Coding theory for scalable media delivery

Michael Luby

RaptorQ is a product of Qualcomm Technologies, Inc.
Application layer erasure coding complements traditional error coding

Forward Error Correction Technology

“Error Coding”
Protects against Data Corruption

• Vast majority of current use of FEC
• Probably what you’re familiar with
• Typically applied at layers 1 or 2
• Usually performed in hardware
• PHY-FEC (physical layer FEC)

“Erasure Coding”
Protects against Data Lost in Transmission

• Commercial application relatively new
• Applied above layer 2
• Complement to Error Coding
• Typically performed in software
• AL-FEC (application layer FEC)
Packet transmission

Stream of packets

Received corrupted packet is discarded

Can identify received packet payloads from packet headers
Application Layer erasure codes

Source data

Erasure encode

Packetize

Transmit

Depacketize

Erasure decode

Source data
AL-FEC and PHY-FEC are complementary

**AL-FEC**
Packet loss protection over small to large block
Flexible time diversity
Flexible amount of protection

**PHY-FEC**
Correct or discard corrupted packet data over small block
Fixed time diversity
Fixed amount of protection
AL-FEC and PHY-FEC working together

- PHY-FEC corrects noise and interference
- AL-FEC “interleaves” and corrects erasures
  - Longer block length (“interleavers”) → better performance

![Diagram showing AL-FEC and PHY-FEC interaction](image-url)

- AL-FEC
  - CR = 0.8
  - 4.3 Mbit/s
  - p=0%

- PHY-FEC works
- 16QAM
  - CR = 1/3
  - 5.3 Mbit/s
  - p=14%

- PHY-FEC fails
What is a fountain code?

- Generate as much encoding as desired
- Recover source from the minimal possible encoding
  - It doesn’t matter what is received or lost
  - It only matters that enough is received
Fountain codes – erasure codes without a rate

- Fountain codes have no predetermined rate

- For fountain codes, for a fixed source data size
  - Erasure code design is extendable to provide any code rate
  - All code rates use the same extendable erasure code design
  - Particular encoded symbols are generated independently of one another
  - Number of encoded symbols that can be generated on the fly is unconstrained
LT encoding

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source block</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Degree</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Degree Distribution
LT encoding

Source block

Choose 2 random source symbols

XOR source symbols

Choose degree = 2

Insert header, and send

<table>
<thead>
<tr>
<th>Degree</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Degree Distribution
LT encoding

Source block

Choose 1 random source symbol

Copy source symbol

Insert header, and send

Choose degree = 1

Degree Distribution

<table>
<thead>
<tr>
<th>Degree</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Degree

Prob
LT encoding

Choose 4 random source symbols

XOR source symbols

Insert header, and send

Degree Distribution

<table>
<thead>
<tr>
<th>Degree</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Choose degree = 4

Source block
Belief propagation decoding

Collect enough encoded symbols and set up graph between encoded symbols and source symbols to be decoded.
Belief propagation decoding

Identify encoded symbol with one unrecovered neighbor
STOP if none exists
Belief propagation decoding

Unrecovered source symbol value is the value of all recovered neighbors XORed into the encoded symbol value.
Belief propagation decoding

Identify encoded symbol with one unrecovered neighbor
STOP if none exists
Belief propagation decoding

Unrecovered source symbol value is the value of all recovered neighbors XORed into the encoded symbol value
Belief propagation decoding

Identify encoded symbol with one unrecovered neighbor
STOP if none exists
Belief propagation decoding

Unrecovered source symbol value is the value of all recovered neighbors XORed into the encoded symbol value.
Belief propagation decoding

Source Block (recovered)
Intuition for Soliton degree distribution

- Consider a symbol of degree $d$
  - Releases when exactly 1 of its $d$ neighbors remains unrecovered
  - Degree $d$ releases when $1/d$ fraction of the symbols remain to be decoded

- A probability distribution on degrees so release distribution is uniform
  - $p(d)$ “covers” the interval $1/(d-1)$ to $1/d$ of the uniform distribution
  - Length of interval $1/(d-1)$ to $1/d$ is $\frac{1}{d \cdot (d-1)}$
  - For $d = 2, 3, ...$

\[ p(d) = \frac{1}{d \cdot (d-1)} \]
Choosing from Soliton distribution

\[ y \in \mathbb{R} [0,1] \]

\[ d = \begin{cases} 
\left\lfloor \frac{1}{y} \right\rfloor & \text{if } y \geq \frac{1}{k} \\
1 & \text{if } y < \frac{1}{k}
\end{cases} \]
Raptor codes in standards

- Raptor codes (IETF RFC 5053, 3GPP, DVB, ITU, ATIS)
  - Systematic fountain codes
  - Linear time encoding and decoding
  - Standardized – 3GPP MBMS, DVB-H IPDC
  - Good recovery properties – like a random code over GF(2)
  - Good flexibility
    - Up to 8,192 source symbols
    - Up to 65,384 source + repair symbols

- RaptorQ codes (IETF RFC 6330)
  - Systematic fountain codes
  - Linear time encoding and decoding
  - Great recovery properties – like a random code over GF(256)
  - Great flexibility
    - Up to 56,403 source symbols
    - Up to 16,777,216 source + repair symbols (essentially unlimited)
Major *technical* features (first appearance)

- LT code (Raptor RFC 5053)
  - fountain property
- Pre-coding (Raptor RFC 5053)
  - linear time
- Inactivation decoding (Raptor RFC 5053)
  - linear time
- Systematic construction (Raptor RFC 5053)
  - encoding includes original source
- Larger finite fields (RaptorQ RFC 6330)
  - reduced reception overhead
- Permanent inactivations (RaptorQ RFC 6330)
  - reduced reception overhead

*Raptor Codes. Foundations and Trends in Communications and Information Theory*  
A. Shokrollahi, M. Luby  
In comparison to other typical alternative FEC technologies, Raptor codes are an order of magnitude or more less complex.
Raptor codes overhead

Raptor code (IETF RFC 5053)
RaptorQ code (IETF RFC 6330)

- $k$ = number of source symbols in source block
- Valid for all supported values of $k$
- Valid for all loss probabilities: 1% to 99%
3GPP LTE Broadcast (eMBMS) Service Layer

**LTE Broadcast**

- **Reliable object delivery using FEC codes**
  - IETF RMT Suite
  - FLUTE, ALC PI, LCT BB, FEC BB, Raptor
- **Video streaming delivery**
  - MPEG-DASH Suite
- **Broadcast file delivery services**
- **Broadcast streaming services**

**HTTP Adaptive Streaming**

- MPEG-DASH Suite
  - HTTP 1.1
Why Fountain codes?
Some applications that you may (not) have thought about
Mobile File Delivery Services Over Cellular Network

LTE broadcast (eMBMS)

Challenge: reliable file delivery to mobile devices

Applications:
• Streaming
• Delivery of popular content
  • Media
  • Games

 Receivers
• Millions of mobile devices
LTE broadcast offload service for HTTP

ONLY broadcast repair symbols

Content
(10 source symbols)

Device

Receive 16 symbols

Decoding not possible
File download completion using HTTP 1.1 byte range requests

Origin Server

On Demand Caching

HTTP edge cache server

Device 1

Device 2

Device 3

Received 5 repair symbols
Need 5 additional symbols to decode

Received 9 repair symbols
Need 1 additional symbol to decode

Received 7 repair symbols
Need 3 additional symbols to decode

Request 5 symbol prefix of content

Request 1 symbol prefix of content

Request 3 symbol prefix of content

Decode

Decode

Decode
LTE broadcast offload service for HTTP

Content package

30 symbols total

Device 1

Media content A  Media content B  Media content C

Device 2

Media content A  Media content B  Media content C

Device 3

Media content A  Media content B  Media content C

ONLY broadcast repair symbols

10 symbols
Just-in-Time recovery

- LTE broadcast data stored directly on the SD card
  - Original multimedia data is never stored on the SD card
  - SD card stores one copy of file – not two!

- Multimedia content available immediately after reception
  - Avoids FEC decode post-processing of file after reception

- Just-in-Time recovery
  - Based on user actions – player requests data to playback multimedia content
  - Relevant data read from SD card, FEC decoded, provided directly to player
  - Trick play response time
    - Playback starts after one sub-block of data read from SD card and FEC decoded
    - Size of sub-block and decode speed determines the response time
Just-in-Time recovery avoids costly post-processing
Portions of multimedia never played back are never processed
Storage usage is minimized – avoids double the writes to SD card
Just-in-Time recovery advantages

Good user experience – media available immediately after reception
Minimizes UE CPU, I/O, and storage resources
Just-in-Time recovery demo

- **Demonstrates**
  - Just-in-Time recovery (importance of sub-blocking to support this)
  - Only repair symbols sent in the original broadcast session
    - Provides ability to efficiently combine with HTTP-based repair service
    - Provides ability to provide broadcast/HTTP hybrid services

- **Demo parameters**
  - Elephant’s Dream – 91.3 MB file
  - Partitioned into 10 source blocks (each of size 9.13 MB)
    - Provides reasonable network efficiency
  - Each source block is partitioned into 41 sub-blocks (each of size 223 KB, Symsize = 36 bytes, K ~ 6300)
    - Read encoded data from SD card and decode sub-block when requested by app
  - Original broadcast session
    - Transmit repair symbols only
    - 20% packet loss according to Markov model applied before reception
Thank you

Follow us on:  

For more information on Qualcomm, visit us at:  
www.qualcomm.com & www.qualcomm.com/blog

For more information on Raptor products or to request your free RaptorQ™ Evaluation Kit please go to www.qualcomm.com/raptor

For questions please email <raptorsupport@qti.qualcomm.com>

© 2013-2014 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved. Qualcomm, Snapdragon, Gobi, and RaptorQ are trademarks of Qualcomm Incorporated, registered in the United States and other countries. All Qualcomm Incorporated trademarks are used with permission. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable.

Qualcomm Incorporated includes Qualcomm’s licensing business, QTL, and the vast majority of its patent portfolio. Qualcomm Technologies, Inc., a wholly-owned subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of Qualcomm’s engineering, research and development functions, and substantially all of its product and services businesses, including its semiconductor business.