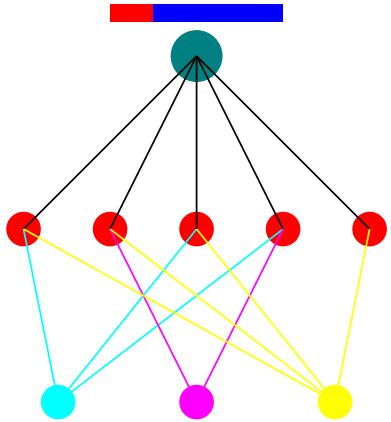
Efficient Content Delivery using Codes



Amin Shokrollahi

digitalfountain

Content

- Goals and Problems
- TCP/IP, Unicast, and Multicast
- Solutions based on codes
- Applications
- LT Codes

Goal

Want to transport data from a transmitter to one or more receivers over IP reliably.

Current solutions have limitations when amount of data is large and

- Number of receivers is large (point-multipoint transmission)
  - Video on Demand, new versions of computer games
- Or, network suffers from unpredictable and transient losses
  - Satellite, wireless
- Or, connection from transmitter to receiver goes over many hops
  - Software company with development sites around the globe

Cost Measures and Scalability

Cost measures:

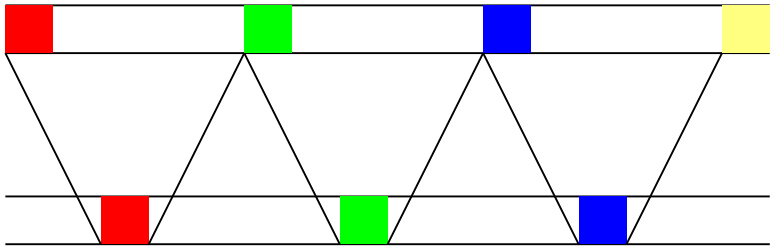
- Number of servers
- Outgoing server bandwidth
- Bandwidth utilization

A solution is called *scalable* if its cost does not increase with the number of recipients. (Server-scalable, bandwidth-scalable.)

Are interested in scalable and reliable solutions which maximize bandwidth utilization.

TCP over IP

TCP ensures reliability using acknowledgements.



To avoid acknowledging all packets, a small *window* is used which constitutes packets in transit.

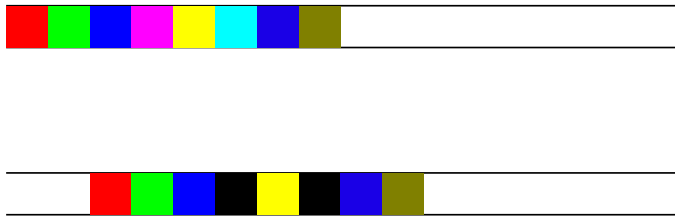
TCP/IP

TCP/IP is

- Reliable
- Not server scalable
  - Processing of ack's increases with number of receivers.
- Not bandwidth scalable
  - Each recipient requires a different connection.
- Not throughput maximizing
  - Loss rate of 2% reduces bandwidth utilization by factor of 5, instead of 0.98.

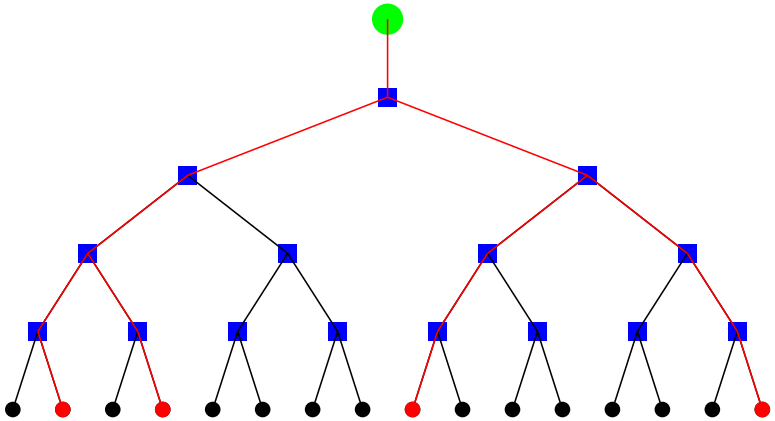
UDP Unicast

No back-channel to server.



- Not reliable
- Server scalable
- Not bandwidth scalable
- Maximizes throughput

UDP Multicast



- Is not reliable,
- Is server scalable,
- Is bandwidth scalable,
- Maximizes throughput.

Channel Model

On a computer network data is sent as packets.

Each packet has an identifier which identifies the entity it is coming from and its position within that entity.

Each packet has a CRC checksum to check its integrity.

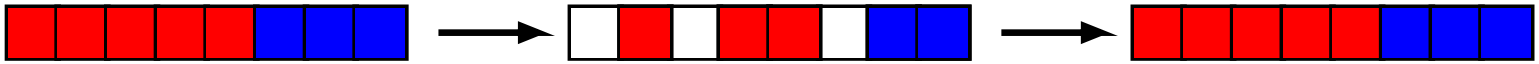
Corrupted packets can be regarded as lost.

Can concentrate on the erasure channel as a model for transmitting packets.

Solution: Codes

Want to have the advantages of Multicast and TCP/IP, but not their disadvantages.

Encode the original data and send encoded version across the network.



Reconstruction is possible if not too many packets were lost.

Reliability $\rightarrow$ Coding.

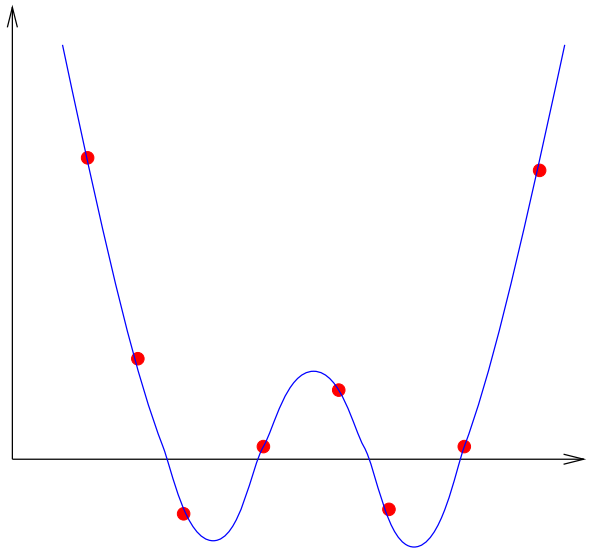
Scalability $\rightarrow$ Multicast (or unicast).

Reed-Solomon Codes

- Choose pairwise different values $x_1, x_2, \ldots, x_n$ in $\mathbb{F}_q$.

- RS-Code of dimension k is the image of the map

$$\mathbb{F}_q[x]_{<k} \to \mathbb{F}_q^n, \qquad f \mapsto (f(x_1), f(x_2), \ldots, f(x_n)).$$



Reed-Solomon Codes: Summary

Encoding and decoding complexity per bit is $\ell \cdot k \cdot (1 - k/n)$.

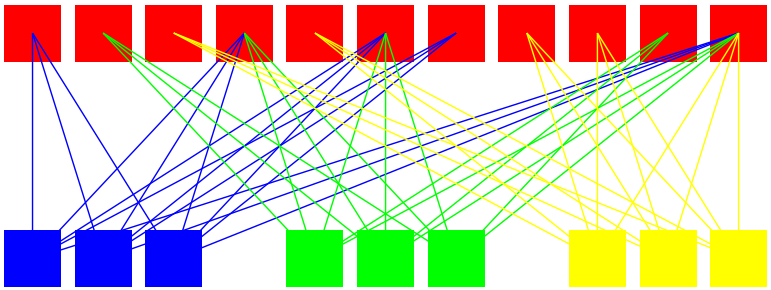
Reasonable encoding, decoding, and reception speeds if file is small, few redundant symbols, and worst possible loss rate of network known in advance.

But: Need lots of redundant symbols in many applications.

Need codes for which encoding and decoding complexity per bit is constant, and which do not need packet interleaving.

How to Make Reed-Solomon Codes more Efficient?

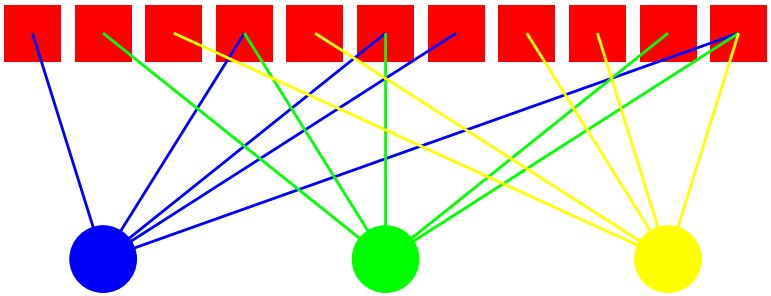
Use small blocks to compute RS redundant symbols from.



Choose the blocks randomly.

How to Make Reed-Solomon Codes more Efficient?

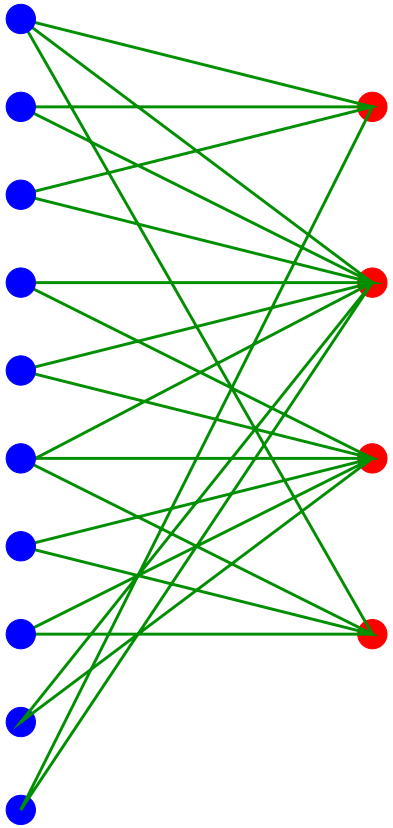
Use small blocks to compute RS redundant symbols from.



Choose the blocks randomly.

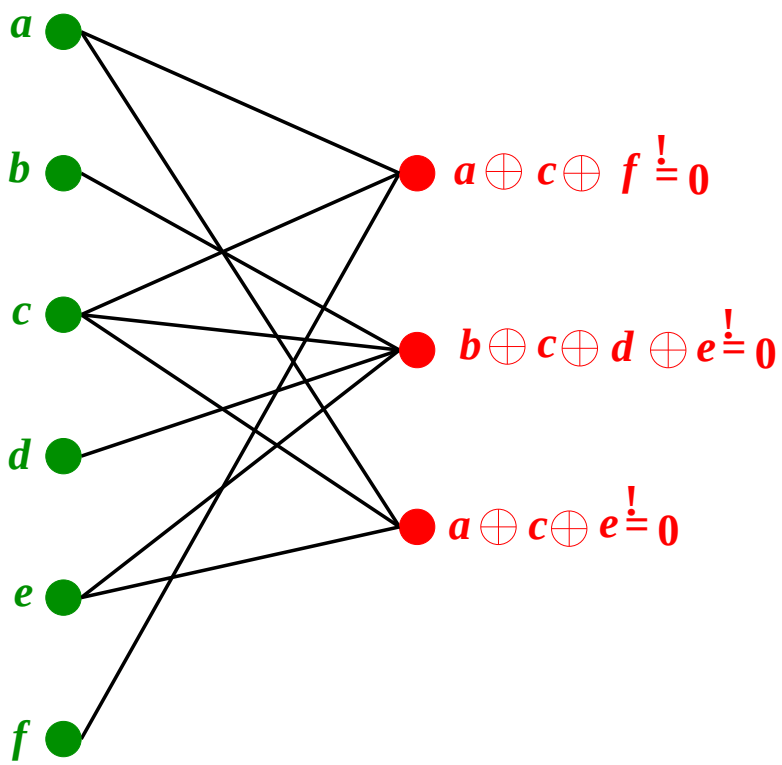
LDPC Codes

Constructed from sparse bipartite graphs.



Left nodes are called message nodes, right nodes are called check nodes.

Construction



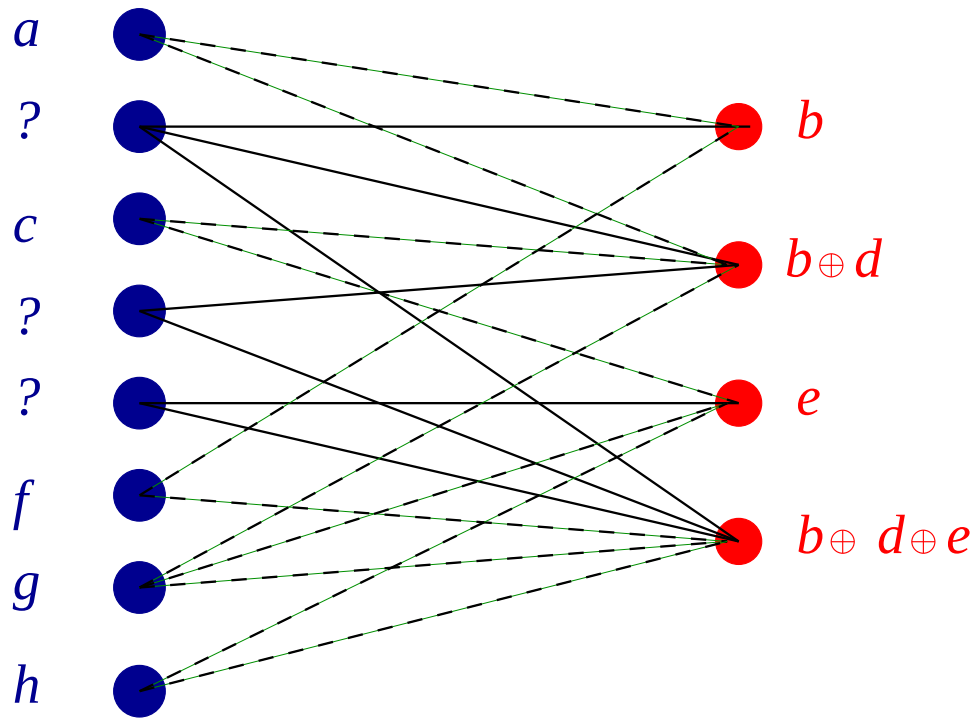
Every binary linear code has such a representation, but not every code can be represented by a sparse graph.

Encoding/decoding times?

Decoding

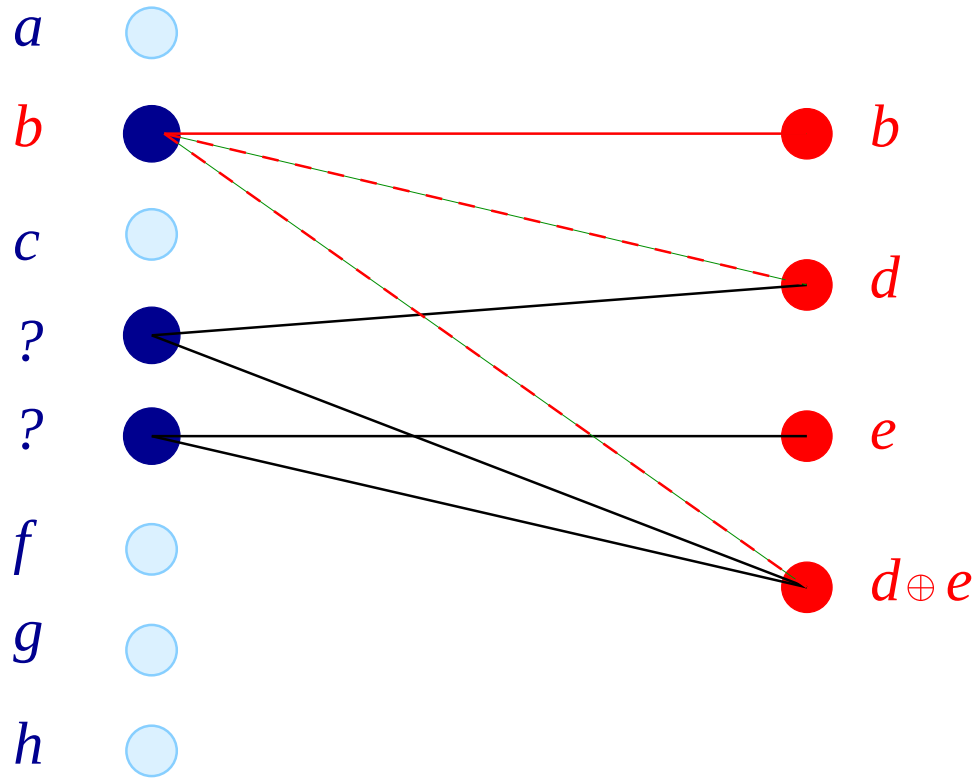
Luby-Mitzenmacher-Shokrollahi-Spielman-Stemann, 1997:

Phase 1: Direct recovery



Decoding

Phase 2: Substitution



Decoding Time

If each symbol is interpreted as a packet with b bits, then we need $\#E \cdot b$ operations to produce $n \cdot b$ bits, where $\#E$ is the number of edges in the graph, and n is the block-length of the code.

Number of operations per bit is

$$\#E/n$$

which is constant for sparse graphs.

The Inverse Problem

We have a fast decoding algorithm.

Want the design the graph in such a way that many erasures are correctable with this algorithm.

Tornado Codes

Choose design parameter D:

$$\lambda(x) := \frac{1}{H(D)}\left(x + \frac{x^2}{2} + \cdots + \frac{x^D}{D}\right)$$

$$\rho(x) := \exp\left(\mu(x-1)\right),$$

$H(D) = 1 + 1/2 + \cdots + 1/D$, $\mu = H(D)(1-R)/\left(1 - 1/(D+1)\right)$.

$$p_0\lambda(1-\rho(1-x)) = p_0\lambda(1-\exp(-\mu x)) < -\frac{p_0}{H(D)}\ln(\exp(-\mu x))$$

$$= \mu\frac{p_0}{H(D)}x < x.$$

This is true for $p_0 < H(D)/\mu = (1-R)(1-1/(D+1))$.

Tornado Codes: Efficiency

Need $k \cdot (1 + \varepsilon)$ entries of a codeword to recover the codeword.

Per-bit running time of encoding is $O(\log(1/\epsilon))$.

Per-bit running time of the decoder is $O(\log(1/\epsilon)/R)$.

Tornado codes scale to arbitary lengths and do not requrie packet interleaving.

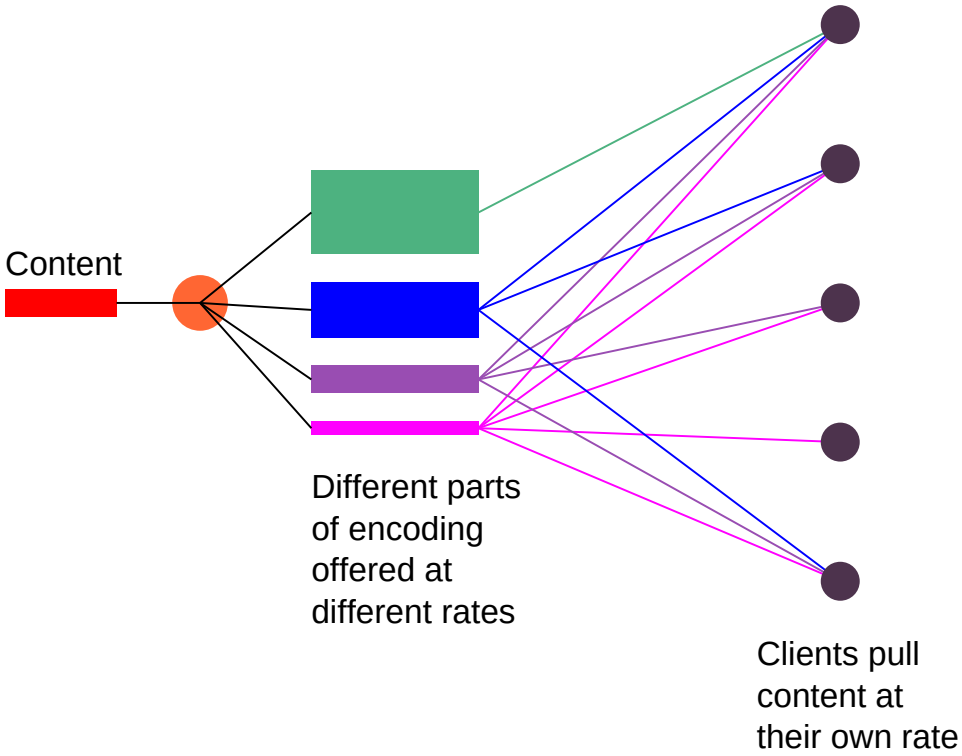
However, need a good guess of the loss rate of the channel.

Practical Applications

- Receiver driven rate control
- Multipath delivery
- Download from multiple servers
- Peer-to-peer networks
- Video-on-Demand
- Fault tolerant storage
- .....

Receiver Driven Rate Control

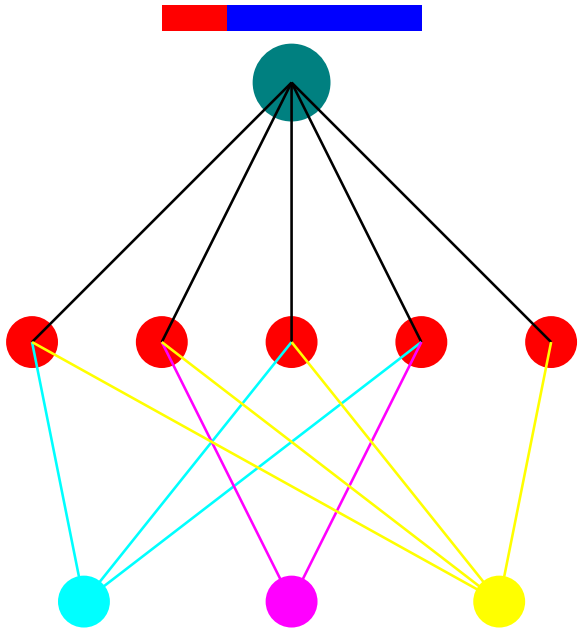
Network traffic generated by UDP should be fair to other network flows like TCP.



A lot of loss, by design. Avoiding duplicate packets is difficult.

Path Diversity

Send data across different paths to receiver to bypass links that are down.



Avoiding duplicate packets is difficult.

Shortcomings of Traditional Codes

Protocols involving erasure correcting codes are excellent candidates for replacing TCP/IP in certain applications. However, traditional codes have many disadvantages:

- One has to have a good guess on the loss rate of the clients; this is very difficult in scenarios like mobile wireless.

- Many applications require coordination between senders and receivers to avoid reception of duplicate packets.

- Many applications require codes of very small rate. But, the running time of fast codes like Tornado codes is proportional to the block-length, rather than the length of the original content.

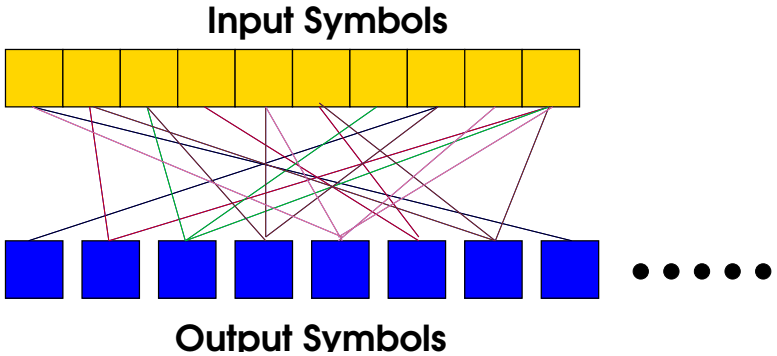
What we Really Want

To design a completely receiver driven and scalable system one needs codes

- That adapt themselves to the individual loss rates of the clients; clients with more loss need longer to recover the content;
- For which the decoding time depends only on the length of the content;
- That achieve capacity of the erasure channel between server and any of the clients;
- That have fast encoding and decoding algorithms.

Beyond Tornado: LT Codes

Michael Luby has invented a class of codes that achieves all these goals.



Their per-bit complexity of encoding/decoding is $O(\log(k))$. (This is optimal.)

If a receiver has loss rate p, then the code corresponding to that receiver has rate $1 - p - c/\sqrt{k}$.

LT Codes: Average Weight

Why does the average weight of an input symbols have to be $\Omega(\log(K))$?

This of the K input symbols as K bins, and of the edges as E balls.

Balls thrown randomly into the bins.

What is the probability that at least one bin not covered? $(1 - 1/K)^E$

For this probability to be at most $1/n$, E has to be $K \ln(K)$.

Analysis

The correct design of LT codes depends on the underlying degree distribution.

The analysis of LT codes uses novel techniques from probability theory quite different from analysis techniques for LDPC codes.

At Digital Fountain, we have developed novel techniques for computing the error probability of the decoder, and can routinely handle problems where K is the hundreds of thousands.

Correctly designed LT codes, solve the content delivery problems described in this lecture in a clean, scalable, and reliable way.