Erasure Codes in Software

With applications for Online Gaming

Chris Taylor, MSEE
mrcatid@gmail.com
March 14, 2014

Overview

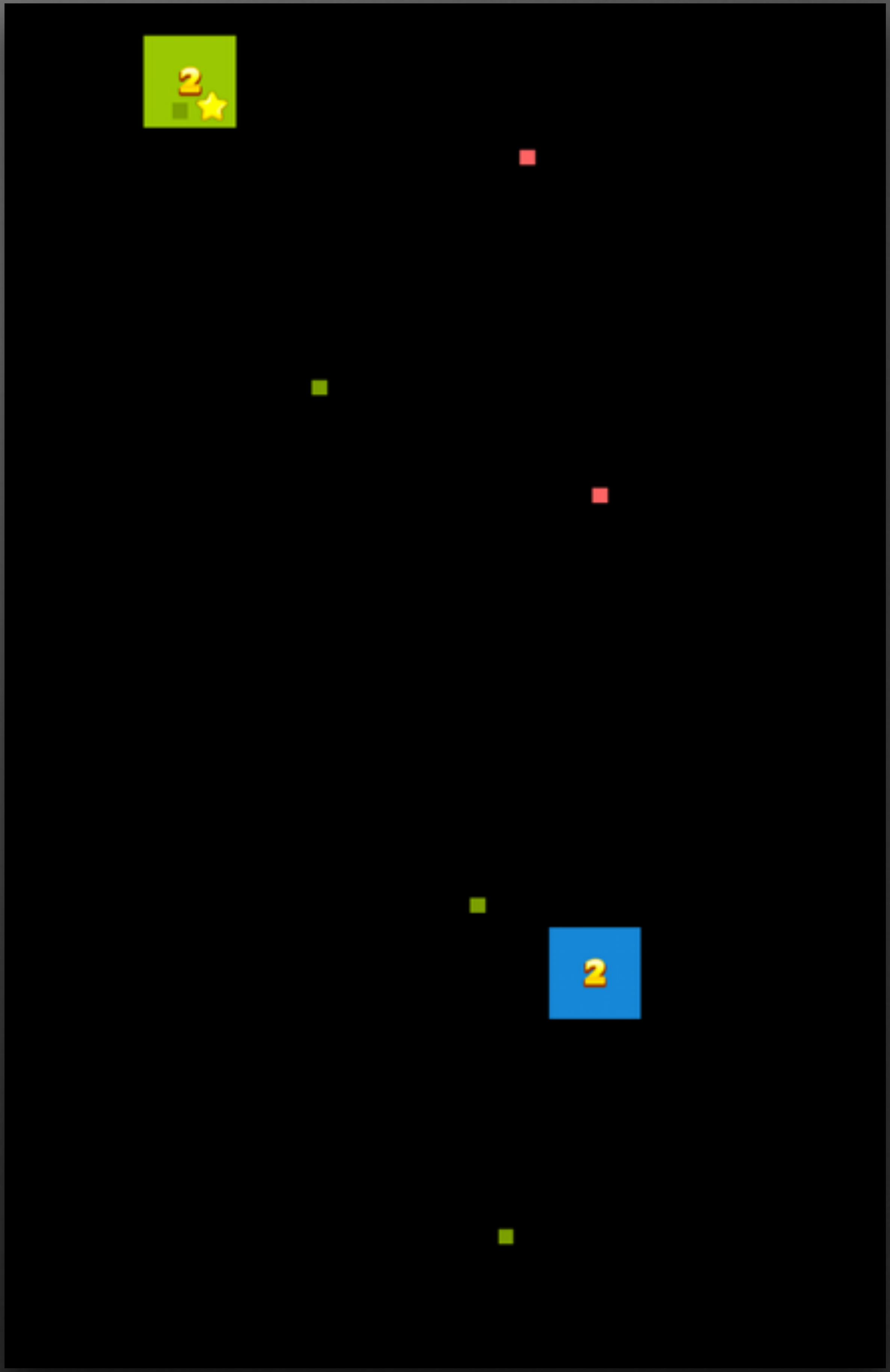
- Erasure codes in software are fast with low-overhead
- Advantages of using erasure codes in online games:
  - > 50% faster delivery over UDP than over TCP
  - > 50% less bandwidth used than naive redundancy
  - > 50% better recovery rate than parity redundancy
- Improves the experience for multiplayer mobile apps

Squares
Android Game

http://dkop.us/squares.apk

(built with the Game Closure DevKit)

See: “Realtime Multiplayer Game”
Think: “Low-Latency”



Game Packet Types

- **Unordered, Unreliable** (UDP)
  - Example: Time synchronization
  - Loss recovery? Not desired.
- **Ordered, Unreliable** (UDP)
  - Example: Avatar Position updates
  - Loss recovery? Just send another update!

- **Unordered, 99% Reliable** (UDP)
    - Examples: Bomb fired, Voice chat
    - *Erasure codes are useful in this case!*
- **Ordered, Reliable** (TCP)
    - Examples: Chat messages, File downloads
    - A hybrid scheme over rUDP could be used to reduce latency here. But not terribly exciting.

Erasure Code 411

Use Redundant packets to fill in for missing packets.

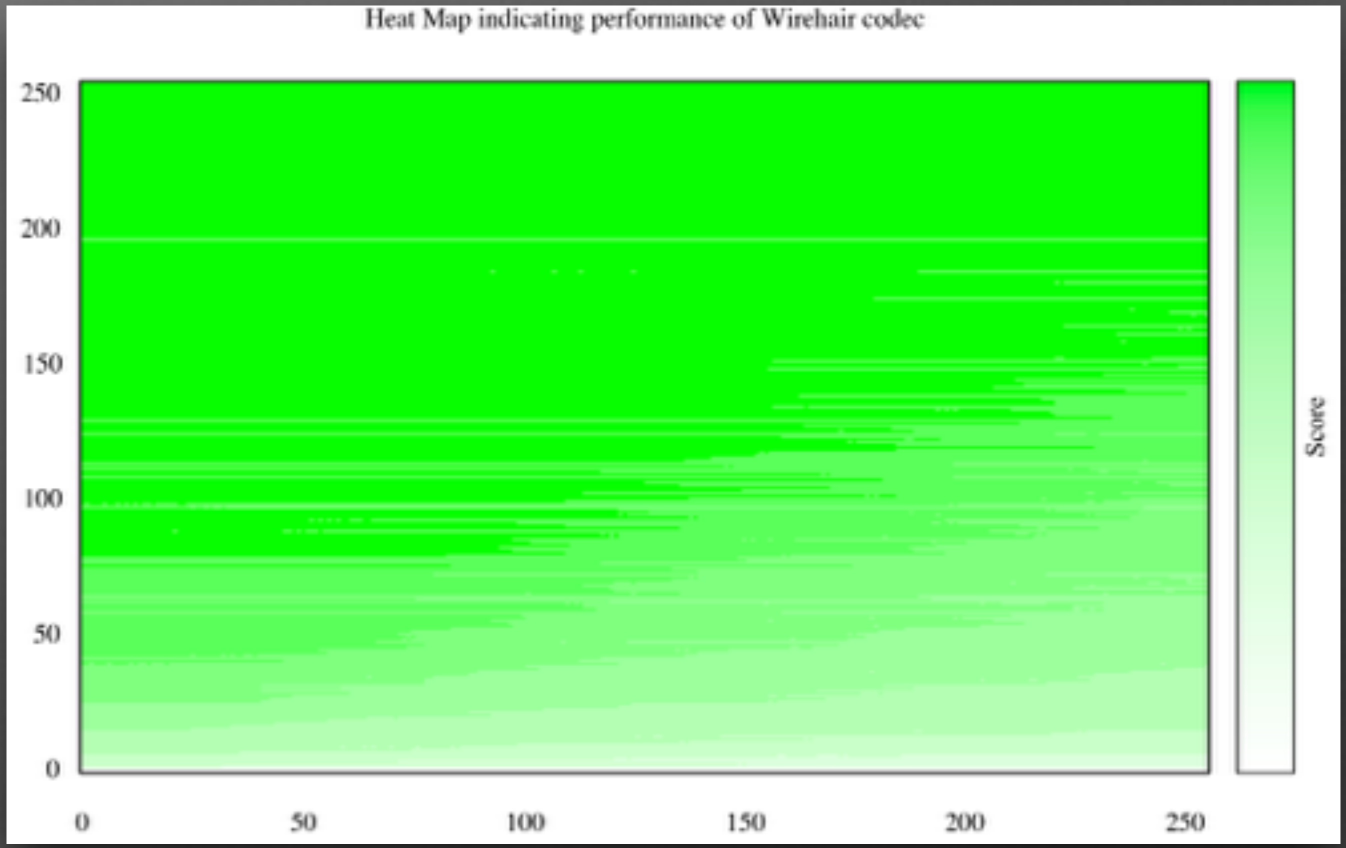
- Which Original packets are protected by each Redundant packet?
- How much bandwidth for Redundant packets?
- Pick one of three dominant types of software erasure codes:
    - Parity: XOR a set of packets together, recover from single loss
    - Reed-Solomon: 100% recovery rate given enough data
    - Low Density Parity Check: Random (but >97%) recovery

My Open-Source Software

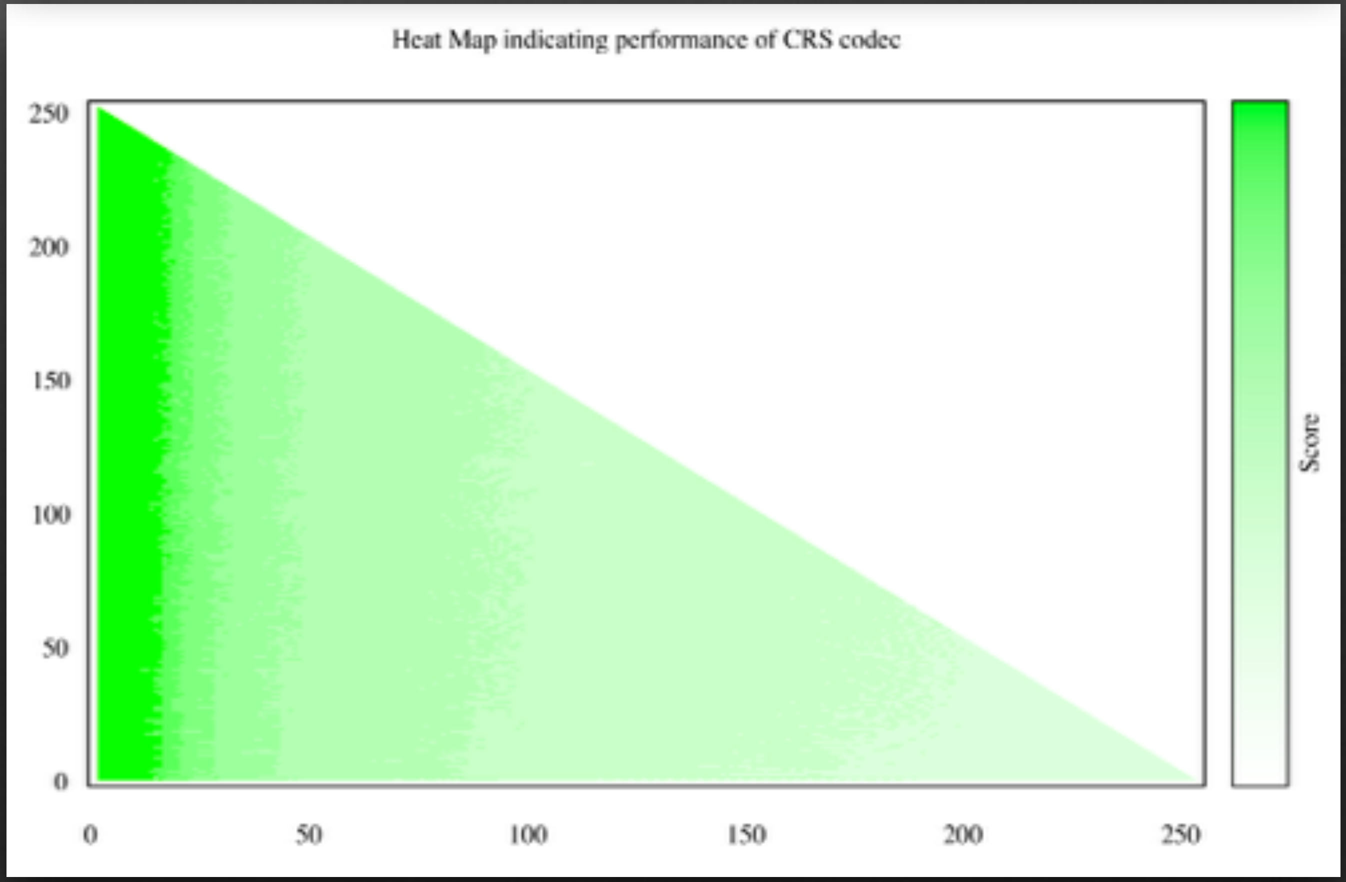
- "Cauchy" RS codes in C++: Longhair (new!)
  - Encodes >400 MB/s (<40 usec) for K < 30 packets.
  - http://github.com/catid/longhair
- "Raptor" LDPC codes in C++: Wirehair (2 years old)
  - Encodes >200 MB/s for protecting K > 30 packets.
  - http://github.com/catid/wirehair

Performance Showdown

Wirehair (files):



Longhair (streams):



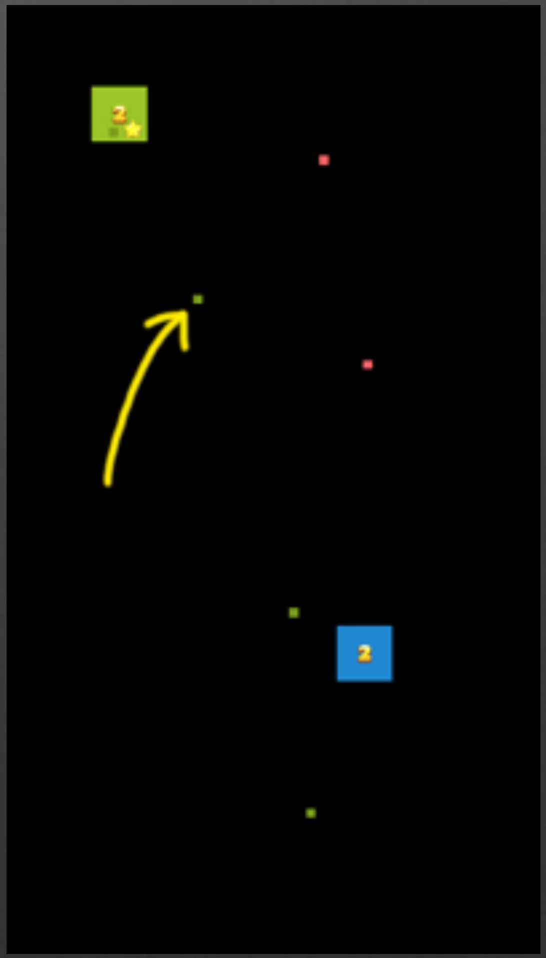
Back to Square One

- Squares.APK sends data in three modes:
    - 0 : Unreliable, Unordered : Time synch
    - 1 : Unreliable, Ordered : Position data, JSON blobs
    - 2 : 99% Reliable, Unordered : Bomb fired!

> Bomb Packets <

- Low latency is essential to quickly display the bomb for everyone.
- Lost bomb packets cause severely-visible desynchronization between players.
- But 1 in 1000 is okay: Desired <0.1% loss.
- Solution: Add redundancy, *but how?*
    - Send each bomb packet twice?

      —> Twice the bandwidth. =(
    - Low bandwidth solution: **Erasure codes.**



Shorthair Codec

- My “Shorthair” codec uses erasure codes for streaming:

    1. Measures packet loss and estimates amount of redundancy required to reach a target loss rate (e.g. 0.1%).

    2. Interleaves sending redundant data for previous RTT/2 of data while delivering latest RTT/2 of data.

- Worst-case RTT delay after loss. Average case RTT/2 delay.

- Low overhead: 5 bytes per packet. Uses Longhair.

- https://github.com/catid/shorthair

Loss Statistics from Sequence Number Holes

- UDP packets may arrive out of order. Average packet loss rate changes.
- Sequence number ranges are tracked in bins. Track only 3 bins:

```
            |<---delay-->|<---delay--->|<---delay--->|

Bin 0 :    ^START       ^STOP         ^SEND

Bin 1 :                 ^START        ^STOP         ^SEND

Bin 2 :                               ^START        ^STOP
```

- Whenever statistics are requested, a new bin is started, the last bin is frozen, and the last-last bin is delivered.
  - **Result:** RTT and packet loss statistics are delivered periodically.

Calculate Redundancy

- Statistically modeled just like repeated coin flips. Let $p = P(\text{loss}) = 0.03$
- This is a Binomial random variable X. Let r = #redundant, k = #original

$$P(\text{Total erasure code failure}) = P(X > r), \qquad X \sim \text{Binomial}(k+r, p)$$
$$\mu = E[X] = (k+r)p, \quad \sigma^2 = SD[X] = \sqrt{((k+r)p(1-p))}$$

- X is approximated by $Y \sim \text{Normal}(\mu, \sigma^2)$
  - Works when $(k+r)p \geq 10$, $(k+r)(1-p) \geq 10$. Otherwise use exact calc.
- So: $P(X > r) \approx P(Y \geq r + 0.5)$, which is much easier to code in C:

```
u = (k+r)*p; s = sqrt(u * (1-p));
Pr = 0.5 * erfc(INVSQRT2 * (r - u - 0.5) / s);
```

Shorthair Protocol

- <SeqNo [2 bytes]>
- <Out-of-Band [1 bit] || CodeGroup [7 bits]>
- OOB=1 packets stop here, but Original data also has:
  - <ID [1 byte]> <(k - 1) [1 byte]>
  - Redundant packets (ID >= k) also have:
    - <(m - 1) [1 byte]> <Original Length [2 bytes]> {data}

Shorthair Schedule

Redundant packets interleaved with originals:

	56	57	58	59	60	61
R0	X	X	X	send here		
R1	X	X	X			send here
R2				X	X	X
R3				X	X	X

Brief Glimpse at the Future

Redundant packets interleaved with originals:

	56	57	58	59	60	61
R0	X	X	X	send here		
R1		X	X	X	send here	
R2			X	X	X	send here
R3				X	X	X

Review

- Erasure codes in software are fast with low-overhead
- Advantages of using erasure codes in online games:
    - > 50% faster delivery over UDP than over TCP
    - > 50% less bandwidth used than naive redundancy
    - > 50% better recovery rate than parity redundancy
- Improves the experience for multiplayer mobile apps

“The future is already here.
— It’s just not evenly distributed.”

-William Gibson, 2003