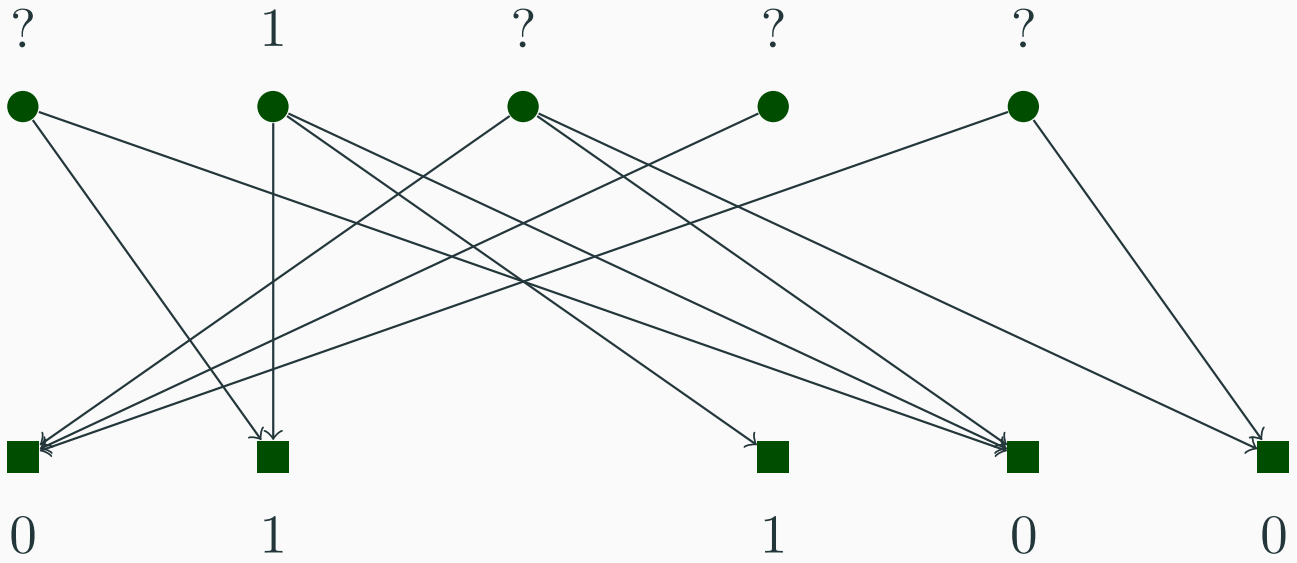
Tanner graph of LT codes



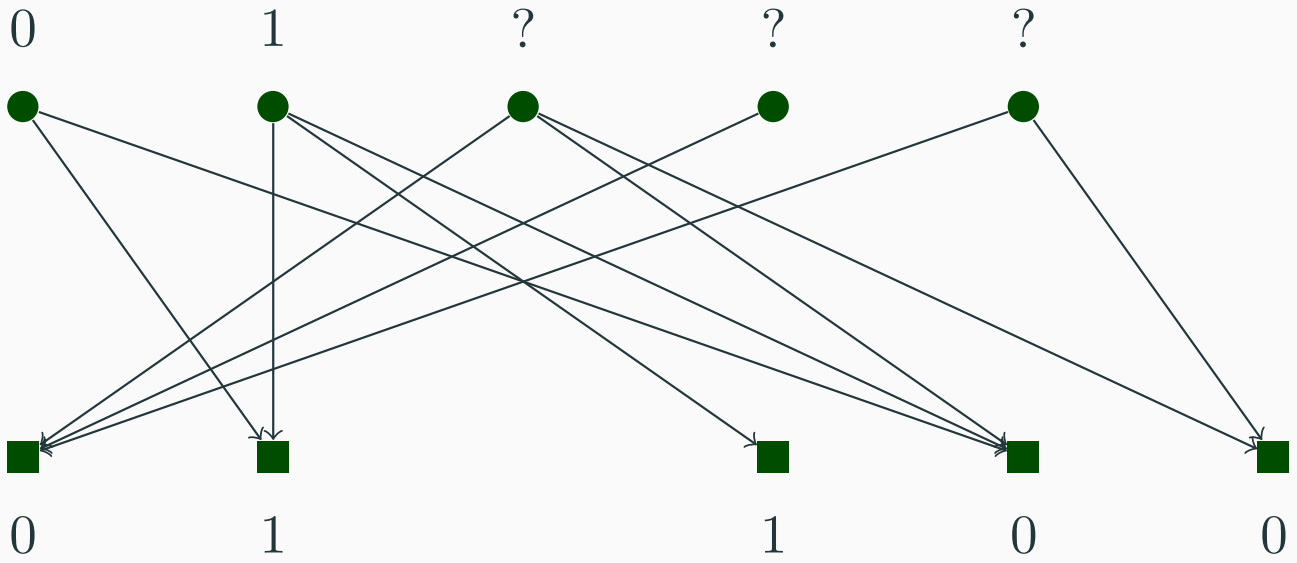
Tanner graph of LT codes



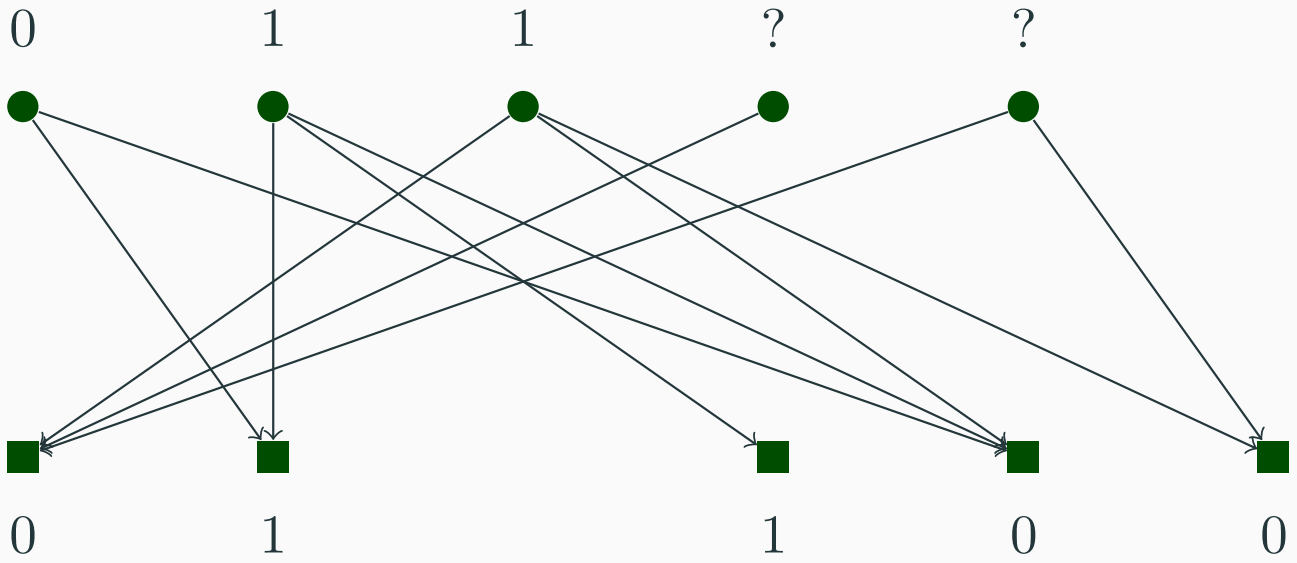
Tanner graph of LT codes



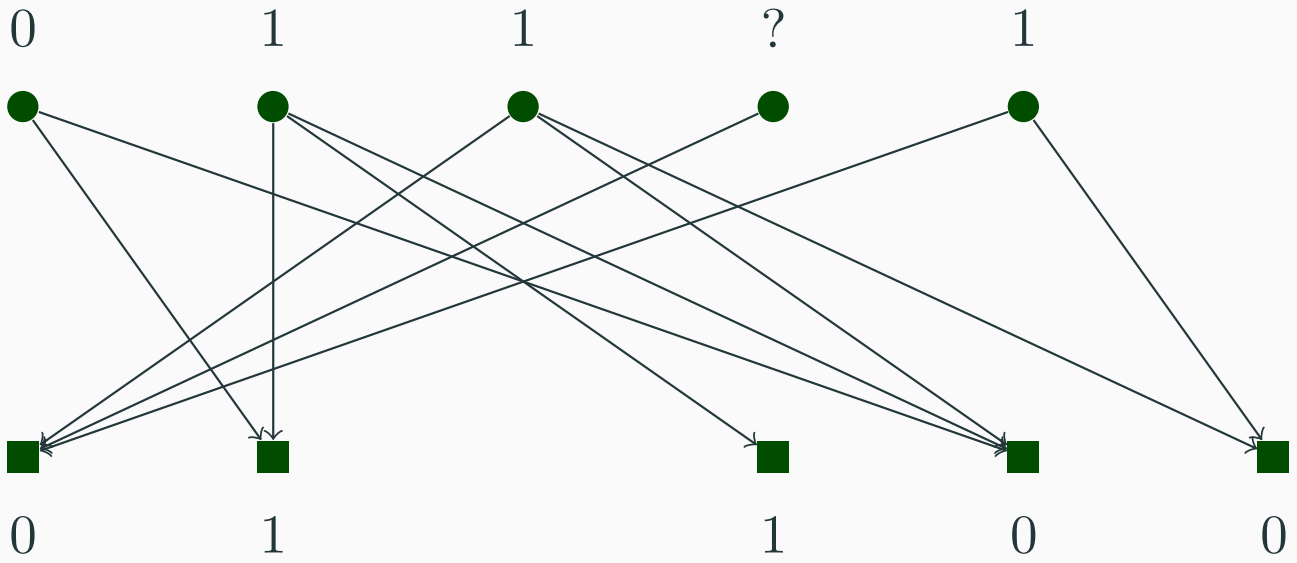
Tanner graph of LT codes



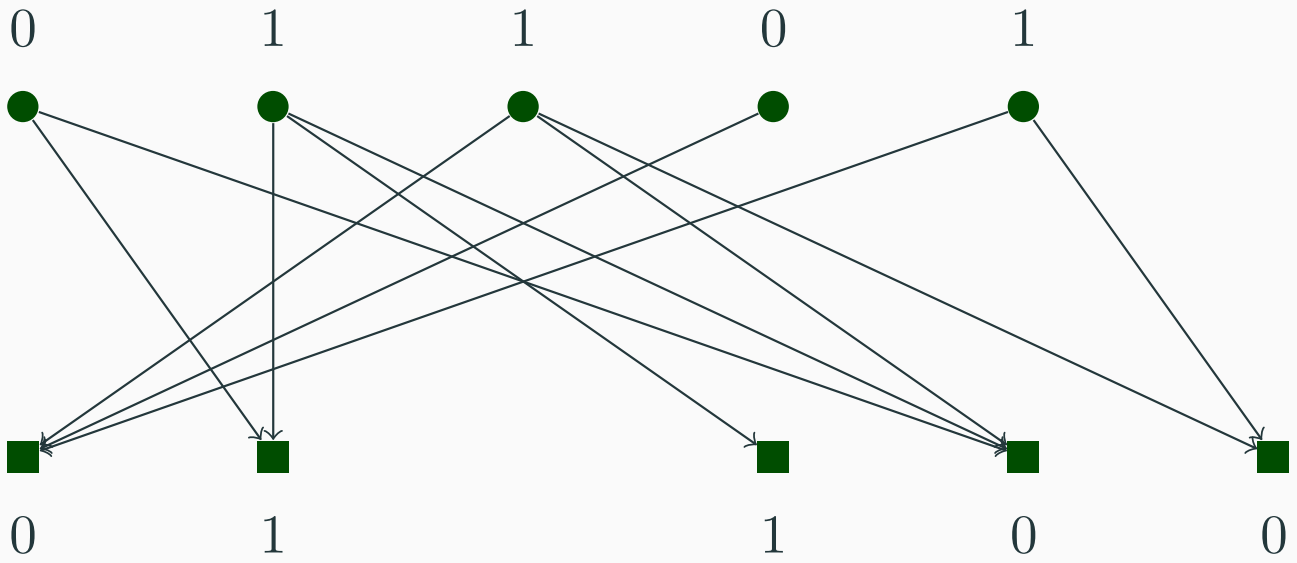
Tanner graph of LT codes



Tanner graph of LT codes



Tanner graph of LT codes



A bound on degree distribution

Theorem
For an LT code with k input packets and n coded packets, if there exists a decoding algorithm with $P_e \leq k^{-c}$, then $\mathbb{E}[\Psi] \geq c' \frac{k}{n} \ln k$.

- So when n is close to k, $\mathbb{E}[\Psi] \geq c' \ln k$.
- Luby showed that there exists a degree distribution such that
  1. $\mathbb{E}[\Psi] = O(\log(k))$,
  2. the BP decoding succeeds with vanishing error probability for n coded packets with $\frac{n-k}{k} \to 0$.

Degree distribution of LT codes

- Ideal Soliton distribution:

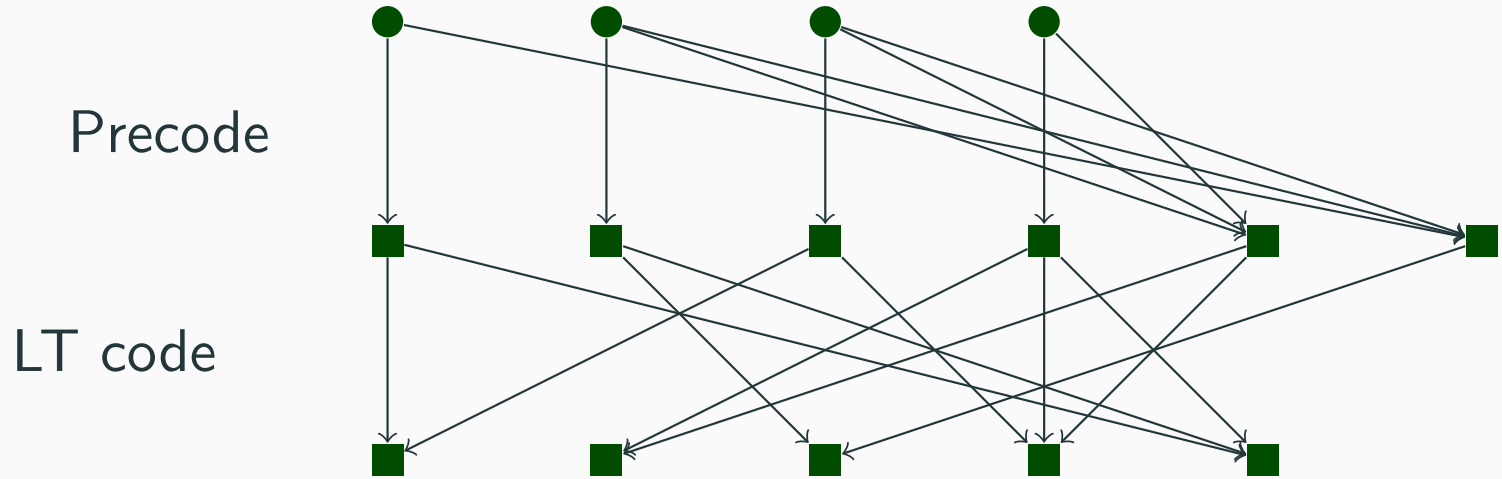
$$\rho_1 = 1/k$$
$$\rho_i = \frac{1}{i(i-1)}, \forall i = 2, 3, \ldots, k.$$

- Robust Soliton distribution:

Raptor codes

Raptor codes [Shokrollahi 2000]

- The trick: precode
- Encoding/decoding complexity $O(1)$ per packet



Degree distribution of Raptor codes

- BP decoding recovers at least $1 - \eta$ fraction of the (intermediate) input packets.
- The maximum degree $D \leq 1/\eta$. So $\mathbb{E}[\Psi] = O(1)$.
- The gap $\frac{n-k}{k}$ can be any positve value but is not vanishing for a fixed degree distribution when $k \to \infty$.

Performance analysis

- Asymptotic analysis: performance when $k \to \infty$.
  - Tree analysis [LMS98]
  - Differential equation approach (see [Wor99])
- Finite-length analysis: performance when k is relative small.
  - Iterative formula for the distribution of the decoder status [KLS04]

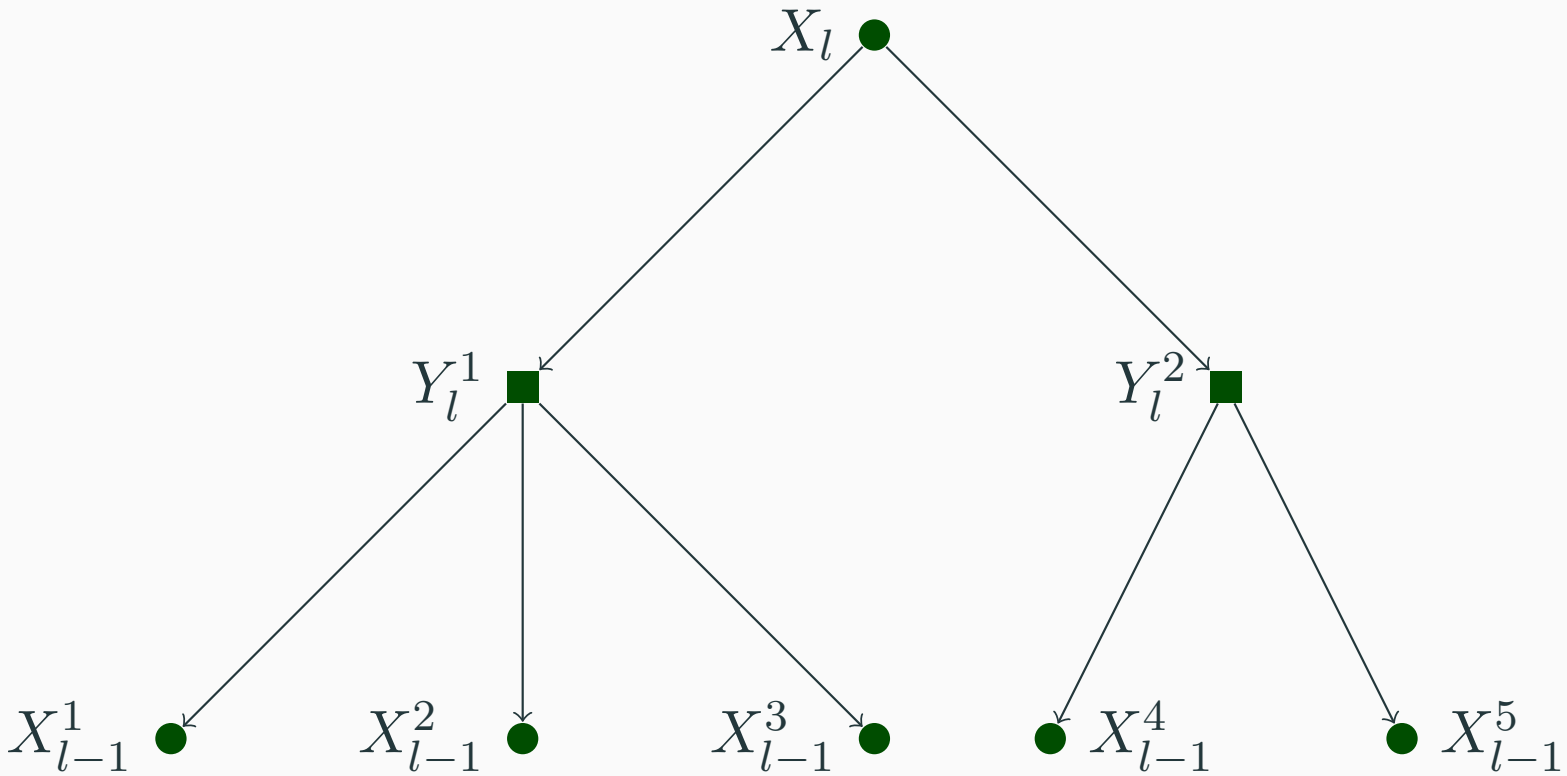
[LMS98] M. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of Random Processes via And-Or Tree Evaluation", in Proc. *SODA*, 1998, pp. 364–373.

[Wor99] N. C. Wormald, "The differential equation method for random graph processes and greedy algorithms," Karonsky and Proemel, eds., Lectures on Approximation and Randomized Algorithms PWN, Warsaw, pp. 73–155, 1999.

[KLS04] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT- codes," in Proc. IEEE ISIT'04, 2004.

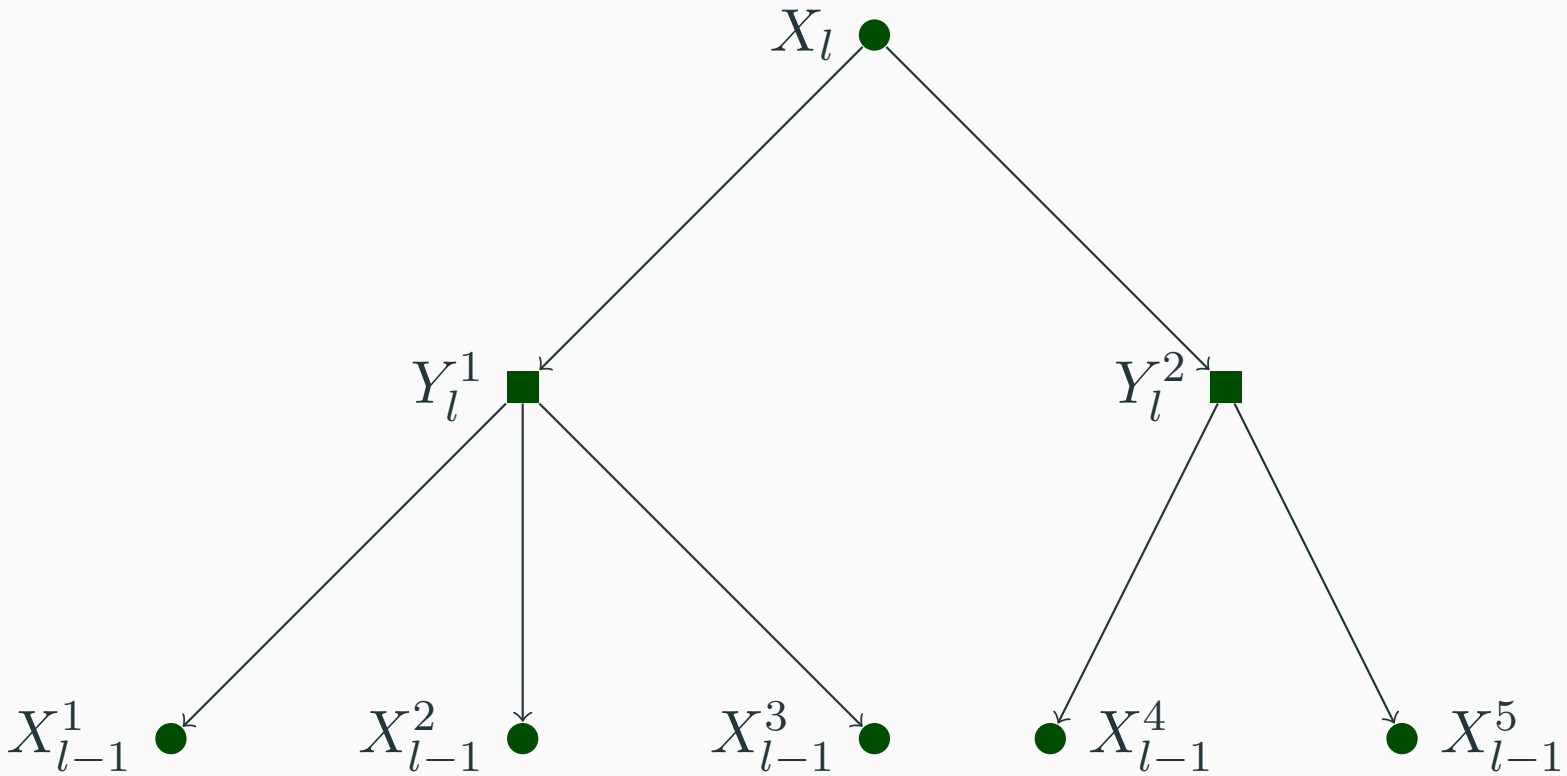
Tree based analysis

And-or tree



And-or tree

$$X_l = Y_l^1 \vee Y_l^2,\ Y_l^1 = X_{l-1}^1 \wedge X_{l-1}^2 \wedge X_{l-1}^3,\ Y_l^2 = X_{l-1}^4 \wedge X_{l-1}^5.$$



And-or tree

- Random variables in the same level are independent.
- Let $x_l = \Pr\{X_l^i = 1\}$, $y_l = \Pr\{Y_l^i = 1\}$.
- Let α_i (β_i) be the probability that an OR (AND) node has i children.

Lemma

$$x_l = 1 - \alpha(1 - \beta(x_{l-1})).$$

Proof.

$$\begin{aligned} x_l &= 1 - \sum_i \alpha_i (1 - y_l)^i \\ &= 1 - \alpha(1 - y_l) \end{aligned} \qquad \begin{aligned} y_{l+1} &= \sum_i \beta_i x_l^i \\ &= \beta(x_l). \end{aligned}$$

□

Variable node degree distribution

- The variable node degree distribution converges to a Poisson distribution

$$\Lambda_k = \frac{(a/C)^k e^{-a/C}}{k!},$$

where $a = \sum_i i\Psi_i$ and $C = K/n$.

Computation graph

- Computation graph T_l: Choose an edge (v, w) and then T_l is the subgraph induced by v and all neighbors of v within distance $2l$ after deleting the edge (v, w).
- A variable node has $i-1$ children with probability $\lambda_i = i\Lambda_i / \sum_i i\Lambda_i$, and a check node has $i-1$ children with probability $\psi_i = i\Psi_i / \sum_i i\Psi_i$.
- The performance of T_l converges to the and-or tree of level $2l+1$ generated using $\{\lambda_i\}$ and $\{\psi_i\}$, when $k \to \infty$.

Tree analysis of LT codes

- Let x_i be the probability that the variable nodes in the $2i$th level is decodable. Using the and-or-tree Lemma, $x_{l+1} = 1 - \lambda(1 - \psi(x_l))$, where

$$\begin{aligned} \psi(z) &= \sum_i \psi_i z^{i-1} = \Psi'(z)/\Psi'(1) = \Psi'(z)/a \\ \lambda(z) &= \sum_i \lambda_i z^{i-1} = \Lambda'(z)/\Lambda'(1) = \exp((z-1)a/C). \end{aligned}$$

Sufficient condition

- To guarantee the success of decoding with high probability, we can require

$$x < 1 - \exp(-\frac{1}{C}\Psi'(x)), \quad \text{for } x \in [0, 1-\eta],$$

which implies

$$\Psi'(x) + C\ln(1-x) > 0, \quad \text{for } x \in [0, 1-\eta].$$

- Let $D = \lfloor 1/(1-\eta) \rfloor - 1$. For any $C < 1$, let

$$\Psi(x) = C\left((1/C-1)x + \sum_{i=2}^{D-1} \frac{x^i}{(i-1)i} + \frac{x^D}{D-1}\right).$$

Differential equation approach

Differential equation: The general method

- Compute the expected changes in random variables of the process per unit time at time t and, regarding the variables as continuous
- Write down the differential equations suggested by the expected changes
- Use large deviation theorems to show that with high probability the solution of the differential equations is close to the values of the variables.

Differential equation: Raptor codes

- Let R_d be the number of edges of degree d. Then

$$E[R_d(t+1) - R_d(t)|R(t)] = (R_{d+1}(t) - R_d(t))\frac{d}{k-t} \quad d > 1$$

$$E[R_1(t+1) - R_1(t)|R(t)] = (R_2(t) - R_1(t))\frac{1}{k-t} - 1 + O(1/k)$$

- The differential equation approach guides us to consider the following system of differential equations

$$\frac{d\rho_d(\tau)}{d\tau} = [\rho_{d+1}(\tau) - \rho_d(\tau)]\frac{d}{C-\tau} \quad d > 1$$

$$\frac{d\rho_1(\tau)}{d\tau} = [\rho_2(\tau) - \rho_1(\tau)]\frac{1}{C-\tau} - 1,$$

  where $\rho_d(0) = d\Psi_d$ and $C = k/n$.

- $R_d(t) \to n\rho_d(t/n)$ (in probability) uniformly for all t before decoding stops.

Solve the system of differential equations

- Since we hope to decode the decoding stops after $t > (1-\eta)K$, we want

$$\rho_1(\tau) > 0, \quad t \in [0, (1-\eta)C].$$

- Solving the system of differential equations, we get

$$\rho_1(\tau) = (1-\tau/C)(\Psi'(\tau/C) + C\ln(1-\tau/C)).$$

- Therefore,

$$\Psi'(x) + C\ln(1-x) > 0, \quad x \in [0, 1-\eta],$$

which is the same requirement as we have obtained using tree analysis.

Finite-length analysis

Finite-length analysis

- Define R^t as number of decodable input symbols at time t
- $P^t_{\text{stop}} = \Pr\{R^t = 0, R^\tau > 0, \tau < t\}$.

Finite-length analysis

- Define R^t as number of decodable input symbols at time t
- $P^t_{\text{stop}} = \Pr\{R^t = 0, R^\tau > 0, \tau < t\}$.
- Define C^t as the number of undecodable received symbols at time t.
- Let $\Lambda^t_{c,r} = \Pr\{C^t = c, R^t = r, R^\tau > 0, \tau < t\}$.

Initial status ($t = 0$)

$$\begin{aligned}
\Lambda^0_{c,r} &= \Pr\{R^0 = r | C^0 = c\} \Pr\{C^0 = c\} \\
&= \Pr\{R^0 = r | C^0 = c\} \mathsf{Bi}(n, c, 1 - \Psi_1) \\
&= \Pr\{|A_{n-c}| = r\} \mathsf{Bi}(n, c, 1 - \Psi_1)
\end{aligned}$$

where $A_{n-c} = \{X_1, X_2, \ldots, X_{n-c}\}$ with X_i be i.i.d. and uniformly distributed in $\{1, 2, \ldots, k\}$.

Recursive formula

$$
\begin{aligned}
&\Lambda^t_{c,r|n} \\
&= \Pr\left\{C^t = c, R^t = r, R^\tau > 0, \tau < t\right\} \\
&= \sum_{c',r'>0} \Pr\left\{C^t = c, C^{t-1} = c', R^t = r, R^{t-1} = r', R^\tau > 0, \tau < t\right\} \\
&= \sum_{c',r'>0} \Pr\left\{C^t = c, R^t = r, |C^{t-1} = c', R^{t-1} = r', R^\tau > 0, \tau < t-1\right\} \Lambda^{t-1}_{c',r'|n} \\
&= \sum_{c',r'>0} \underbrace{\Pr\left\{R^t = r | C^t = c, C^{t-1} = c', R^{t-1} = r', E_{t-1}\right\}}_{(a)} \times \\
&\quad \times \underbrace{\Pr\left\{C^t = c | C^{t-1} = c', R^{t-1} = r', E_{t-1}\right\}}_{(b)} \Lambda^{t-1}_{c',r'|n}.
\end{aligned}
$$

- (a) can be calculated similar to $\Pr\{R^0 = r | C^0 = c\}$.
- (b) can be calculated similar to $\Pr\{C^0 = c\}$.