Coding for Network Coding

Lecture 8

Shenghao Yang

Institute for Interdisciplinary Information Sciences
Tsinghua University

12 Nov, 2013
1. Recall random linear network coding
2. Fountain codes with recoding
3. Batched sparse codes
4. Chunked codes
Recall random linear network coding

- **Power**: decentralized, linear operation, capacity achieving for erasure networks, ...

- **Issues**
  - Coding vector overhead
  - Computation complexity
  - Storage cost
  - ...

Complexity of Linear Network Coding

- Dense encoding: $O(TK)$ per packet.
- Gaussian elimination decoding: $O(K^2 + TK)$ per packet.
- Network coding: $O(TK)$ per packet. Buffer $K$ packets.
Reduce coding vector overhead

- Non-coherent transmission
- Coding with small chunks (or generations, batches, ...)
  - Predefined chunks
  - Online chunks
Reduce computational/storage cost

- Sparse encoding, BP decoding
- Chunk based encoding/decoding
- Limit buffer size
Outline

1. Recall random linear network coding
2. Fountain codes with recoding
3. Batched sparse codes
4. Chunked codes
Fountain Codes with Coding in Intermediate Nodes

encoding

network coding
P2P and line networks

- Network coding changes the degree distribution of the received packets such that the low decoding complexity cannot be guaranteed.
- Works for special cases: P2P file sharing [CHKS09] [TF11] and line networks [PFS05] [GS08].
  - Difficult to extend.
  - In the intermediate nodes, computational cost is $O(TK)$ per packet and storage cost is $K$ packets.

1. Recall random linear network coding
2. Fountain codes with recoding
3. Batched sparse codes
4. Chunked codes
Batched Sparse (BATS) Codes

Apply a “matrix fountain code” at the source node:

1. Obtain a degree $d$ by sampling a degree distribution $\Psi$.
2. Pick $d$ distinct input packets randomly.
3. Generate a batch of $M$ coded packets using the $d$ packets.

Transmit the batches sequentially.

$$X_i = \begin{bmatrix} b_{i1} & b_{i2} & \cdots & b_{id_i} \end{bmatrix}$$

$$G_i = B_i G_i.$$
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.

\[ Y_i = X_i H_i \]
Belief Propagation Decoding

1. Find a check node $i$ with degree $i = \text{rank}(G_iH_i)$.
2. Decode the $i$th batch.
3. Update the decoding graph. Repeat 1).

The linear equation associated with a check node: $Y_i = B_iG_iH_i$. 
A Sufficient Condition

Define

$$\Omega(x) = \sum_{r=1}^{M} \bar{h}_r \sum_{d=r+1}^{D} d\psi_d I_{d-r,r}(x) + \sum_{r=1}^{M} r\psi_r \sum_{s \geq r} \bar{h}_s,$$

where $\bar{h}_r$ is related to the rank distribution of $H$ and $I_{a,b}(x)$ is the regularized incomplete beta function.

Theorem

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

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

\[
\begin{align*}
\text{max} & \quad \theta \\
\text{s.t.} & \quad \Omega(x) + \theta \ln(1 - x) \geq 0, \quad 0 < x < 1 - \eta \\
& \quad \psi_d \geq 0, \quad d = 1, \ldots, D \\
& \quad \sum_d \psi_d = 1.
\end{align*}
\]

- \( D = \lceil M/\eta \rceil \)
- Solver: Linear programming by sampling some \( x \).
Complexity of Sequential Scheduling

<table>
<thead>
<tr>
<th>Source node encoding</th>
<th>$O(TM)$ per packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination node decoding</td>
<td>$O(M^2 + TM)$ per packet</td>
</tr>
<tr>
<td>Intermediate Node buffer</td>
<td>$O(TM)$</td>
</tr>
<tr>
<td>Intermediate Node network coding</td>
<td>$O(TM)$ per packet</td>
</tr>
</tbody>
</table>

$T$: length of a packet  
$K$: number of packets  
$M$: batch size
The optimal values of $\theta$ is very close to $E[\text{rank}(H)]$. It can be proved when $E[\text{rank}(H)] = M \Pr\{\text{rank}(H) = M\}$. 
Outline

1. Recall random linear network coding
2. Fountain codes with recoding
3. Batched sparse codes
4. Chunked codes
Using chunks to reduce complexity [CWJ03]
- Encoding complexity: $O(TKL)$
- Decoding complexity: $O(KL^2 + TKL)$

Buffer requirement in the intermediate nodes?

---

Scheduling of Chunks

- Sequential scheduling of chunks
  - Protocol overhead
  - Not scalable for multicast

Intermediate network nodes cache K packets. Less efficient when a major fraction of all the chunks have been decoded.

Scheduling of Chunks

- **Sequential scheduling of chunks**
  - Protocol overhead
  - Not scalable for multicast

- **Random scheduling of chunks** [MHL06]
  - Intermediate network nodes cache $K$ packets.
  - Less efficient when a major fraction of all the chunks have been decoded.

---

Overlapped chunked codes

- Improve the throughput of random scheduling
- Cannot reduce the buffer size

encoding

network coding


Expander graph: Design of overlapping

(a) (b) (c)

\[ C_1 = \{1, 2, 7, 10, 11\} \]
\[ C_2 = \{2, 3, 8, 12, 13\} \]
\[ C_3 = \{3, 4, 9, 14, 15\} \]
\[ C_4 = \{4, 5, 8, 16, 17\} \]
\[ C_5 = \{5, 6, 7, 18, 19\} \]
\[ C_6 = \{1, 6, 7, 20, 21\} \]

Gamma codes: LDPC in chunks
