Raptor Codes for use in Opportunistic Error Correction

Timon Zijnge

Twente University

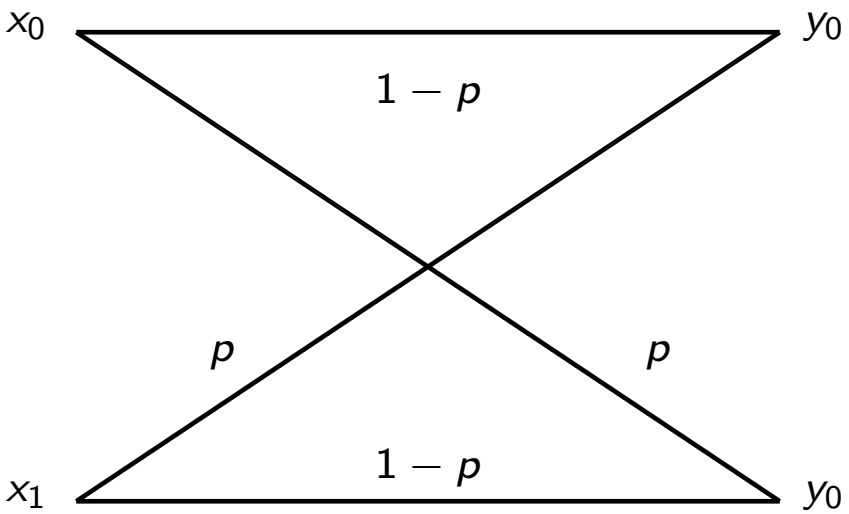
January 14, 2010

Introduction to error correction

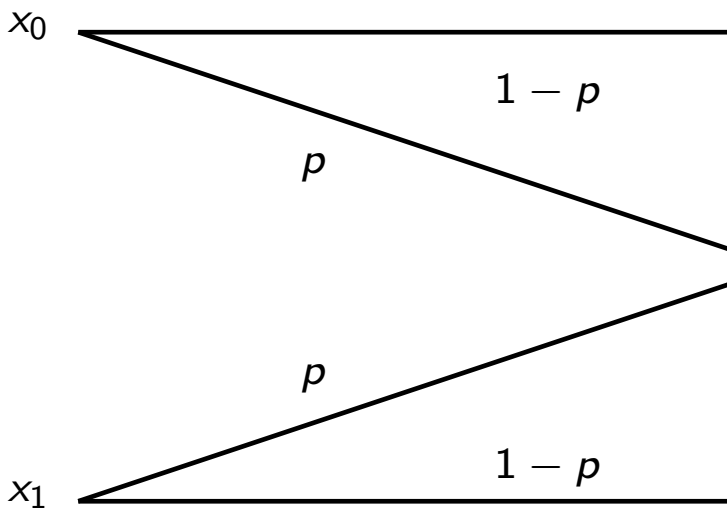
General model of a communication system.

Introduction to error correction

Binary Symmetric Channel



Binary Erasure Channe



Opportunistic Error Correction

- Retrieve information when the channel allows it
- For OFDM systems such as WLAN
- Reduction in receiver power consumption by using lower resultion ADC
- Based on Fountain codes

Outlook

- LDPC codes
- Fountain codes

LDPC codes

Linear block codes

Encoding

$\mathbf{t} = \mathbf{G}^T\mathbf{s}$

Decoding

$\mathbf{Ht} = \mathbf{0}$

$\mathbf{s}$ = Source message

$\mathbf{t}$ = Codeword

$\mathbf{G}$ = Generator matrix

$\mathbf{H}$ = Parity-check matrix

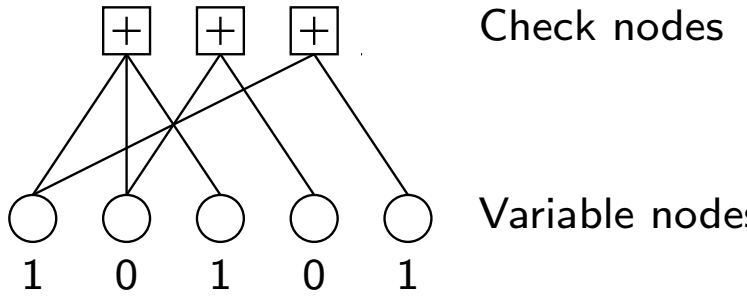
$\mathbf{GH}^T = \mathbf{0}$

Parity-check matrix

Matrix decoding ($\mathbf{Ht} = \mathbf{0}$)

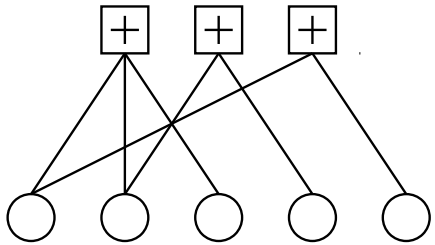
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Graph decoding



Error decoding using message passing

1. Initialization
2. Check nodes to variable nodes
3. Variable nodes to check nodes



Check nodes to variable nodes

$$r_{mn}^{0} = \sum_{x_{n'}:n' \in \mathcal{N}(m)\setminus n} P(z_m \mid x_n = 0, x_{n'} : n' \in \mathcal{N}(m)\setminus n) \prod_{x_{n'}:n' \in \mathcal{N}(m)\setminus n} q_{mn'}^{x_{n'}}$$

$$r_{mn}^{1} = \sum_{x_{n'}:n' \in \mathcal{N}(m)\setminus n} P(z_m \mid x_n = 1, x_{n'} : n' \in \mathcal{N}(m)\setminus n) \prod_{x_{n'}:n' \in \mathcal{N}(m)\setminus n} q_{mn'}^{x_{n'}}$$

Variable nodes to check nodes

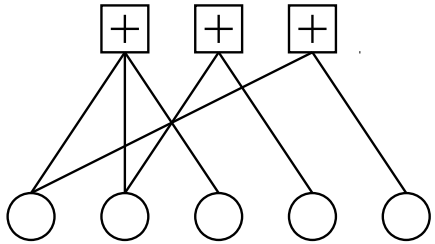
...

Decision

...

Error decoding using message passing

1. Initialization
2. Check nodes to variable nodes
3. Variable nodes to check nodes



Check nodes to variable nodes

...

Variable nodes to check nodes

$$q_{mn}^0 = \alpha_{mn} p_n^0 \sum_{m' \in \mathcal{M}(n) \setminus m} r_{m'n}^0$$

$$q_{mn}^1 = \alpha_{mn} p_n^1 \sum_{m' \in \mathcal{M}(n) \setminus m} r_{m'n}^1$$

Decision

$$q_n^0 = \alpha_n p_n^0 \sum_{m' \in \mathcal{M}(n)} r_{mn}^0$$

$$q_n^1 = \alpha_n p_n^1 \sum_{m' \in \mathcal{M}(n)} r_{mn}^1$$

Erasure decoding using message passing

- Same principle of sending messages between variable node and check nodes.
- Faster decoding because of simple binary calculations and exact knowledge of correctly received bits

LDPC properties

- Low decoding cost due to sparse parity-check matrix (few connections between nodes in the graph)
- Very efficient for large block lengths (near Shannon limit)
- Iterative decoding

General

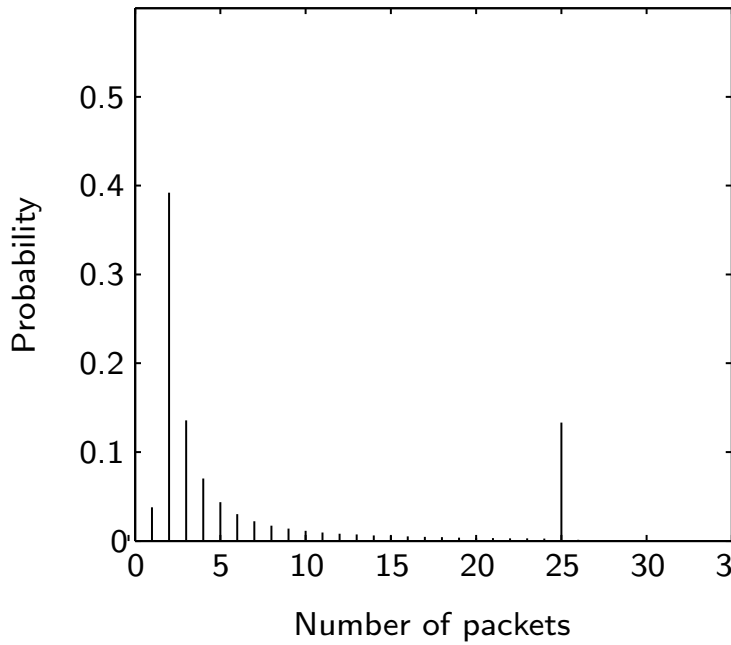
- Collecting drops of water from a fountain in a bucket
- Rateless
- Low overhead
- Useful in systems like many-to-one or one-to-many
- Useful in opportunistic error correction

LT codes

1. Divide source message into source packets
2. Sum a random number of randomly chosen source packets to create an encoded packet
3. Continue creating encoded packets until source file is recovered by receiver
4. Decode by summing encoded packets

LT codes

1. Divide source message into source packets
2. Sum a random number of randomly chosen source packets to create an encoded packet
3. Continue creating encoded packets until source file is recovered by receiver
4. Decode by summing encoded packets



LT codes

1. Divide source message into source packets
2. Sum a random number of randomly chosen source packets to create an encoded packet
3. Continue creating encoded packets until source file is recovered by receiver
4. Decode by summing encoded packets

Encoding:

$$t_1 = s_1 \oplus s_2 \oplus s_3$$
$$t_2 = s_1 \oplus s_2$$
$$t_3 = s_1$$

Decoding:

$$s_1 = t_3$$
$$s_2 = t_2 \oplus s_1$$
$$s_3 = t_1 \oplus s_1 \oplus s_2$$

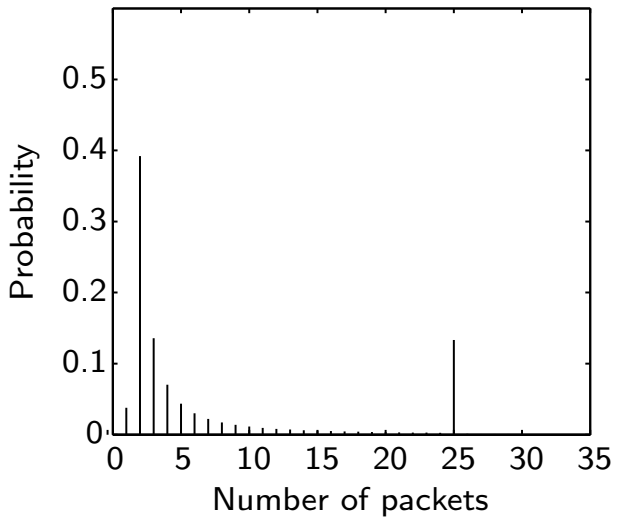
LT code properties

Picture of decoded packets vs received packets.

- Good performance for large block length (overhead goes to zero)
- Decoding cost scales with $k \log k$ for block length k

Raptor codes

We want the decoding cost to be less than $k \log k$
Solution: Use a truncated degree distribution



Raptor codes

Show images of LT performace and weakened LT performance Stop decoding after receiving x packets and use a pre-code to recover the packets not recovered by the weakened LT code

Raptor codes

Options for pre-codes:

- LDPC
- Reed-Solomon

LDPC has linear decoding cost. RS potentially has better erasure correcting capabilities, depending on the erasure distribution.
RS is a symbol code. A codeword is split up in symbols. A symbol is corrupted when one or more bits of the symbol are erased. Therefore RS performance is best for burst erasures, not for random erasures as one would expect in a Raptor code. It is not important which symbols are disturbed, as long as no more than a certain number are disturbed.

Results

Show table with results. discuss results, mention blocklength

Future work

- Compare results of LT codes and Raptor codes with respect to block length, packet delay, encoding and decoding complexity.
- Implement code on testbed

Questions