

Zigzag decodable 噴水符号の復号反復回数削減

野崎 隆之

山口大学

情報理論研究会

2016/1/18

概要

噴水符号

マルチキャスト向けの消失訂正符号

(応用例) ネットワークでの (動画) 放送

(既存研究) LT 符号, Raptor 符号

これまでの研究 [Nozaki2014]

Zigzag decodable (ZD) 噴水符号の提案

利点 Raptor 符号よりも, オーバヘッド・復号消失率・空間計算量が小さい

欠点 復号反復回数が多い (復号時間が大きい)

本研究の目的と結果

ZD 噴水符号の復号反復回数の削減

(結果) 復号性能を保ったまま復号反復回数の削減に成功

Outline

- 1 噴水符号
- 2 Raptor 符号
- 3 ZD 噴水符号・復号法
- 4 従来復号法の問題点と改善法

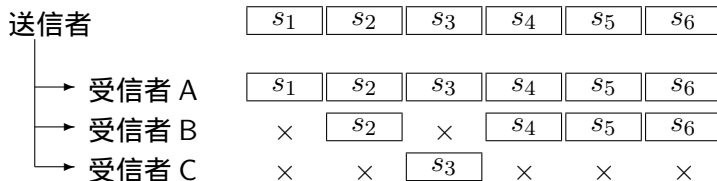
噴水符号 (1: マルチキャストとその問題点)

マルチキャスト

特定の複数のユーザにデータを送信する

問題点

- 送信者は受信者の再送要求に応じることができない
- 受信者によって通信路の状況が異なる

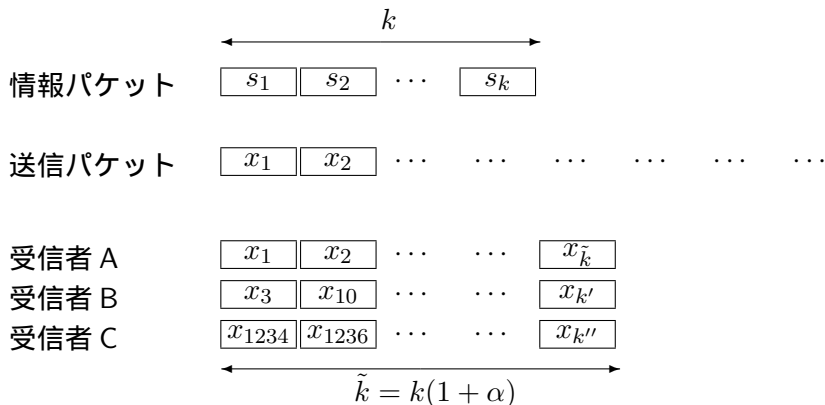


噴水符号 (2: 概要)

[送信者] k 個の情報パケットを符号化し、十分大きな数の送信パケットを作成

[受信者] 任意の $k(1 + \alpha)$ 個の送信パケットを受信し、復号

⇒ オーバヘッド $\alpha \cdot$ 復号消失率を小さくしたい



噴水符号 (3: 既存研究, Raptor 符号)

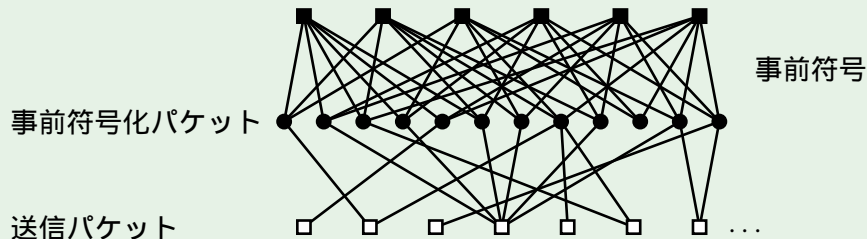
Raptor 符号 ($\mathcal{C}, \Omega(x)$)

■ 符号化

- 1 事前符号 \mathcal{C} を用いて, 情報パケットから事前符号化パケットを作成
- 2 事前符号化パケットを LT 符号 ($\Omega(x)$) で符号化し, 送信

■ 復号

- 受信パケットと事前符号のタナーグラフをもとにファクターグラフを作成し, パケット毎にピーリング (PA) 復号



ZD 噴水符号 (1: Zigzag decodable code [Gollakota 2008])

符号化

情報パケットに対して排他的論理和とシフト演算を用いる

情報パケット	<table border="1"><tr><td>$s_{1,1}$</td><td>$s_{1,2}$</td><td>\dots</td><td>$s_{1,\ell}$</td></tr><tr><td>$s_{2,1}$</td><td>$s_{2,2}$</td><td>\dots</td><td>$s_{2,\ell}$</td></tr></table>	$s_{1,1}$	$s_{1,2}$	\dots	$s_{1,\ell}$	$s_{2,1}$	$s_{2,2}$	\dots	$s_{2,\ell}$	<table border="1"><tr><td>$s_{1,1}$</td><td>$s_{1,2}$</td><td>\dots</td><td>$s_{1,\ell}$</td><td></td></tr><tr><td></td><td>$s_{2,1}$</td><td>\dots</td><td>$s_{2,\ell-1}$</td><td>$s_{2,\ell}$</td></tr></table>	$s_{1,1}$	$s_{1,2}$	\dots	$s_{1,\ell}$			$s_{2,1}$	\dots	$s_{2,\ell-1}$	$s_{2,\ell}$
$s_{1,1}$	$s_{1,2}$	\dots	$s_{1,\ell}$																	
$s_{2,1}$	$s_{2,2}$	\dots	$s_{2,\ell}$																	
$s_{1,1}$	$s_{1,2}$	\dots	$s_{1,\ell}$																	
	$s_{2,1}$	\dots	$s_{2,\ell-1}$	$s_{2,\ell}$																
送信パケット	<table border="1"><tr><td>$x_{1,1}$</td><td>$x_{1,2}$</td><td>\dots</td><td>$x_{1,\ell}$</td></tr></table>	$x_{1,1}$	$x_{1,2}$	\dots	$x_{1,\ell}$	<table border="1"><tr><td>$x_{2,1}$</td><td>$x_{2,2}$</td><td>\dots</td><td>$x_{2,\ell}$</td><td>$x_{2,\ell+1}$</td></tr></table>	$x_{2,1}$	$x_{2,2}$	\dots	$x_{2,\ell}$	$x_{2,\ell+1}$									
$x_{1,1}$	$x_{1,2}$	\dots	$x_{1,\ell}$																	
$x_{2,1}$	$x_{2,2}$	\dots	$x_{2,\ell}$	$x_{2,\ell+1}$																

復号法 (Zigzag 復号法)

- 符号化パケットの先頭から順に復号する。
- ビット毎のピーリング復号法とみなせる

1 $s_{1,1} = x_{2,1}$

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

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

ZD 噴水符号 (2: 符号化法)

提案噴水符号 $(\mathcal{C}, \Omega(x), \Delta(x))$

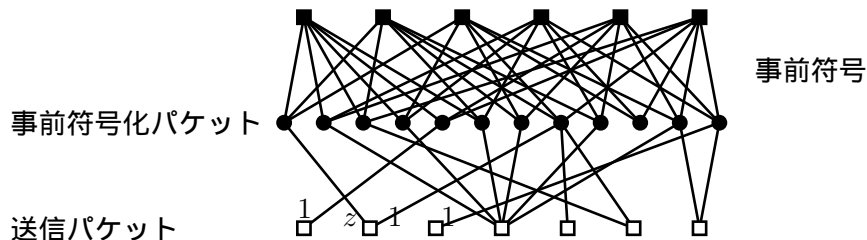
- 1 事前符号 \mathcal{C} を用いて, 情報パケット (s_1, s_2, \dots, s_k) から事前符号化パケット (a_1, a_2, \dots, a_n) を作成
- 2 次数分布 $\Omega(x)$ に従って送信パケットの次数 d を選ぶ
- 3 相異なる d 個の事前符号化パケットを一様に選び, そのインデックスを (j_1, \dots, j_d) と書く
- 4 シフト分布 $\Delta(x) := \sum_{i=0}^D \Delta_i x^i$ に基づき d 個のシフト量 $\delta_1, \dots, \delta_d$ を独立に選ぶ.
- 5 以下を送信パケットとして送信

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

パケットの多項式表現 $a_j(z) = a_{j,1} + a_{j,2}z + a_{j,3}z^3 + \dots + a_{j,\ell}z^{\ell-1}$

ZD 噴水符号 (3: 復号法)

受信者毎に，事前符号と受信パケットからファクターグラフを作成



復号法

- 1 パケット毎に PA 復号 (次数 1 の検査ノードから復号)
- 2 ビット毎に ZD 復号

数値実験による性能比較

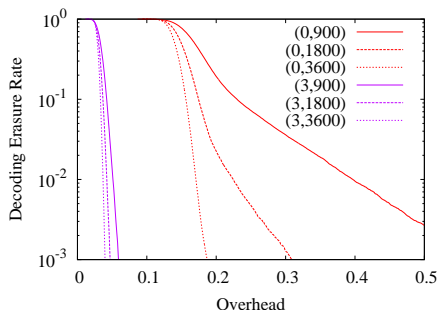


Table: 復号時間 [sec] の比較 ($\alpha = 0.12$)

	Raptor	ZD 噴水
$\ell = 100$	0.17039	0.43909
$\ell = 1000$	0.17039	14.0874

Figure: 復号性能の比較 (赤 : Ratpor 符号, 紫 : 提案法) $k = 900, 1800, 3600$

提案法の利点・欠点

利点 : オーバーヘッドと復号誤り率が低い

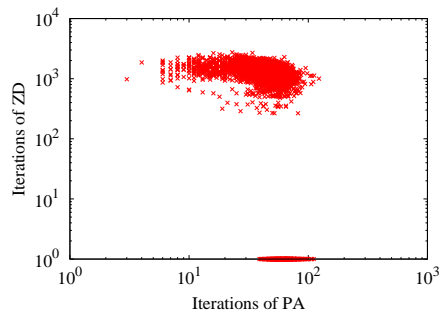
欠点 : 復号時間が大きい

復号法の改良

目標：復号誤り率を保ったまま復号反復回数を削減

原因と解決策：

- パケット毎の復号法による情報の復元が少ない
⇒ パケット毎の復号法を強力なものに変更する



反復回数の散布図

($\ell = 1000$, $\alpha = 0.12$)

反復回数

パケット毎の PA : 50 回程度

ビット毎の ZD : 1000 回程度

方針

ビット毎の復号の反復回数を削減

LDPC 符号の復号法

PA (Peeling Algorithm) [Luby et al. 1997]

- LDPC 符号に対するファクターグラフ上の復号アルゴリズム
- **次数 1** の検査ノードを利用
- **後退代入のみ** を利用した線形方程式の解法

TEP (Tree-structure Expectation Propagation) [Olmos et al. 2010]

- LDPC 符号に対するファクターグラフ上の復号アルゴリズム
- **次数 1 と 2** の検査ノードを利用
- 後退代入 + **部分的に前方消去** を利用した線形方程式の解法

G-TEP (Generalized TEP) [Salamanca et al. 2013]

- LDPC 符号に対するファクターグラフ上の復号アルゴリズム
- 全ての検査ノードを利用
- ガウスの消去法を利用した解法

研究のアイデア

復号法の比較

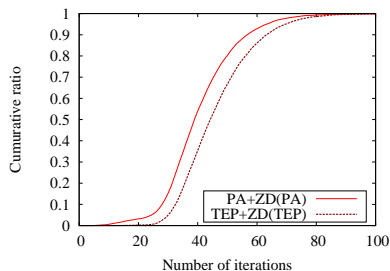
復号法	復号処理	利用するノード	性能	計算量	グラフの密度
PA	後退代入のみ	次数 1	小	小	減少
TEP	一部前進消去	次数 1,2	中	中	非増加
G-TEP	ガウスの消去法	任意	大	大	増加

研究のアイデア

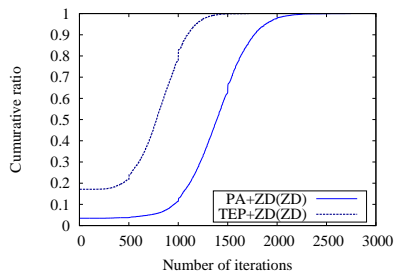
TEP 復号法を ZD 噴水符号のパケット毎の復号へ拡張

- 復号誤り率は劣化しない (PA より復号性能が高いので)
- ZD 復号の反復回数を減らせる
(パケット毎の復号で多く復元できるので)
- (欠点) 復号アルゴリズムがやや煩雑

数値実験 (1: 反復回数)



パケット毎の復号の反復回数
 $\ell = 1000, \alpha = 0.12$

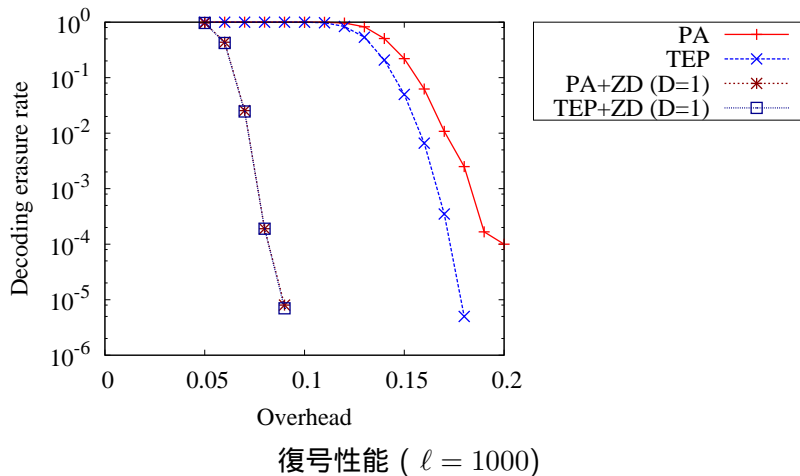


ビット毎の復号の反復回数

復号の反復回数の削減に成功

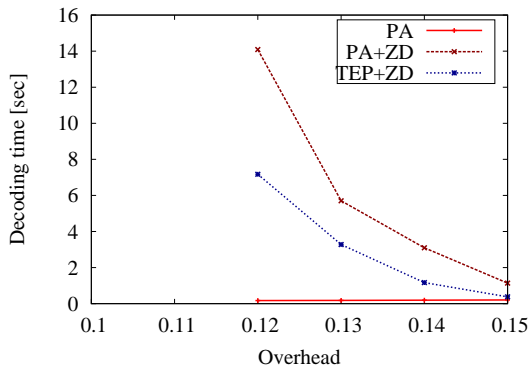
- パケット毎の復号は 5 回程度増加
- ビット毎の復号は 600 回程度減少

数値実験 (2: 復号性能)



- 復号性能に劣化がない

数値実験 (3: 復号時間)



復号時間 ($\ell = 1000$)

- 復号時間の削減に成功

まとめと今後の課題

まとめ

ZD 噴水符号に対して、復号時間の削減に成功した

- TEP 復号法を ZD 噴水符号へ拡張

今後の課題

ZD 噴水符号に関する研究

- 復号性能解析 (密度発展法)
- 次数分布 $\Omega(x)$ の最適化
- シフト分布 $\Delta(x)$ の最適化
- 復号法の更なる改良 (ZD 復号を変更)