Prioritized Luby Transform (LT) codes

Simon S. Woo
Michael K. Cheng

Jet Propulsion Laboratory, CA
California Institute of Technology, CA

CISS 2008
Princeton, NJ
March 20th 2008

JPL

Outline

- Introduction
- Background
- Prioritized LT Encoding Scheme
- Analysis and Results
- Conclusion and Future work

Luby Transform (LT) codes

- Luby invented rateless codes (fountain codes), where code rate (k/n) approaches to zero

- It is very efficient for correcting erasures with very small overheads as k increases

- Related work
  - Tornado Codes(2001)
  - Raptor Codes (2003)
  - Unequal error protection of LT codes has been considered in [5] (2007)

- Current work
  - Prioritized LT coding scheme extends the work from [5] and provide a systematic approach to enhance the likelihood of receiving high priority data while maintaining good overall decoding performance

Degree distributions and Encoding

- k : information bits
- n : encoded bits
- $r = n - k$: overhead
- $p_1, p_2, \ldots, p_k$: discrete probabilities describing the degrees so that $\sum_{i=1}^{k} p_i = 1$

Each encoded bit is generated as follows:
1) Select a degree (random variable), d, according to the degree distribution $\{p_1, p_2, \ldots, p_k\}$
2) Choose a d-element subset uniformly from $\{1,2,\ldots, k\}$
3) XOR the information bits in positions specified by 2)

Probability Distribution of degrees

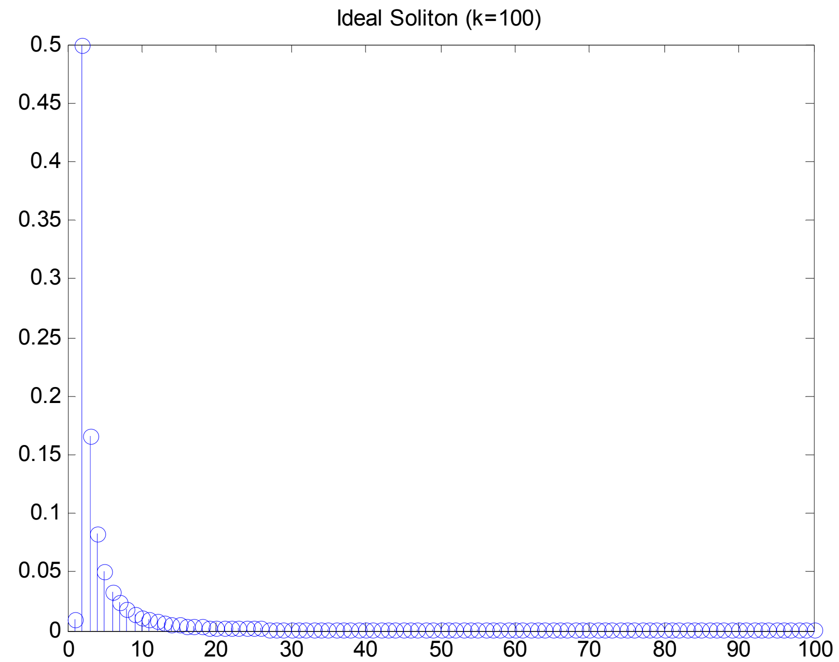
(Ideal Soliton Distribution)

$$\{\rho(1), \rho(2), \ldots, \rho(k)\}$$

$$\rho(1) = \frac{1}{k}$$

$$\rho(i) = \frac{1}{i \cdot (i-1)} = \frac{1}{i-1} - \frac{1}{i} \quad \text{for } i = 2,3,\ldots,k$$

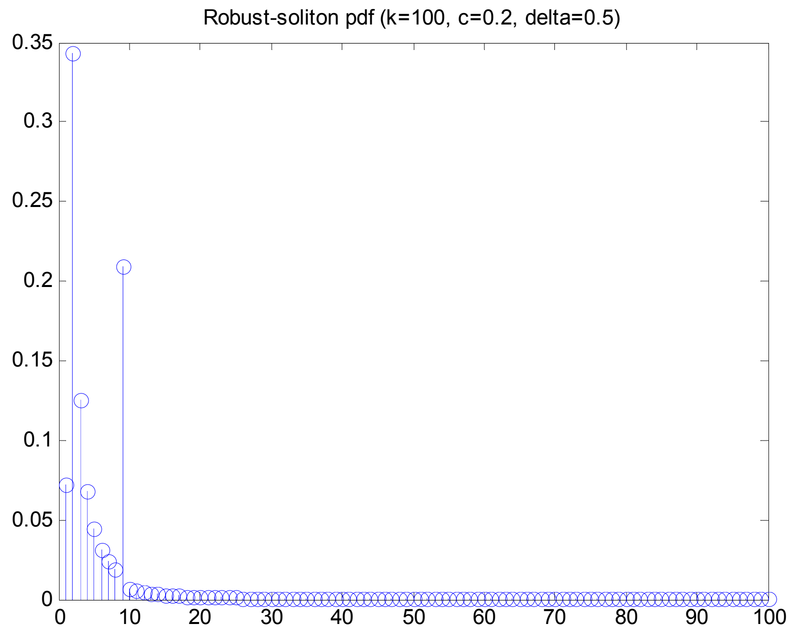
$$E[d] = \sum_{i=1}^{k} i \cdot \rho(i) = \sum_{j=1}^{k} \frac{1}{j} = H_k$$



Robust-Soliton Distribution

Let's define new probability distribution

for degrees



$$\{\mu(1), \mu(2), \ldots, \mu(k)\}$$

$$\mu(i) = \frac{\rho(i) + \tau(i)}{\beta} \quad \text{for } i = 1, 2, \ldots, k,$$

where

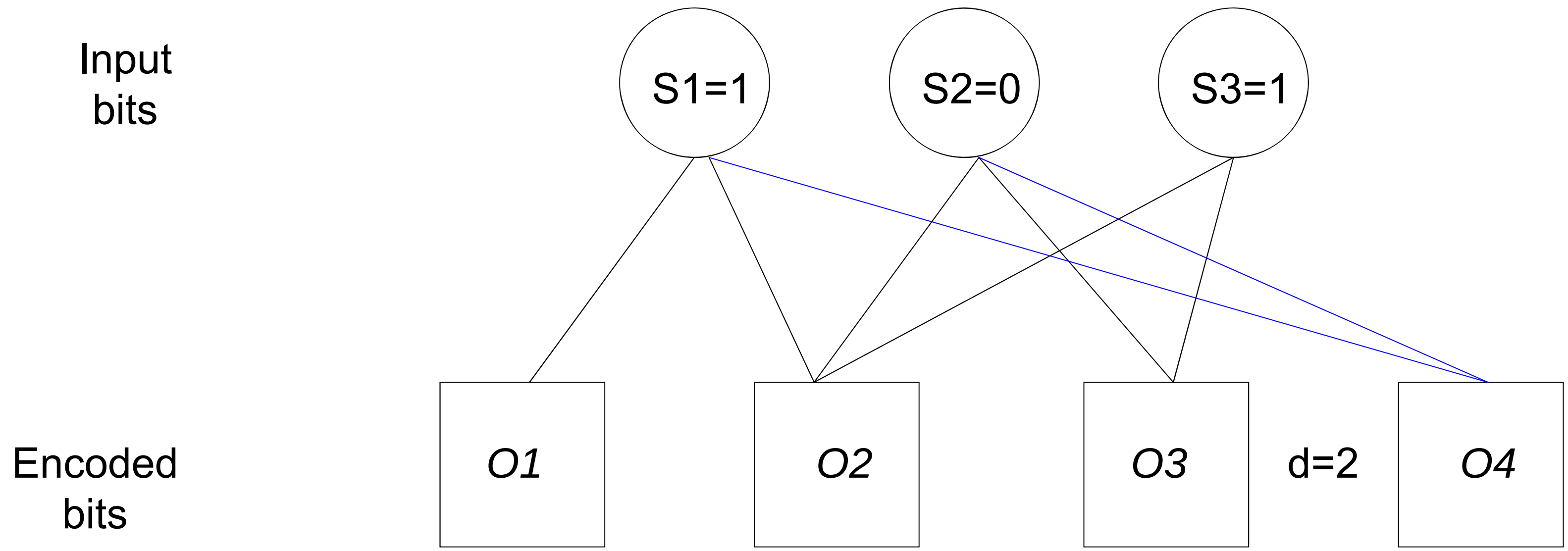
$$\beta = \sum_{i=1}^{k} \{\rho(i) + \tau(i)\},$$

$$\tau(i) = \begin{cases} (c\ln(k/\delta)\sqrt{k})/(ik), \text{for i} = 1, \ldots, \{k/(c\ln(k/\delta)\sqrt{k})\} - 1 \\ \{(c\ln(k/\delta)\sqrt{k})\ln(c\ln(k/\delta)\sqrt{k})/\delta\}/(ik), \text{i} = \{k/(c\ln(k/\delta)\sqrt{k})\} \\ 0, \text{i} = \{k/(c\ln(k/\delta)\sqrt{k})\} + 1, \ldots, k \end{cases}$$

- $\mu(1)$ does not vanish as much as $\rho(1)$ as *k* increases
- Also, increased probability of having higher degrees adds efficiency of decoding

Encoding LT codes

LT Encoding: $k = 3$, $n = 4$



O1=S1=1

O2 = (S1) +(S2)+(S3)=0

O3 = (S2) +(S1)=1

O4 = (S1) +(S2)=1

Encoded output = {1,0,1,1}

Outline

- Introduction
- Background
- Prioritized LT Encoding Scheme
- Analysis
- Results
- Conclusion and Future work

Priority LT codes Scheme

Goal : In communications, some of data are more important than other. If decoder fails, we want to decode high priority data with high reliability. LT codes is designed to maximize the overall decoding performance

Prioritized LT codes:

- Without losing the fidelity of overall decoding performance, we want to maximize the data return of certain portion of data.
- Redesign the "uniform bit selection" part to increase the likelihood receiving certain portion of data (science/command)
- Modify encoder only

Let's define the following parameters:

- h : the number of high priority bits
- $t = h/k$: fraction of high priority bits over total information bits
- $p_1, p_2, \ldots, p_k$: discrete probabilities using the *Robust-Soliton* distribution $\sum_{i=1}^{k} p_i = 1$
- $q_1, q_2, \ldots, q_h$: discrete probabilities describing the degrees so that $\sum_{i=1}^{h} q_i = 1$ for high priority bits using the *Robust-Soliton* distribution

Algorithm

Initialize *count* = 1

For $i \in \{1,2,\ldots, n\}$,

1) Select a random degree *d* according to the *Robust-Soliton* degree distribution, $\{p_1, p_2, \ldots, p_k\}$

if *d* = 1,

choose *d* uniformly from high priority data group $\{1,2, \ldots, h\}$

else if *d* = 2,

If *count* ≤ Ω

choose *d* uniformly from high priority data group $\{1,2,\ldots, h\}$ and

count = *count* + 1

else

choose *d* uniformly from entire set of information bits $\{1,2,\ldots, k\}$

else if *d* ≥ 3

choose *d* uniformly from entire set of information bits $\{1,2,\ldots, k\}$

2) XOR information bits obtained from 1) to generate x_i

Intuition:

-Assign lower degrees (degree 1 or 2) to high priority data so

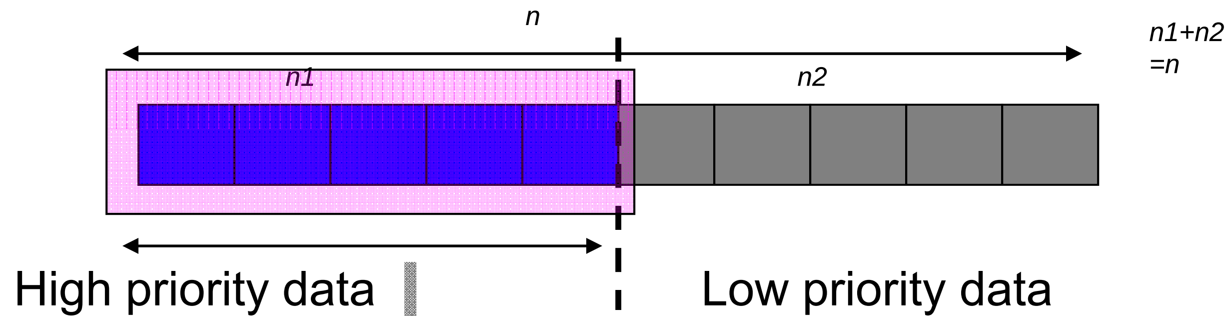
that decoding process can

start from high priority group

(Constraints on the selections of degree 1 or degree 2 encoded bits)

Illustration of Systematic degree calculation

Key Idea: Mix of regularity (high priority data) and non-regularity (low priority data) at the encoder



The number of degree 2 encoded bits impacts decoding performance significantly

Calculate degree distribution for high priority block (*n1*)

To maintain good high priority data decoding performance

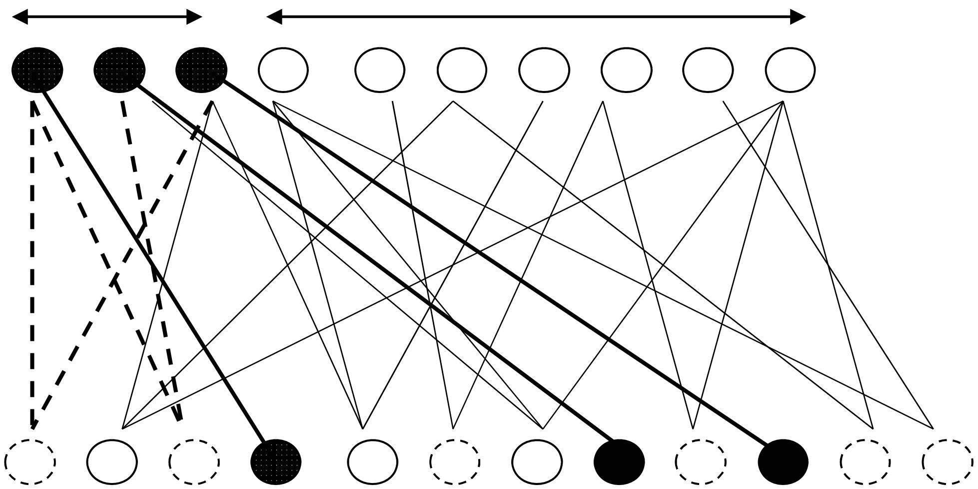
Calculate degree distribution for whole block (*n*)

To maintain good overall packet decoding performance

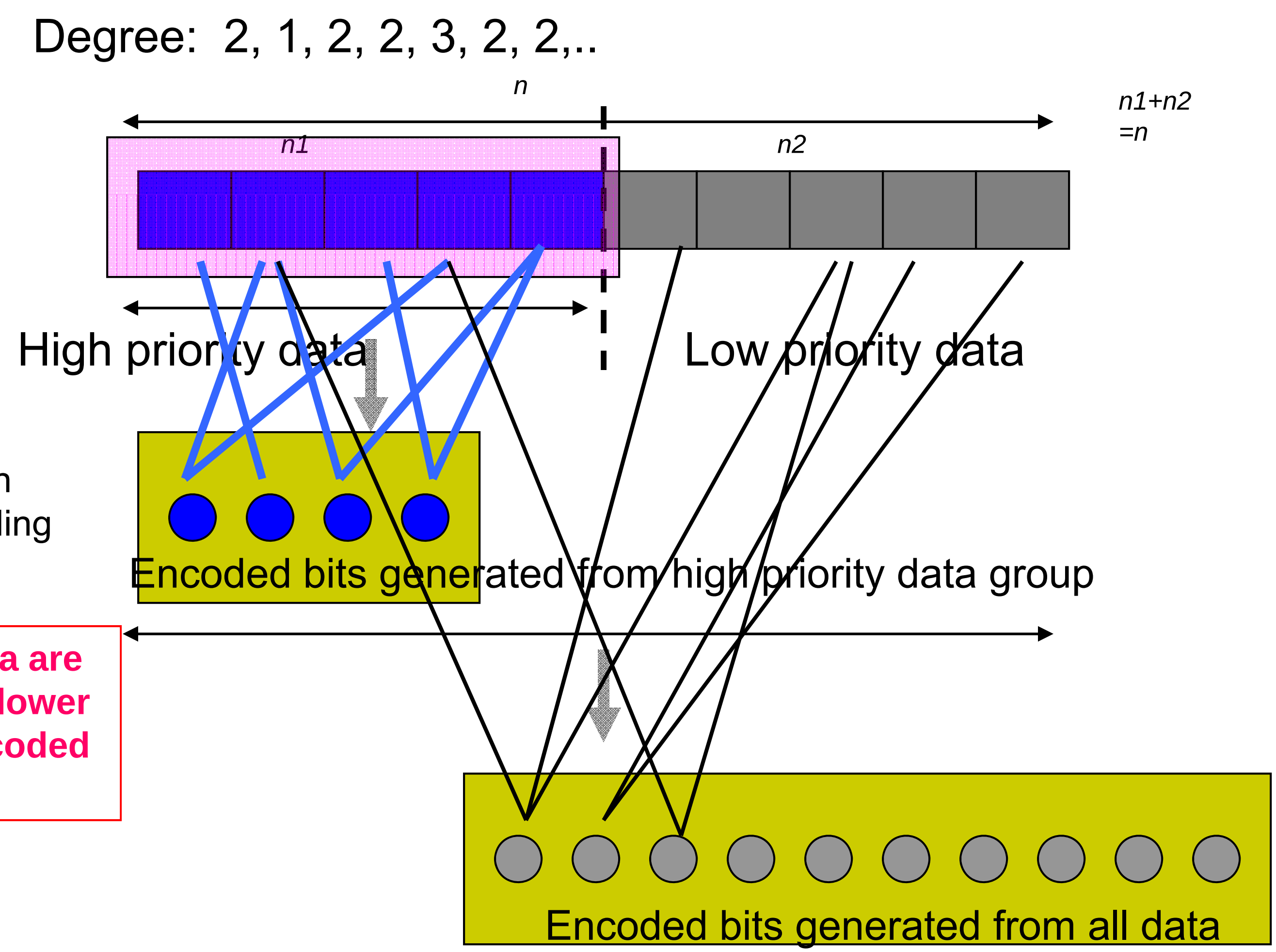
Enforce decoding high priority data first, by introducing non-uniformity while maintaining good full packet decoding performance

Final Encoded bits

k = 10, n = 12, h = 3, $\Omega = 2$



Encoder Structure



Maintain good high priority data decoding performance

High priority data are clustered at the lower degrees and decoded faster

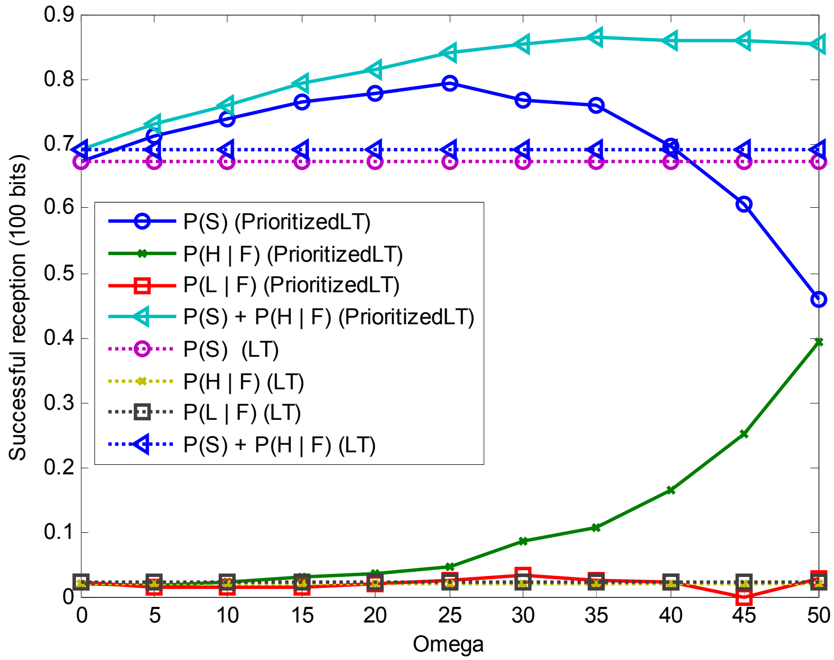
Maintain good overall packet decoding performance

Outline

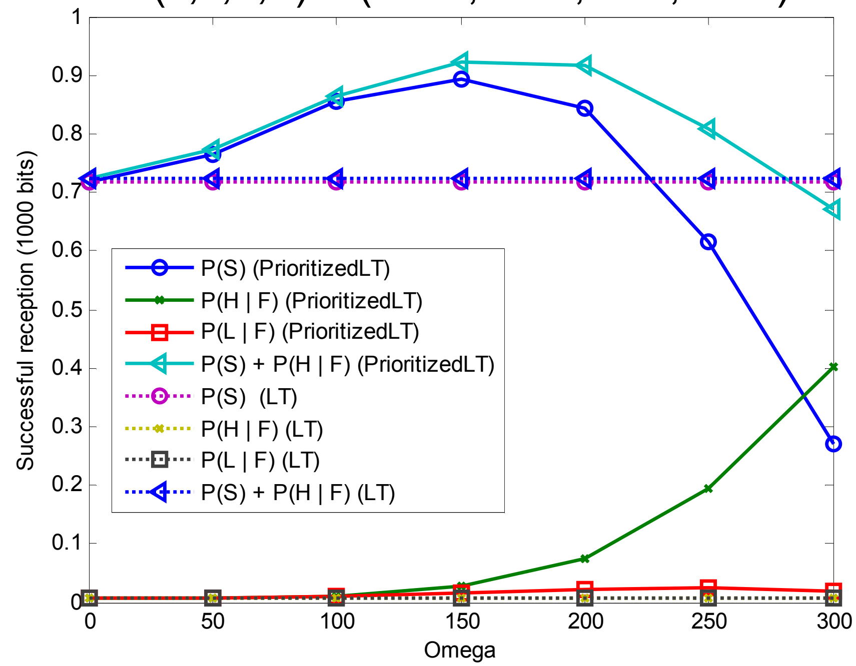
- Introduction
- Background
- Prioritized LT Encoding Scheme
- Analysis and Results
- Conclusion and Future work

Omega

(*n*,*k*,*h*,*e*) = (150,100, 40, 0.1)



(*n*,*k*,*h*,*e*) = (1200,1000, 400, 0.05)



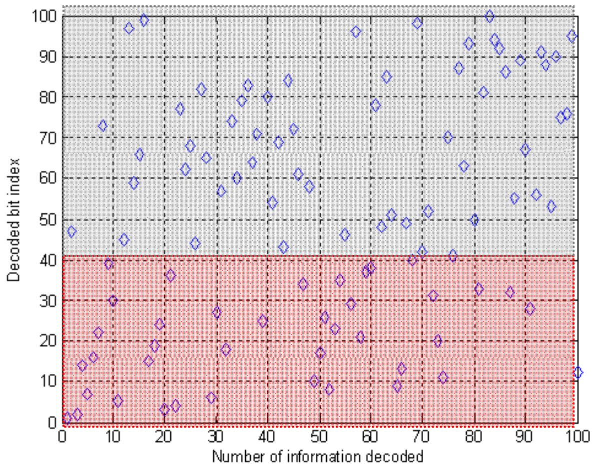
$\Omega \sim h/2 = k/4$

- *P* (successful decoding of all data) = *P* (*S*)
- *P* (decoding failure) = 1 - *P* (*S*) = *P* (*F*)
- *P*(successful decoding of high priority data) = *P*(*S*) + *P*(successful decoding of more than 90 percent of high priority data | decoding failures) = *P* (*S*) + *P* (*H* | *F*)
- *P*(successful decoding of low priority data) = *P*(*S*) + *P*(successful decoding of more than 90 percent of low priority data | decoding failures) = *P* (*S*) + *P* (*L* | *F*)

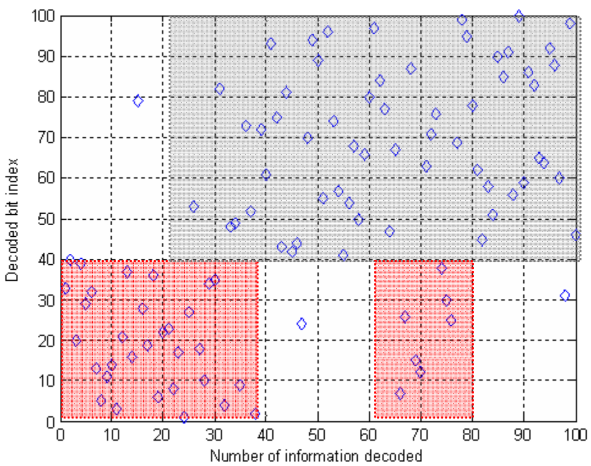
Successful Decoding Process

(n,k,h,e) = (150,100,40,0.01)

LT

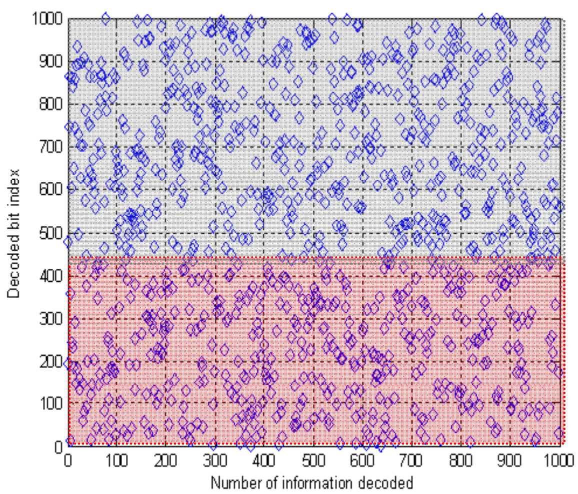


Prioritized LT

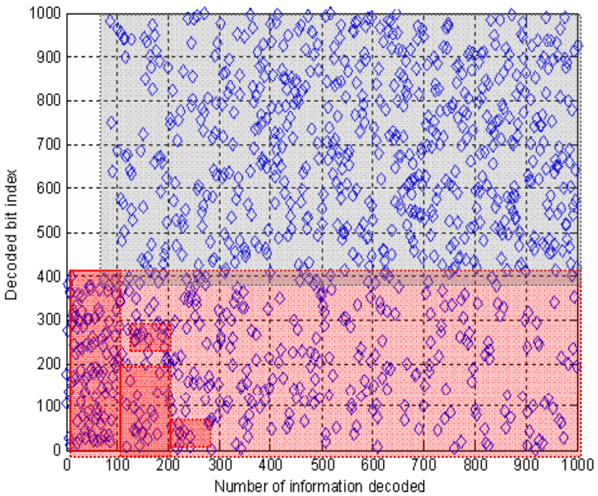


(n,k,h,e) = (1200,1000,400,0.01)

LT



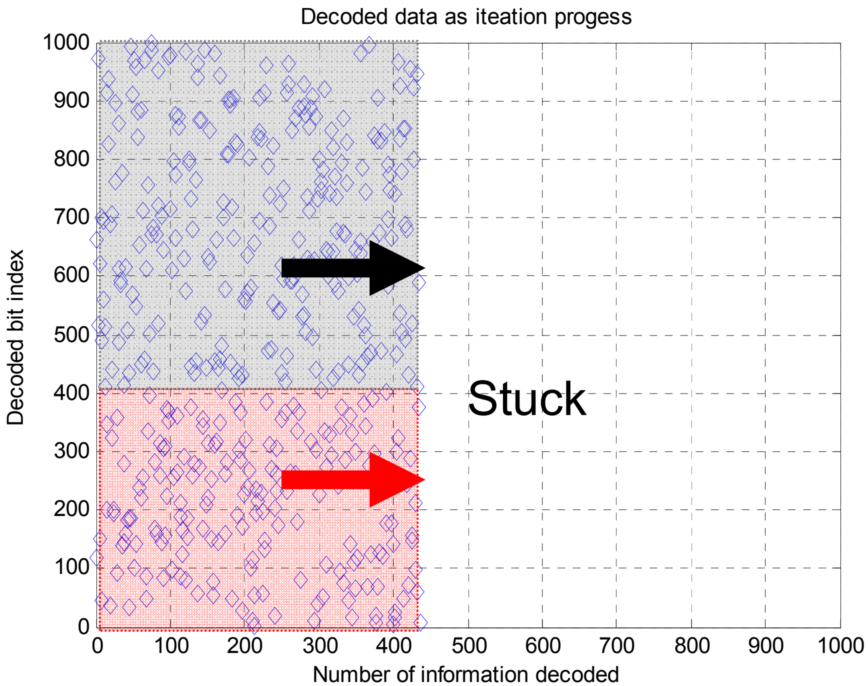
Prioritized LT



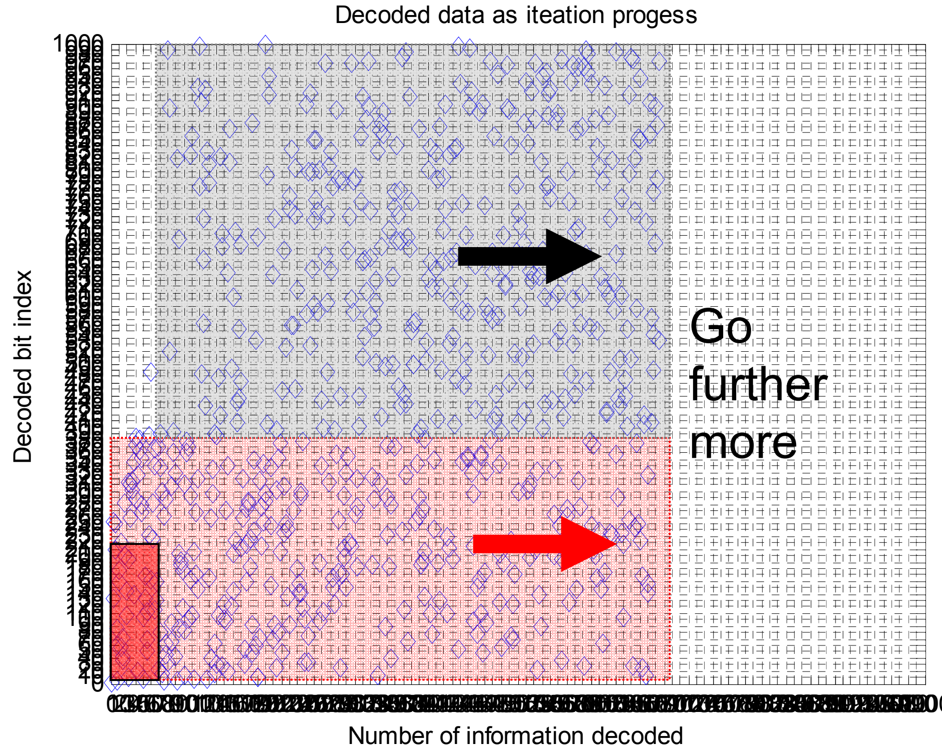
Decoding Failure

(*n*,*k*,*h*) = (1100,1000,400)

LT

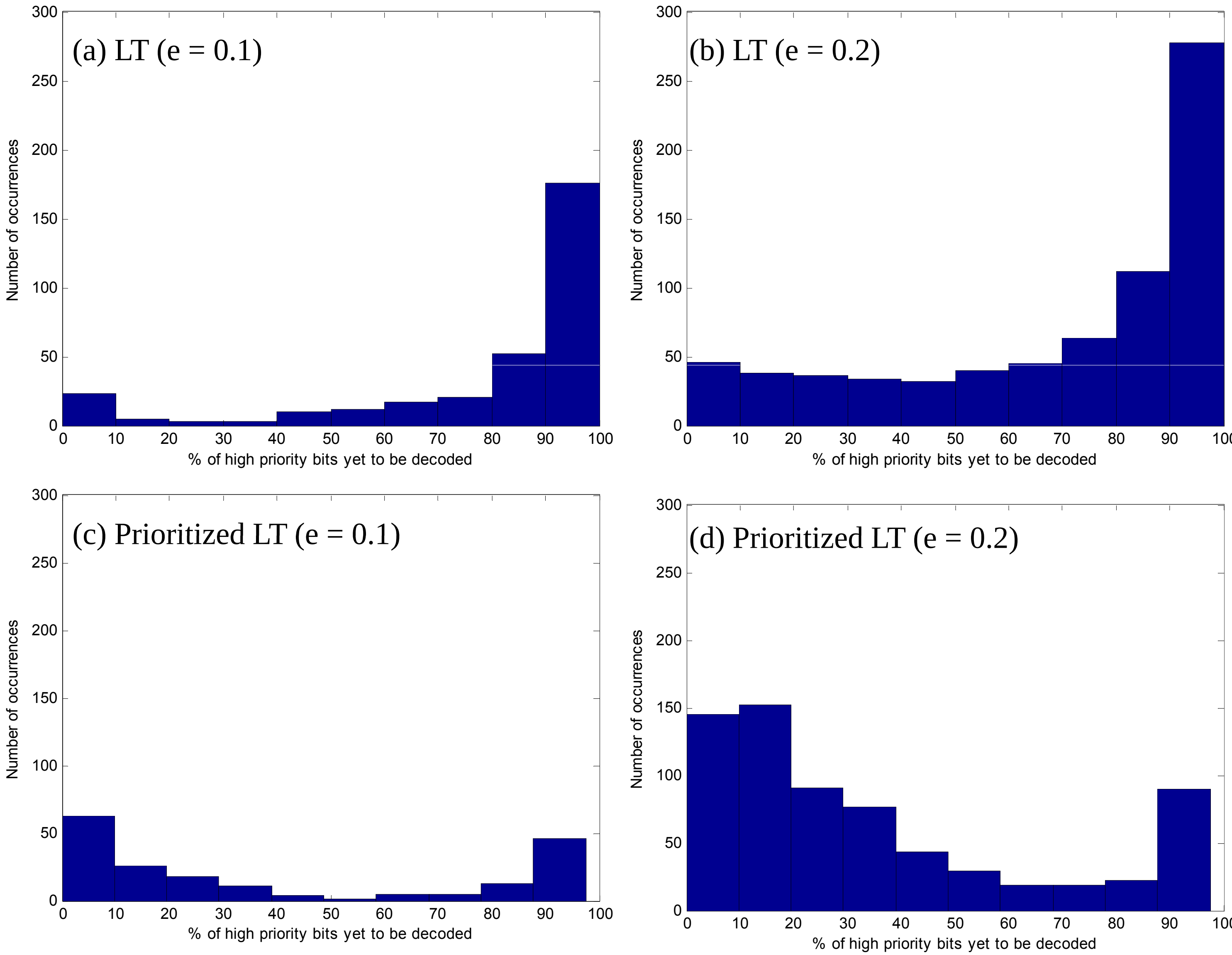


Prioritized LT



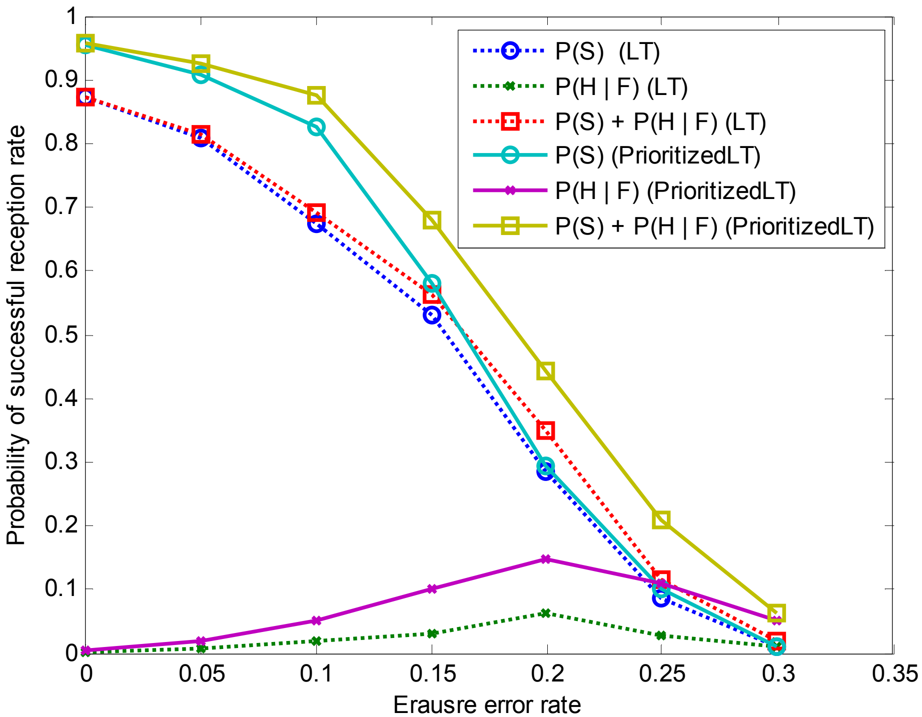
This means decoding more data

Failure distributions

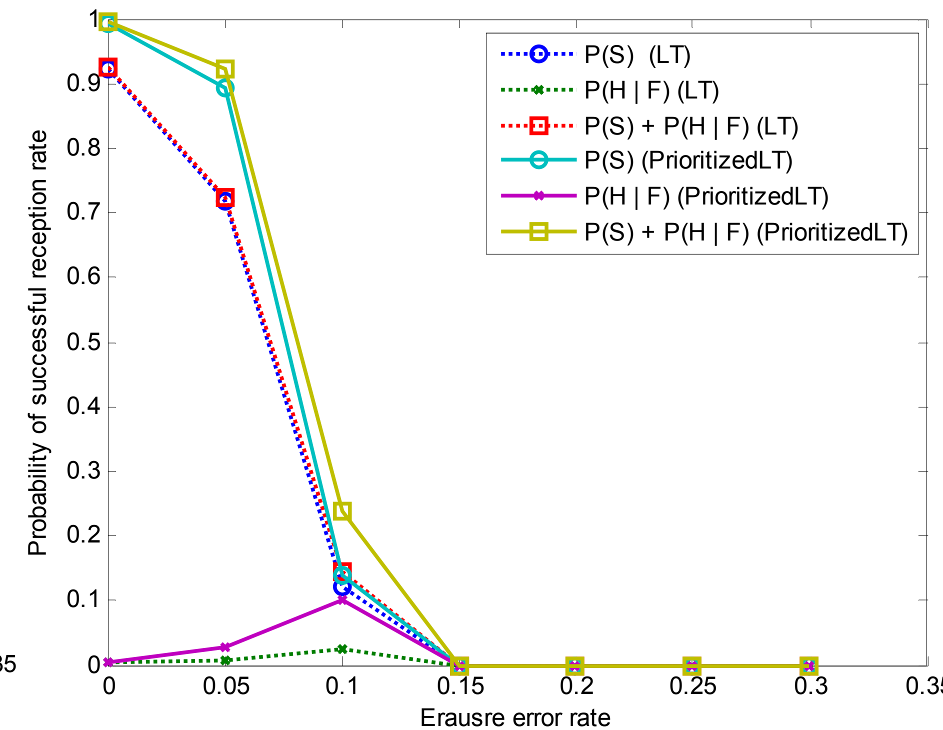


Overall decoding performance

(*n*,*k*,*h*)=(150,100,40)



(*n*,*k*,*h*)=(1200,1000,400)



Conclusion and Future work

- We obtained the experimental result using the sub-optimal algorithm to increase the likelihood of decoding high priority data (and it performs better than the original LT codes)
- Prioritized LT codes scales well up to high priority data size less than the 50% of total information bits
- This scheme requires modifications on LT encoder
- Potential benefits include sending video, audio, or any data types with high priority data
- LT codes can be combined with ARQ scheme and will be effective for channel with high bandwidths and long propagation delay (due to minimal feedback)
- This scheme works well for finite block LT codes but questionable for rateless approach
- Applying AWGN channel and used with other channel codes
- Rateless approach

References

[1] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, and Daniel A. Spielma n, “Efficient Erasure Correcting Codes,” IEEE Transactions on Information Theory, VOL. 47, NO. 2, FEBRUARY 2001

[2] Amin Shokrollahi, "Raptor Codes," IEEE Transactions on Information Theory, vol. 52, pp. 2551-2567, 2006.

[3] R. M. Karp, M. Luby and A. Shokrollahi, "Finite-Length Analysis of LT codes,” in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, pp. 39.

[4] E. Maneva, and A. Shokrollahi, “New Model for Rigorous Analysis of LT codes,” in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, Seattle, WA, Jul. 2006, pp. 2677-2679.

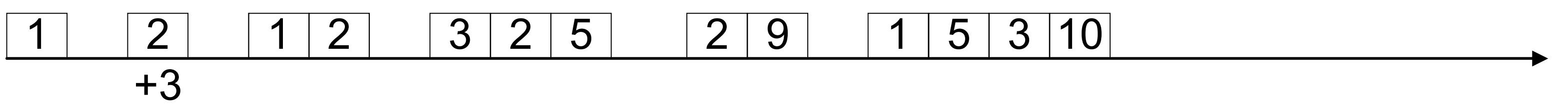
[5] N. Rahnavard, B. N. Vellambi, and F. Fekri, “Rateless Codes with Unequal Error Protection Property,” in *Proc. IEEE Trans. on Information Theory,* vol. 53, no4, pp.1 521-1532, Apr. 2007

Coupon Collectors Problem II

Different way of formulating problem

There are *k* different types of coupons one wants to collect and how many (XOR) bits need to be send to collect all coupons

(Hint: We do not throw away the same information)



Expected number of bits need to be send are drastically reduced by sending linear combinations !