Network Coding for Cost, Reliability and Ease of Management

Muriel Médard

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

EPFL
Overview

• Network coding
• Distributed randomized network coding
• Erasure and failure reliability
• Distributed optimization:
  – General case
  – Wireless case
• Further directions
Collaborators

- MIT LIDS: Desmond Lun, Fang Zhao, Anna Lee, Ebad Ahmed, Clifford Choute, Hyunjoo (Jenny) Lee
- MIT CSAIL: David Karger, Ben Leong
- University of Illinois Urbana-Champaign: Ralf Koetter, Niranjan Ratnakar
- California Institute of Technology: Michelle Effros, Siddharth Jaggi
- Lucent Bell Labs: Supratim Deb (previously LIDS), Tracey Ho (previously LIDS, UIUC, joining Caltech)

Supported by NSF ITR, DARPA
Network coding

- Canonical example [ACLY00]

- No longer flows, but information
Randomized network coding

• The effect of the network is that of a transfer matrix from sources to receivers
• To recover symbols at the receivers, we require sufficient degrees of freedom – an invertible matrix in the coefficients of all nodes

\[ Y_{jm} = \alpha_{jm}^i Y_{ij} + \alpha_{jm}^{hj} Y_{hj} + \alpha_{jm}^j X_j \]
Distributed random network coding

- The realization of the determinant of the matrix will be non-zero with high probability if the coefficients are chosen independently and randomly.

- Probability of success over field $F \approx 1 - \frac{1}{|F|}$

- Randomized network coding can use any multicast subgraph which satisfies min-cut max-flow bound for each receiver [HKMKE03, HMSEK03, WCJ03] for any number of sources, even when correlated [HMEK04]
Robustness to failures and erasures

- For multicast recovery, the code in the interior of the network need not be changed [KM01, HMK03]
- What about packet erasures - probabilistic link failures?
Erasure reliability

$\varepsilon_{12}$: Erasure probability on link (1, 2).
$\varepsilon_{23}$: Erasure probability on link (2, 3).

End-to-end erasure coding:
- Capacity is $(1 - \varepsilon_{12})(1 - \varepsilon_{23})$ packets per unit time.

As two separate channels:
- Capacity is $\min(1 - \varepsilon_{12}, 1 - \varepsilon_{23})$ packets per unit time.
- Can use block erasure coding on each channel. But delay is a problem.
Random erasure approach

- For erasures, correlated or not, we can in the multicast case deal with *average* flows uniquely [LME04], [LMK05], [DGPHE04]:
  - Nodes store received packets in memory
  - Random linear combinations of memory contents sent out at every transmission opportunity (without waiting for full block)

- We obtain delay expressions using in effect a generalization of Jackson networks for the innovative packets:
  - Keep track of the propagation of *innovative packets* - packets whose *auxiliary encoding vectors* (transformation with respect to the packets injected into the node’s memory) are linearly independent across cuts
  - For Poisson arrivals, propagation of innovative packets through any node forms a stable $M/M/1$ queueing system in steady-state

- Scheme can be operated *ratelessly* - can be run indefinitely until successful reception
Network coding for cost
Network coding for cost
Network coding for cost

Cost of trees = 26
Network coding for cost

Cost of network coding = 23
Relation between network coding and flows

\[(z_{ij}, x_{ij}^{(1)}, x_{ij}^{(2)})\]
Optimization

minimize $f(z)$
subject to $z \in Z$,

$z_{ij} \geq x_{ij}^{(t)} \geq 0, \quad \forall (i, j) \in A, \ t \in T,$

$$
\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} = \sigma_i^{(t)},
$$

$\forall \ i \in N, \ t \in T,$

where

$$
\sigma_i^{(t)} = \begin{cases} 
R & \text{if } i = s, \\
-R & \text{if } i = t, \\
0 & \text{otherwise.}
\end{cases}
$$

[LMHK04]
Optimization

• For any convex cost functions [LRKMA05]
• The vector $z$ is part of a feasible solution for the optimization problem if and only if there exists a network code that sets up a multicast connection in the graph $G$ at average rate arbitrarily close to $R$ from source $s$ to terminals in the set $T$ and that puts a flow arbitrarily close to $z_{ij}$ on each link $(i, j)$
• Proof follows from min-cut max-flow necessary and sufficient conditions
• Polynomial-time
• Can be solved in a distributed way
• Steiner-tree problem can be seen to be this problem with extra integrality constraints
Distributed approach

• Consider the problem

\[
\begin{align*}
\text{maximize} & \quad \sum_{t \in T} q^{(t)}(p^{(t)}) \\
\text{subject to} & \quad \sum_{t \in T} p_{ij}^{(t)} = a_{ij} \quad \forall (i, j) \in A, \\
& \quad p_{ij}^{(t)} \geq 0 \quad F^{(t)} \quad \forall (i, j) \in A, t \in T,
\end{align*}
\]

where

\[
q^{(t)}(p^{(t)}) = \min_{x^{(t)} \in F^{(t)}} \sum_{(i, j) \in A} p_{ij}^{(t)} x_{ij}^{(t)},
\]

• We have that \( F^{(t)} \) is the bounded polyhedron of points \( x^{(t)} \) satisfying the conservation of flow constraints and capacity constraints

[LRKMAL05]
# Wireline examples

<table>
<thead>
<tr>
<th>Network</th>
<th>Approach</th>
<th>Average multicast cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2 sinks</td>
</tr>
<tr>
<td>Sprint (us)</td>
<td>DST approximation</td>
<td>30.2</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>22.3</td>
</tr>
<tr>
<td>Ebone (eu)</td>
<td>DST approximation</td>
<td>28.2</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>20.7</td>
</tr>
<tr>
<td>Tiscali (eu)</td>
<td>DST approximation</td>
<td>32.6</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>24.5</td>
</tr>
<tr>
<td>Abovenet (us)</td>
<td>DST approximation</td>
<td>27.2</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>21.8</td>
</tr>
</tbody>
</table>

Obtained using Rocketfuel
Distributed approach

- Consider a subgradient approach
- Start with an iterate $p[0]$ in the feasible set
- Solve subproblem in previous slide for each $t$ in $T$
- We obtain a new updated price
- Use projection arguments to relate new price to old
- Use duality to recover coded flows from price
Distributed approach

\[ p_{ij}[n + 1] := \arg \min_{v \in P_{ij}} \sum_{t \in T} (v^{(t)} - (p_{ij}^{(t)}[n] + \theta[n]x_{ij}^{(t)}[n]))^2 \]

for each \((i, j) \in A\), where \(P_{ij}\) is the \(|T|\)-dimensional simplex

\[ P_{ij} = \left\{ v \left| \sum_{t \in T} v^{(t)} = a_{ij}, \ v \geq 0 \right. \right\} \]

and \(\theta[n] > 0\) is an appropriate step size

\(p_{ij}[n + 1]\) is set to be the Euclidean projection of

\(p_{ij}[n] + \theta[n]x_{ij}[n]\) onto \(P_{ij}\)
Recovering the primal

- Problem of recovering primal from approximation of dual
- Use approach of [SC96] for obtaining primal from subgradient approximation to dual
- The conditions can be coalesced into a single algorithm to iterate in a distributed fashion towards the correct cost
- There is inherent robustness to change of costs, as in classical distributed Bellman-Ford approach to routing
Wireless case

- Wireless systems have a multicast advantage
- Omnidirectional antennas: $i \rightarrow j$ implies $i \rightarrow k$ “for free”
- Same distributed approach holds, with some modification to the conditions to take into account multicast advantage without double counting transmission
Wireless results

- Random multicast connections in random networks
  - MIP algorithm of Wieselthier et al. (*MONET*, 2002)
  - Significant energy use improvement

<table>
<thead>
<tr>
<th>Network size/Approach</th>
<th>Average multicast energy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 sinks</td>
</tr>
<tr>
<td>20 nodes</td>
<td></td>
</tr>
<tr>
<td>MIP algorithm</td>
<td>30.6</td>
</tr>
<tr>
<td>Network coding</td>
<td>15.5</td>
</tr>
<tr>
<td>40 nodes</td>
<td></td>
</tr>
<tr>
<td>MIP algorithm</td>
<td>24.4</td>
</tr>
<tr>
<td>Network coding</td>
<td>14.5</td>
</tr>
</tbody>
</table>
Distributed operation

Average cost of random 4-terminal multicasts in 30-node wireless networks, using the decentralized subgraph optimization algorithms and centralized MIP algorithm. For modified primal recovery method, $N_\alpha = 30$. 
## Distributed operation

<table>
<thead>
<tr>
<th>Network size</th>
<th>No. of terminals</th>
<th>Average multicast energy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 nodes</td>
<td>2</td>
<td>26.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>31.9</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>37.7</td>
</tr>
<tr>
<td>40 nodes</td>
<td>2</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>29.3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>35.1</td>
</tr>
<tr>
<td>50 nodes</td>
<td>2</td>
<td>22.6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>27.3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>32.8</td>
</tr>
</tbody>
</table>
Mobility

Average cost of a random 4-terminal multicast in a 30-node mobile wireless network, with $N_s = 50$. For the modified primal recovery method, we used $N_a = 20$ and for the look-back primal recovery method, we used $N_a = 50$. 
Further directions

- Intersection of signal processing and network coding
- Instantiating correlated data coding [HMEK04] and decoding [CME05] – generalization of Slepian-Wolf distributed compression to the network
- Interference issues
- Creating protocols for distributed optimization – analogy with relation of Distributed Bellman-Ford and OSPF
- Convergence time issues
- Pricing issues
- Delay issues in erasure networks [LME04]
- Dynamic aspects - DP formulation [LMK05]
- Non-multicast [KRT05]
- Data dissemination, gossip [DM04]
- Limited codes [LM05]
- Robustness to Byzantine failures [HLKMEK04]
Extensions

• Can be extended to any strictly convex cost
• Primal-dual optimization
• Asynchronous, continuous-time algorithm
• Question: how many messages need to be exchanged for costs to converge?
## Wireline examples

<table>
<thead>
<tr>
<th>Network</th>
<th>Approach</th>
<th>Average multicast cost</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 sinks</td>
<td>4 sinks</td>
<td>8 sinks</td>
</tr>
<tr>
<td>Sprint (us)</td>
<td>DST approximation</td>
<td>30.2</td>
<td>46.5</td>
<td>71.6</td>
<td>127.4</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>22.3</td>
<td>35.5</td>
<td>56.4</td>
<td>103.6</td>
</tr>
<tr>
<td>Ebone (eu)</td>
<td>DST approximation</td>
<td>28.2</td>
<td>43.0</td>
<td>69.7</td>
<td>115.3</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>20.7</td>
<td>32.4</td>
<td>50.4</td>
<td>77.8</td>
</tr>
<tr>
<td>Tiscali (eu)</td>
<td>DST approximation</td>
<td>32.6</td>
<td>49.9</td>
<td>78.4</td>
<td>121.7</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>24.5</td>
<td>37.7</td>
<td>57.7</td>
<td>81.7</td>
</tr>
<tr>
<td>Abovenet (us)</td>
<td>DST approximation</td>
<td>27.2</td>
<td>42.8</td>
<td>67.3</td>
<td>75.0</td>
</tr>
<tr>
<td></td>
<td>Network coding</td>
<td>21.8</td>
<td>33.8</td>
<td>60.0</td>
<td>67.3</td>
</tr>
</tbody>
</table>

Obtained using Rocketfuel