Network Coding Overview

Mark Perillo

June 11, 2007
Network Coding

• In a given graph, the amount of traffic that one can transport is defined by the max-flow, min-cut theorem
Network Coding

- Traditionally, the way nodes moved data from source(s) to destination(s) is through packet “routing”
- Unfortunately, in some situations, simple packet routing does not allow you to achieve the capacity predicted by the MF/MC Theorem
Network Coding

S wants to send 2 bits to both D1 and D2
Network Coding

Unicasting: No problem
Network Coding

We would need to get bits a and b to node y so that it can forward them to D2 and D1, respectively.
But if we use network coding rather than packet forwarding, we can XOR $a$ and $b$ and meet our 2 bit traffic requirement.
Network Coding

S1 wants to send 1 bits to D1

S2 wants to send 1 bits to D2
Network Coding

There is only one way for S1 to send its bit to D1 using routing.

This disallows the S2-D2 communication.
But using network coding, we can support S1-D1 and S2-D2.
Benefits of Network Coding

• Bandwidth reduction
  – If any single destination can be supported in isolation for a given set of link capacities, then all destinations can be supported simultaneously
  – NC allows us to maximize multicast capacity
  – Fewer transmissions also means better energy-efficiency
Benefits of Network Coding

• Simplicity
  – Finding the optimal routing to support required throughput in a multicast network is NP-complete
  – Network coding solution can be found in polynomial time
Benefits of Network Coding

• Robustness
  – Consider an many-to-all broadcast network
    • How to get all source packets to all receiving nodes?
      • Shouldn’t flood (too much overhead)
      • Would rather resend probabilistically
      • If packets are routed only, we may need very high rebroadcast rates to ensure all packets are delivered to everyone: $O(N \log N)$
        • With network coding, this is reduced to $O(N)$
  – Distributed Network Storage
NC for Distributed Network Storage

• Dimakis et al., IPSN 2005.
• Wang et al., IWQoS 2006 (Partial Network Coding).
NC for Distributed Network Storage

- Dimakis et al., IPSN 2005.
- Wang et al., IWQoS 2006 (Partial Network Coding).
Distributed Network Storage

- Sensors have little memory
  - Can store ~1 packet
- Options:
  - Querying node finds data, data routed back
    - High latency?
  - Data all routed back to central node
    - Not scalable
  - Forward data throughout network with Random NC
Drawbacks of Network Coding

• Packet Latency
  – In packet routing, packets can be sent along right away
  – In network coding, an intermediate node must wait for a number of packets so that they can be combined, increasing the latency of the first packet
Drawbacks of Network Coding

• Robustness to Lost Packets
  – If one packet is lost, it could affect the decoding of a number of the original packets
Linear Network Coding

- Node receives $m$ packets $(X_1, \ldots, X_m)$ that are a combination of $n$ original packets $(M_1, \ldots, M_n)$
- Each $X_j$ has an encoding vector $(g_{j1}, \ldots, g_{jn})$
- Such that $X_j = \sum_{i=1}^{n} g_{ji}M_i$

\[G = \begin{bmatrix} 1 & 5 & 2 & 0 \\ 9 & 4 & 5 & 0 \\ 7 & 4 & 1 & 0 \end{bmatrix}\]

\[X_1 = M_1 + 5M_2 + 2M_3 + 0M_4\]

\[X_4 = 0M_1 + 7M_2 + 5M_3 + 6M_4\]
Linear Network Coding

• Intermediate node may encode using a randomly chosen $H$ matrix such that

$$X'_j = \sum_{j=1}^{m} h_{j,j} X_j$$

• New encoding vector is

$$g'_{ji} = \sum_{j=1}^{m} h_{j,j} g_{ji}$$
Linear Network Coding

• Decoding is very simple
  – We need to solve a linear system of $m$ equations and $n$ unknowns
  – We need $m \geq n$ (#original packets) linearly independent combinations
  – We can use Gaussian elimination once we have enough equations (i.e., packets)
Galois Field Math

- We must define a Galois field of size 2^s and an irreducible polynomial
  
  e.g., \( F_{2^8} \)  
  
  \[ 1 + Z^2 + Z^3 + Z^4 + Z^8 \]

- Packets are broken into symbols of s bits
- Encoding is carried out identically for each symbol (zero pad if necessary)
- Conveniently, Matlab’s Communications Toolbox is equipped with functions for Galois field operations (add, multiply, matrix inversion, etc.)
Galois Field Math

\[ M_1 = X_1 + 5X_2 \]

\[ \begin{array}{cccccccc}
6 & 45 & 62 & 88 & 12 & 51 & 156 & 112 \\
192 & 215 & 66 & 39 & 172 & 98 & 116 & 201 \\
\end{array} \]

\[ 1X_1 = (00000001) \times (00000110) = (1) \times (Z^1 + Z^2) = Z^1 + Z^2 = 00000110 = 6 \]

\[ (Z^1 + Z^2) \mod (1 + Z^2 + Z^3 + Z^4 + Z^8) = Z^1 + Z^2 = 00000110 = 6 \]

\[ 5X_2 = (00000101) \times (11000000) = (1 + Z^2) \times (Z^6 + Z^7) = Z^6 + Z^7 + Z^8 + Z^9 \]

\[ (Z^6 + Z^7 + Z^8 + Z^9) \mod (1 + Z^2 + Z^3 + Z^4 + Z^8) = (1 + Z + Z^2 + Z^5 + Z^6 + Z^7) = 11100111 = 231 \]

\[ 1X_1 + 5X_2 = 00000110 + 11100111 = 11100001 = 225 \]
Linear Network Coding

• Linear codes are optimal in multicast networks with one source for a large enough $q$ (Galois field size)

• A finite field of size $|T|$ is good enough
• Coefficients can be found in polynomial time
• But linear network codes are not optimal in general
Random Linear Coding

\[ P_{\text{success}} \geq (1 - \frac{d}{q})^v \]

- \( d \): number of receivers
  - Smaller is better
- \( v \): number of edges with independent randomized coefficients
  - Smaller is better
- \( q \): Galois field size
  - Larger is better
Random Coding

• Benefits
  – Simple, can adapt easily to network topology

• Drawbacks
  – Must include encoding matrix in packet
  – Decoding efficiency

Optimizations

c_{ij}: Channel Capacity
z_{ij}: Data actually being sent on link(i,j)
x_{ij}^{(t)}: Data being sent on link(i,j) for t’s benefit
R: traffic rate
Optimizations

• To minimize energy consumption:

\[
\text{minimize} \quad \sum_{(i,j) \in E} a_{ij} z_{ij} \\
\text{subject to} \\
z_{ij} \geq x_{ij}^{(t)}, \quad \forall (i,j) \in E \\
\sum_{j((i,j) \in E)} x_{ij}^{(t)} - \sum_{j((j,i) \in E)} x_{ji}^{(t)} = \sigma_i^{(t)}, \quad \forall i \in N, t \in T \\
c_{ij} \geq x_{ij}^{(t)} \geq 0, \quad \forall (i,j) \in E t \in T \\
\sigma_i^{(t)} = \begin{cases} 
R & i = s \\
-R & i = t \\
0 & \text{otherwise}
\end{cases}
\]

\(c_{ij}\): Channel Capacity
\(z_{ij}\): Data actually being sent on link \((i,j)\)
\(x_{ij}^{(t)}\): Data being sent on link \((i,j)\) for \(t\)'s benefit
\(R\): traffic rate

Cost of transmitting over link \((i,j)\)
Optimizations

• In wireless networks, there is an inherent multicast advantage
  – A single transmitted packet can be received by multiple receivers

• Thus, our network must be modeled as a “hypergraph” rather than a plain old graph
Hypergraph

Hyperlink \((i, J)\)

\(i = 1\)

\(J = \{2, 3, 4, 5, 6\}\)
Power Minimization for Wireless Multicast Using Network Coding

minimize \[ \sum_{(i,j) \in E} a_{ij} z_{ij} \]

subject to
\[ z_{ij} \geq \sum_{j \in J} x_{ij}^{(t)} \quad \forall (i,J) \in E, t \in T \]
\[ \sum_{\{J \mid (i,J) \in E\}} \sum_{j \in J} x_{ij}^{(t)} - \sum_{\{j \mid j \in E, i \in I\}} x_{ji}^{(t)} = \sigma_i^{(t)} \quad \forall i \in N, t \in T \]
\[ x_{ij}^{(t)} \geq 0 \quad \forall (i,J) \in E, j \in J, t \in T \]
\[ \sigma_i^{(t)} = \begin{cases} R & i = s \\ -R & i = t \\ 0 & \text{otherwise} \end{cases} \]

\(c_{ij}\): Channel Capacity
\(z_{ij}\): Data actually being sent on hyperarc \((i,J)\)
\(x_{ij}^{(t)}\): Data being sent on hyperarc \((i,J)\) and received by \(j\) for \(t\)'s benefit
\(R\): traffic rate
Optimization

• Problem: There are an exponential number of possible hyperarcs
• LP’s are only solvable in polynomial time if number of constraints, variables both polynomial
• Solution: Consider only hyperarcs in which we sequentially add next closest node
Optimization
Optimization
Optimization
Optimization

• Now we have at most $N^2$ hyperarcs
  – Less if we have a fixed number of discrete power levels
• However, we are not optimizing for receive energy up to this point
• When calculating $a_{ij}$, we must consider the receive energy for all $j$ in $J$
• But in reality, a node doesn’t have to listen just because the transmitter is sending and within range
• Options:
  – Go back to considering all possible hyperarcs
  – Reformulate the problem slightly differently
Power Minimization for Wireless Multicast Using Network Coding

minimize \[ \sum_{(i,J) \in E} a_{ij} z_{ij} + \sum_{(i,J) \in E} \sum_{j \in J} b_{ij} r_{ij} \]
subject to
\[ z_{ij} \geq \sum_{j \in J} x_{ijj}^{(t)}, \quad \forall (i,J) \in E, t \in T \]
\[ r_{ijj} \geq x_{ijj}^{(t)}, \quad \forall (i,J) \in E, j \in J, t \in T \]
\[ \sum_{\{J|j,J \in E\}} \sum_{j \in J} x_{ijj}^{(t)} - \sum_{\{j|j,J \in E, i \in I\}} x_{ijj}^{(t)} = \sigma_{i}^{(t)}, \quad \forall i \in N, t \in T \]
\[ x_{ijj}^{(t)} \geq 0, \quad \forall (i,J) \in E, j \in J, t \in T \]
\[ \sigma_{i}^{(t)} = \begin{cases} R & i = s \\ -R & i = t \\ 0 & otherwise \end{cases} \]

\( c_{ij} \): Channel Capacity
\( z_{ij} \): Data actually being sent on hyperarc \((i,J)\)
\( x_{ijj}^{(t)} \): Data being sent on hyperarc \((i,J)\) and received by \( j \) for \( t \)'s benefit
\( r_{ijj} \): Data being sent on hyperarc \((i,J)\) that is actually received by \( j \)
\( R \): traffic rate
Optimization

• Including the $r_{ijj}^{(t)}$ variable sets us up nicely to be able to solve a different optimization program: maximizing network lifetime
Lifetime Maximization for Wireless Multicast Using Network Coding

\[ \text{maximize} \quad L \]

subject to

\[ e_i \geq \sum_{\substack{J \mid i, J \in E \setminus \{J\mid J, J \in E\}}} \sum_{j \in J} r_{ij} b_{ij} + \sum_{\substack{J \mid i, J \in E \setminus \{J\mid J, J \in E\}}} z_{iJ}, \quad \forall i \in N \]

\[ z_{iJ} \geq \sum_{j \in J} x_{ij}^{(t)}, \quad \forall (i, J) \in E, t \in T \]

\[ r_{ij} \geq x_{ij}^{(t)}, \quad \forall (i, J) \in E, j \in J, t \in T \]

\[ x_{ij}^{(t)} \geq 0, \quad \forall (i, J) \in E, j \in J, t \in T \]

\[ \sum_{\substack{J \mid i, J \in E \setminus \{J\mid J, J \in E\}}} \sum_{j \in J} x_{ij}^{(t)} - \sum_{\substack{J \mid j, J \in E, i \in I \setminus \{J\mid J, J \in E\}}} x_{ij}^{(t)} = \sigma_{i}^{(t)}, \quad \forall i \in N, t \in T \]

\[ \sigma_{i}^{(t)} = \begin{cases} L & i = s \\ -L & i = t \\ 0 & \text{otherwise} \end{cases} \]

\[ c_{ij} : \text{Channel Capacity} \]
\[ z_{ij} : \text{Data being sent on hyperarc } (i, J) \]
\[ x_{ij}^{(t)} : \text{Data being sent on hyperarc } (i, J) \text{ and received by } j \text{ for } t \text{'s benefit} \]
\[ r_{ij} : \text{Data being sent on hyperarc } (i, J) \text{ that is actually received by } j \]
\[ R : \text{traffic rate} \]
\[ e_i : \text{initial energy} \]
Optimization

• Multiple sources
  – Can use the same formulation with more flow constraints
  – Does this really maximize lifetime? No
  – Optimization for multiple sources is much more complex
  – But we will at least get a better solution than our best solution just using routing
Optimization

• 1 “multicasts” to 3
• 3 “multicasts” to 1
• Energy-efficient solution:
  – 1 and 3 send data to 2
  – 2 XORs messages and broadcasts
COPE

• COPE (Coding Opportunistically):
  – Katti et al, Allerton 2005/SIGCOMM 2006

• Opportunistic Listening
  – Nodes listen to channel in promiscuous mode
  – Send packet reports periodically

• Opportunistic Coding
  – Maximize # of native packets delivered in a single transmission
  – Don’t delay for incoming packets

• Additional MAC Gain
COPE

(a) B can code packets it wants to send

B's Output Queue
P4 P3 P2 P1
C

C's Packet Pool
P4 P1

A
P4 P3
A's Packet Pool

D
P3 P1
D's Packet Pool

(b) Next hops of packets in B's queue

<table>
<thead>
<tr>
<th>Packets in B's Queue</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>A</td>
</tr>
<tr>
<td>P2</td>
<td>C</td>
</tr>
<tr>
<td>P3</td>
<td>C</td>
</tr>
<tr>
<td>P4</td>
<td>D</td>
</tr>
</tbody>
</table>

(c) Possible coding options

<table>
<thead>
<tr>
<th>Coding Option</th>
<th>Is it good?</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 + P2</td>
<td>Bad Coding (C can decode but A can't)</td>
</tr>
<tr>
<td>P1 + P3</td>
<td>Better Coding (Both A and C can decode)</td>
</tr>
<tr>
<td>P1 + P3 + P4</td>
<td>Best Coding (Nodes A, C, and D can decode)</td>
</tr>
</tbody>
</table>
COPE

• Opportunistic Listening
  – Nodes listen to channel in promiscuous mode
  – Send packet reports periodically

• Opportunistic Coding
  – Maximize # of native packets delivered in a single transmission
  – Don’t delay for incoming packets

• Additional MAC Gain
COPE

- 802.11 MAC tries to be fair
- Routing Scheme: 2 sends 2X as much traffic as 1,3
- NC: Each node sends at same rate
  - Better suited to 802.11 MAC
COPE

• Snooping on channel is a good mechanism for reducing bandwidth, but is not good for energy-efficiency
• COPE does not take advantage of all coding opportunities
• COPE does not necessarily send traffic along routes that make for good energy-efficiency
Coding-Aware Routing

• ROCX (Routing with Opportunistically Coded Exchanges):
  – Ni et al, SECON 2006

• An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing
  – Sengupta et al, Infocom, 2007
Coding-Aware Network Routing

(a) Coding-oblivious routing

(b) Coding (and interference) aware routing

coding opportunity
Coding-Aware Routing

• In real networks, there exists a tradeoff in how to forward different traffic flows
  – Route packets close to each other for coding opportunities
  – Route packets further from each other to avoid interference
Areas for Work in Network Coding

• Previous Work presented here shows
  – How to opportunistically code packet for a given routing structure
  – That this scheme is not optimal

• But no work exists that shows how to set up paths that best create coding opportunities
  – Previous work just shows optimization results
Areas for Work in Network Coding

• Once we have such distributed protocols, how do different link/node costs affect network setup decisions?