From Fountain to BATS: Realization of Network Coding

Shenghao Yang

Jan 26, 2015
Shenzhen

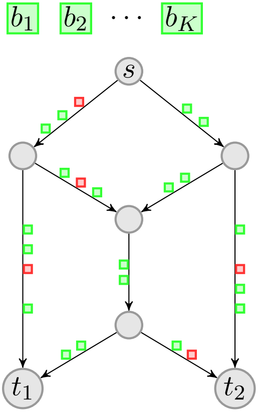
Outline

1. Outline

2. Single-Hop: Fountain Codes
   - LT Codes
   - Raptor codes: achieving constant complexity

3. Multi-Hop: BATS Codes
   - Random Linear Network Coding
   - BATS Codes

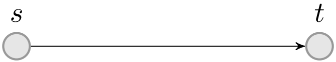
File Transmission through Packet Networks



Network features

- Many wireless links
- Loss due to interference/fading
- Limited feedbacks
- Node capability constraint
- Multiple destinations
- ...

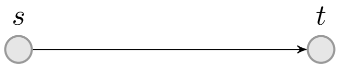
Single-hop Network



The network link has a packet loss rate 0.2.

- Capacity: $1 - 0.2 = 0.8$.
- Capacity achieving approaches:
  - retransmission
  - forward error correction

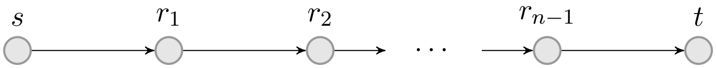
Single-hop Network



The network link has a packet loss rate 0.2.

- Capacity: $1 - 0.2 = 0.8$.
- Capacity achieving approaches:
  - retransmission
  - fountain codes

Multi-hop Networks



All links have a packet loss rate 0.2.

Intermediate Operation	Maximum Rate
forwarding	$0.8^n \to 0$
network coding	0.8

Outline

- Fountain codes and BATS codes
    - rateless
    - capacity achieving
    - low encoding/decoding complexity
    - (for BATS) low network coding complexity
- BATS Protocol
    - real-world issues
    - experimental results

Outline

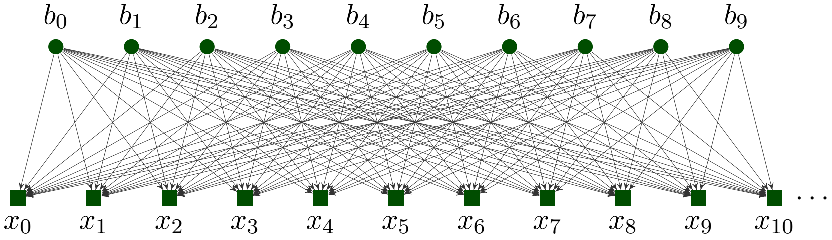
What are fountain codes?

- Transmit a file of K packets: $b_1, b_2, \ldots, b_K \in \mathbb{F}_q^T$.
- Encoder generates potentially infinite number of coded packets.
- The file can be recovered from any subset of N coded packets, where N is slightly larger than K.
- Also known as *rateless codes*.

Single-hop network

Random linear codes

- Encoding: $x_j = \sum_{i=1}^{K} \alpha_{j,i} b_i$ where $\alpha_{j,i}$ are randomly chosen from $\mathbb{F}_q$.
- Coefficient vector: $[\alpha_{j,1}, \alpha_{j,2}, \ldots, \alpha_{j,K}]^\top$.
- Decoding: collects K coded packets with linearly independent coding vectors.

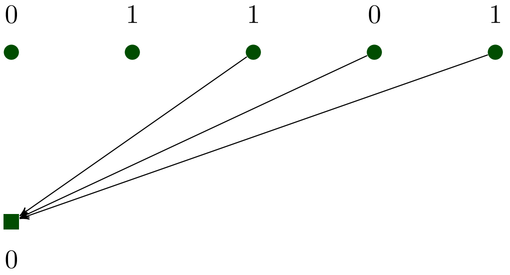


Classes of fountain codes

- Complexities of random linear codes
  - Encoding: $O(KT)$ per packet
  - Decoding: $O(K^2 + KT)$ per packet
- LT codes (Luby 1998): $O(T \log K)$ per packet
- Raptor codes (Shokrollahi 2000): $O(T)$ per packet

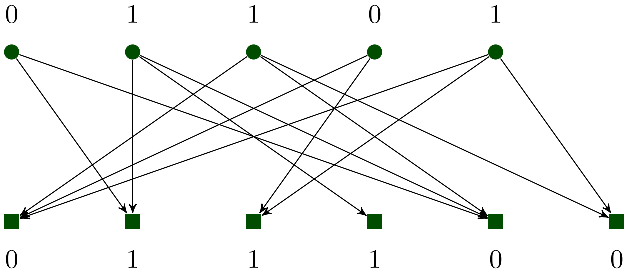
LT codes: encoding

1. pick a degree d by sampling a degree distribution $\Psi = (\Psi_1, \Psi_2, \ldots, \Psi_K)$.
2. uniformly at random pick d input packets.
3. generate a coded packet by linearly combinate of the d input packets.



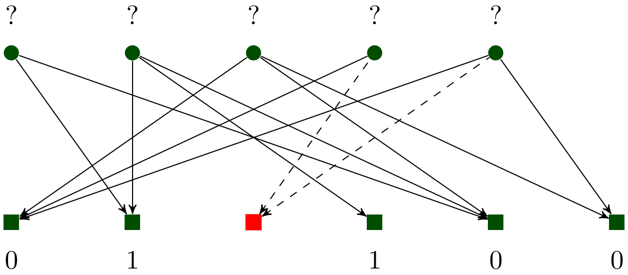
LT codes: encoding

1. pick a degree d by sampling a degree distribution $\Psi = (\Psi_1, \Psi_2, \ldots, \Psi_K)$.
2. uniformly at random pick d input packets.
3. generate a coded packet by linearly combinate of the d input packets.
4. repeat 1 - 3.



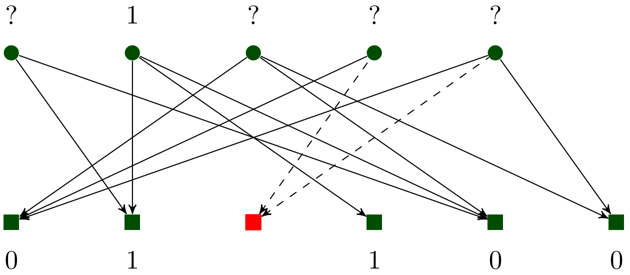
Belief propagation decoding

1. find a coded packet with degree one, which recovers the corresponding input packet.
2. substitute the recovered input packet into the other coded packets that it involves.
3. repeat 1 - 2 until there is no coded packets with degree one.



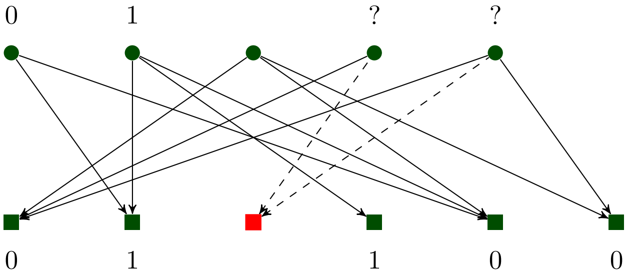
Belief propagation decoding

1. find a coded packet with degree one, which recovers the corresponding input packet.
2. substitute the recovered input packet into the other coded packets that it involves.
3. repeat 1 - 2 until there is no coded packets with degree one.



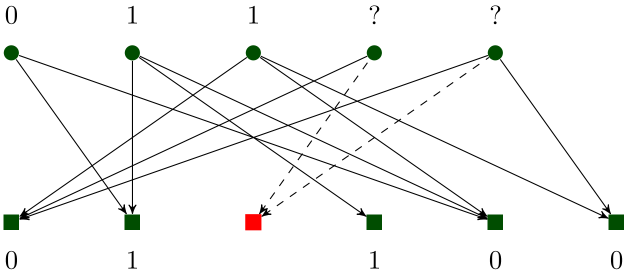
Belief propagation decoding

1. find a coded packet with degree one, which recovers the corresponding input packet.
2. substitute the recovered input packet into the other coded packets that it involves.
3. repeat 1 - 2 until there is no coded packets with degree one.



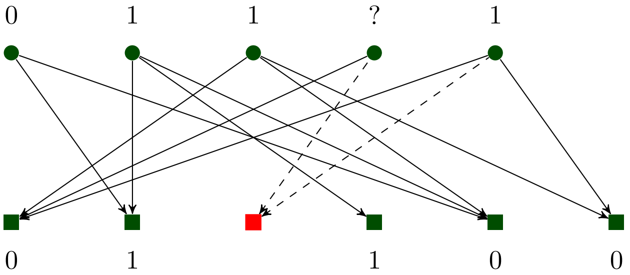
Belief propagation decoding

1. find a coded packet with degree one, which recovers the corresponding input packet.
2. substitute the recovered input packet into the other coded packets that it involves.
3. repeat 1 - 2 until there is no coded packets with degree one.



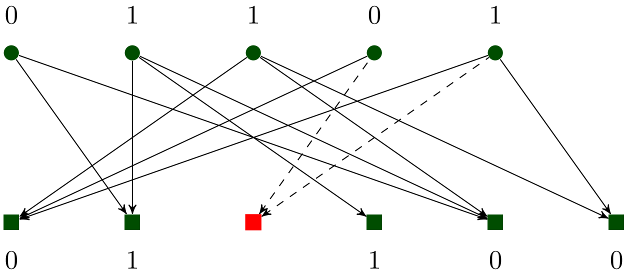
Belief propagation decoding

1. find a coded packet with degree one, which recovers the corresponding input packet.
2. substitute the recovered input packet into the other coded packets that it involves.
3. repeat 1 - 2 until there is no coded packets with degree one.



Belief propagation decoding

1. find a coded packet with degree one, which recovers the corresponding input packet.
2. substitute the recovered input packet into the other coded packets that it involves.
3. repeat 1 - 2 until there is no coded packets with degree one.



Degree distribution of LT codes

Proposition

For an LT code with K input packets and n coded packets, if there exists a decoding algorithm with $P_e \leq K^{-c}$, then $\mathrm{E}[\Psi] \geq c' \frac{K}{n} \ln K$.

- So when n is close to K, $\mathrm{E}[\Psi] \geq c' \ln K$.
- Luby showed that there exists a degree distribution such that
  1. $\mathrm{E}[\Psi] = O(\log(K))$,
  2. the BP decoding succeeds with vanishing error probability for n coded packets, and
  3. $\frac{n-K}{K} \to 0$.

Soliton distribution

- Ideal soliton distribution

$$\begin{aligned}\rho(1) &= 1/K \\ \rho(d) &= \frac{1}{d(d-1)}, \quad d = 2, 3, \ldots, K.\end{aligned}$$

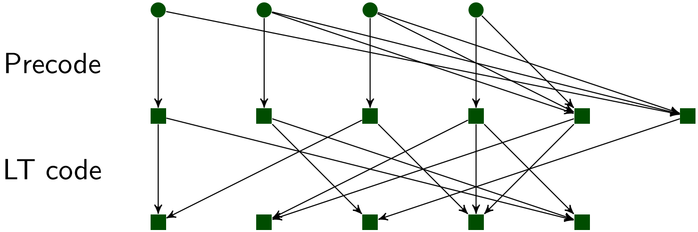
- Robust soliton distribution: $\rho(d) + \tau(d)$ with normalization

$$\tau(d) = \begin{cases} \frac{S}{K}\frac{1}{d} & \text{for } d = 1, 2, \ldots, (K/S) - 1 \\ \frac{S}{K}\log(S/\delta) & \text{for } d = K/S \\ 0 & \text{for } d > K/S \end{cases}$$

where $S = c\log(K/\delta)\sqrt{K}$.

Raptor codes

- The original inputs packets are first encoded by a precode (an erasure correction code).
- The intermediate coded packets are further encoded by an LT code (with different degree distribution from the original one).
- BP decoder recovers a fraction of the intermediate coded packets, from which the precode can recover the original input packets.



Degree distribution of Raptor codes

- BP decoding recovers at least η fraction of the (intermediate) input packets.
- The maximum degree $D \leq 1/(1-\eta)$. So $\mathrm{E}[\Psi] = O(1)$.
- The gap $\frac{n-K}{K}$ can be any positive value but is not vanishing for a fixed degree distribution when $K \to \infty$.

Performance analysis

- Asymptotic analysis: performance when $K \to \infty$.
  - Tree analysis [LMS98]
  - Differential equation approach (see [Wor99])
- Finite-length analysis: performance when K is relative small.
  - Iterative formula for the distribution of the decoder status

[LMS98] M. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of Random Processes via And-Or Tree Evaluation", in Proc. *SODA*, 1998, pp. 364–373.

[Wor99] N. C. Wormald, "The differential equation method for random graph processes and greedy algorithms," Karonsky and Proemel, eds., Lectures on Approximation and Randomized Algorithms PWN, Warsaw, pp. 73–155, 1999.

- To guarantee the success of decoding with high probability, we require

$$\Psi'(y) + \theta \ln(1-y) > 0, \quad \text{for } y \in [0, 1-\eta].$$

- Let $D = \lfloor 1/(1-\eta) \rfloor - 1$. For any $\theta < 1$, the degree distribution

$$\Psi(x) = \theta \left((1/\theta - 1)x + \sum_{i=2}^{D-1} \frac{x^i}{(i-1)i} + \frac{x^D}{D-1} \right)$$

satisfies the above requirement.

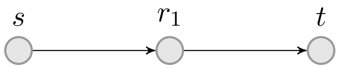
Outline

1. Outline

2. Single-Hop: Fountain Codes
   - LT Codes
   - Raptor codes: achieving constant complexity

3. Multi-Hop: BATS Codes
   - Random Linear Network Coding
   - BATS Codes

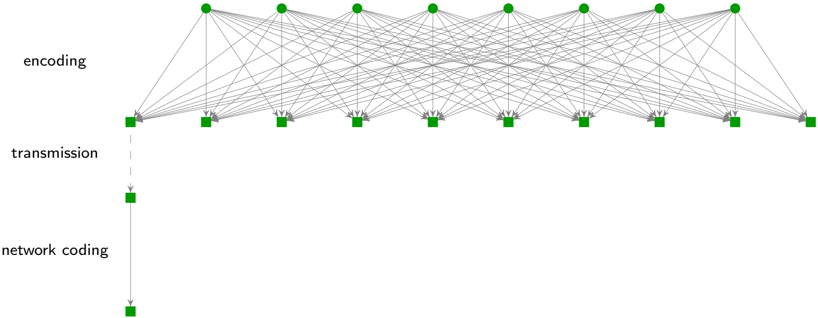
Two-hop network



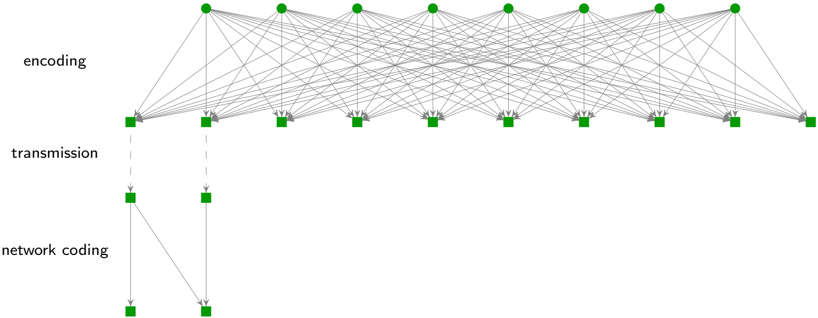
Both links have a packet loss rate 0.2.

Intermediate Operation	Maximum Rate
forwarding	0.64
network coding	0.8

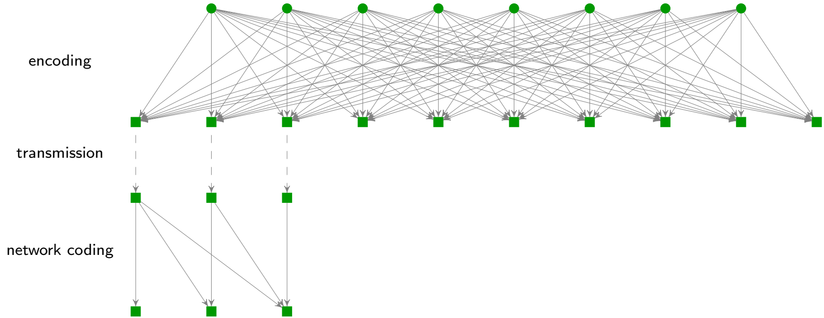
Random linear network coding



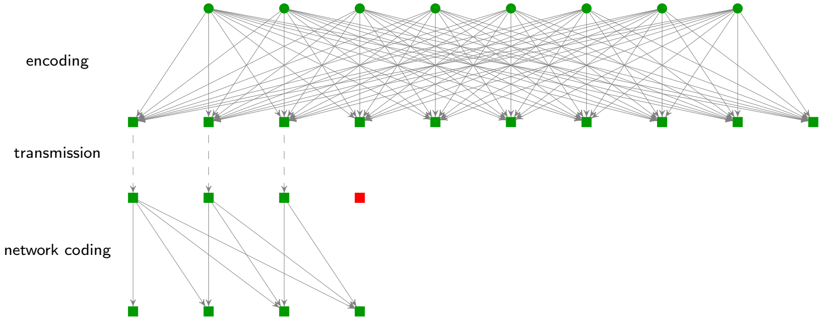
Random linear network coding



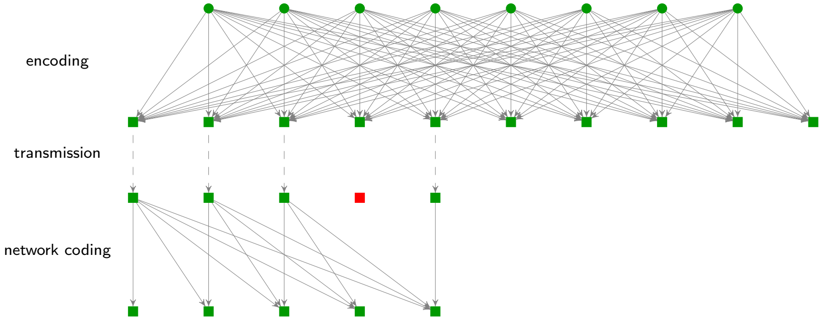
Random linear network coding



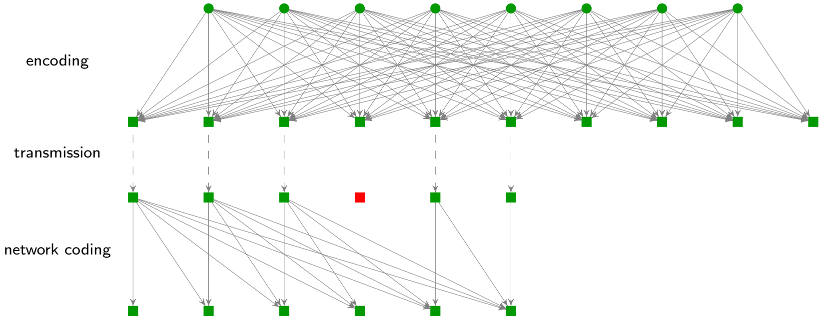
Random linear network coding



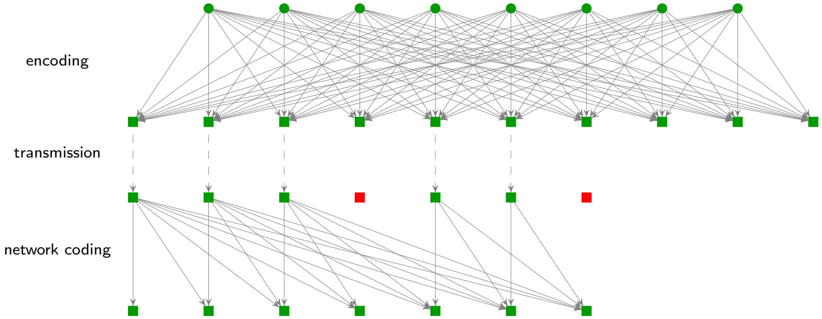
Random linear network coding



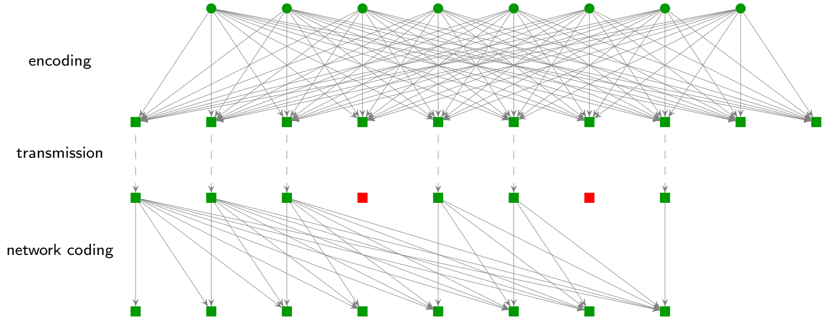
Random linear network coding



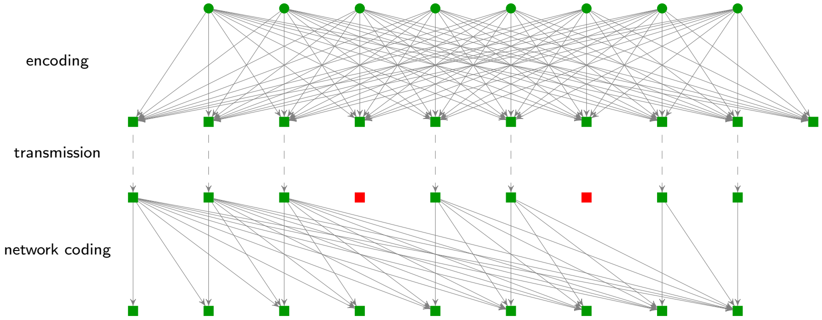
Random linear network coding



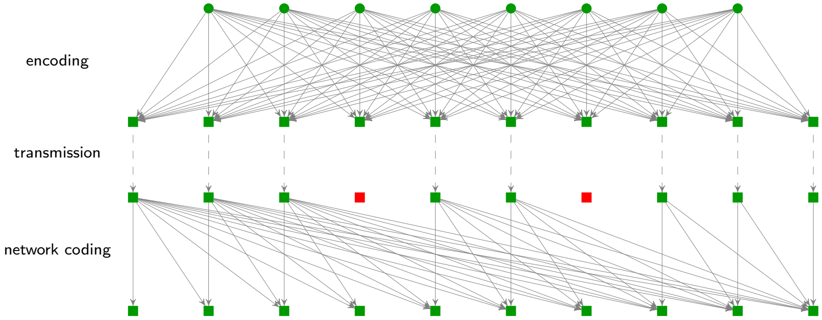
Random linear network coding



Random linear network coding



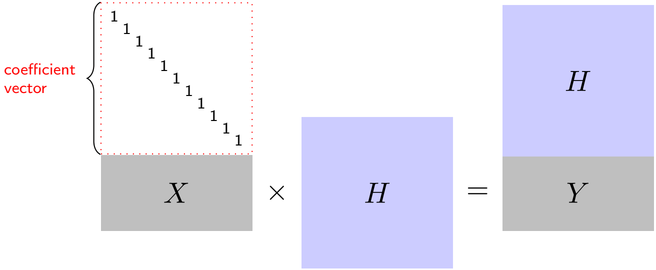
Random linear network coding



Coefficient vector overhead

$$X \times H = Y$$

Coefficient vector overhead



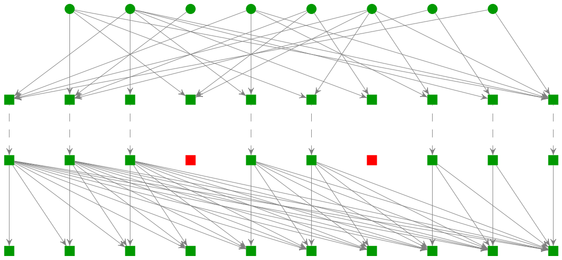
Complexity of linear network coding

- K: number of input packets
- Encoding: $\mathcal{O}(K)$ per packet.
- Decoding: $\mathcal{O}(K^2)$ per packet.
- Network coding: $\mathcal{O}(K)$ per packet. Buffer K packets.

Previous approach 1

Sparse encoding: Modifying fountain codes [PFS05, CHKS09, GS08, TF11]

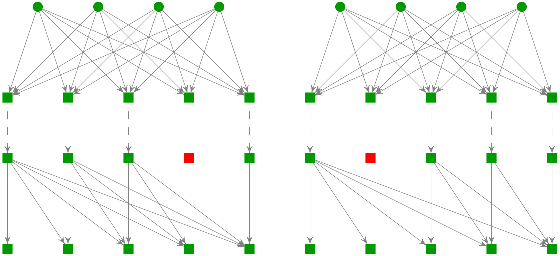
- Network coding changes the degree distribution.
- Cannot reduce Coefficient vector overhead.



Previous approach 2

Chunked encoding [CWJ03, MHL06, SZK09, HB10, LSS11]

- Disjoint chunks are not efficient.
- Heuristic designs of overlapped chunks.



New approach: coding for network coding

- BATS codes [YY11, YY14]
  - Combine fountain codes with chunks.
  - Rateless codes.
- Coding-based chunked codes [Tang12, MAB12, YT14]
  - Using LDPC codes to construct chunks.
  - Fixed-rate codes.

[YY11] S. Yang and R. W. Yeung, "Coding for a network coded fountain," ISIT 2011.

[YY14] S. Yang and R. W. Yeung, "Batched sparse codes," IEEE Trans. Inform. Theory, vol. 60, no. 9, Sep. 2014.

[Tang12] B. Tang, S. Yang, Y. Yin, B. Ye and S. Lu, "Expander graph based overlapped chunked codes", ISIT 2012.

[YT14] S. Yang and B. Tang, "From LDPC to chunked network codes", ITW 2014.

New approach: coding for network coding

- BATS codes [YY11, YY14]
  - Combine fountain codes with chunks.
  - Rateless codes.
- Coding-based chunked codes [Tang12, MAB12, YT14]
  - Using LDPC codes to construct chunks.
  - Fixed-rate codes.

[YY11] S. Yang and R. W. Yeung, "Coding for a network coded fountain," ISIT 2011.

[YY14] S. Yang and R. W. Yeung, "Batched sparse codes," IEEE Trans. Inform. Theory, vol. 60, no. 9, Sep. 2014.

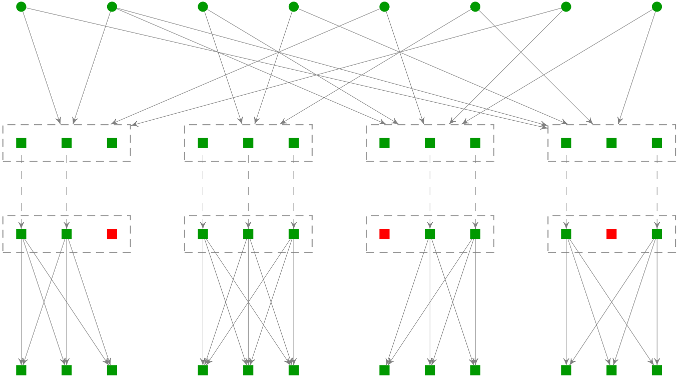
[Tang12] B. Tang, S. Yang, Y. Yin, B. Ye and S. Lu, "Expander graph based overlapped chunked codes", ISIT 2012.

[YT14] S. Yang and B. Tang, "From LDPC to chunked network codes", ITW 2014.

Batched Sparse (BATS) Codes

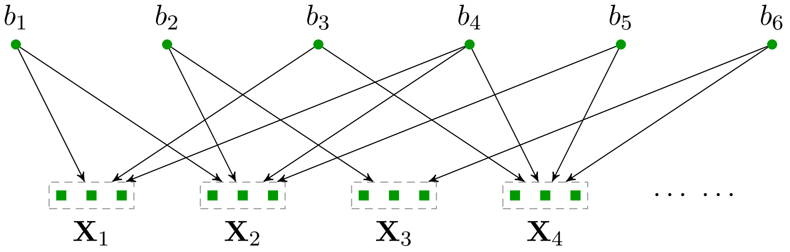
outer code

inner code
(network code)



Outer Code

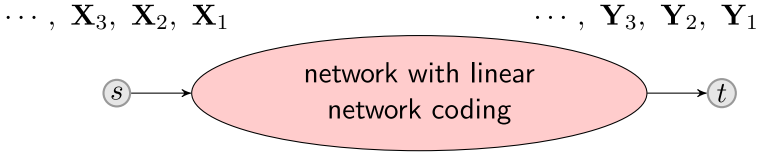
- Apply a "matrix fountain code" at the source node:
  1. Obtain a degree d by sampling a degree distribution Ψ.
  2. Pick d distinct input packets randomly.
  3. Generate a batch of M coded packets using the d packets.
- Transmit the batches sequentially.



$$\mathbf{X}_i = \begin{bmatrix} b_{i1} & b_{i2} & \cdots & b_{id_i} \end{bmatrix} \mathbf{G}_i = \mathbf{B}_i \mathbf{G}_i.$$

Inner Code

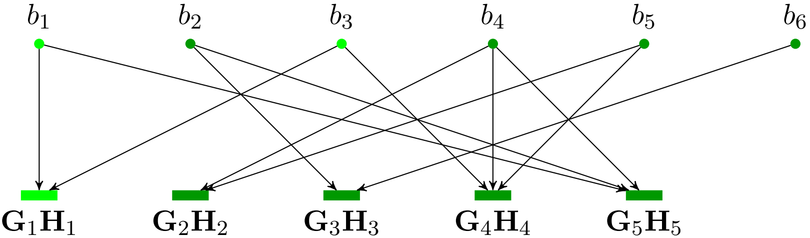
- The batches traverse the network.
- Encoding at the intermediate nodes forms the inner code.
- Linear network coding is applied in a causal manner within a batch.



$$\mathbf{Y}_i = \mathbf{X}_i\mathbf{H}_i,\ i = 1, 2, \ldots.$$

Belief Propagation Decoding

1. Find a check node i with $\text{degree}_i = \text{rank}(\mathbf{G}_i\mathbf{H}_i)$.
2. Decode the ith batch.
3. Update the decoding graph. Repeat 1).



The linear equation associated with a check node: $\mathbf{Y}_i = \mathbf{B}_i\mathbf{G}_i\mathbf{H}_i$.

Asymptotic Analysis

Theorem

Consider a sequence of decoding graph BATS(K, n, Ψ) *with constant* $\theta = K/n$. *The BP decoder is asymptotically error free if the degree distribution satisfies*

$$\Omega(x) + \theta \ln(1-x) > 0 \quad \text{for } x \in (0, 1-\eta),$$

where $\Omega(x)$ *is related to degree distribution* Ψ *and the rank distribution of the transfer matrices.*

Degree-Distribution Optimization

$$
\begin{aligned}
\max \quad & \theta \\
\text{s.t.} \quad & \Omega(x) + \theta \ln(1-x) \geq 0, \quad 0 < x \leq 1-\eta \\
& \Psi_d \geq 0, \quad d = 1, \cdots, D \\
& \sum_d \Psi_d = 1.
\end{aligned}
$$

- $D = \lceil M/\eta \rceil$
- Solver: Linear programming by sampling x.

Practical Design

- Precode: achieve constant complexity
- Inactivation decoding: reduce coding overhead when K is small
- Finite-length analysis [NY13]

[NY13] T. C. Ng and S. Yang, Finite length analysis of BATS codes, in Proc. IEEE NetCod 2013.

Complexity

Source node encoding		$\mathcal{O}(1)$ per packet
Destination node decoding		$\mathcal{O}(1)$ per packet
Intermediate Node	buffer	$\mathcal{O}(1)$
	network coding	$\mathcal{O}(1)$ per packet
Coeff. vector overhead		M symbols per packet

Achievable Rates for Line Networks