# An Overview of the Use of Distributed Mechanisms in Network Coding

**Muriel Médard**

**Laboratory for Information and Decision Systems**

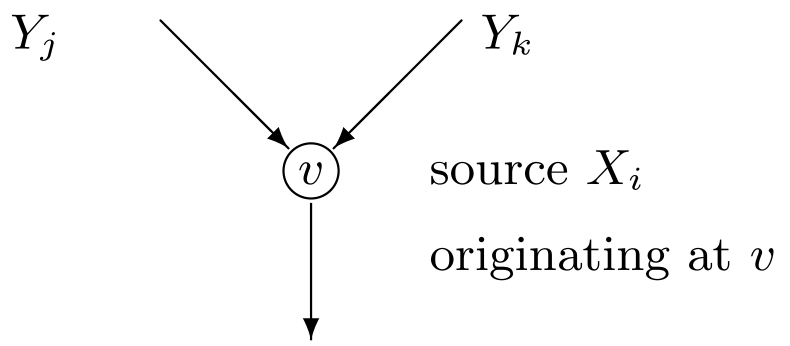
**Massachusetts Institute of Technology**

**Deb, Effros, Ho, Karger, Koetter, Lun, Médard, Ratnakar**

## Distributed Methods for Multicast Network Coding

- Can we build codes in a distributed manner?
- Can we have a distributed deployment of network coding that is cost efficient?
- How can we disseminate information in the absence of source information?

# Randomized Network Coding for Multicast




$$Y_l = a_{1,3}X_i + f_{1,3}Y_j + f_{2,3}Y_k$$

Determining feasibility: min-cut max-flow bound satisfied for each receiver [ACLY00]

For a feasible $d$-receiver multicast for independent or linearly correlated sources [HKMKE03, HMSEK03, WCJ03]

- choose code coefficients $a_{i,j}$, $f_{l,j}$ for $\eta$ links independently and uniformly over $F_q$
- success probability is at least $(1 - d/q)^\eta$ for $q > d$.

## Randomized Network Coding

- Randomized network coding can use any subgraph which satisfies min-cut max-flow bound for each receiver
- Receiver nodes can decode if they receive as many independent linear combinations as the number of source processes
- Differs from traditional networking approaches which first do source/diversity coding followed by routing of coded information
- Closely related to random codes for compression

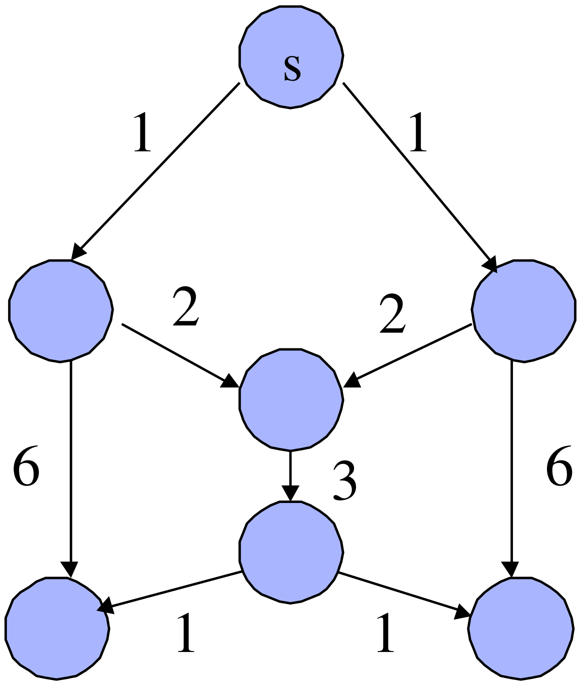
## Random Coding and Robustness

- For multicast recovery, the random code in the **interior** of the network need not be changed [KM01, HMK03]
- Robustness to corruption - what happens when a compromised node can transmit nefarious signals? Randomized distributed network coding can be used to achieve Byzantine modification detection using a simple polynomial hash functions included in transmitted packets. The modifications are detected with high probability [HLKMEK04]

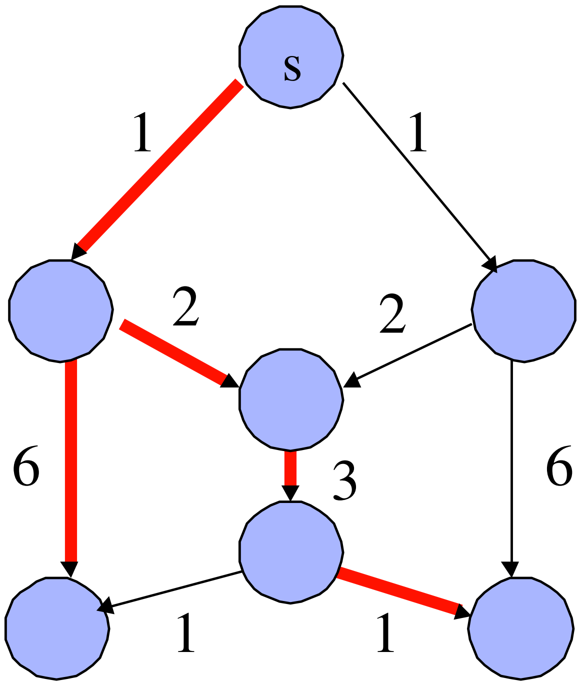
# Network Coding for Cost

- Random coding in effect decouples routing decisions and code selection decisions in the multicast case
- Are there any true benefits to obtaining a decentralized solution for coding if the choice of subgraphs must itself be centralized?
- Could a decentralized approach, of the Bellman-Ford type, be applied, even though we are not dealing with point-to-point routes?

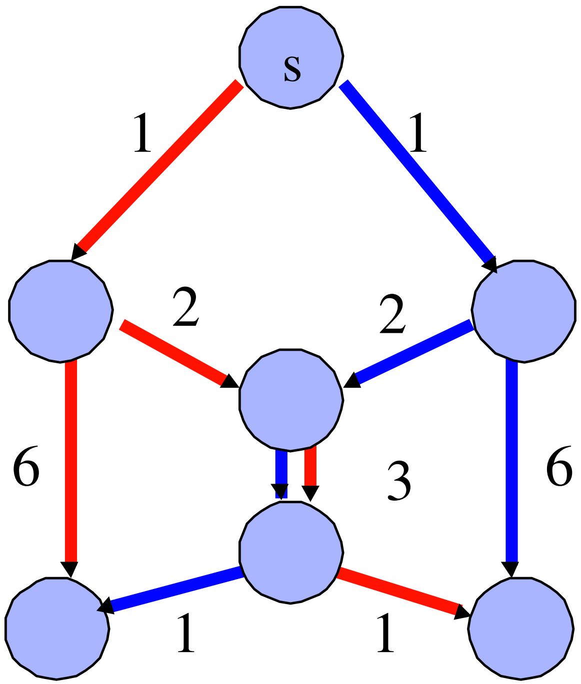
# Network Coding for Cost



# Network Coding for Cost

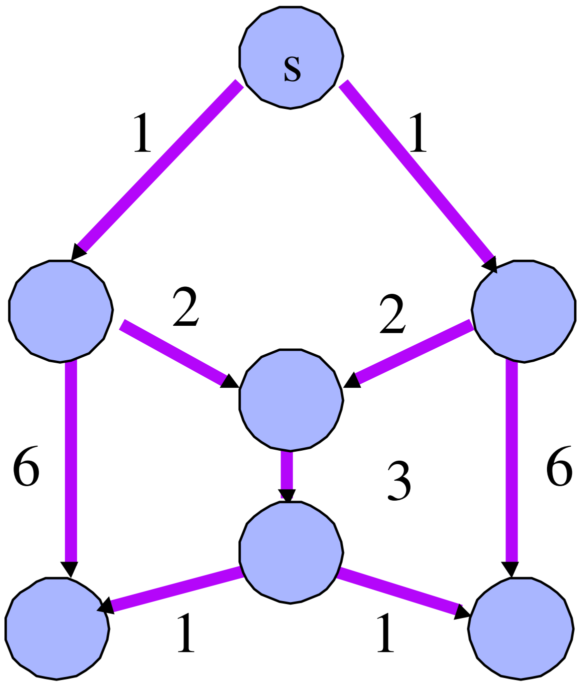


## Network Coding for Cost




Cost of trees = 26

# Network Coding for Cost




Cost of code = 23

# Network Coding for Cost

- Without coding, the problem of multicast is the Steiner tree problem over dags, possibly with decompositions into several trees
- An immediately attractive approach would be to overlay trees to create codes, attempting to increase overlaps and counting only once several uses of a link - code is built automatically
- Complexity is high and does not make use of distributed random code construction, which works well in practice
- A linear (or convex) program statement of the problem (polynomial-time) can be solved in a **distributed manner** [LMHK04, LRKMLA05]

## A LP-based Solution

$$
\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{(i,j)\in A} a_{ij} z_{ij} \\
\text{subject to} \quad & z_{ij} \geq x_{ij}^{(t)}, \ \forall\ (i,j) \in A,\ t \in T, \\
& \textstyle\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t)} =
\end{aligned}
$$

$$
\begin{cases}
R \text{ if } i = s, \\
-R \text{ if } i = t, \quad \forall\ i \in N,\ t \in T, \\
0 \text{ otherwise},
\end{cases}
\tag{1}
$$

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

## A LP-based Solution

- The vector $z$ is part of a feasible solution for the LP problem if and only if there exists a network code that sets up a multicast connection in the graph $G$ at 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 almost immediately from min-cut max-flow necessary and sufficient conditions
- Polynomial-time
- Steiner-tree problem can be seen to be this problem with extra integrality constraints

# A Distributed Approach

Consider the dual problem

$$\begin{array}{rlll} \text{maximize} & \sum_{t \in T} q^{(t)}(p^{(t)}) & & \\ \text{subject to} & \sum_{t \in T} p_{ij}^{(t)} = a_{ij} & \forall\ (i,j) \in A, & \\ & p_{ij}^{(t)} \geq 0 & \forall\ (i,j) \in A,\ t \in T, & \end{array} \tag{2}$$

where

$$q^{(t)}(p^{(t)}) = \min_{x^{(t)} \in F^{(t)}} \sum_{(i,j) \in A} p_{ij}^{(t)} x_{ij}^{(t)}, \tag{3}$$

and $F^{(t)}$ is the bounded polyhedron of points $x^{(t)}$ satisfying the conservation of flow constraints and capacity constraints

# Subgradient Approach

- Consider a subgradient approach
- Start with an iterate $p[0]$ in the feasible set
- Solve subproblem (3) for each $t$ in $T$ to obtain $x[n]$

$$p_{ij}[n+1] := \arg\min_{v \in P_{ij}} \sum_{t \in T} (v^{(t)} - (p_{ij}^{(t)}[n] + \theta[n] x_{ij}^{(t)}[n]))^2 \quad (4)$$

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

$$P_{ij} = \left\{ v \left| \sum_{t \in T} v^{(t)} = a_{ij},\ v \geq 0 \right. \right\} \quad (5)$$

  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 $P_{ij}$

## Step Size Selection

- $u := p_{ij}[n] + \theta[n]x_{ij}[n]$
- We index the elements of $T$ such that $u^{(t_1)} \geq u^{(t_2)} \geq \ldots \geq u^{(t_{|T|})}$
- Take $k^*$ to be the smallest $k$ such that

$$\frac{1}{k}\left(a_{ij} - \sum_{r=1}^{t_k} u^{(r)}\right) \leq -u^{(t_{k+1})} \tag{6}$$

or set $k^* = |T|$ if no such $k$ exists

- Projection is achieved by

$$p_{ij}^{(t)}[n+1] = \begin{cases} u^{(t)} + \frac{1}{k^*}\left(a_{ij} - \sum_{r=1}^{t_{k^*}} u^{(r)}\right) & \text{if } t \in \{t_1, \ldots, t_{k^*}\}, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

## Distributed Approach - Bringing it Together

- 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

# Application - Wireless Networks

- Omnidirectional antennas - when transmitting from node $i$ to node $j$, we get transmission to all nodes whose distance from $i$ is less than that from $i$ to $j$ "for free"
- We consider energy efficiency
- We do not consider interference (bursty set-up, for instance)
- We impose an ordering $\preceq$ on the set of outgoing links from node $i$, such that $(i, j) \preceq (i, k)$ if and only if $a_{ij} \leq a_{ik}$
- Typically, the set of outgoing links from $i$ will be the set of all nodes within a certain, fixed radius of $i$ and the cost $a_{ij}$ of the link between nodes $i$ and $j$ will be proportional to their distance raised to some power $\alpha$, where $\alpha \geq 2$

## LP for Wireless Network

- Owing to the omnidirectionality of the antennas, flow can be pushed from $i$ to $j$ by pushing it to any node $k$ such that $(i,k) \in A$ and $(i,k) \succeq (i,j)$
- Thus, the maximum flow $x_{ij}^{(t)}$ that can be pushed for a given $t$ in $T$ is

$$z_{ij} + \sum_{\{k|(i,k)\in A,(i,k)\succeq(i,j)\}\setminus\{j\}} \left(z_{ik} - x_{ik}^{(t)}\right) \quad (8)$$

- Hence

$$\sum_{\{k|(i,k)\in A,(i,k)\succeq(i,j)\}} \left(z_{ik} - x_{ik}^{(t)}\right) \geq 0 \quad (9)$$

for all $t \in T$.

## Usefulness of LP

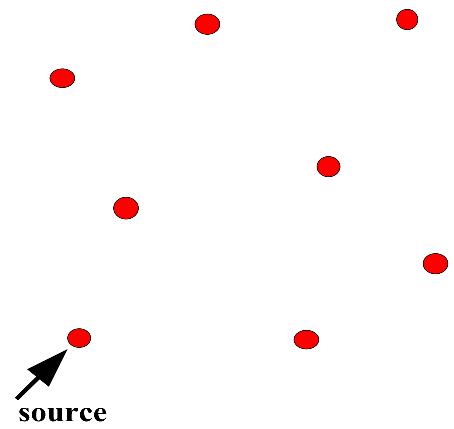
- We can extend this approach to other types of cost functions, for instant typical cost functions used to represent cost of congestion
- Can use to obtain equivalence of distances in networks, extend minimum first derivative length approaches or other convex cost considerations
- Many open issues: asynchronicity, speed of convergence, compatibility with times associated with routing-based solutions, pricing of resources and economic incentives

## Distributed Sources

Robustness to source location - what happens when the sources of traffic may not be readily identified?

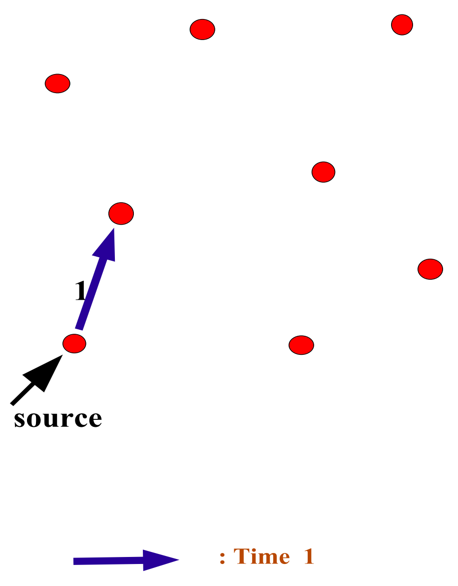
We may apply random network coding for message dissemination in networks, speeding the dissemination of $\Theta(N)$ messages from $O(Nlog(N))$ to $O(N)$ [DM04]

# The Original Gossip Problem



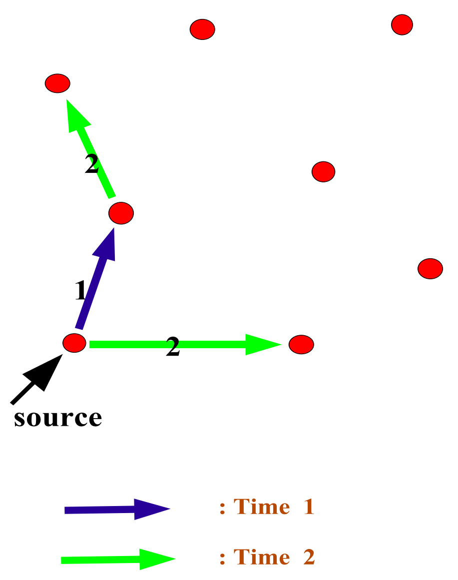

- A group of $N$ people/nodes. Somebody has a message/rumor
- How much time does it take for the rumor to disseminate completely?

# The Original Gossip Problem




- A group of $N$ people/nodes. Somebody has a message/rumor
- How much time does it take for the rumor to disseminate completely?

# The Original Gossip Problem




- A group of $N$ people/nodes. Somebody has a message/rumor
- How much time does it take for the rumor to disseminate completely?

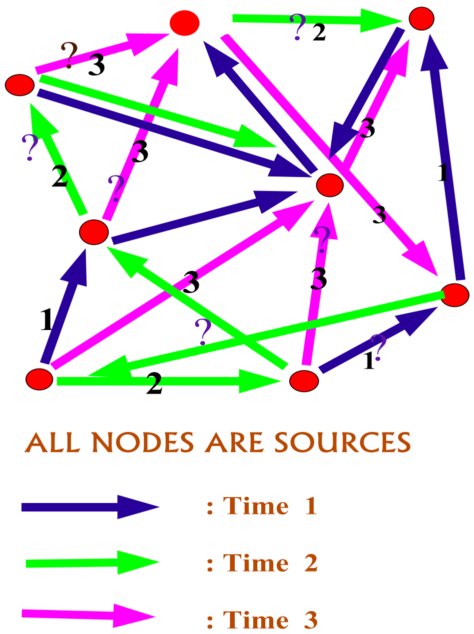
## Our Problem: The Multiple Message Case

- Suppose there are $k = \Theta(N)$ messages to start with
- Every node has only one of the $k$ messages

  Sensor network

  File downloading from distributed storage locations in the network
- Goal: To disseminate all the messages among all the nodes
- Can we do better by disseminating the messages simultaneously? How?

# Main Issue: The Multiple Message Case




- Only one message can be transmitted per communication
- Communication Protocol: What to transmit? Nodes do not know the requirements of the communication partner

## Multiple Message Dissemination: A Closer Look

- If there is an omniscient central controller that decides who transmits what, complete dissemination occurs in $\Omega(N)$ rounds
- Takes $O(\ln(N))$ time if entire data-base exchange is allowed (almost the single message framework)

  Assumes unlimited bandwidth between nodes
- Can we disseminate the messages in $O(N)$ transmissions in a decentralized manner?

  Nodes only have local knowledge

  A sequential approach takes $\Theta(N \ln(N))$ time

## A Distributed Random Coding Approach

- $O(N)$ rounds with approach, provided we allow a slight overhead with every transmission. The nodes collect linear combination of the messages, building up **degrees of freedom**
- We allow random algebraic mixing of the messages rather than treating them as mere transportation elements
- An uncoded approach does poorly because:

  Once all the nodes have $O(N)$ messages, a new message is likely to be an old one (coupon collector problem)

  Probability of getting a new message (from virtually any node) goes down with the number of messages collected

## Conclusions

- Distributed methods appear to have a natural place in network coding for multicast applications
- Network coding may in effect render several problems for multicasting more easily implemented in a distributed fashion
- Other naturally distributed settings lend themselves well to network coding approaches:
  - distributed storage
  - networks with varying costs
  - networks with erasures