Rateless Codes

Jie Ren

2013/1/23

ASPITRG, Drexel University

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel

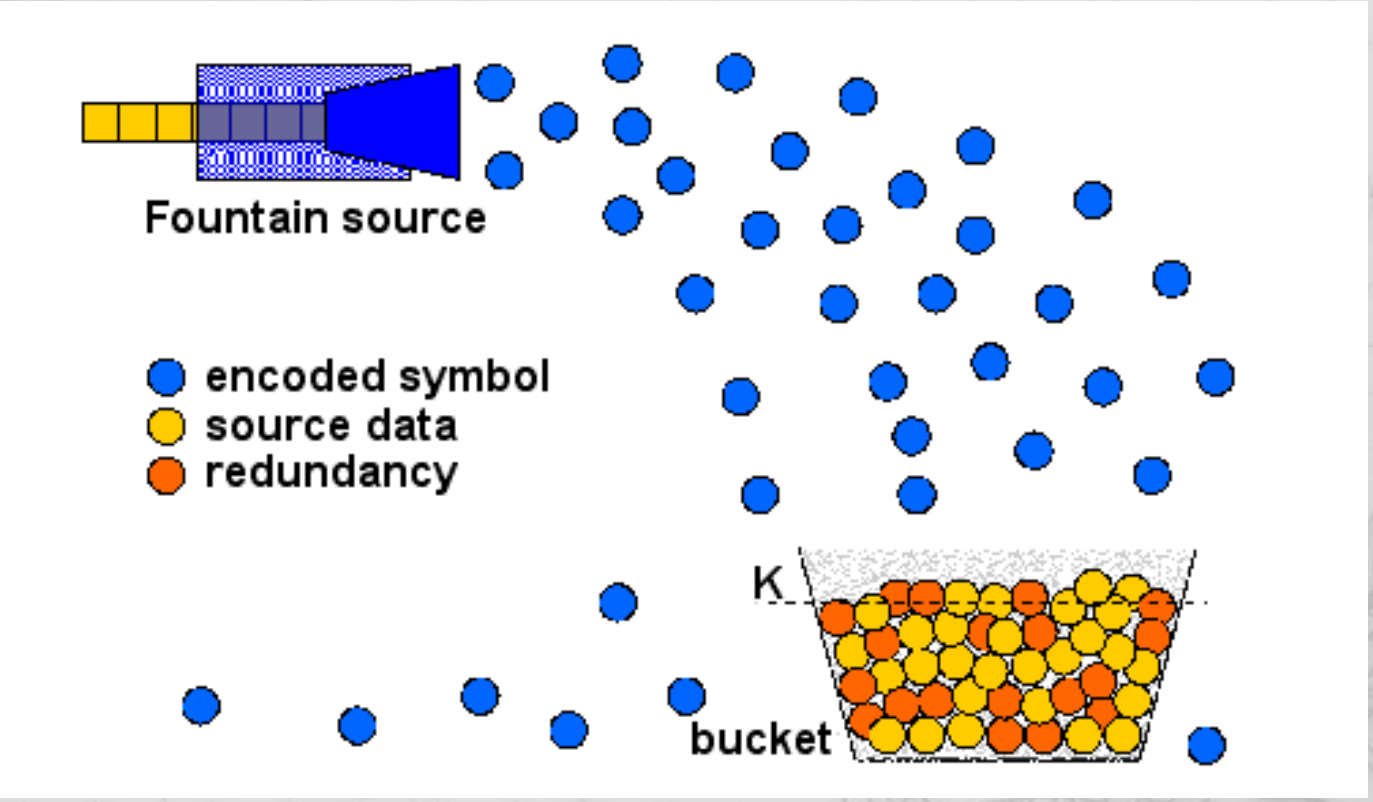
Outline

- Introduction of Rateless Codes
  - Basic Idea of Rateless Codes
  - Applications of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel

Outline

- Introduction of Rateless Codes
  - Basic Idea of Rateless Codes
  - Applications of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel

Basic Idea of Rateless Codes



Outline

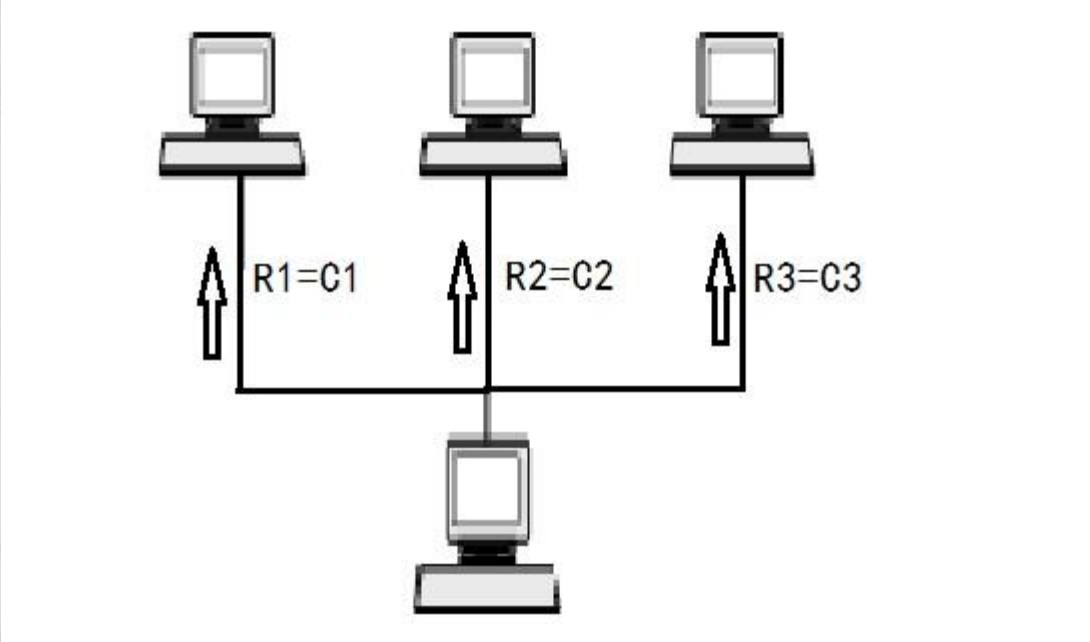
- Introduction of Rateless Codes
  - Basic Idea of Rateless Codes
  - Applications of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel

Applications of Rateless Codes

- One-to-many problem

  Each channel achieve its capacity

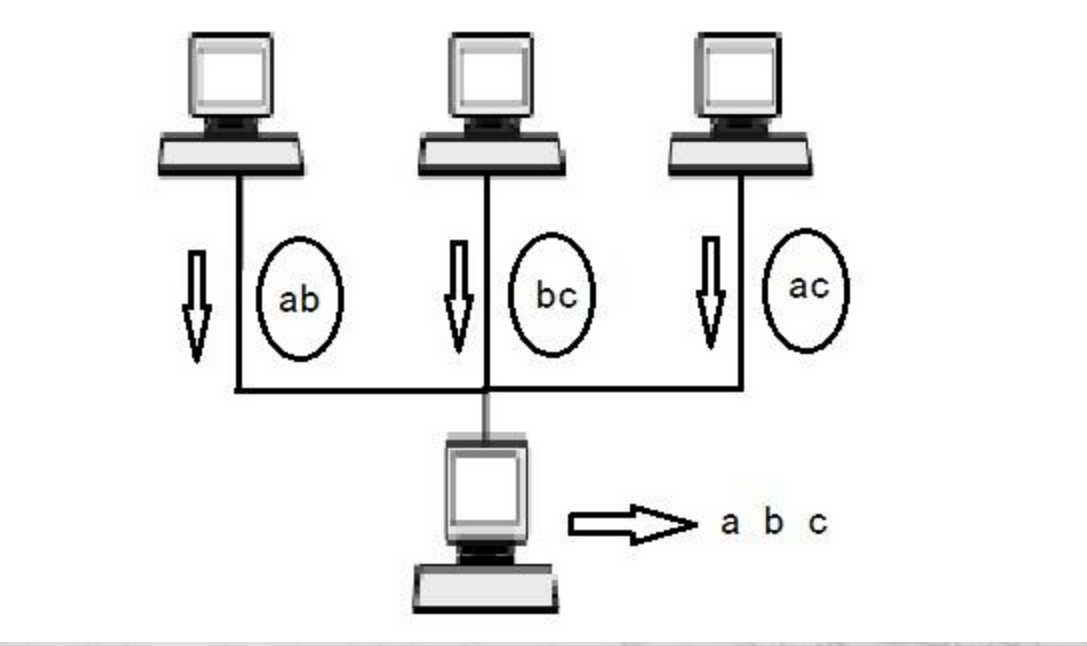
  No need of feed back



Applications of Rateless Codes

- Many-to-one problem

  Minimize delivery of redundant packets



Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel

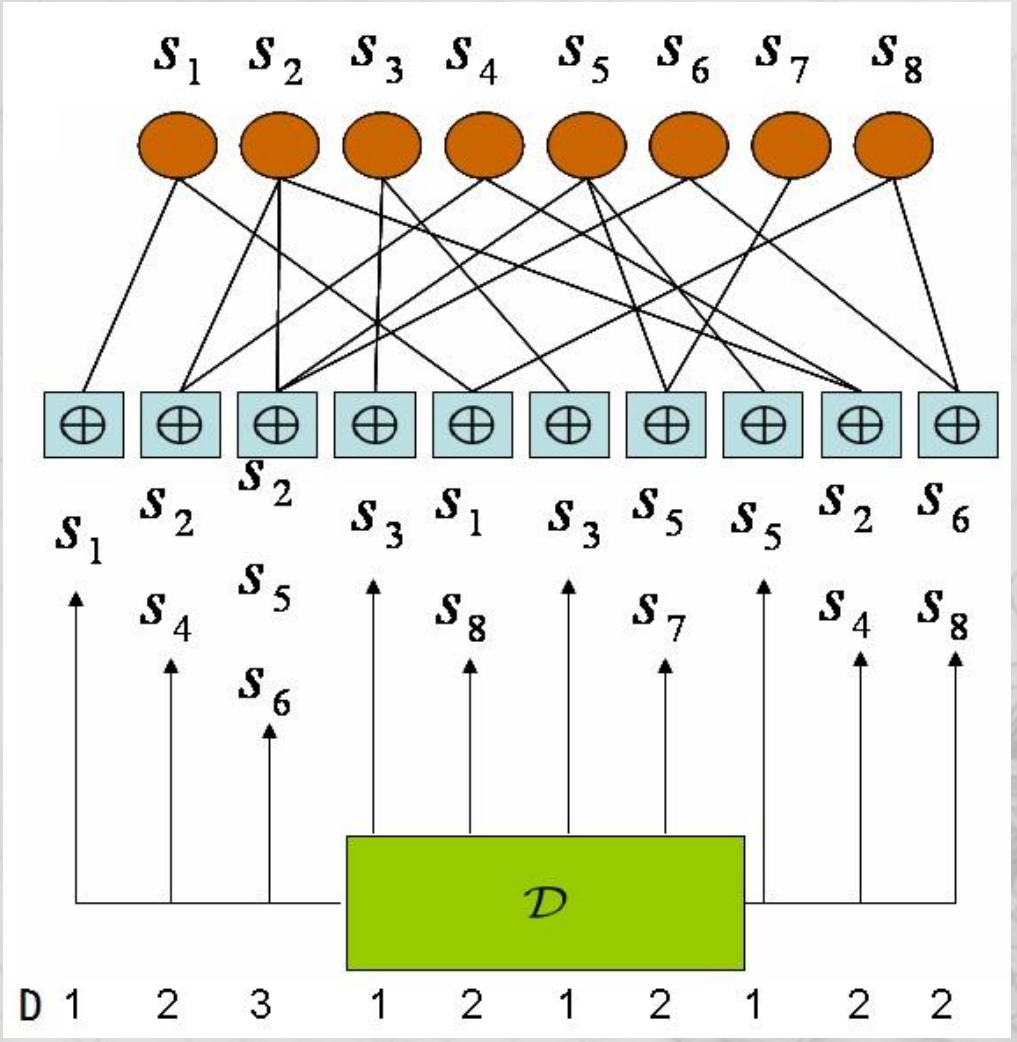
Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
  - Enc/Dec
  - Performance
  - Proof
- Rateless Codes for AWGN Channel

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
  - Enc/Dec
  - Performance
  - Proof
- Rateless Codes for AWGN Channel

Rateless Codes for Erasure Channel



Rateless Codes for Erasure Channel

- Encoder

  k: input symbols of length l

  K: encoding symbols

  d: degree of encoding symbol

  Randomly choose the value d

  Choose d input symbols as neighbors(~uni)

  Calculate the XOR of the d neighbors

Rateless Codes for Erasure Channel

- Decoder

  1. Recover the encoding symbols with d=1
  2. Remove the input symbol as a neighbor

  Repeat

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
  - Enc/Dec
  - Performance
  - Proof
- Rateless Codes for AWGN Channel

Performance

- Encoding Complexity

$$O(\ln(k/\delta))$$

- Decoding Complexity

$$O(k * \ln(k/\delta))$$

- k input symbols can be recovered from any

$$k + O(\sqrt{k}\ln^2(k/\delta))$$

of the encoding symbols with probability 1-δ

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
  - Enc/Dec
  - Performance
  - Proof
- Rateless Codes for AWGN Channel

Proof

- The LT process
- The LT Distributions
- The Ideal Soliton Distribution
- The Robust Soliton Distribution

Proof

- The LT process
- The LT Distributions
- The Ideal Soliton Distribution
- The Robust Soliton Distribution

The LT process

- Definitons
  - Encoding Symbol
  - Input Symbol
  - Neighbor
  - Release Encoding Symbol
  - Ripple Set
  - Cover Set

The LT process

Procedures

Step 1 : release encoding symbols with d=1, put the input symbols in ripple

Step 2 : pick one from ripple, process it, put it in cover set

Step 3 : remove this symbol

Repeat

The LT process

The LT process ends when :

the ripple is empty at the end of some step

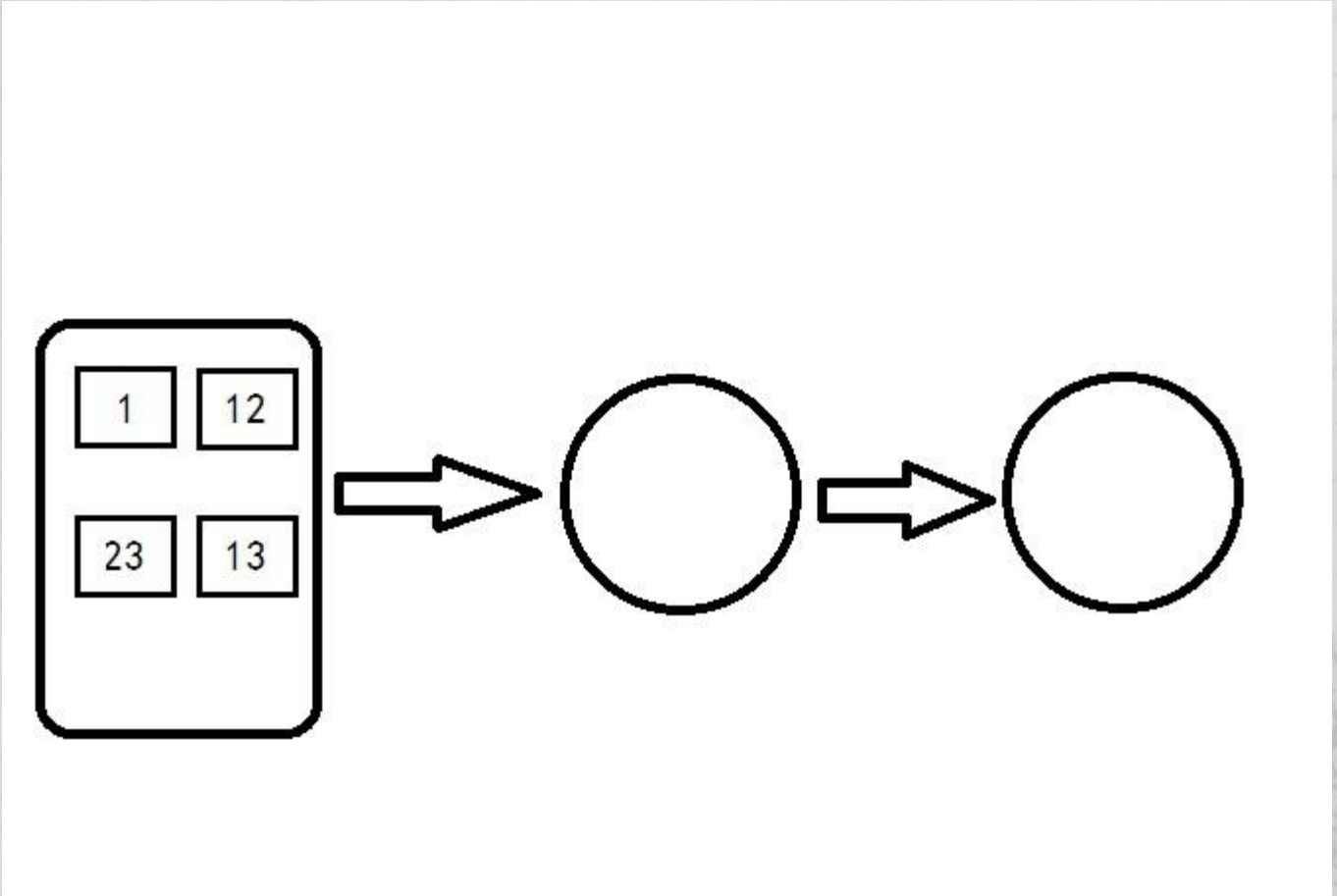
The LT process fails if :

there is at least one uncovered input symbol at the end

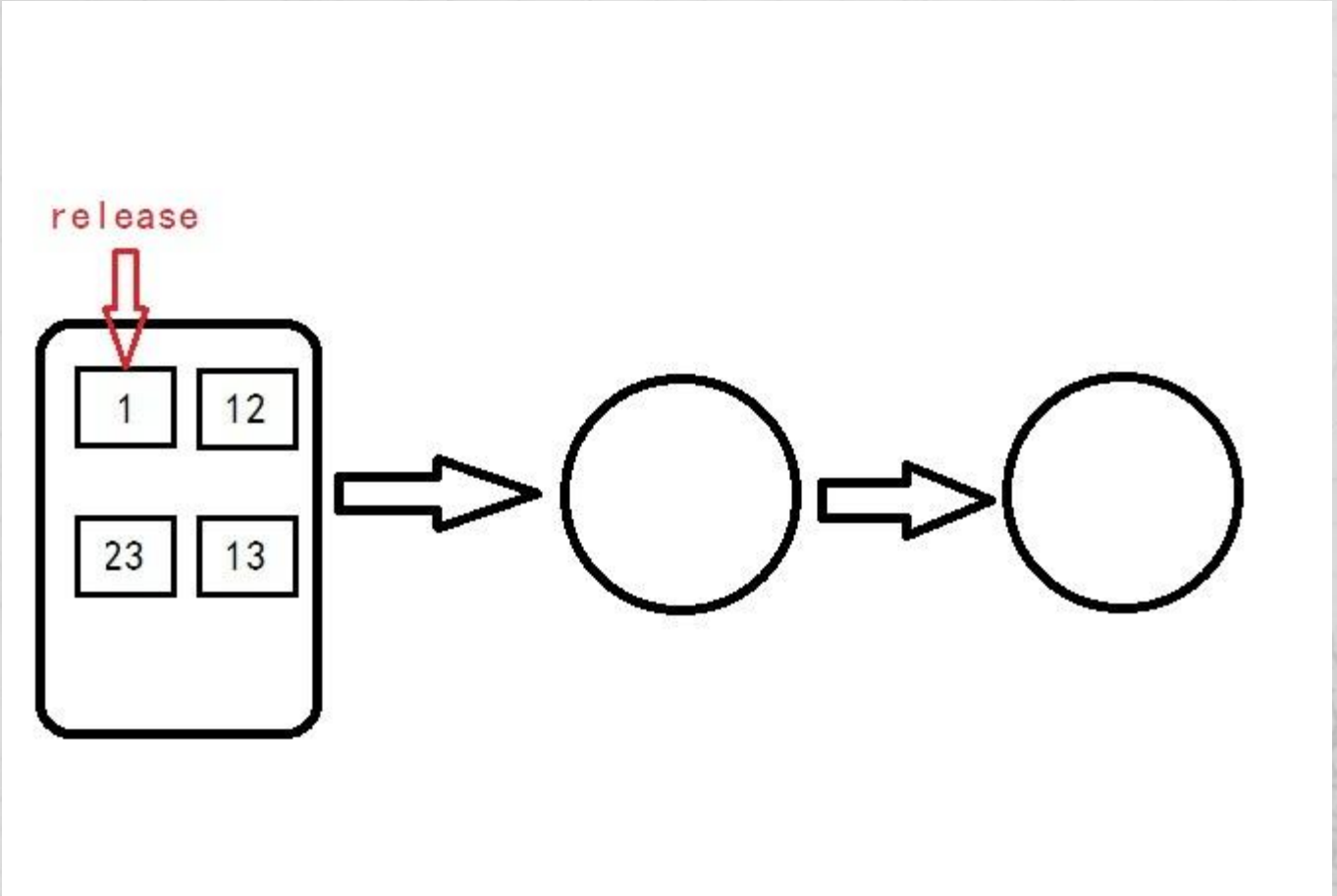
The LT process succeeds if :

all input symbols are covered at the end

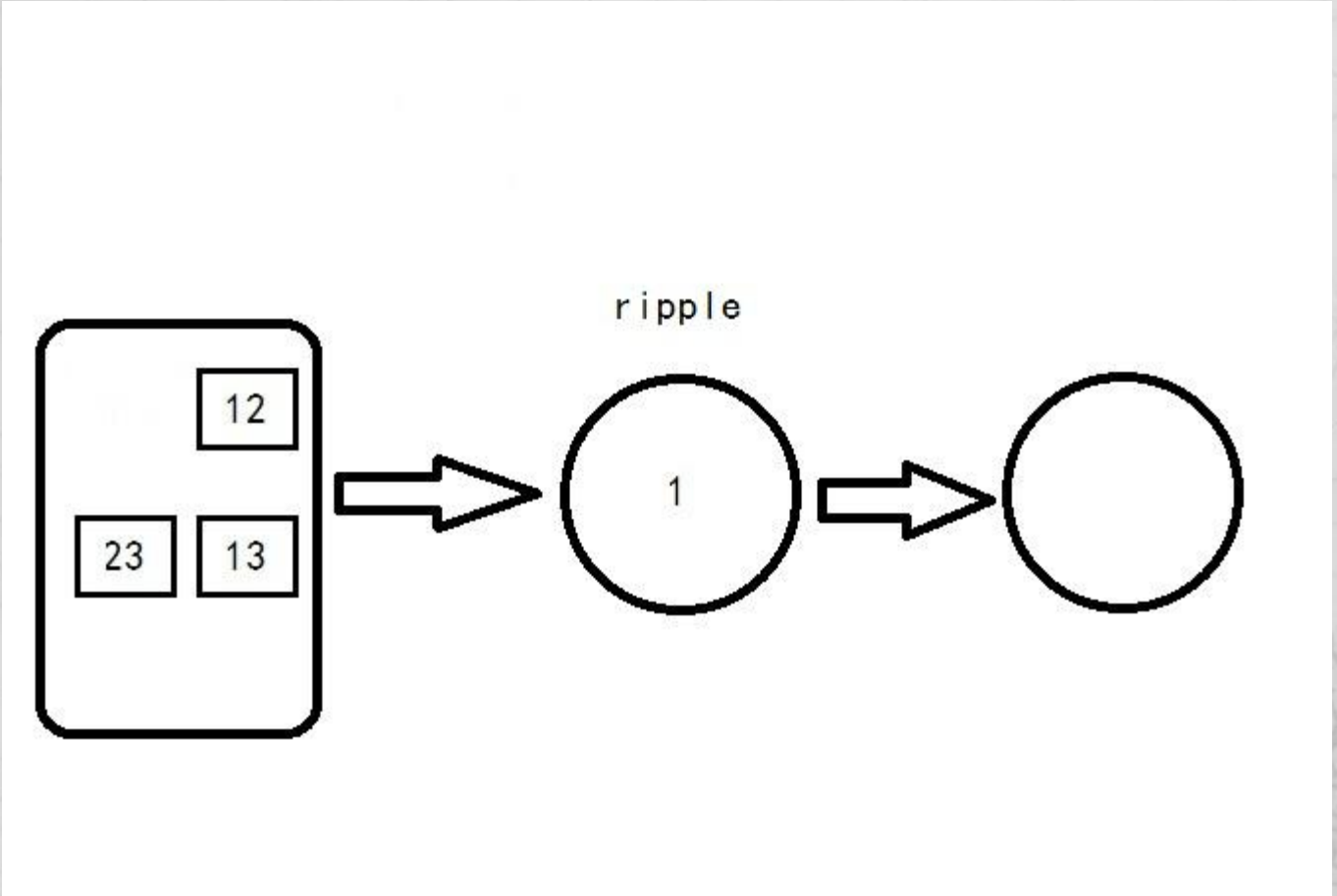
The LT process



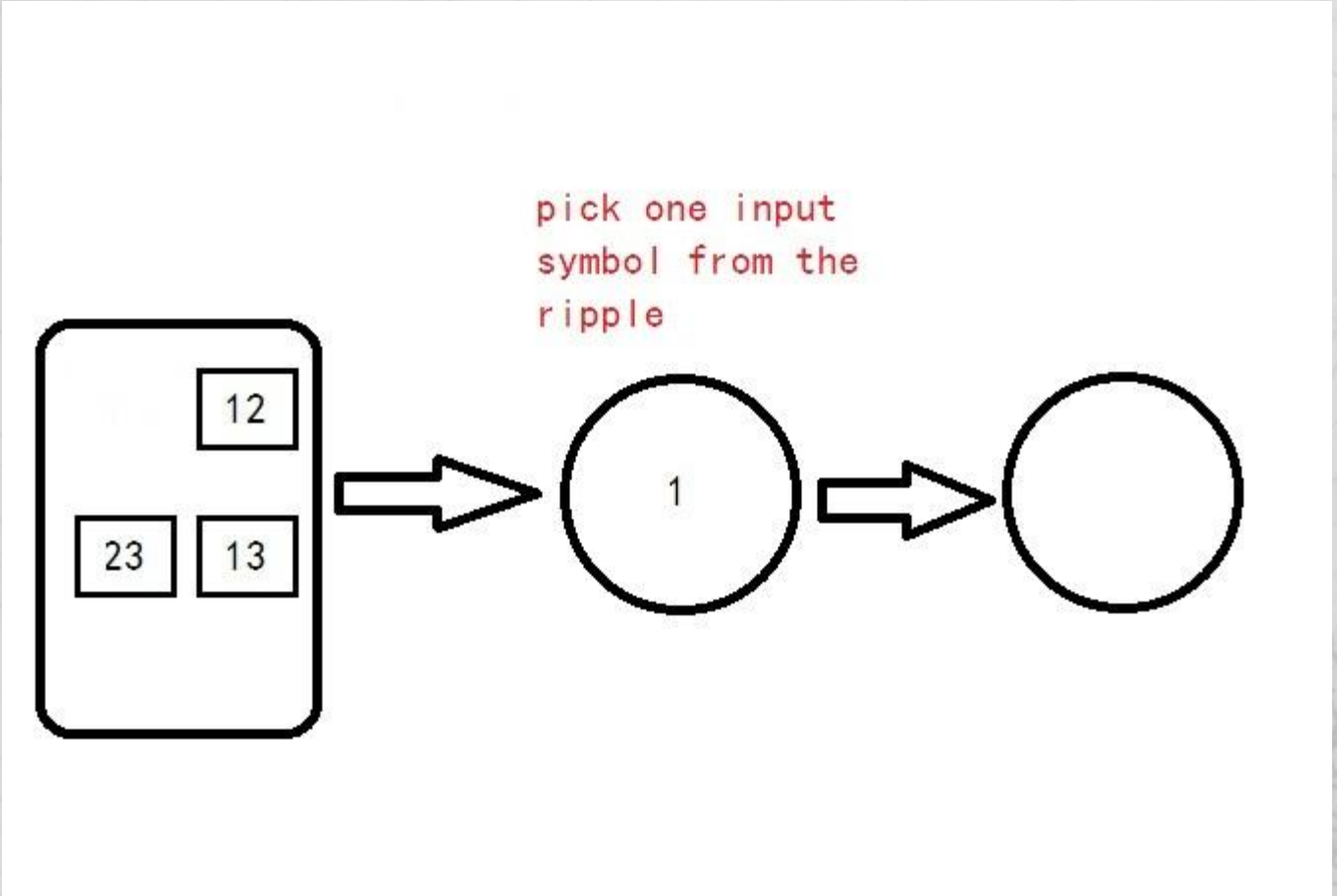
The LT process



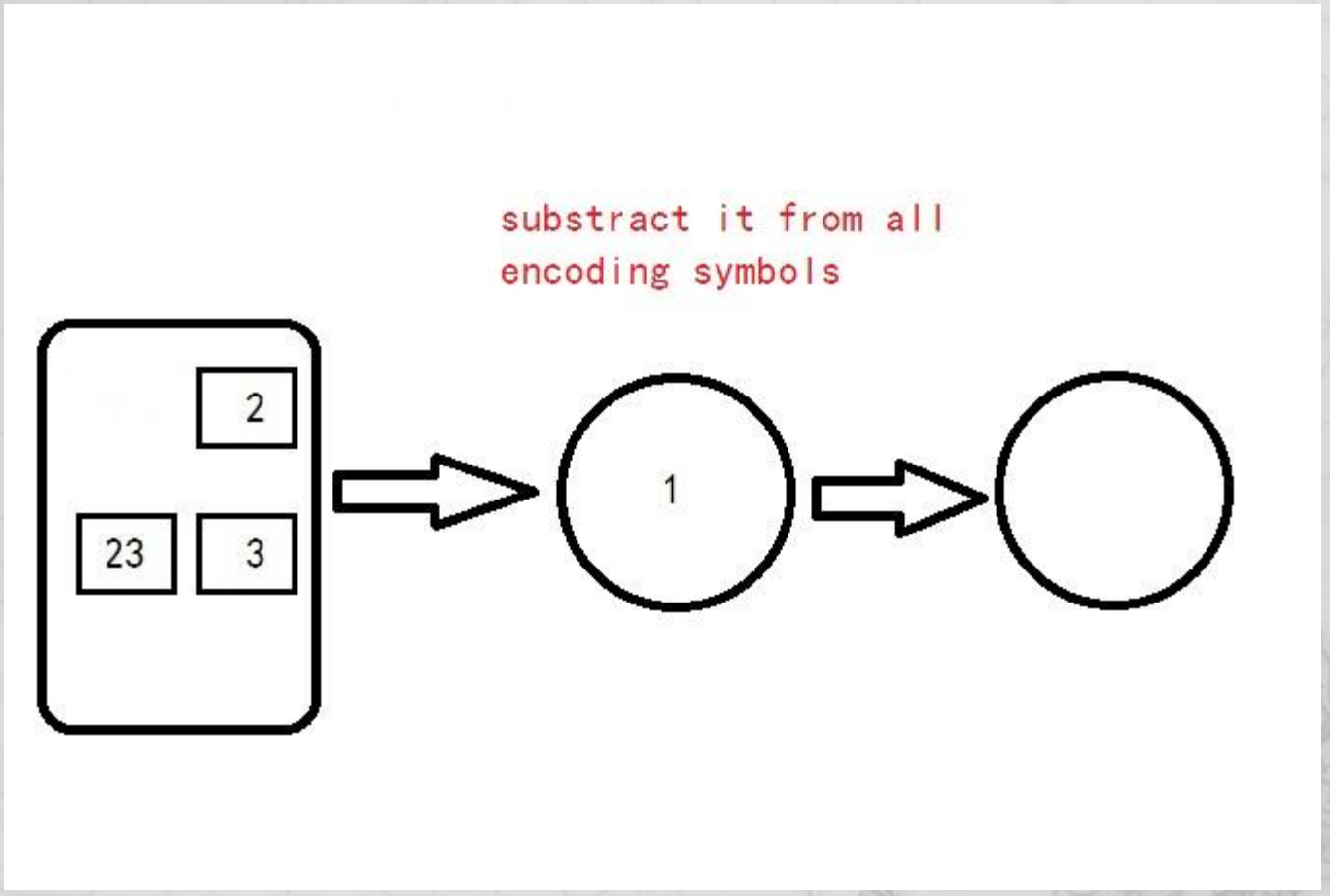
The LT process



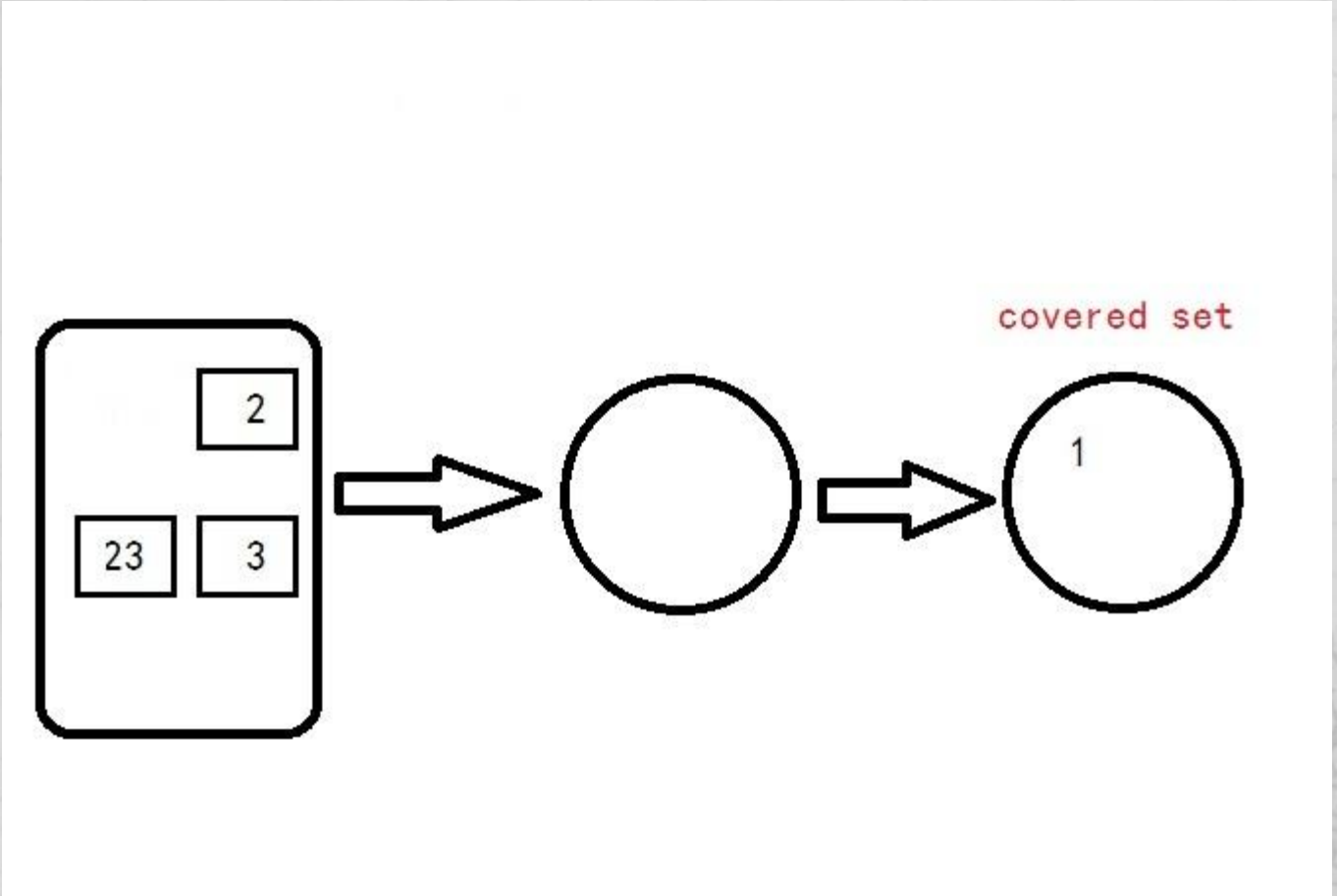
The LT process



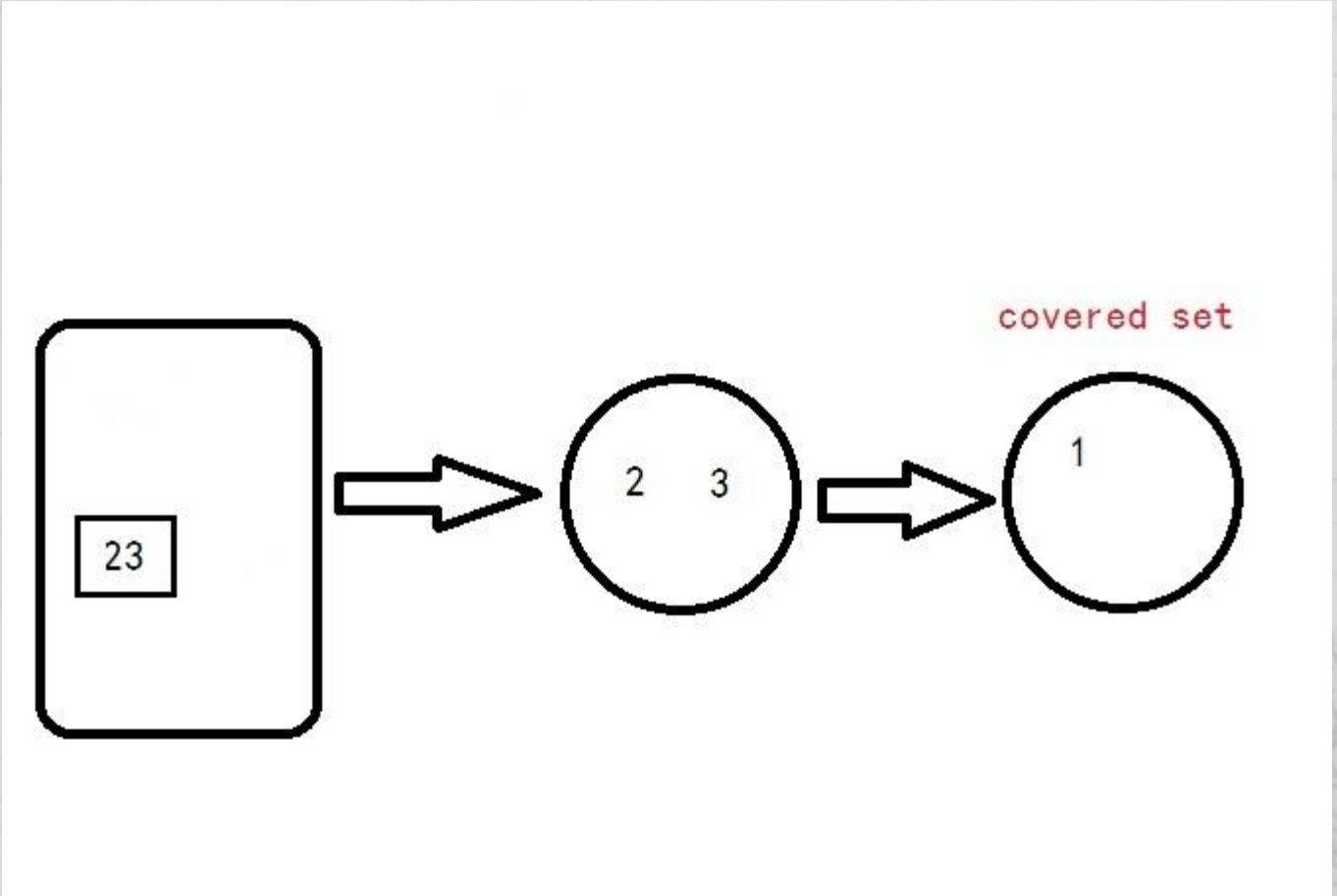
The LT process



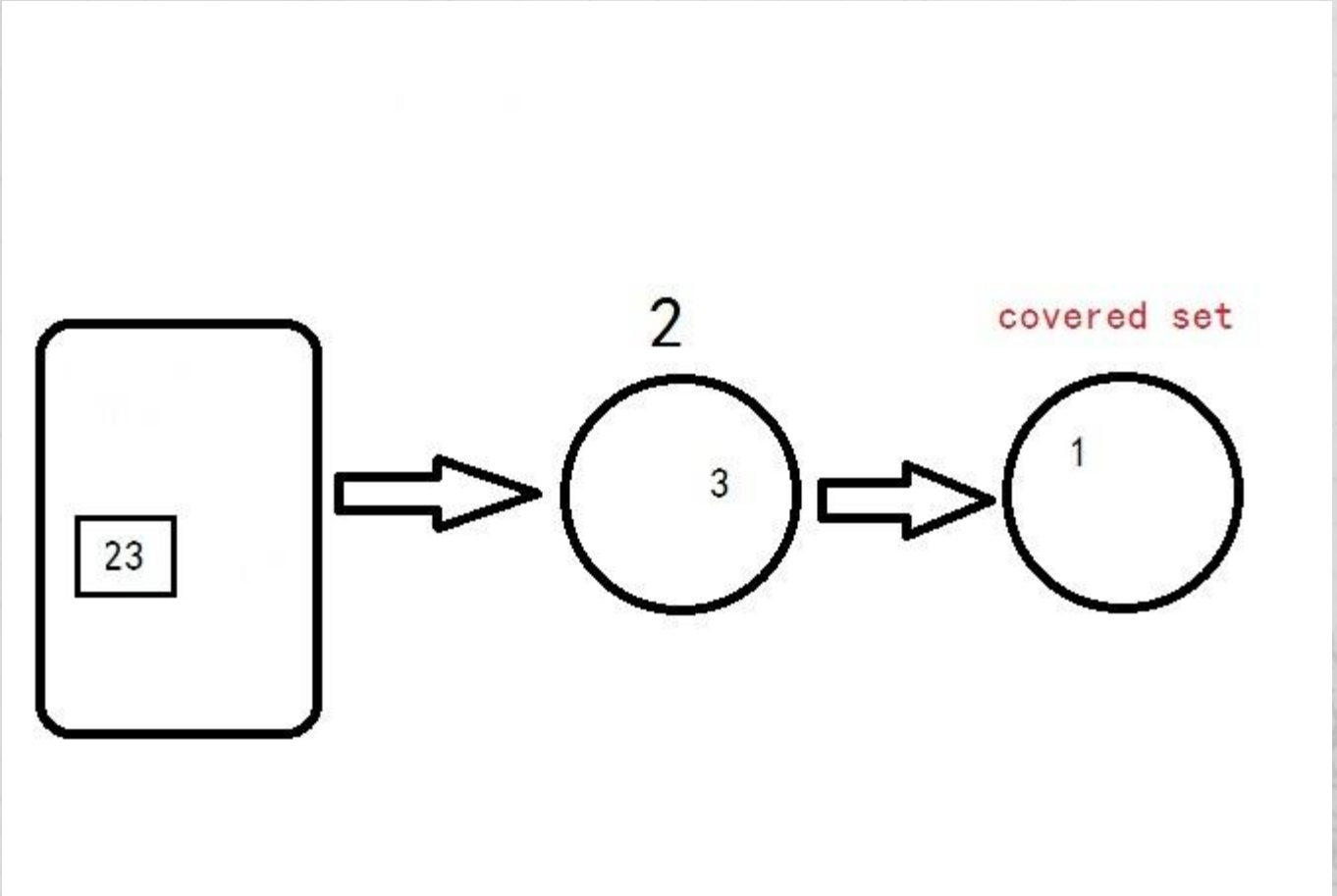
The LT process



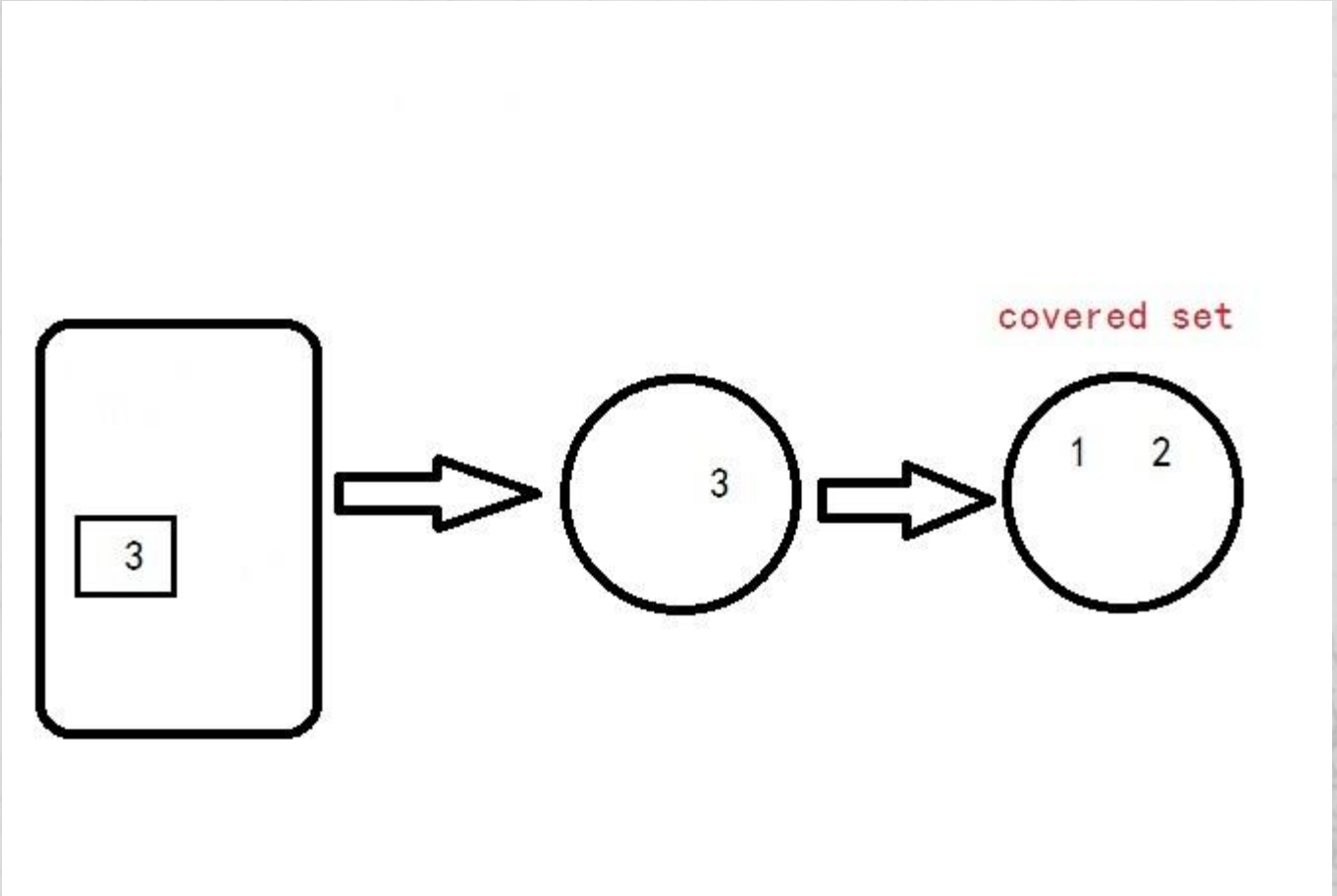
The LT process



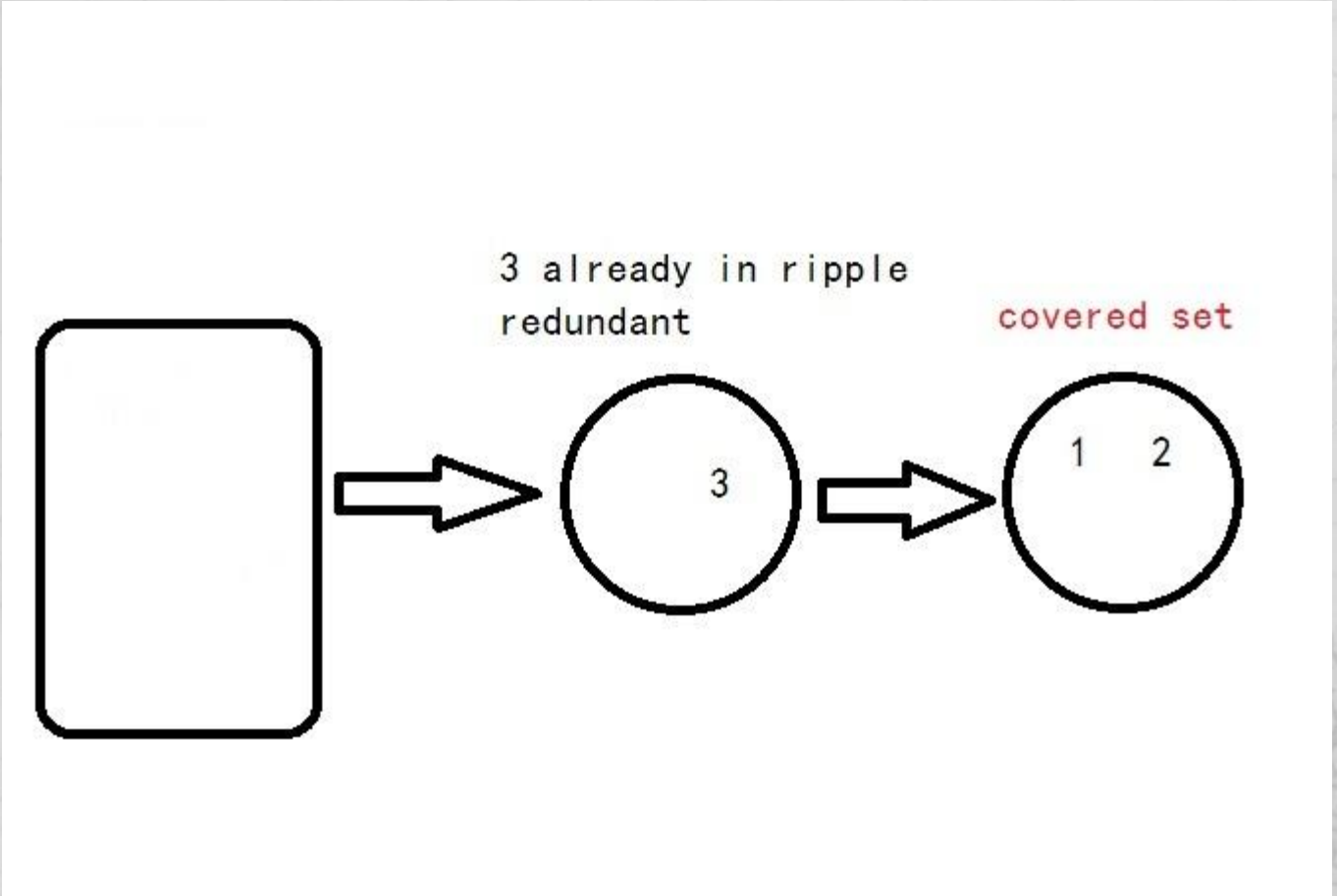
The LT process



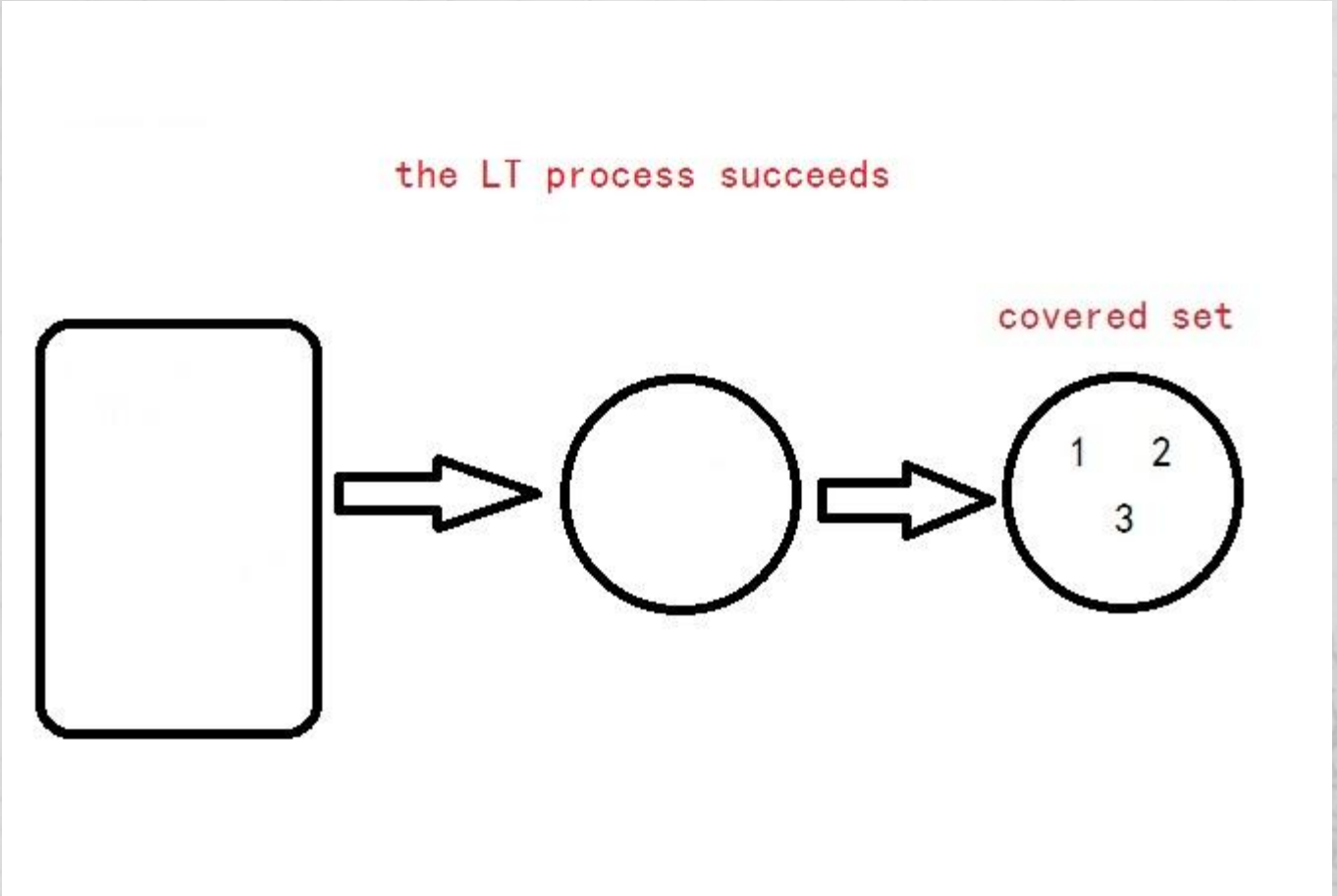
The LT process



The LT process



The LT process



The LT process

- Small fraction of the input symbols in the ripple at the beginning
- The goal of the degree distribution design:
  - "Slowly" release encoding symbols
  - Keep the ripple small
  - Prevent redundant coverage
- Why is that?

The LT process

Consider the three situation with 4 input symbols

Situation 1:

{1} {1,2} {2,3} {3,4} good distribution

Situation 2:

{1} {1,2} {1,3} {1,4} bad distribution

Situation 3:

{1} {1,2} {1,2,3} {1,3,4}

seems good but bad actually

Proof

- The LT process
- The LT Distributions
- The Ideal Soliton Distribution
- The Robust Soliton Distribution

The LT distribution

Two design goals of the degree distribution

- As few encoding symbols as possible on average
- The average degree of the encoding symbols is as low as possible

The LT distribution

Degree Release Probability : q(i,L)

Overall Release Probability : r(L)

$$r(L) = \sum_i r(i,L) = \sum_i p(i) \cdot q(i,L)$$

Degree Release Probability

$$q(1,k) = 1$$

For i = 2,…,k, for all L=k-i+1,…,1

$$q(i,L) = \frac{i(i-1)L\prod_{j=0}^{i-3} k-(L+1)-j}{\prod_{j=0}^{i-1} k-j}$$

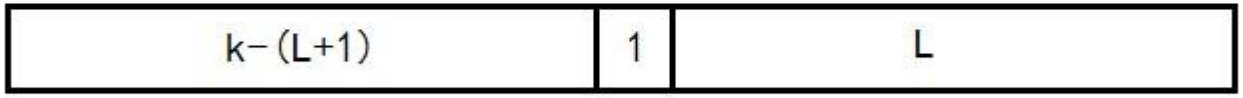
For all other i and L

$$q(i,L) = 0$$

Degree Release Probability

$$q(i,L) = \frac{i(i-1)L\prod_{j=0}^{i-3} k - (L+1) - j}{\prod_{j=0}^{i-1} k - j}$$

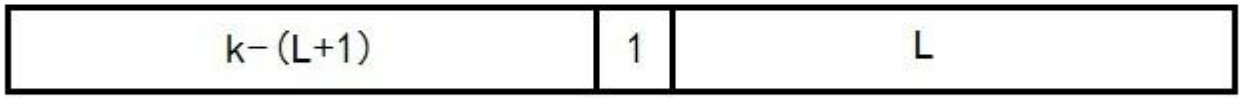
Degree Release Probability



k-(L+1) symbols are covered
k-Lth symbol is under processing
L symbols remain uncovered

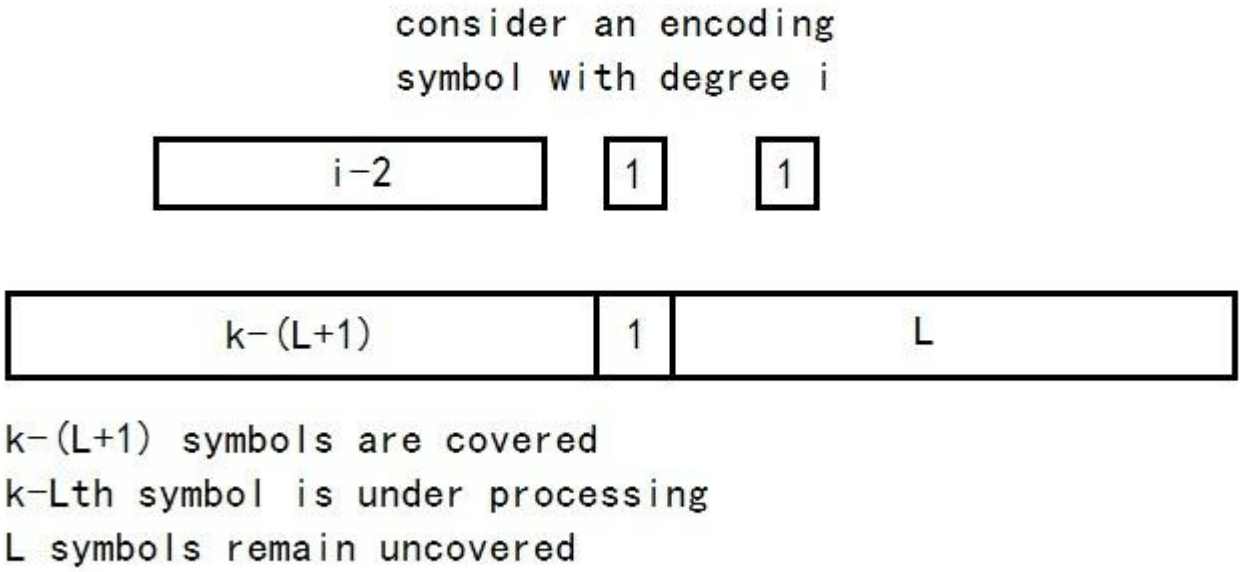
Degree Release Probability

consider an encoding
symbol with degree i



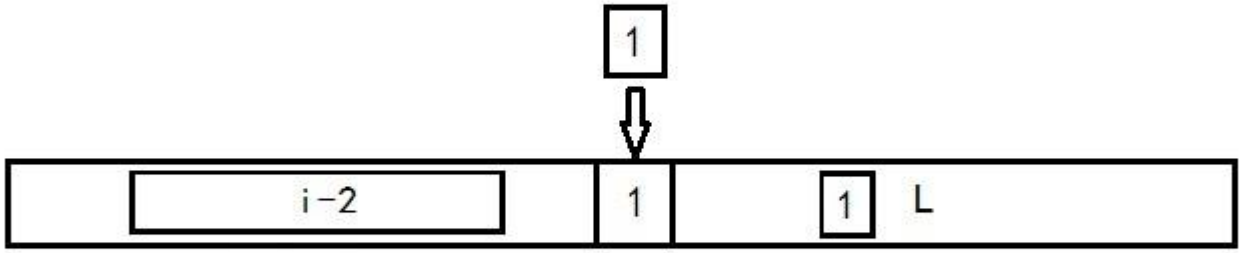
k-(L+1) symbols are covered
k-Lth symbol is under processing
L symbols remain uncovered

Degree Release Probability



Degree Release Probability

an encoding symbol with degree i is released if
i-2 symbols are covered
1 symbol is under processing
Another one is uncovered



k-(L+1) symbols are covered
k-Lth symbol is under processing
L symbols remain uncovered

Degree Release Probability

i-2 symbols are covered:

$$\binom{k-(L+1)}{i-2}\binom{L+1}{2}/\binom{k}{1}$$

1 symbol is covering, 1 remains uncovered:

$$\binom{L}{1}/\binom{L+1}{2}$$

Degree Release Probability

$$q(i,L) = \frac{\binom{k-(L+1)}{i-2}\binom{L+1}{2}}{\binom{k}{1}} * \frac{\binom{L}{1}}{\binom{L+1}{2}}$$

$$= \frac{i(i-1)L\prod_{j=0}^{i-3} k-(L+1)-j}{\prod_{j=0}^{i-1} k-j}$$

Degree Release Probability

An encoding symbol is chosen to be d=i and is released when L input symbols remain uncovered

$$r(i, L) = p(i) \cdot q(i, L)$$

Overall Release Probability

$$r(L) = \sum_i r(i, L) = \sum_i p(i) \cdot q(i, L)$$

Proof

- The LT process
- The LT Distributions
- The Ideal Soliton Distribution
- The Robust Soliton Distribution

The Ideal Soliton Distribution

For i=1

$$\rho(1) = 1/k$$

For i=2,…,k,

$$\rho(i) = 1/i(i-1)$$

The Ideal Soliton Distribution

Theorem 1 (uniform release probability)

For the Ideal Soliton distribution,

$$r(L) = \frac{1}{k} \quad \text{for all } L = k, \ldots, 1$$

Proof of Theorem 1

For L = k

All encoding symbols with d=1 are released

$$r(L) = \sum_i r(i, L) = \sum_i p(i) \cdot q(i, L)$$

$$r(L) = p(1) \cdot q(1, L) = 1 \cdot \frac{1}{k}$$

Proof of Theorem 1

For L = k-1,…,1

$$r(i,L) = p(i) \cdot q(i,L)$$

$$r(i,L) = \frac{1}{i(i-1)} \cdot \frac{i(i-1)L\prod_{j=0}^{i-3} k-(L+1)-j}{\prod_{j=0}^{i-1} k-j}$$

$$r(i,L) = \frac{L\prod_{j=0}^{i-3} k-(L+1)-j}{\prod_{j=0}^{i-1} k-j}$$

Proof of Theorem 1

For L = k-1,…,1

$$r(i,L) = \frac{L\prod_{j=0}^{i-3} k - (L+1) - j}{\prod_{j=0}^{i-1} k - j}$$

$$k \cdot r(i,L) = \frac{L\prod_{j=0}^{i-3} k - (L+1) - j}{\prod_{j=1}^{i-1} k - j}$$

Proof of Theorem 1

Rewrite the formula

$$k \cdot r(i, L) = \frac{L \prod_{j=0}^{i-3}(k-1) - L - j}{\prod_{j=0}^{i-2}(k-1) - j}$$

$$= \frac{(k-1) - L}{k-1} \cdots \frac{(k-1) - (i-3) - L}{(k-1) - (i-3)} \cdot \frac{L}{(k-1) - (i-2)}$$

Proof of Theorem 1

Consider:

Throw balls uniformly among k-1 bins

Eliminating each bin once occupied

i-1th ball first comes into one of L designated bins

$$= \frac{(k-1)-L}{k-1} \cdots \frac{(k-1)-(i-3)-L}{(k-1)-(i-3)} \cdot \frac{L}{(k-1)-(i-2)}$$

Proof of Theorem 1

$$k \cdot r(L) = k \cdot \sum_{i=1}^{k-L+1} r(i, L) = 1$$

Hence,

$$r(L) = \frac{1}{k}$$

Proof

- The LT process
- The LT Distributions
- The Ideal Soliton Distribution
- The Robust Soliton Distribution

The Robust Soliton Distribution

Let $R = c \cdot \ln(k/\delta)\sqrt{k}$ for suitable c>0

Define:

$$\tau(i) = \begin{cases} \frac{R}{ik} & \text{for } i = 1, \ldots, \frac{k}{R} - 1 \\ \frac{R \ln\left(\frac{R}{\delta}\right)}{k} & \text{for } i = k/R \\ 0 & \text{for } i = \frac{k}{R}, \ldots, k \end{cases}$$

The Robust Soliton Distribution

Add the ideal Soliton distribution and normalize to get the robust Solition distribution:

$$\beta = \sum_{i=1}^{k} \rho(i) + \tau(i)$$

For all i = 1,…,K

$$\mu(i) = (\rho(i) + \tau(i))/\beta$$

The Robust Soliton Distribution

Idea:

Keep the ripple size = R instead of 1

Using $K=k\beta$ encoding symbols instead of k

The Robust Soliton Distribution

How we design this Robust Soliton Distribution?

A released encoding symbol adds to the ripple $\frac{L}{(L-R)}$

Requires $\frac{(L-R)}{L}$ released symbols to add one to the ripple

The Robust Soliton Distribution

Density of encoding symbols with degree i=k/L should be proportional to

$$\frac{L}{i(i-1)(L-R)} = \frac{k}{i(i-1)(k-iR)}$$

$$= \frac{1}{i(i-1)} + \frac{R}{(i-1)(k-iR)} \approx \rho(i) + \tau(i)$$

The Robust Soliton Distribution

Theorem 2 (number of encoding symbols)

$$K = k + O(\sqrt{k} \cdot \ln^2(k/\delta))$$

The Robust Soliton Distribution

Theorem 3 (average degree of an encoding symbol)

$$D = O(\ln (k/\delta))$$

The Robust Soliton Distribution

Euler-Mascherorri Constant

$$\gamma = \lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k} - \ln(n)$$

$$\gamma = 0.577\ldots$$

The Robust Soliton Distribution

Theorem 4 (high probability recovery)

The decoder fails to recover the data with probability at most δ from a set of K encoding symbols

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel
  - Channel and System Model
  - Code Structure
  - Perfect Rateless Codes

Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel
  - Channel and System Model
  - Code Structure
  - Perfect Rateless Codes

Rateless Codes for AWGN Channel

- Channel Model

$$Y_m^N = \beta X_m^N + Z_m^N, \quad m = 1,2, \ldots,$$

β channel gain

σ^2 noise variance

β and σ^2 are known a priori at the Dec

P power limit

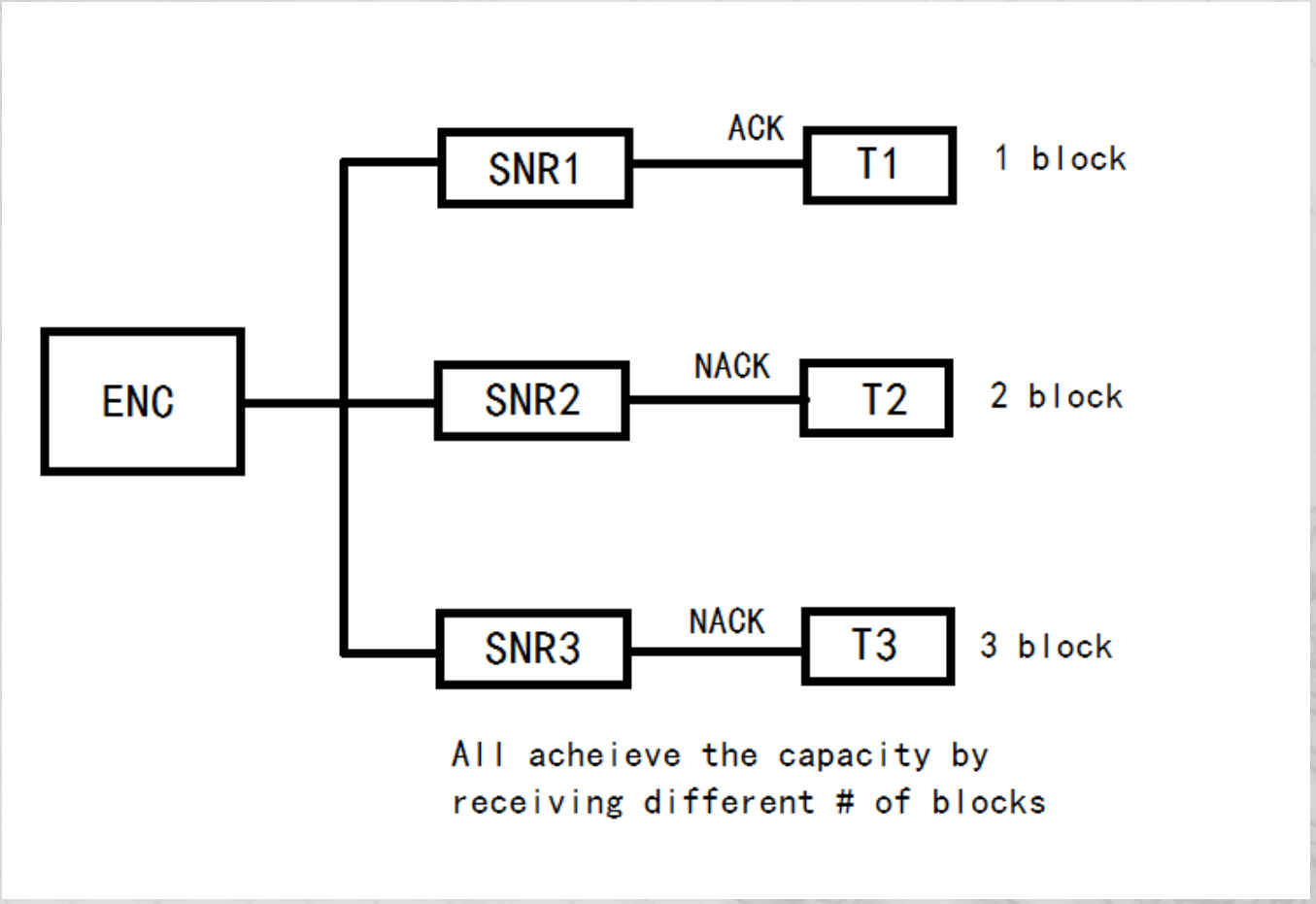
a sequence of code blocks $x_1(w), x_2(w), \ldots,$

receiver get sufficient blocks of y to recover w

receiver always starts receiving from m=1

Rateless Codes for AWGN Channel

- System Model



Outline

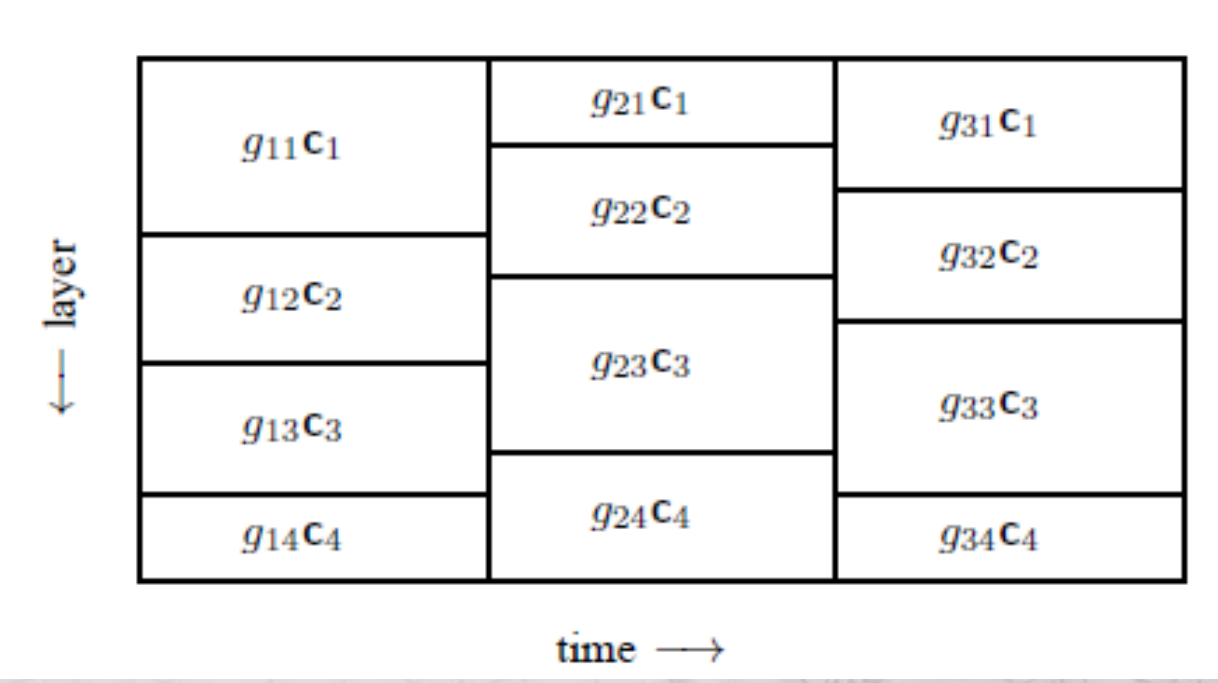
- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel
  - Channel and System Model
  - Code Structure
  - Perfect Rateless Codes

Rateless Codes for AWGN Channel

- Layering
- Repetition
- Time-varying Weighting

Rateless Codes for AWGN Channel

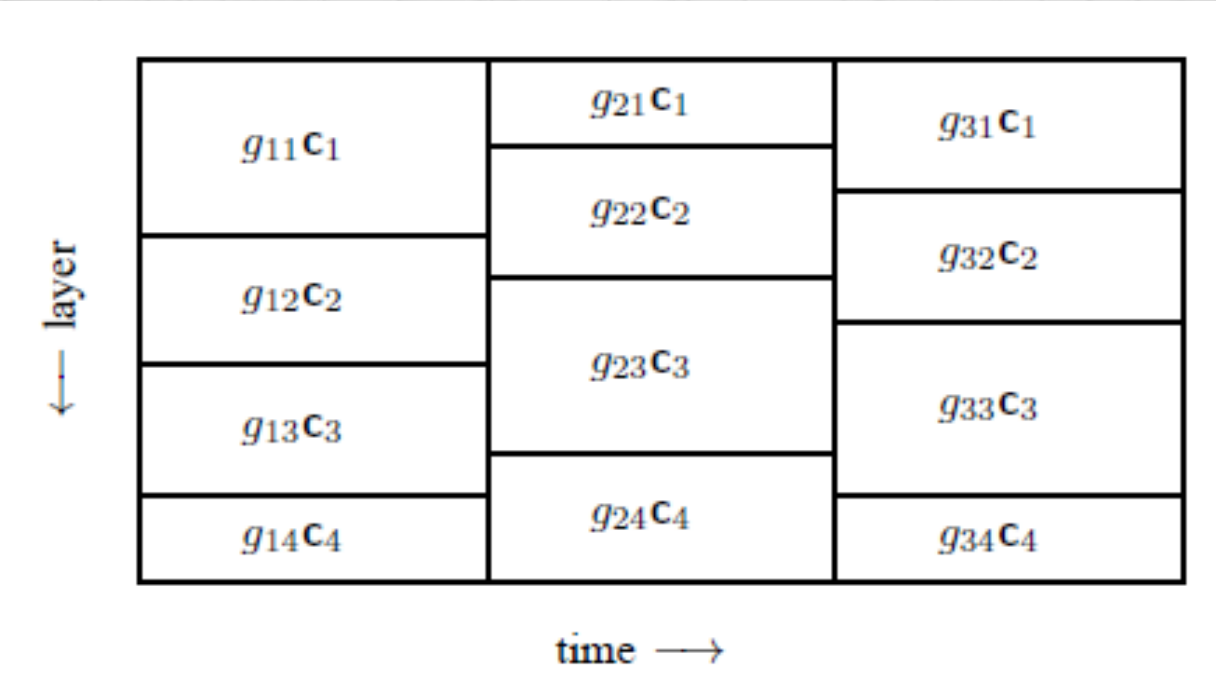
- Layering
- Repetition
- Time-varying Weighting



Practical Rateless Codes for AWGN

$$x = g_{11}c_1 + g_{12}c_2 + g_{13}c_3 + g_{14}c_4$$

$$|g_{11}|^2 + |g_{12}|^2 + |g_{13}|^2 + |g_{14}|^2 = P$$



Outline

- Introduction of Rateless Codes
- Rateless Codes for Erasure Channel
- Rateless Codes for AWGN Channel
  - Channel and System Model
  - Code Structure
  - Perfect Rateless Codes

Perfect Rateless Codes

- Use a 2-layers design as an example

  two channel gain thresholds

  $\beta \geq \alpha_1$

  decoding requires one block

  $\alpha_1 \geq \beta \geq \alpha_2$

  decoding requires two blocks

  $\alpha_2 > \beta$

  decoding is not possible

Perfect Rateless Codes

One-block decoding

$$Y_1^N = \alpha_1 X_1^N + Z_1^N$$

Two-block decoding

$$Y_1^N = \alpha_2 X_1^N + Z_1^N$$

$$Y_2^N = \alpha_2 X_2^N + Z_2^N$$

Perfect Rateless Codes

SNR thresholds

$$\mathrm{SNR}_1 = \mathrm{P}\alpha_1^2/\delta^2$$

$$\mathrm{SNR}_2 = \mathrm{P}\alpha_2^2/\delta^2$$

Channel capacity

$$\mathrm{I}_1 = \log\left(1 + \mathrm{SNR}_1\right)$$

$$\mathrm{I}_2 = 2\log\left(1 + \mathrm{SNR}_2\right)$$

Perfect Rateless Codes

The smallest of channel gain we can achieve occur when

$$I_1 = I_2 = R$$

Reference

[1] Uriz Erez, Mitchell D. Trott, Gregory W. Wornell “Rateless Coding for Gaussian Channels” IEEE Trans. Inform. Theory, vol. 58, pp. 530-547, Feb. 2012.

[2] M.G. Luby, “LT Codes,” The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.