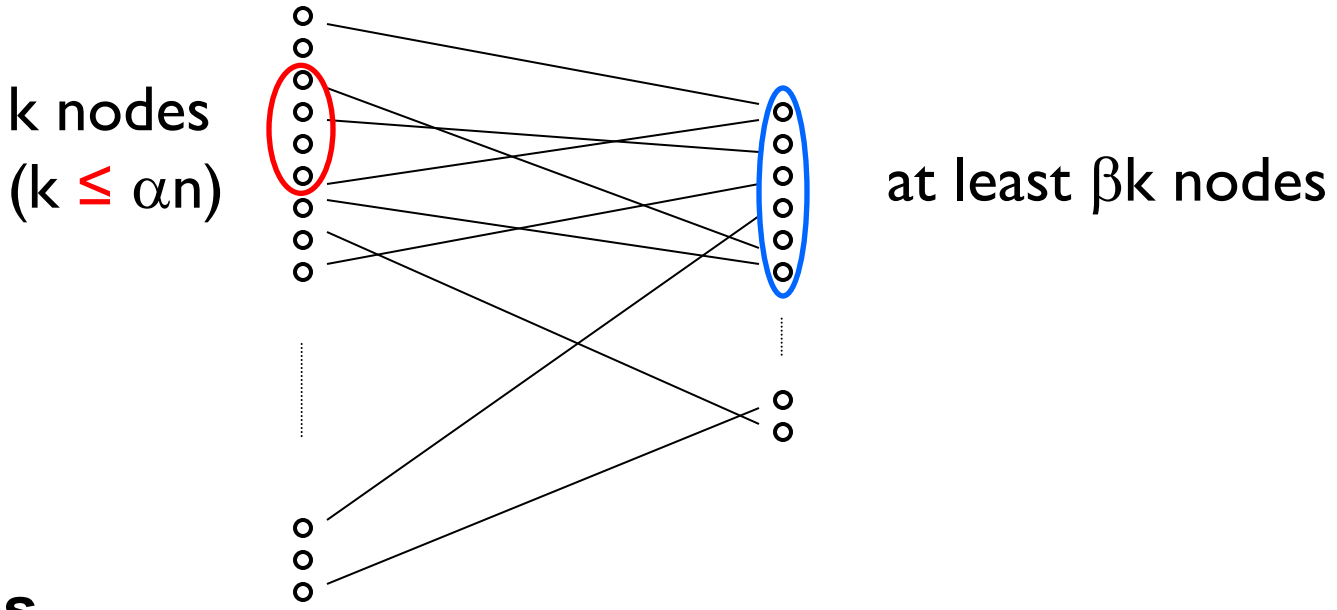
15-853:Algorithms in the Real World

- LDPC (Expander) codes
- Tornado codes
- Fountain codes and Raptor codes

Scribe volunteer?

Recap: (α, β) Expander Graphs (bipartite)



Properties

- **Expansion:** every small subset ($k \leq \alpha n$) on left has many ($\geq \beta k$) neighbors on right
- **Low degree –** not technically part of the definition, but typically assumed

Expander Graphs

Useful properties:

- Every (small) set of vertices has many neighbors
- Every balanced cut has many edges crossing it
- A random walk will quickly converge to the stationary distribution (rapid mixing)
- Expansion is related to the eigenvalues of the adjacency matrix

Recap: Expander Graphs: Constructions

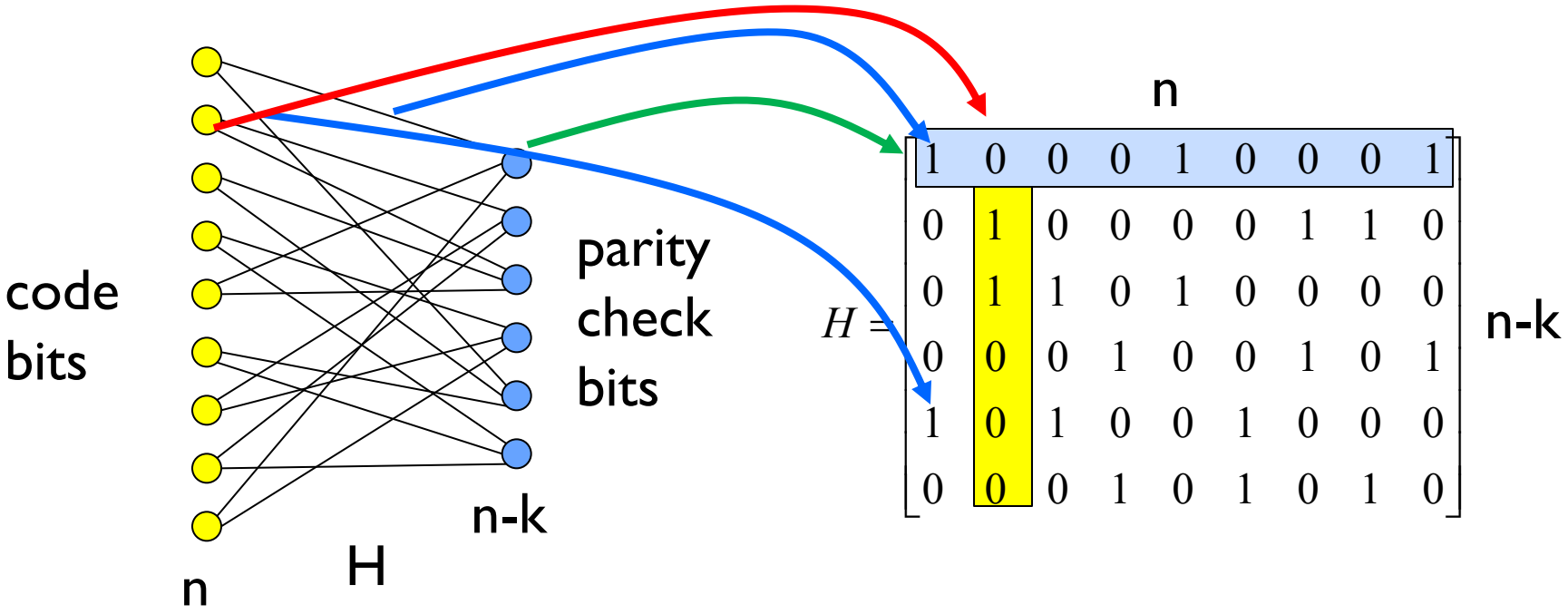
Theorem: For every constant 0 < c < 1, can construct bipartite graphs with

n nodes on left,
cn on right,
d-regular (left),

that are $(\alpha, 3d/4)$ expanders, for constants α and d that are functions of c alone.

"Any set containing at most alpha fraction of the left has (3d/4) times as many neighbors on the right"

Recap: Low Density Parity Check (LDPC) Codes



Each row is a vertex on the right and each column is a vertex on the left.

A codeword on the left is valid if each right "parity check" vertex has parity 0.

The graph has O(n) edges (**low density**)

Recap: Distance of LDPC codes

Consider a d-regular LPDC with (α,3d/4) expansion.

Theorem: Distance of code is greater than αn.

Proof. (by contradiction)

Linear code; distance= min weight of non-0 codeword.

Assume a codeword with weight w ≤ αn.

Let W be the set of 1 bits in codeword

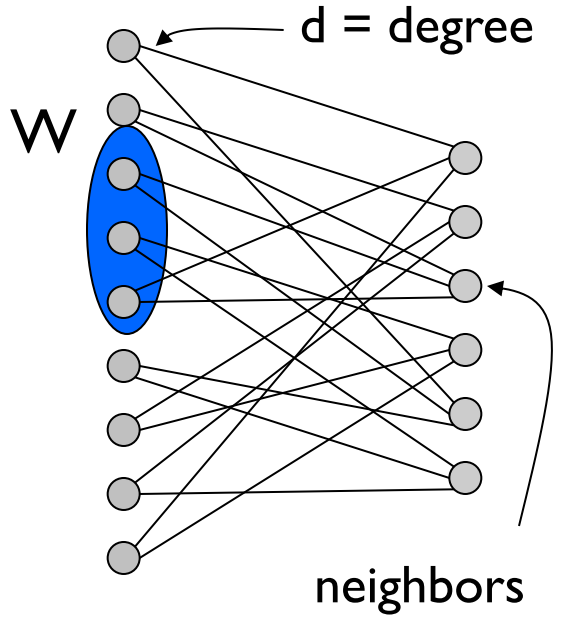
#edges = wd

#neighbors on right >3/4*wd

Max #neighbors with >1 edge from W?

#unique neighbors = wd/2

So at least one neighbor sees a single 1-bit. Parity check would fail!



Recap: Correcting Errors in LPDC codes

We say a vertex is *unsatisfied* if parity $\neq 0$

Algorithm:

While there are unsatisfied check bits

1. Find a bit on the left for which more than d/2 neighbors are unsatisfied
2. Flip that bit

Converges:

Since every step reduces unsatisfied parity by at least 1.

Running time:

Runs in linear time (for constant maximum degree on the right).

Recap: Correcting Errors in LPDC codes

Theorem: Always exists a node > d/2 unsatisfied neighbors if we're not at a codeword.

Proof: (by contradiction)

Suppose not. (Let d be odd.) Let S be the corrupted bits.

Each such bit has majority of satisfied neighbors

(sat. neighbors see at least two corrupted bits on left)

(unsat. neighbors may see only one corrupted bit on left)

Each corrupt bit give $1 to each unsat nbr, $½ to sat nbr.

Total money given < 3d/4 |S|.

Each node in N(S) collects $1 at least.

Total money collected at least |N(S)|.

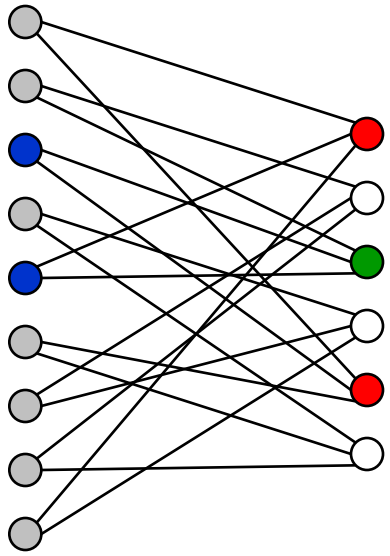
So |N(S)| < 3d/4 |S|. Contradicts expansion.

Coverges to closest codeword

Theorem: Assume (α,3d/4) expansion. If # of error bits is less than αn/4 then simple decoding algorithm converges to closest codeword.

Proof: let:

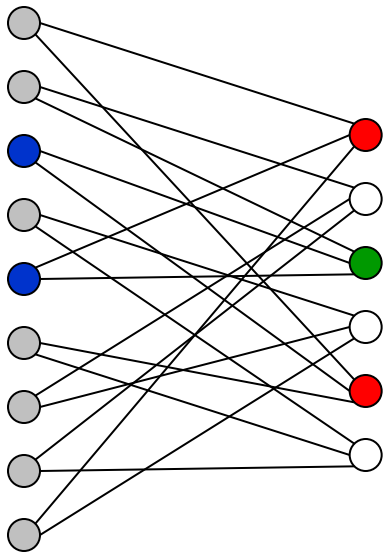
- u_i = # of unsatisfied check bits on step i
- r_i = # corrupt code bits on step i
- s_i = # satisfied check bits with corrupt neighbors on step i



Q: What do we have to show about r_i?

We know that u_i decrements on each step, but what about r_i?

Proof continued:



- u_i = unsatisfied
- r_i = corrupt
- s_i = satisfied with corrupt neighbors

$$u_i + s_i > \frac{3}{4} dr_i \quad \text{(by expansion)}$$

$$2s_i + u_i \le dr_i \quad \text{(by counting edges)}$$

$$\frac{1}{2} dr_i \le u_i \quad \text{(by substitution)}$$

$u_i < u_0$ (steps decrease u) $\qquad u_0 \le dr_0$ (by counting edges)

Therefore: $r_i < 2r_0$ i.e. number of corrupt bits cannot more than double

If we start with at most $\alpha n/4$ corrupt bits we will never get $\alpha n/2$ corrupt bits --- but the distance is αn. So converge to closest codeword.

More on decoding LDPC

- Simple algorithm is only guaranteed to fix half as many errors as could be fixed but in practice can do better.
  - Fixing (d-1)/2 errors is NP hard

- "Hard decision decoding" vs "soft decision decoding"
- Soft decision decoding
  - Probabilistic channel model (e.g., Binary Symmetric Channel) <board>
  - Goal: to compute *maximum a posteriori* (MAP) probability of each code bit conditioned on parity checks being met

More on decoding LDPC

- Soft decision decoding as originally specified by Gallager is based on **belief propagation**---determine probability of each code bit being 1 and 0 and propagate probs. back and forth to check bits.
  - Belief propagation algorithm gives MAP only if the graph is cycle free
  - As the minimum cycle length increases comes closer and closer

Encoding LPDC

Encoding can be done by generating G from H and using matrix multiply. (Remember, c = xG).

What is the problem with this?

Various more efficient methods have been studied

Let's see one approach to efficient coding and decoding.

TORNADO CODES

Luby Mitzenmacher Shokrollahi Spielman 2001

Tornado codes

Goal: low (linear-time) complexity encoding and decoding

We will focus on **erasure** recovery

- Each bit either reaches intact, or is lost.
- We know the positions of the lost bits.

The random erasure model

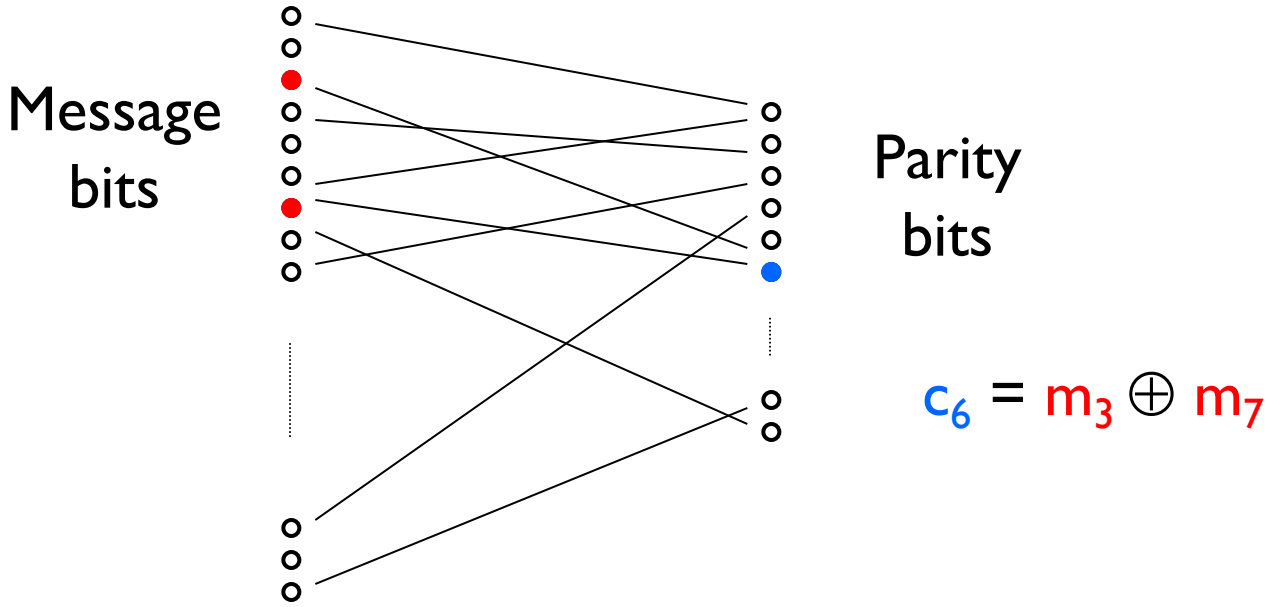
Random erasure model:

- Each bit is erased with some probability p (say ½ here)
- Known: a random linear code with rate < 1-p works (why?)

Makes life easier for the explanation here.

Can be extended to worst-case error, and bit corruption with extra effort.

[see e.g., Spielman1996]

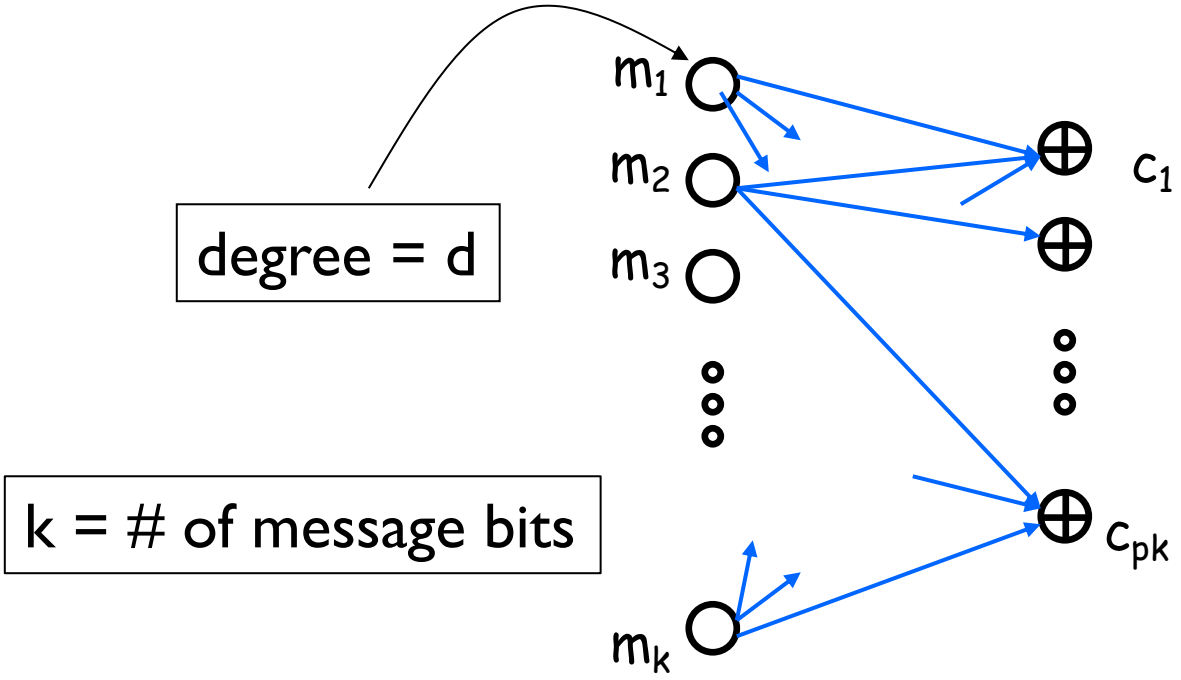


Similar to standard LDPC codes but parity bits are not required to equal zero.

(i.e., the graph does not represent H anymore).

Tornado codes

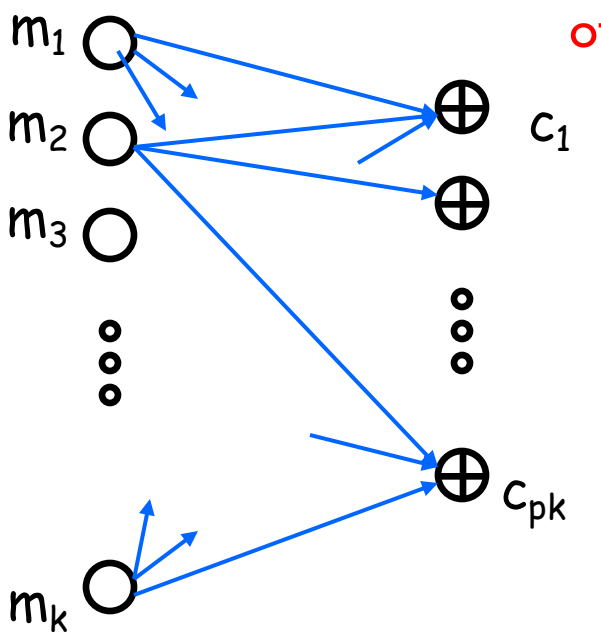
- Have d-left-regular bipartite graphs with k nodes on the left and pk on the right.



- Let's again assume 3d/4-expansion.

Tornado codes: Encoding

Why is it linear time?



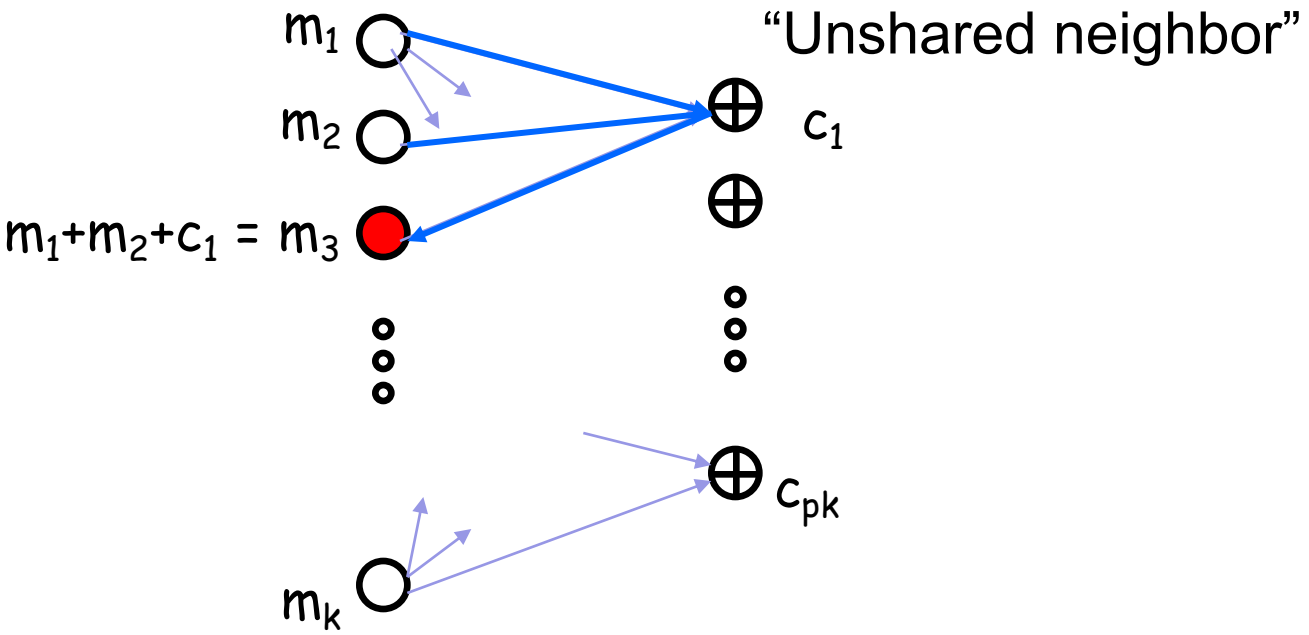
Computes the sum modulo 2 of its neighbors

Tornado codes: Decoding

First, assume that all the parity bits are intact

Find a parity bit such that only one of its neighbors is erased (an *unshared neighbor*)

Fix the erased bit, and repeat.



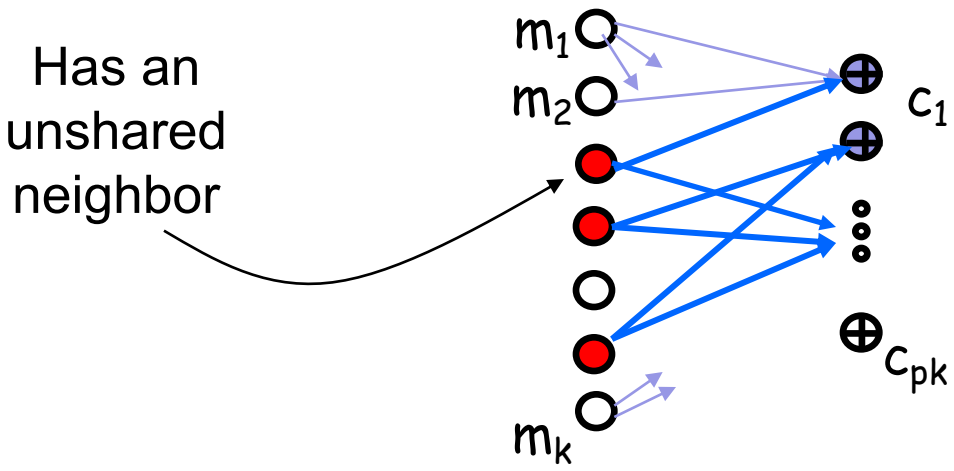
Tornado codes: Decoding

Want to always find such a parity bit with the “Unshared neighbor” property.

Consider the set of corrupted message bit and their neighbors.

Suppose this set is small.

=> at least one message bit has an unshared neighbor.



Tornado codes: Decoding

Can we always find unshared neighbors?

Expander graphs give us this property if expansion > d/2

(similar argument to one above)

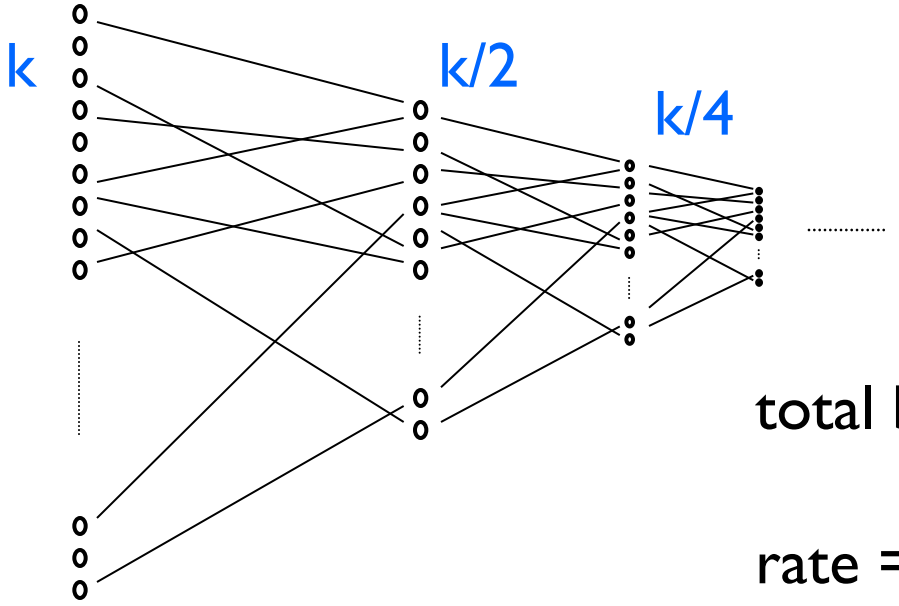
Also, [Luby et al] show that if we construct the graph from a specific kind of degree distribution, then we can always find unshared neighbors.

What if parity bits are lost?

Ideas?

Cascading

- Use another bipartite graph to construct another level of parity bits for the parity bits
- Final level is encoded using RS or some other code



stop when $k/2^t$ "small enough"

total bits $n \leq k(1 + \frac{1}{2} + \frac{1}{4} + \ldots)$

$= 2k$

rate = $k/n = \frac{1}{2}$.

(assuming p =1/2)

Tornado codes enc/dec complexity

Encoding time?

- for the first t stages : |E| = d x |V| = O(k)
- for the last stage: poly(last size) = O(k) by design.

Decoding time?

- start from the last stage and move left
- Last stage is O(k) by design
- Rest proportional to |E| = O(k)

So get very fast (linear-time) coding and decoding.

100s-10,000 times faster than RS

FOUNTAIN & RAPTOR CODES

Luby, "LT Codes", FOCS 2002

Shokrollahi, "Raptor codes", IEEE/ACM Transactions on Networking 2006

The random erasure model

We will continue looking at recovering from erasures

Q: Why erasure recovery is quite useful in real-world applications?

Hint: Internet

Packets over the Internet often gets lost (or delayed) and packets have sequence numbers!

Applications in the real world

- Internet Engineering Task Force (IETF) standards for object delivery over the Internet
  - RFC 5053, RFC 6330 (RaptorQ)

- Over the years RaptorQ has been adopted into a number of different standards: cellular networks, satellite communications, IPTV, digital video broadcasting

Fountain Codes

- Randomized construction – so there is going to be a probability of failure to decode

- A slightly different view on codes: New metrics
  1. Reception overhead
     - how many symbols more than k needed to decode
  2. Probability of failure to decode

Q: These metrics for RS codes?

Perfect? Why look for beyond?

1. Encoding and decoding complexity high
2. **Need to fix "n" beforehand**

Fountain Code: Ideal properties

1. Source can generate any number of coded symbols
2. Receiver can decode message symbols from any subset with small reception overhead and with high probability
3. Linear time encoding and decoding complexity

"Digital Fountain"

LT Codes

- First practical construction for Fountain Codes
- Graphical construction
- Encoding algorithm
  - Goal: Generate coded symbols from message symbols
  - Steps:
    - Pick a degree d randomly from a “degree distribution”
    - Pick d distinct message symbols
    - Coded symbols = XOR of these d message symbols

LT Codes: Encoding

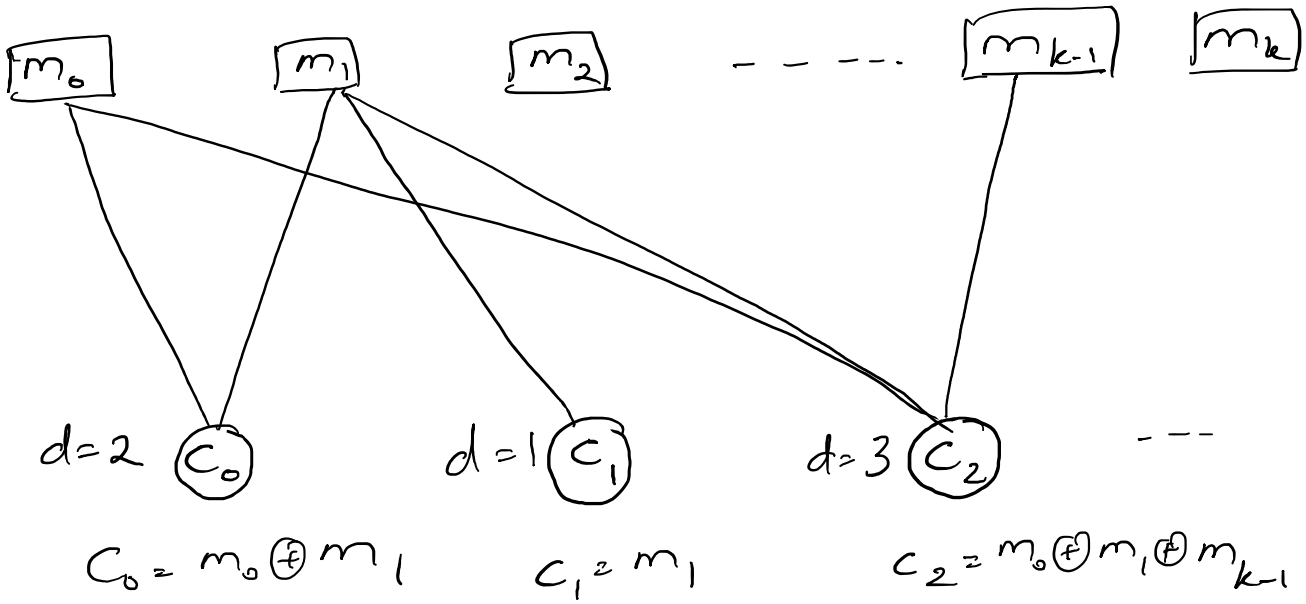
Pick a degree d randomly from a "degree distribution"

Pick d distinct message symbols

Coded symbols = XOR of these d message symbols

Message symbols

Coded symbols



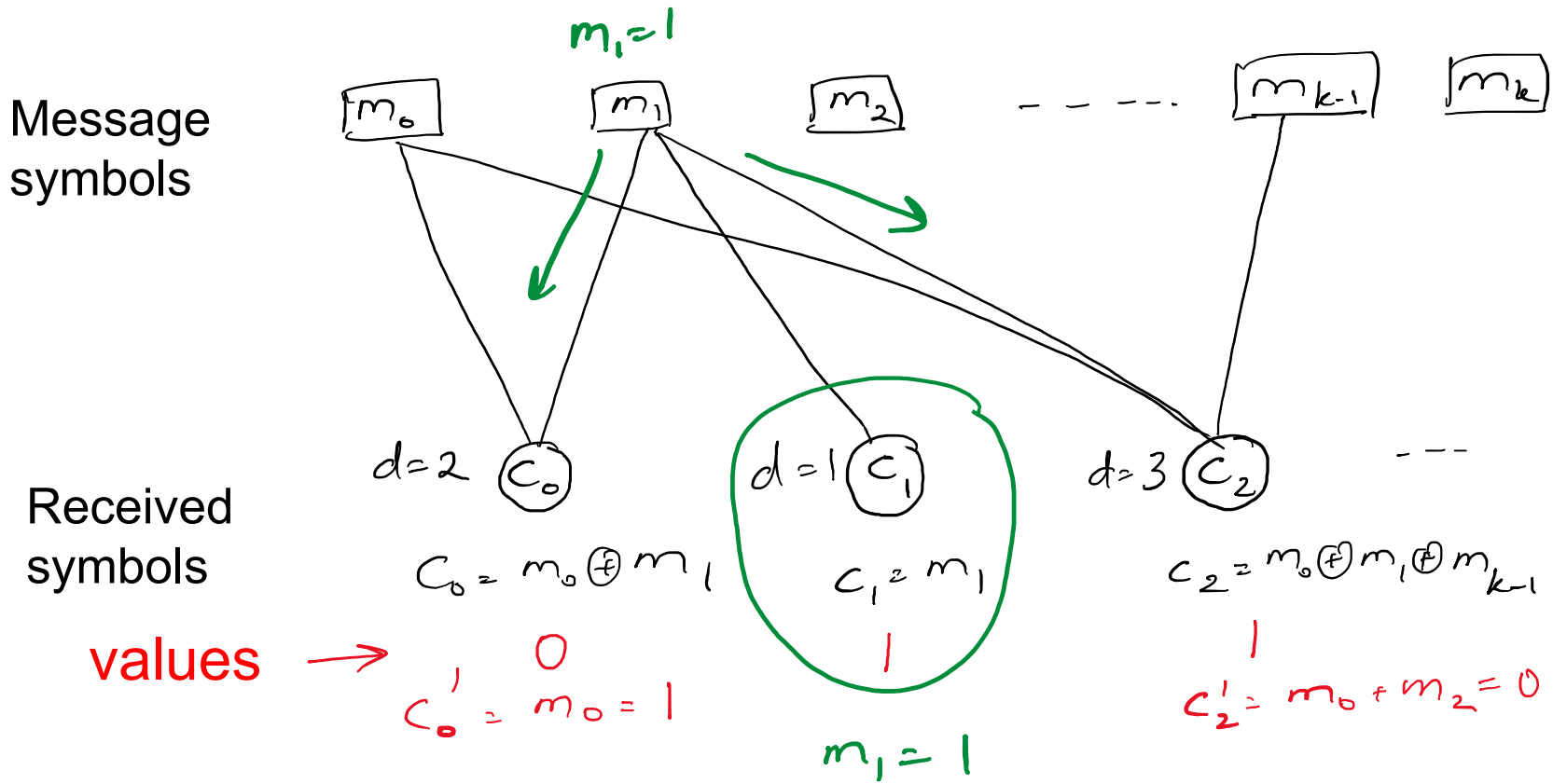
LT Codes: Decoding

Goal: Decode message symbols from the received symbols

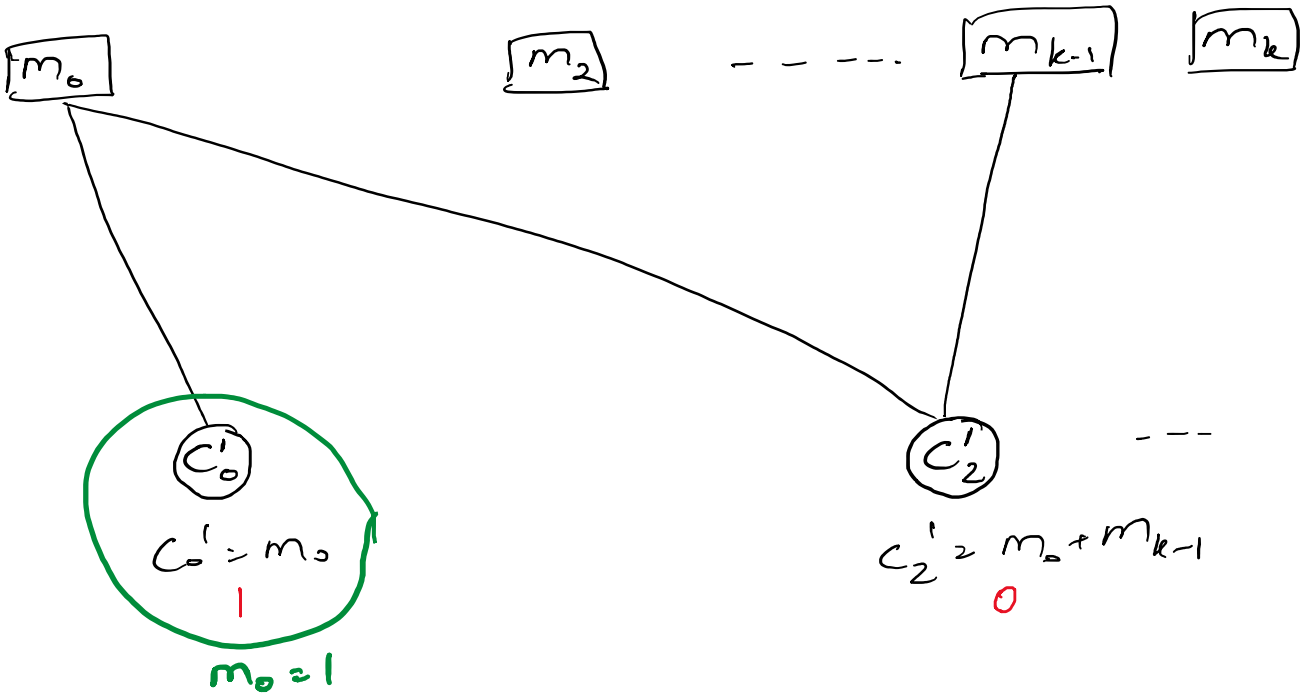
Algorithm: Repeat following steps until failure or stop successfully

1. Among received symbols, find a coded symbol of degree 1
   - Q: What does degree =1 mean?
2. Decode the corresponding message symbol
3. XOR the decoded message symbol to all other recieved symbols connected to it
4. Remove the decoded message symbols and all its edges from the graph
5. Repeat if there are unrecovered message symbols

LT Codes: Decoding



LT Codes: Decoding



Encoding and Decoding Complexity

Think: Number of XORs

Q: Encoding complexity?

#Edges in the graph

Q: Decoding complexity

#Edges in the graph restricted to received symbols

Q: #Edges is determined by what?

Degree distribution

Degree distribution

Denoted by $P_D(d)$ for $d = 1,2,\ldots,k$

Q: Simplest degree distribution?

"One-by-one" distribution: Pick only one source symbols for each encoding symbol.

$$P_D(d) = \begin{cases} 1 & \text{if } d=1 \\ 0 & \text{o.w} \end{cases}$$

Q: What is the excepted reception overhead?

Reminds you of any classical problem in probability?

Coupon collector problem! **Reception overhead: k ln k**

Huge overhead: k=1000 => 10x overhead!!

Degree distribution

Q: How to fix this issue?

Think about this…

We will continue in the next lecture.