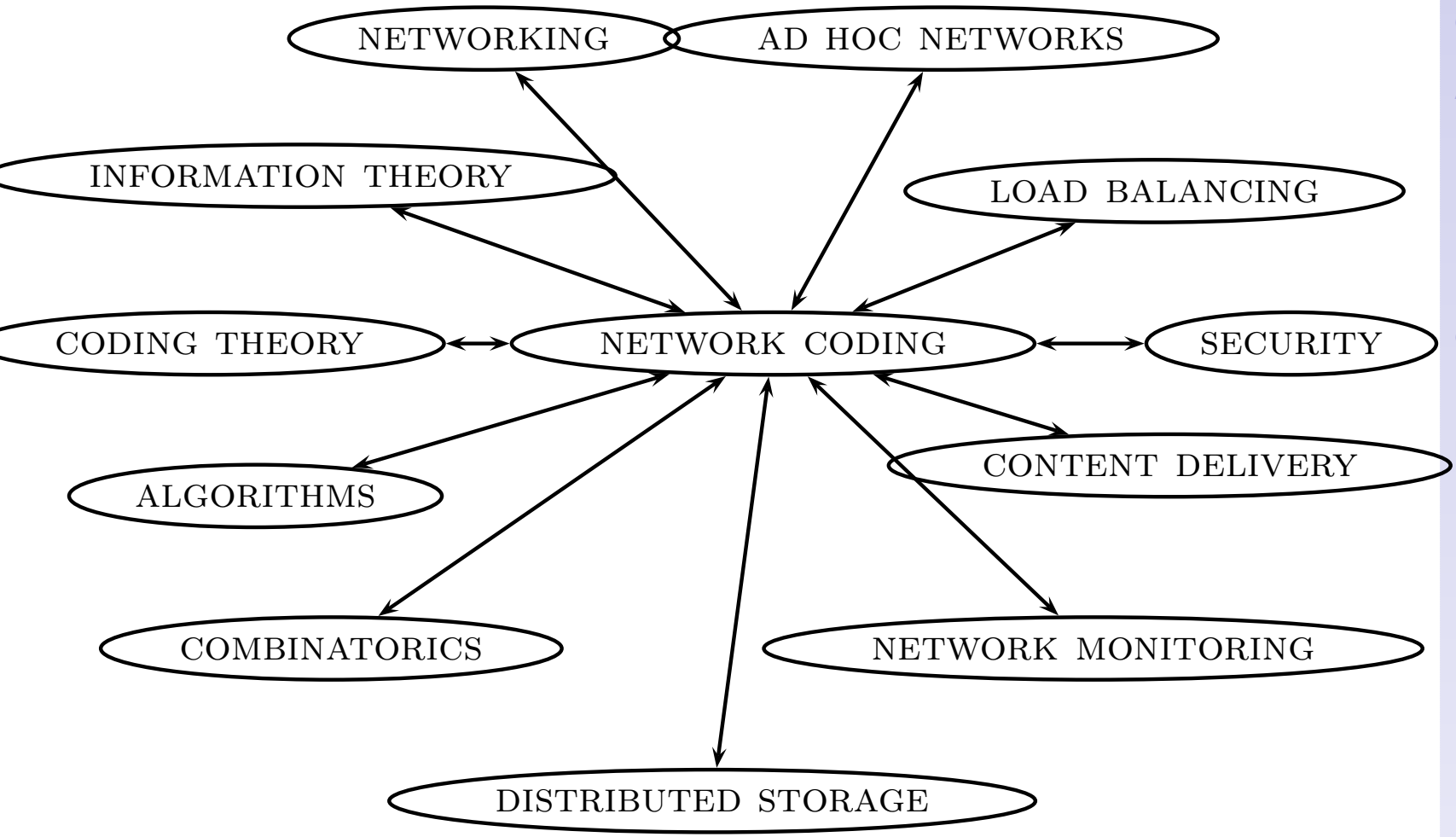
NETWORK CODING TUTORIAL

Emina Soljanin

Comm. & Stat. Research, Bell Labs

Murray Hill, June 2007

CONNECTIONS WITH OTHER AREAS



Coupons & Numbers

CDN

Avalanche

Propagation

Protocols

Advantages

Challenges

Channel Codes

Fountain Codes

Distributed Storage

COUPON COLLECTION

Examples:

- Coupon collection is a probabilistic model used to study, *e.g.*,
  1. collecting pictures of soccer players placed in cereal boxes,
  2. collecting file fragments distributed throughout a network.
- Recall hard-to-get pictures and trading with friends.

Problem formulation:

- A population of N elements is sampled with replacement.
- We are interested in
  1. Sample size S_r necessary for acquiring r distinct elements.
  2. Waiting time W_r to acquire the r-th distinct element.
  3. Number of different coupons C_s in a sample of size s.

NUMBER COLLECTION

- Suppose the population consists of numbers $x_1, \ldots, x_N$.
- You need to collect all N of them.
- You are offered two sampling strategies:
  1. Draw a number with replacement.
  2. Draw a random linear combination of the numbers
     (the coefficients and the result of combining will be revealed).
- Which strategy would you choose?
- Which strategy would you choose if
  - you have problems with inverting matrices, or
  - you can only afford to draw fewer than N times, or
  - the numbers are elements of some finite field?

COLLECTING COUPONS

- Waiting time W_r to acquire the r-th distinct element:
  - Success happens with probability $p_r = 1 - (r-1)/N$.
  - Bernoulli trials:
    $E\{W_r\} = \sum_{k=1}^{\infty} k \cdot p_r (1-p_r)^{k-1} = 1/p_r = N/(N-r+1)$.
- Sample size S_r necessary for acquiring r distinct elements:

$$E\{S_r\} = E\{W_1 + W_2 + \cdots + W_r\}$$
$$= N\left(\frac{1}{N} + \frac{1}{N-1} + \cdots + \frac{1}{N-r+1}\right) \gtrapprox N \log \frac{N+1}{N-r+1}$$

- Collection size C_s in a sample of size s.
  - Probability of not drawing a fixed element is $\left[(N-1)/N\right]^s$.
  - $E\{C_s\} = N\left[1 - (1-1/N)^s\right] \gtrapprox N\left(1 - e^{-s/N}\right)$.

COLLECTING NUMBERS OVER $\mathbb{F}_q$

- Waiting time W_r to acquire the r-th innovative coupon:
  - Success happens with probability $p_r = 1 - q^{r-1}/q^N$.
  - Bernoulli trials:

$$\mathrm{E}\{W_r\} = \sum_{k=1}^{\infty} k \cdot p_r (1-p_r)^{k-1} = \frac{1}{p_r} = \frac{q^N}{q^N - q^{r-1}} < \frac{q}{q-1}.$$

- Sample size S_r necessary for acquiring r innovative coupons:

$$\begin{aligned}\mathrm{E}\{S_r\} &= \mathrm{E}\{W_1 + W_2 + \cdots + W_r\} \\ &= \frac{q^N}{q^N} + \frac{q^N}{q^N - q} + \cdots + \frac{q^N}{q^N - q^{r-1}} < r \cdot \frac{q}{q-1}\end{aligned}$$

- How helpful is it to acquire fewer than N innovative coupons?

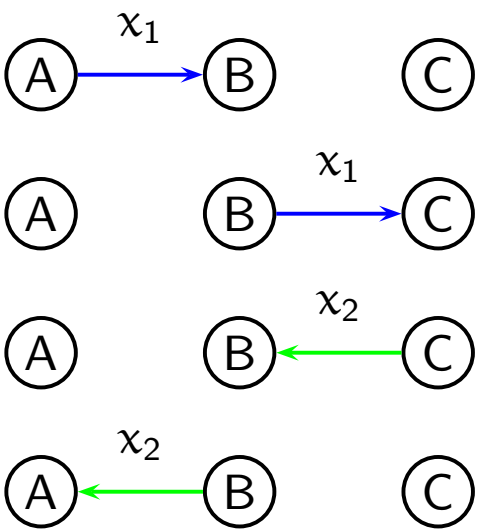
DISTRIBUTED COUPONS

- Suppose the coupons are distributed throughout the network.
- Example:

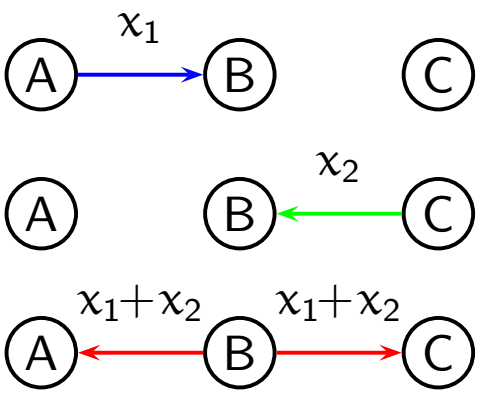


A has x_1 and C has x_2; all three nodes want both x_1 and x_2.

Without Coding



With Coding



Content Distribution Networks

- Traditional (Akamai):
  - A client directly requests and receives content from a server.
  - Content is duplicated and distributed throughout the network.
  - Not scalable, flash crowds, server failure.
- Peer-to-Peer (BitTorrent):
  - Nodes receiving content assist in its further distribution.
  - Inherently scalable:

    New nodes bring additional transmission & computing power.
  - Less sensitive to a single point of failure.
  - MS Avalanche resembles BitTorrent with coding.

BitTorrent/Avalanche – Topology

- For each file, there is an overlay distribution network of peers.
- The origin of the file (source) is the first node in the network.
- As they join the network, peers connect to 4 to 8 neighbors.
- Each node keeps local topological information.
- A peer may join and leave the network at any time.
- A peer may drop a neighbor and connect to another nodes.
- There are special nodes:
  - Registrar: enables peer discovery and keeps track of topology.
  - Logger: aggregates peers' and registrar trace messages.
  - Seeds: peers containing complete content.

BitTorrent/Avalanche – Communications

- The source splits the file into N blocks.
- A peer in possession of a block acts as a server for that block.
- Functionality of peers:
  - network transport maintains connectivity to peers
  - content manager maintains the actual content transfer
- Content propagation strategies (choosing a block to transfer):
  - In BitTorrent, a peer transfers
    - a random block in the beginning of the distribution, and
    - a local rarest after a few blocks have been downloaded, or
    - a global rarest in small P2P networks.
  - In Avalanche,

    a peer transfers a random linear combination of his blocks.

Coupons & Numbers

CDN

Avalanche

Propagation

Protocols

Advantages

Challenges

Channel Codes

Fountain Codes

Distributed Storage

Blocks and Generations

- The source splits the file into N blocks/packets.
- In experiments, N is between 1000 and 2000.
- These packets are grouped in generations (segments).
- The generation tag is placed in the packet header.
- Only packets in the same generation are linearly combined.
- In experiments, a typical generation size is 50.

Packets and Symbols

- The file data is carried by multiple network packets of L bits.
- m consecutive bits are treated as a symbol of $\mathbb{F}_{2^{\mathrm{m}}}$.
- In (non-vector) encoding,
  each symbols in a packet is multiplied by the same number in $\mathbb{F}_{2^{\mathrm{m}}}$ prior to adding to other (multiplied) packets.
- Typically, L is 1400×8 bits, m is 8 or 16.

Content Propagation within a Generation

- Encoding/Decoding takes place within a generation.
- The source transfers linear combinations of the file blocks.
- Peers transfer linear combinations of already received packets.
- Consequently,
  each packet carries a linear combination of the file blocks.
- This linear combination is described by the coding vector.
- The coding vector is placed in the packet header.
- For generation size h, the overhead caused by coding is $\mathrm{h}/\frac{\mathrm{L}}{\mathrm{m}}$.
- For the generation size 50, the overhead is $50/1400 \approx 3\%$.

Collecting Packets within a Generation

- Peers collect packets with linearly independent coding vectors.
- Let $c(v)$ be the coding vectors of the packets at node v.
- Any coding vector not in the span of $c(v)$ is innovative for v.
- Peers with h independent packets can decode the generation.
- For symbol position i, the node solves a system of equations

$$\mathbf{C}\mathbf{x}_i = \mathbf{b}_i, \quad i = 1, \ldots, L/m,$$

to recover the h content symbols at position i, where

- $\mathbf{x}_i$ is the $h \times 1$ vector of unknown file symbols at position i,
- $\mathbf{C}$ is the $h \times h$ matrix whose rows are the coding vectors,
- $\mathbf{b}_i$ is the $h \times 1$ vector of the packet symbols at position i.

MS Avalanche – A Simple Protocol

- Nodes register demands with their neighbors.
- Sender

  is a peer with a new packet and registered demands by others.
- Receivers

  are all peers who have registered demands with the sender.

Coupons & Numbers

CDN

Avalanche

Propagation

Protocols

Advantages

Challenges

Channel Codes

Fountain Codes

Distributed Storage

A Simple Protocol Transmissions

1. The sender
   - generates random coefficients that he intends to use for linear combining of its packets, and then
   - computes the coding vector and sends it to the receivers.
2. Receivers

   who find the coding vector innovative request the packet.
3. The sender
   - only then forms the linear combination of its packets, and
   - sends the packet to the receivers who requested it.

MS Avalanche – Another Approach

- The peers know the coding vectors of their neighbors.
- For each neighbor, a node
  computes the rank of the matrix of their combined vectors.
- This way peers learn
  who can provide innovative packets and how many.
- In both protocols, the sender
  - does not estimate the packets' levels of representation, but
  - generates a random linear combination of all its packets.

AVALANCHE – Advantages

Increased Reliability

- Peer departures and transmission losses cause packet losses.
- Some packets become rare and can be lost forever.
- Coding alleviates the problem of rare packets. How?

Free Riding

- Free riders use network resources without contributing theirs.
- Tit-for-tat solution:
  allow download at a rate proportional to the rate of upload.
- The solution penalizes new users. Why?
- New users are penalized less if coding is used. Why?

AVALANCHE – Advantages

Churn

- Dynamics of independent arrivals and departures of peers.
- *E.g.*, node has downloaded 80% of the file when a new node B joins the network, and requests packets from their common neighbors. A gets packets it already has.
- With coding, the neighbors can make more of the packets they transmit simultaneously useful to both A and B.

Nodes Behind Walls

- NAT devices and firewalls create unreachable peers.
- Such nodes may advertise content but not share it.
- Coded systems perform surprisingly well even with many unreachable peers.

AVALANCHE – Questionable Advantages

Clustering

- Clusters arise because of the way P2P networks are formed.
- Reconfiguration becomes limited and resources precious.
- Coding makes better use the available resources.

Applications

- Without coding, applications
  keep track of how many and which packets they have.
- With coding, applications
  keep track only of the number of innovative packet.

Coupons & Numbers
CDN
Avalanche
Propagation
Protocols
Advantages
Challenges
Channel Codes
Fountain Codes
Distributed Storage

AVALANCHE – Challenges

Encoding/Decoding Complexity

If the operating field is $\mathbb{F}_{2^m}$ and generation size h, we need

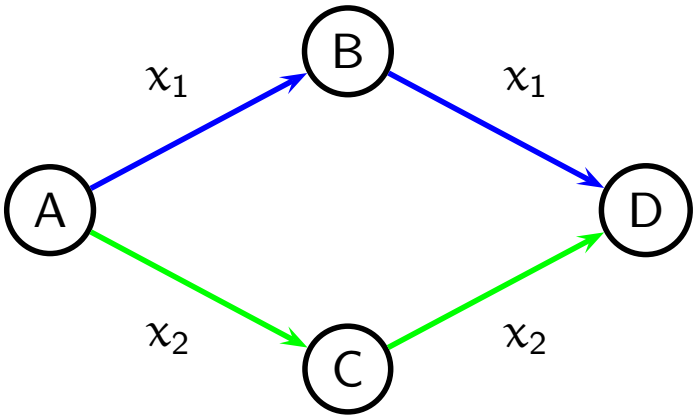
- $\mathcal{O}(h^2)$ operations in $\mathbb{F}_{2^m}$ to form a linear combination.
- $\mathcal{O}(h^3)$ operations in $\mathbb{F}_{2^m}$ for a matrix inversion by Gaussian elimination.
- $\mathcal{O}(m^2)$ binary operations for a multiplication or inversion.

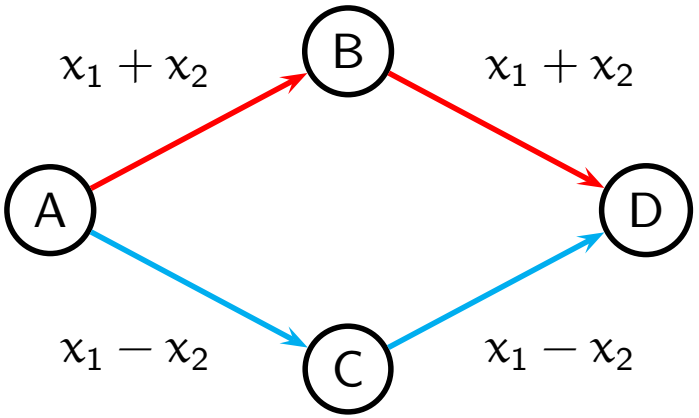
Peer Heterogeneity

- Throughput benefits are unclear for nonuniform demands.
- Nonuniform communication resources create QoS problems.

AVALANCHE – Challenges

Security





- Poisoning attacks by malicious peers more detrimental.
- The network becomes more vulnerable to DoS attacks.
- Problems are hard to solve and even harder to implement.
- *E.g.*, authentication with homomorphic hash functions.

BINARY LINEAR CODES

- Suppose we want to communicate k bits $\omega_1, \ldots, \omega_k$.
- If we encode $\omega_1, \ldots, \omega_k$ into $n = k + 1$ bits $x_1, \ldots, x_k, x_{k+1}$ s.t.

$$x_i = \omega_i \text{ for } 1 \leqslant i \leqslant n-1 \text{ and } x_n = \omega_1 + \cdots + \omega_{n-1},$$

  we can decode all $\omega_1, \ldots, \omega_k$ if one of $x_1, \ldots, x_n$ is erased.
- The generator matrix of this (n, k) code is

$$G = \begin{bmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ldots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 1 \end{bmatrix}_{k \times n}$$

RANDOM LINEAR CODES vs. FOUNTAIN

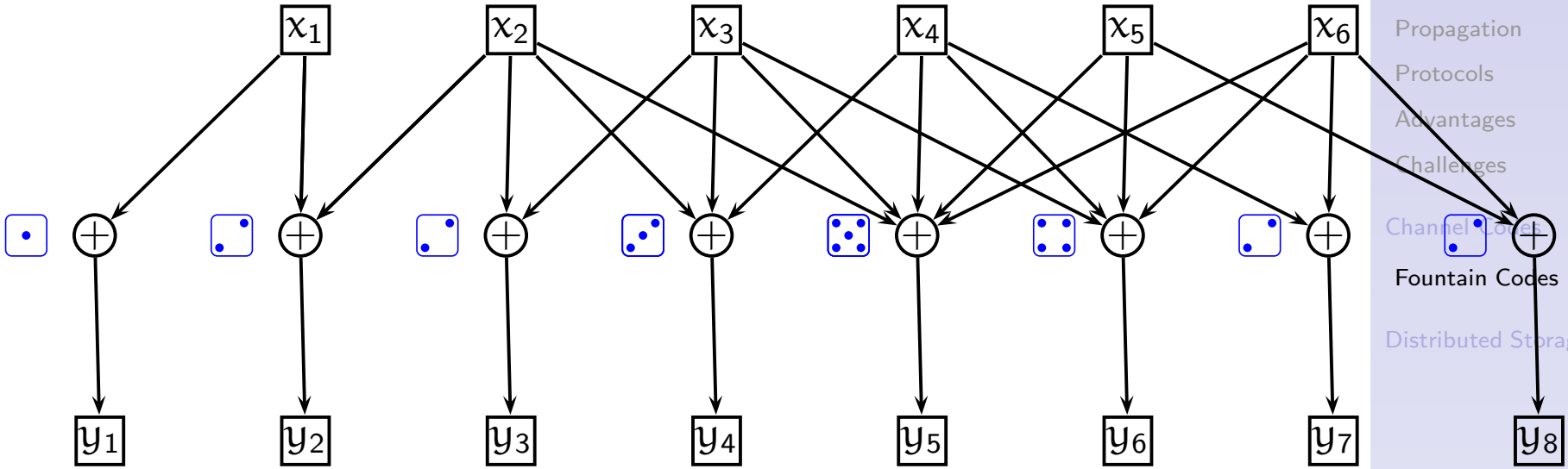
- There are performance/redundancy/complexity tradeoffs.
- Random Linear Codes
  - The $k \times n$ elements of G are chosen uniformly at random.
  - Random codes are good but hard to decode.
  - Erasure decoding complexity is $\mathcal{O}(n^3)$ (Gaussian elimination).
  - General decoding is NP hard.
- Fountain Codes
  - The $k \times n$ elements of G are not chosen entirely at random.
  - Fountain codes are good and have useful structure.
  - Erasure decoding complexity is $\mathcal{O}(n)$ (Gaussian elimination).
  - The most popular are LT codes and Raptor codes.

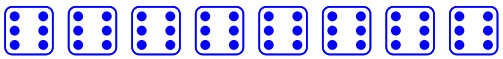
FOUNTAIN CODES

- Enable communications over multiple, unknown channels:
  1. They are rateless, and thus their redundancy can be flexibly adapted to changing physical channel conditions characteristic for mobile wireless.
  2. In the case of multiple erasure channels as in Multimedia Broadcast/Multicast Service (MBMS), they can be made to universally achieve the channel capacity for all erasure rates.
- Fixed-rate codes would not perform well in such settings.
- Fountain codes are easy to encode and decode.
- A class of Fountain codes, Digital Fountain Raptor technology, has been standardized by 3GPP as a single mandatory code for MBMS.

Coupons & Numbers
CDN
Avalanche
Propagation
Protocols
Advantages
Challenges
Channel Codes
Fountain Codes
Distributed Storage

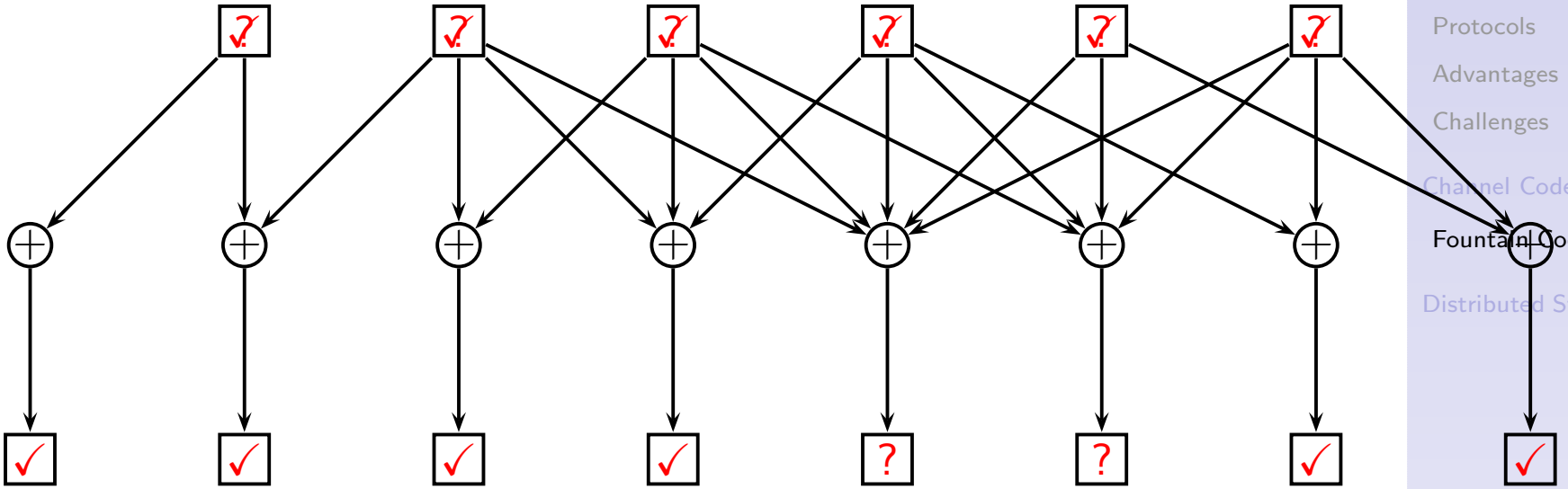
LT CODES – Encoding





face						
Ω	.05	.5	.1	.05	.25	.05

LT CODES – Decoding



- Code design:

  Ω has to be chosen to enable simple decoding under erasures.

- Raptor codes, Fountain codes.

RAPTOR CODES

- Raptor codes are concatenated codes.
- Their inner codes are LT, and the outer codes are high rate.
- Such schemes are useful when the code length is limited.
- Raptor codes do not degrade gracefully.
- This is a problem in heterogeneous networks.

DISTRIBUTED STORAGE

- P2P Distributed Storage Systems
  - Coded fragments of files are spread across the network.
  - Redundancy is accomplished by the means of erasure codes.
  - Can coded fragments be generated in a decentralized way?
  - Can coded fragments be generated with little data transfer?
- Sensor Network Data Persistence
  - Sensor nodes collecting data frequently fail.
  - Data is also vulnerable to loss because of the funneling effect.
  - Distributed compression and erasure codes are helpful.
  - Growth codes are efficient as data accumulates at the sink.