Fountain Codes Based on Zigzag Decodable Coding

Takayuki Nozaki

Kanagawa University

28th Oct. 2014

Overview

Purpose of this study

Constructing fountain codes which have good performance when the number of source packets is finite.

Strength of the proposed system (overhead, space complexity of decoding)

- The proposed system outperforms the Raptor coding system in term of the overhead and space complexity of decoding for the fixed decoding erasure rate.

Drawback of the proposed system (time complexity of decoding)

- The time decoding complexity of proposed system is larger than the Raptor coding system.

Main Idea

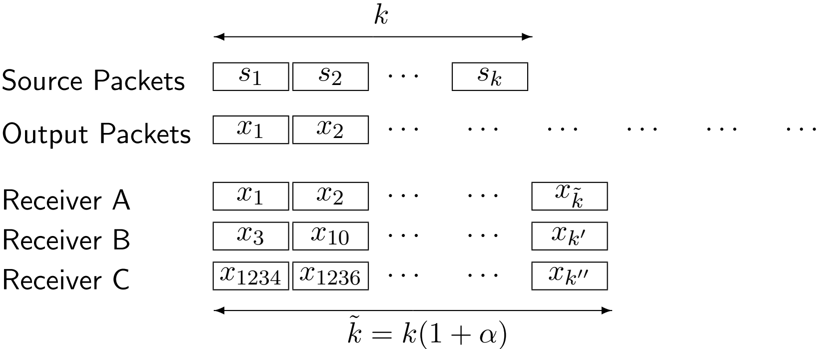
(Raptor code) + (Zigzag decodable code) $\Rightarrow$ Proposed fountain code

Outline

Fountain Code [Byers *et al.* 2002]

- Erasure decoding code
- Solving the retransmission problem on the multicast

Sender encodes k source packets and produces infinite output packets.
Receivers decodes the original message from *arbitrary* $k(1+\alpha)$ output packets.
$\Rightarrow$ Goal: Achieving small overhead α of received packets.



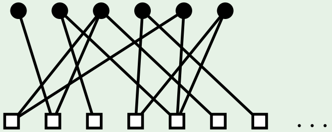
LT Codes ($\Omega(x)$) [Luby 2002]

First code realizing the concepts of the fountain code.

- Encoding
  1. Randomly choose a degree d according to the degree distribution $\Omega(x) = \sum_i \Omega_i x^i$
  2. Randomly choose d distinct source packets
  3. Output bit-wise XOR of the d distinct source packets as an output packet
- Decoding
  1. Construct the factor graph from the received packets
  2. Recover the source packets by using packet-wise peeling algorithm (PA)

Source Packets

Output packets

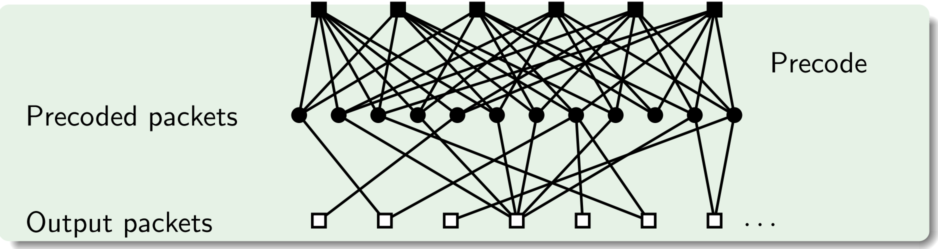


$$\boldsymbol{x}_1 = \boldsymbol{s}_3 + \boldsymbol{s}_5$$
$$= (s_{3,1} + s_{5,1}, s_{3,2} + s_{5,2}, \dots, s_{3,\ell} + s_{5,\ell})$$

Raptor Codes [Shokrollahi 2006]

Raptor code $(\mathcal{C}, \Omega(x))$

- Encoding
  1. Generate the precoded packets from the source packets by using precode $\mathcal{C}$
  2. Generate the output packets from the precoded packets by using LT code $(\Omega(x))$
- Decoding
  1. Construct the factor graph from the received packets and Tanner graph of $\mathcal{C}$
  2. Recover the source packets by using packet-wise PA



Zigzag Decodable Codes (1) [Gollakota & Katabi 2008]

- Originally proposed in the wireless communication to combat the hidden terminal problem
- Applied to the distributed storage system and network coding

Encoding

Generate the encoded packets from the source packets by using shift and bit-wise XOR

Source Packet	$s_{1,1}$	$s_{1,2}$	$\cdots$	$s_{1,\ell}$
	$s_{2,1}$	$s_{2,2}$	$\cdots$	$s_{2,\ell}$
Encoded Packet	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,\ell}$

$s_{1,1}$	$s_{1,2}$	$\cdots$	$s_{1,\ell}$	
	$s_{2,1}$	$\cdots$	$s_{2,\ell-1}$	$s_{2,\ell}$
$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,\ell}$	$x_{2,\ell+1}$

Zigzag Decodable Codes (2)

(Original) Zigzag Decoding

- The decoding starts only left of the encoded packets
- Solving linear equations with one unknown variable (like as PA).

Source Packet	$s_{1,1}$	$s_{1,2}$	$\cdots$	$s_{1,\ell}$		$s_{1,1}$	$s_{1,2}$	$\cdots$	$s_{1,\ell}$	
	$s_{2,1}$	$s_{2,2}$	$\cdots$	$s_{2,\ell}$			$s_{2,1}$	$\cdots$	$s_{2,\ell-1}$	$s_{2,\ell}$
Encoded Packet	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,\ell}$		$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,\ell}$	$x_{2,\ell+1}$

1. $s_{1,1} = x_{2,1}$
2. $s_{2,1} = x_{1,1} - s_{1,1}$
3. $s_{1,2} = x_{2,2} - s_{2,1}$

Modified Zigzag Decoding (bit-wise PA)

If the decoding starts left and right of the encoded packets, the decoding performance is improved.

Representation for the Zigzag Decodable Codes

- Polynomial representation of the packet $\boldsymbol{s}_i = (s_{i,1}, s_{i,2}, \dots, s_{i,\ell})$

$$s_i(z) = s_{i,1} + s_{i,2}z + s_{i,3}z^2 + \cdots + s_{i,\ell}z^{\ell-1}$$

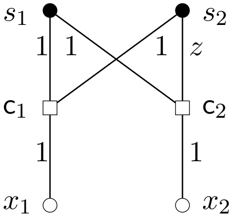
- Polynomial representation for the encoded packets

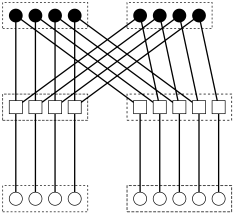
$$x_i(z) = \textstyle\sum_{j=1}^{k} g_{i,j}(z)s_j(z) \qquad (g_{i,j}(z) \in \{0, 1, z, z^2, \dots\})$$

- Matrix representation for the Zigzag decodable code

$$\boldsymbol{x}(z) = \boldsymbol{G}(z)\boldsymbol{s}(z)$$

- Factor graph representation for the Zigzag decodable code





$$G(z) = \begin{pmatrix} 1 & 1 \\ 1 & z \end{pmatrix}$$

Fountain Codes Based on Zigzag Decodable Coding (Encoding)

Proposed fountain code $\mathcal{F}(\mathcal{C}, \Omega(x), \Delta(x))$

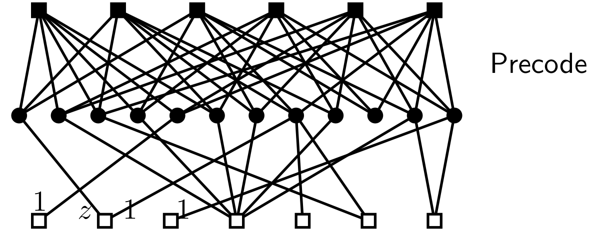
(Precoding) Generate the precoded packets $\boldsymbol{a}$ from the source packets $\boldsymbol{s}$ by using precode $\mathcal{C}$.

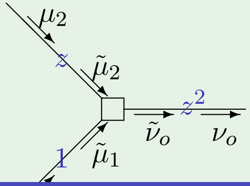
1. Choose a degree d according to the degree distribution $\Omega(x)$.
2. Choose d-tuple of shift amount $(\tilde{\delta}_1, \dots, \tilde{\delta}_d) \in [0, s_m]^d$ in independent of each other according to shift distribution $\Delta(x)$.
3. Define $\delta_i := \tilde{\delta}_i - \tilde{\delta}_{\min}$ for $\forall i$, where $\tilde{\delta}_{\min} := \min_i\{\tilde{\delta}_1, \dots, \tilde{\delta}_d\}$.
4. Choose d distinct precoded packets uniformly. Let $(j_1, j_2, \dots, j_d)$ denote the d-tuple of indexes of the chosen precoded packets.
5. Generate an output packet from the d distinct precoded packets as follows:

$$\sum_{i=1}^{d} z^{\delta_i} a_{j_i}(z).$$

Fountain Codes Based on Zigzag Decodable Coding (Decoding)

1. Generate the factor graph from the received packets and precode.
2. Execute bit-wise peeling algorithm





$\mu_1, \mu_2, \nu \in \{0, 1, ?\}^\ell, \quad ? + * = ?$

1. $\tilde{\mu}_1(z) = \mu_1(z), \quad \tilde{\mu}_2(z) = z\mu_2(z)$
2. $\tilde{\nu}_o(z) = \tilde{\mu}_1(z) + \tilde{\mu}_2(z) + x(z)$
3. $(\nu_{o,1}, \dots, \nu_{o,\ell}) = (\tilde{\nu}_{o,3}, \dots, \tilde{\nu}_{o,k+2})$

Performance Evaluation (1: Overhead)

Overhead α of received packets

$$\alpha = \frac{\#\text{of received packets}}{\#\text{of source packets}} - 1 = \frac{\tilde{k}}{k} - 1$$

Overhead β of received bits

$$\beta = \frac{\#\text{of received bits}}{\#\text{of source bits}} - 1 = \frac{\sum_{i=1}^{\tilde{k}} (\ell + \ell_i)}{k\ell} - 1$$

ℓ : the length of the source packets.

$\ell + \ell_i$: the length of the i-th received packet.

$$\beta = \alpha + \frac{1}{k\ell} \sum_{i=1}^{k(1+\alpha)} \ell_i.$$

For the Raptor codes: $\alpha = \beta$

For the proposed codes: $\alpha \leq \beta$

Performance Evaluation (2: Overhead)

[Proposition 1] Expectation of the length of the received packets

- L : Random variable representing a length of a received packet
- $\Omega(x) = \sum_i \Omega_i x^i$: Degree distribution
- $\Delta(x) = \sum_{i=0}^{s_m} \Delta_i x^i$: Shift distribution

$$\mathbb{E}[L] = \ell + s_m - \sum_{i=0}^{s_m - 1} \Omega(\Delta_{[0,i]}) - \sum_{i=1}^{s_m} \Omega(\Delta_{[i,s_m]}),$$

where $\Delta_{[i,j]} := \sum_{k \in [i,j]} \Delta_k$.

For a fixed α, the expectation of β is

$$(1+\alpha)\frac{\mathbb{E}[L]}{\ell} - 1$$

Since $\mathbb{E}[L] \leq \ell + s_m$, $\beta \to \alpha$ as $\ell \to \infty$.

Performance Evaluation (3: Decoding Erasure Probability)

[Theorem 1] Comparison of decoding erasure probability with Raptor code

$P_{\mathrm{B}}(\alpha, \mathcal{C}, \Omega, \Delta)$: decoding erasure probability for the proposed fountain code $\mathcal{F}(\mathcal{C}, \Omega(x), \Delta(x))$ at the overhead α

For arbitrary $\alpha, \mathcal{C}, \Omega(x), \Delta(x)$, the following holds

$$P_{\mathrm{B}}(\alpha, \mathcal{C}, \Omega, 1) \geq P_{\mathrm{B}}(\alpha, \mathcal{C}, \Omega, \Delta)$$

Note: The fountain code $\mathcal{F}(\mathcal{C}, \Omega(x), 1)$ is equivalent to the Raptor code $(\mathcal{C}, \Omega(x))$

[Corollary 1]

If Raptor code $(\mathcal{C}, \Omega(x))$ achieves $\alpha = 0$, then the proposed fountain code $\mathcal{F}(\mathcal{C}, \Omega(x), \Delta(x))$ also achieves $\alpha = 0$ for arbitrary $\Delta(x)$.

Performance Evaluation (4: Decoding Complexity)

Space Complexity

Space complexity is determined from the total number of factor nodes.
For both the proposed code and Raptor code, $k\ell(2+\beta)+n\ell$.

Time Complexity (Upper bound)

Time complexity is determined from the number of iteration in peeling algorithm.
For the Raptor code, $\alpha k + n$. (Since packet-wise PA)
For the proposed code, $\ell(\beta k + n)$. (Since bit-wise PA)

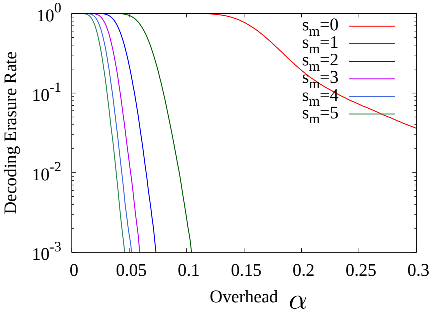
$\Rightarrow$ (main drawback)
The time complexity of the proposed code is larger than Raptor code.

Performance Evaluation (5: Simulation Results)

Settings

- Packet
  Length of packet ℓ : 100 (bits)
- Precode
  (3,30)-regular LDPC code
  The number of source packets: $k = 900, 1800, 3600$
  The number of precoded packets: $n = 1000, 2000, 4000$
- Degree distributions
  $\Omega(x) =$
  $0.007969x + 0.493570x^2 + 0.166220x^3 + 0.072646x^4 + 0.032558x^5 + 0.056058x^8 + 0.037229x^9 + 0.055590x^{19} + 0.025023x^{65} + 0.003135x^{66}$
  $\Delta(x) = \sum_{i=0}^{s_m} \frac{1}{1+s_m} x^i$

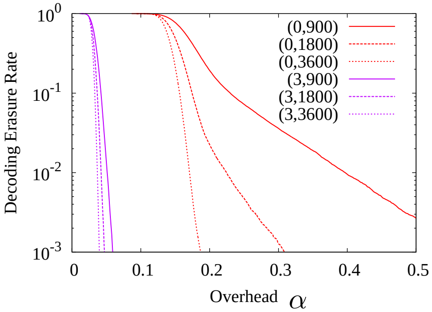
Performance Evaluation (6: Simulation Results)

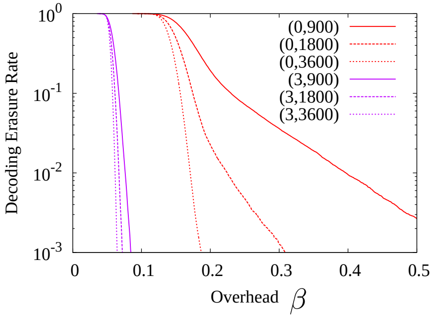


Number of source packets $k = 900$ $(n = 1000)$
Maximum shift amount $s_m = 0, 1, 2, 3, 4, 5$

Raptor code : $s_m = 0$
Proposed code : $s_m = 1, 2, 3, 4, 5$

Performance Evaluation (7: Simulation Results)





Number of source packets $k = 900, 1800, 3600$
Maximum shift amount $s_m = 0, 3$

(Remark)

The space complexity of decoding is $k\ell(2 + \beta) + n\ell$.

Conclusion and Future Works

Conclusion

- Proposing a fountain coding system based on zigzag decodable code
- The proposed fountain code outperforms the Raptor code in term of the overheads and space complexity of decoding

Future Works

- Propose an efficient decoding algorithm
- Analyze the decoding threshold