Do we really need codes?

Is the performance gap between the network coding solution and routing, i.e. packing directed Steiner trees bounded?

A simple example:

Each receiver picks out one of \( \binom{n}{k} \) possible middle layer links.
Do we really need codes?

The transfer matrix from the transmitter to the “cut” has to satisfy that $k \times k$ submatrix has full rank!

⇒ The field size is at least in the same order as $n$ (the MDS conjecture)
Theorem There exist multicast problems with \( T \) receivers such that the minimum field size required for a solution grows as \( O(\sqrt{T}) \).

Theorem There exist multicast problems such that the gap between routing and network coded strategies is arbitrarily large.
How do we find solutions for the Multicast?

\[ C = \{ (v, u_1, \mathcal{X}(v)), (v, u_2, \mathcal{X}(v)), \ldots, (v, u_K, \mathcal{X}(v)) \} \]

\( M \) is a \( |\mathcal{X}(v)| \times K|\mathcal{X}(v)| \) matrix.

\[ m_i(\xi) = \det(M_i(\xi)) \]

Choose the coefficients in \( \overline{F} \) so that all \( m_i(\xi) \) are unequal to zero.

The degree of each \( m_i(\xi) \) is at most \( |\mathcal{X}(v)| \)
An algorithm to find a vector $a$ such that $F(a) \neq 0$ for a polynomial $F$.

Input: A polynomial $F$ in indeterminates $\xi_1, \xi_2, \ldots, \xi_n$, integers: $i = 1, t = 1$

Iteration:

1. Find the maximal degree $\delta$ of $F$ in any variable $\xi_j$ and let $i$ be the smallest number such that $2^i > \delta$. 
2. Find an element $a_t$ in $\mathbb{F}_{2^i}$ such that $F(\xi)|_{\xi_t=a_t} \neq 0$ and let $F \leftarrow F(\xi)|_{\xi_t=a_t}$.

3. If $t = n$ then halt, else $t \leftarrow t + 1$, goto 2).

Output: $(a_1, a_2, \ldots, a_n)$.

The crucial step is 2) which is successful if the fieldsize is larger than the degree of $F$. 
Let \((G, C)\) be a multicast network coding problem with \(T\) receivers and \(R\) symbols transmitted per time unit. There exists a solution for \((G, C)\) over a finite field \(\mathbb{F}_{2^m}\) with

\[ m \leq \lceil \log_2(TR + 1) \rceil. \]

(A more careful analysis shows that a field \(\mathbb{F}_{2^m}\) with \(m \leq \lceil \log_2(T) \rceil\) or \(\mathbb{F} \geq T\))
Multicast:

For any multicast networking problem with $T$ receivers there always exists a solution over an alphabet which is at least as large as $T$.

Conversely:

There exist multicast networking problems with $T$ receivers such that the minimum alphabet size is bounded below by $\sqrt{T} - o(1)$.

(In practice - just try the random approach...)

A different approach...


A flow based approach that carefully constructs a solution in polynomial time.
A different approach...

A solution for acyclic networks is constructed “one link at a time” starting at the source.

Each flow to a receiver is being treated as a set of disjoint paths with the set of edges that was processed last (the frontier set) having to form a full rank matrix.
Flow I

Flow II

Frontier Sets

Frontier Sets full rank at all times

$[1 \ 0]$

$[0 \ 1]$

$[1 \ 0]$

$[0 \ 1]$
Flow I
Flow II
Frontier Sets
Frontier Sets
full rank at all times

\[ e_1 \]
\[ e_2 \]

\[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

\[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \]

find a and b such that we have
Flow I

\[
\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

Flow II

\[
\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}
\]

Flow I

\[
\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}
\]
Flow I

Flow II

Frontier Sets

15
Flow I

Flow II

[1 0]
[0 1]

[1 1]
[0 1]

Frontier Sets

full rank at all times

[1 1]
[0 1]
Flow I

Flow II

Frontier Sets

Full rank at all times
Flow I

Flow II

Frontier Sets

full rank at all times

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$
Full rank at all times

Flow I

Flow II

Frontier Sets

\[
\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = a \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]
Flow I

[1 0]
[0 1]

Flow II

[1 1]
[0 1]

Frontier Sets

20
Flow I

Flow II

Frontier Sets

Full rank at all times
Flow I

Flow II

Frontier Sets

full rank at all times
Flow I

Flow II

Frontier Sets

Full rank at all times
The algorithm of Jaggi, Sanders et al.

A multicast network

The frontier sets of a multicast to three receivers

- [a b c]
- [a' b' c']
- [a'' b'' c'']

[has full rank for all colors]
The algorithm of Jaggi, Sanders et al.

**Theorem** [Jaggi, Sanders, et al] Let $(G, C)$ be a multicast network coding problem with $E$ edges, $R$ symbols to be transmitted simultaneously and $T$ receivers. There exists a linear network coding solution for $(G, C)$ over a finite field $\mathbb{F}$ if $|\mathbb{F}| > T$. Moreover this solution can be found in time $O(E \cdot T \cdot R(R + T))$. 
The algorithm of Jaggi, Sanders et al.

**Theorem** [Jaggi, Sanders, et al.] Let \((G, C)\) be a multicast network coding problem with \(E\) edges, \(R\) symbols to be transmitted simultaneously and \(T\) receivers. There exists a linear network coding solution for \((G, C)\) over a finite field \(\mathbb{F}\) if \(|\mathbb{F}| > T\). Moreover this solution can be found in time \(O(E \cdot T \cdot R(R + T))\).

(In practice - still just try the random approach...)

The “link growth” algorithm can be modified such that it is applicable to all the generalizations of the previous session. Example: “Two-Level Multicast”
Two-Level Multicast
Two-Level Multicast

full rank for all frontier sets
Two-Level Multicast

Reencode \([a,b,c]\) as \(A[abc]\) such that

\[
\begin{bmatrix}
100 \\
001 \\
011 \\
111
\end{bmatrix}
\]

\[
\begin{bmatrix}
101 \\
011 \\
111 \\
001
\end{bmatrix}
\]
An analysis of random assignments is done in the next session.

The flow based algorithm is inherently more efficient than a pure random assignment.

How do we pack flows with as much overlap as possible?

But first: The case of bidirectional links!

Bidirectional links — A case where network coding does not help

Directional links: rate of transmission 0.5 symbols per time unit

Bi-directional links: Rate of transmission is bounded by 6/7
Steiner Tree Packing for Multicast Problems:

Find the set of all Steiner trees $\mathcal{T}$, i.e. trees connecting all receivers with a source in a multicast group.

For a link $e$ and $T \in \mathcal{T}$:

$$I(e, T) = \begin{cases} 1 & \text{e is part of } T \\ 0 & \text{otherwise} \end{cases}$$

The central problem: Find $\lambda(T) \in \mathbb{R}_+$ maximizing

$$\sum_{T \in \mathcal{T}} \lambda(T) \text{ such that } \sum_{T \in \mathcal{T}} \lambda(T)I(e, T) \leq C(e)$$

for all links $e$. 
Bidirectional links

Without network coding: One symbol per time unit

With network coding: Two symbols per time unit

Without network coding: ?

With network coding: ?

A case where network coding does help (even though it's not much)
Bidirectional links

Without network coding: One symbol per time unit

With network coding: Two symbols per time unit

Without network coding: ?

With network coding: Two symbols per time unit (min cut)

A case where network coding does help (even though it's not much)
Bidirectional links

Packing the below trees yields a rate of 1.5 symbols per time unit

(1.875 optimal [Li,Li,Lau])
Bidirectional links

Without network coding: One symbol per time unit

With network coding: Two symbols per time unit

Without network coding: 1.875 symbols per time unit

With network coding: Two symbols per time unit (min cut)

A case where network coding does help (even though it's not much)
[Li, Li, Lau] The ratio between the multicast rates achievable with or without network coding in bidirectional networks is bounded by a factor of two.
Bidirectional links

[Li,Li,Lau] The ratio between the multicast rates achievable with or without network coding in bidirectional networks is bounded by a factor of two.

(The point of network coding here is really complexity!)
With network coding we achieve a capacity of 2 symbols per time unit.

Without network coding we achieve a throughput of 1.786 symbols per unit time.

This comes at a cost of optimizing over 119104 Steiner trees [Li, Li, Lau]
Bidirectional links

He crucial step in a network coding solution for the multicast problem in bidirectional links is to find the best (bidirectional) flows corresponding to each receiver. To this end we formulate a linear program:

Each link $e$ carries two flows (direction $+$ and $-$) $f^{(\ell)}_+(e)$ and $f^{(\ell)}_-(e)$ due to receiver $\ell$. 
Maximize: $f$

Constraints for all $\ell$

\[ f^{(\ell)}_+(e) + f^{(\ell)}_-(e) \leq c(e) \]

\[ \sum_{f^{(\ell)} \text{ flowing into receiver } \ell} f^{(\ell)} = f \]

\[ \sum_{f^{(\ell)} \text{ flowing out of the source}} f^{(\ell)} = f \]

\[ \sum_{f^{(\ell)} \text{ flowing into node } i} f^{(\ell)} = \sum_{f^{(\ell)} \text{ flowing out of node } i} f^{(\ell)} \]
Summary:

- For directed networks the “coding gain” is unbounded

- We “really” need codes

- The necessary multicast fieldsize is bounded as $\sqrt{T} \leq |F| \leq T$

- Two basic methods to find solutions: algebraic and recursively assigning edges

- A natural method: “random assignment” (more about this shortly)
• For bidirectional link the coding gain is bounded by 2

• The main advantage of network coding in complexity.

• More about linear programs shortly!