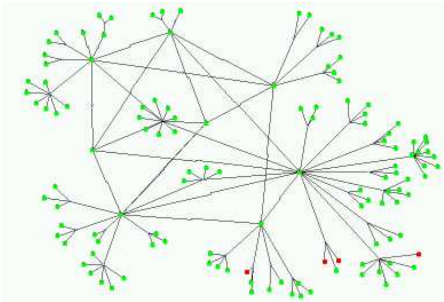
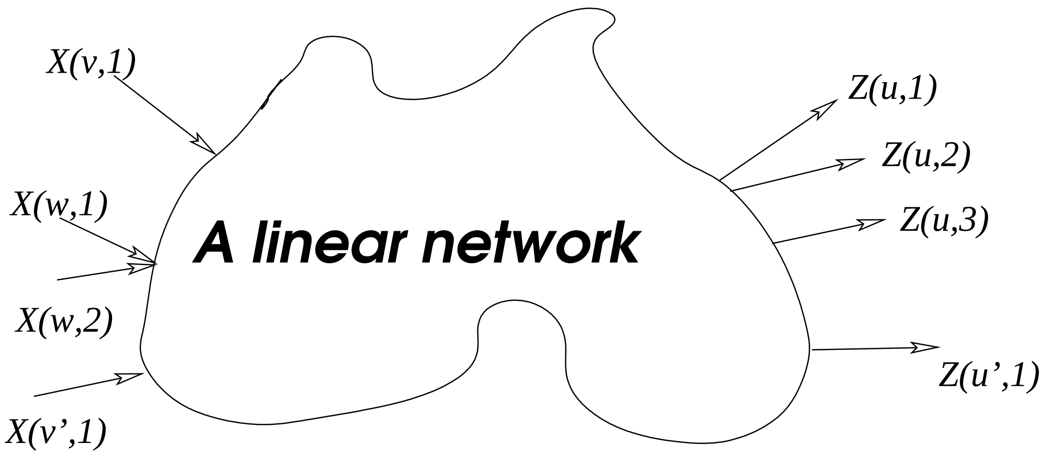
The non-multicast case some observations



Outline

- What do we know about the non-multicast case? (Ralf Koetter)
- A solution by flows (Niranjan Ratnakar)
- Online/Offline algorithms (Tracey Ho)

One reason the non-multicast case is difficult - linear network coding



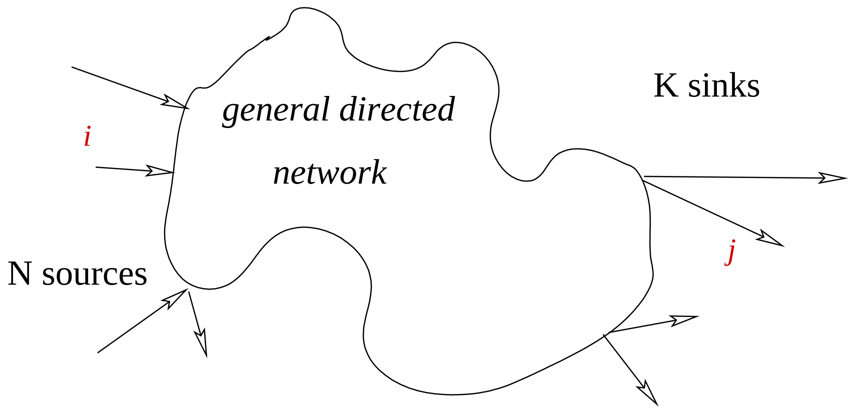
Input vector: $\underline{x}^T = (X(v,1), X(v,2), \ldots, X(v', \mu(v')))$

Output vector: $\underline{z}^T = (Z(u,1), Z(u,2), \ldots, Z(u', \nu(u')))$

Transfer matrix: M, $\underline{z} = M\underline{x}$

(Just as in encoders for convolutional codes the entries of M are polynomials or rational functions over variables $\underline{\xi} = (\xi_1, \xi_2, \ldots, (D),)$

General Problems $(\mathcal{G}, \mathscr{C})$



$$M = \begin{pmatrix} M_{1,1} & M_{1,2} & \dots & M_{1,K} \\ M_{2,1} & M_{2,2} & & M_{2,K} \\ \vdots & M_{i,j} & & \vdots \\ M_{N,1} & M_{N,2} & \dots & M_{N,K} \end{pmatrix}$$

$M_{i,j}$ corresponds to the transfer matrix between source i and sink j.

Theorem [Generalized Min-Cut Max-Flow Condition] Let an acyclic, delay-free scalar linear network problem be given and let $M = \{M_{i,j}\}$ be the corresponding transfer matrix relating the set of input nodes to the set of output nodes. The network problem is solvable if and only if there exists an assignment of numbers to $\underline{\xi}$ such that

1. $\boxed{M_{i,j} = \mathbf{0} \text{ for all source/sink pairs } (v_i, v_j) \text{ of vertices with no demand.}}$

2. If the connections (v_i, v_j) is demanded for $i \in \{i_1, i_2, \ldots, i_\ell\}$ the determinant of $\left[M^T_{i_1,j} M^T_{i_2,j}, \ldots, M^T_{i_\ell,j}\right]$ is nonzero.

The ideal of a network coding problem

Entries in $M_{i,j}$ that have to evaluate to zero: $f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_L(\underline{\xi})$

Determinants of submatrices that have to evaluate to nonzero values:
$g_1(\underline{\xi}), g_2(\underline{\xi}), \ldots, g_{L'}(\underline{\xi})$

$$\mathrm{Ideal}((\mathcal{G}, \mathscr{C})) = \langle f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_L(\underline{\xi}), 1 - \xi_0 \prod_{i=1}^{L'} g_i(\underline{\xi}) \rangle$$

$$\mathrm{Var}((\mathcal{G}, \mathscr{C})) = \{(a_1, a_2, \ldots, a_n) \in \bar{\mathbb{F}}^n :$$
$$f(a_1, a_2, \ldots, a_n) = 0 \ \forall\ f \in \mathrm{Ideal}((\mathcal{G}, \mathscr{C}))\}.$$

The central Theorem

Theorem Let a scalar linear network problem $(\mathcal{G}, \mathscr{C})$ be given. The network problem is solvable if and only if $\mathrm{Var}((\mathcal{G}, \mathscr{C})$ is nonempty or equivalently , the ideal $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ is a proper ideal of $\bar{\mathbb{F}}[\xi_0, \underline{\xi}]$, i.e $\mathrm{Ideal}((\mathcal{G}, \mathscr{C})) \subsetneq \mathbb{F}_2[\xi_0, \underline{\xi}]$.

So why is the gerenal case so much harder?

For the general case we need to find solutions to some system of polynomial equations!

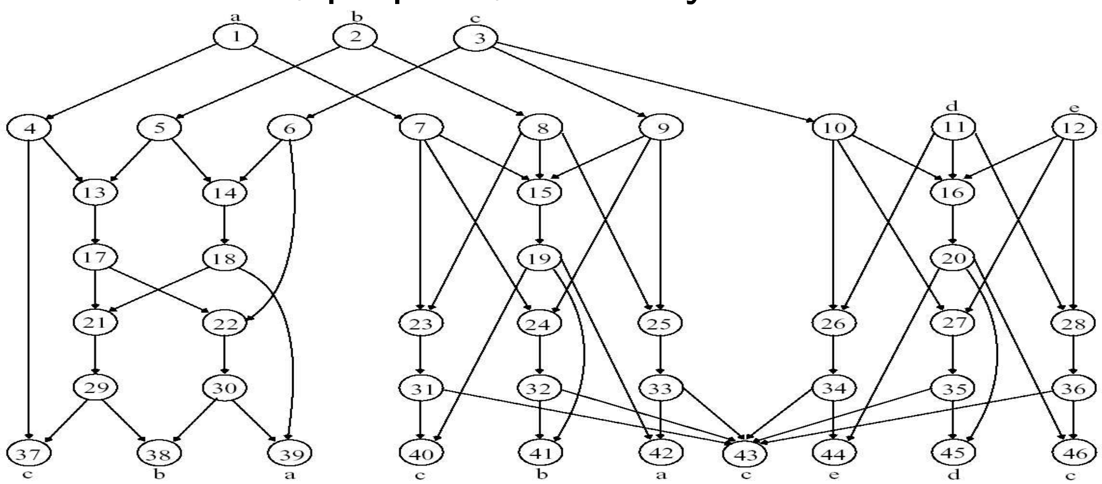
For the multicast case we need to find non solutions to some system of polynomial equations!

Another way to phrase this is: In a multicast setup everybody wants everything so the issue of interference is moot!

For the general case we may have carefully balanced solutions where some unwanted information cancels out in clever ways.....

More bad news...

R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of Linear Coding in Network Information Flow", preprint, February 2004



This network is not solvable over any Galois field, including vector versions thereof

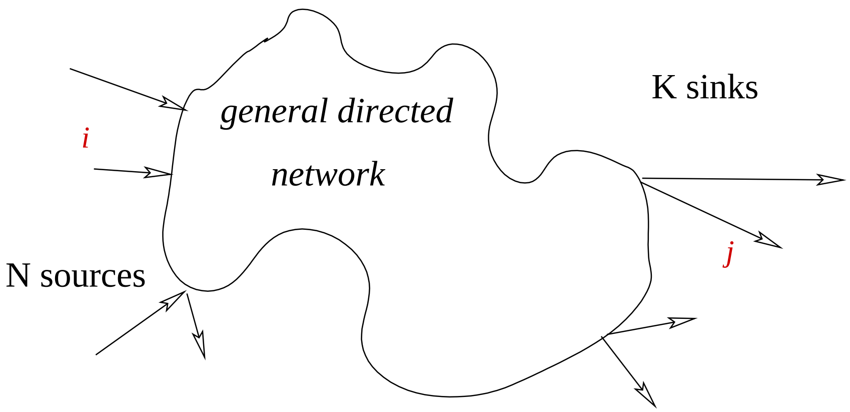
(still the network has a linear feel to it....)

Excellent news...

R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of Linear Coding in Network Information Flow", preprint, February 2004

So far we only have a collection of (very clever) countre examples — let's focus on practical constructions

Some requirements:

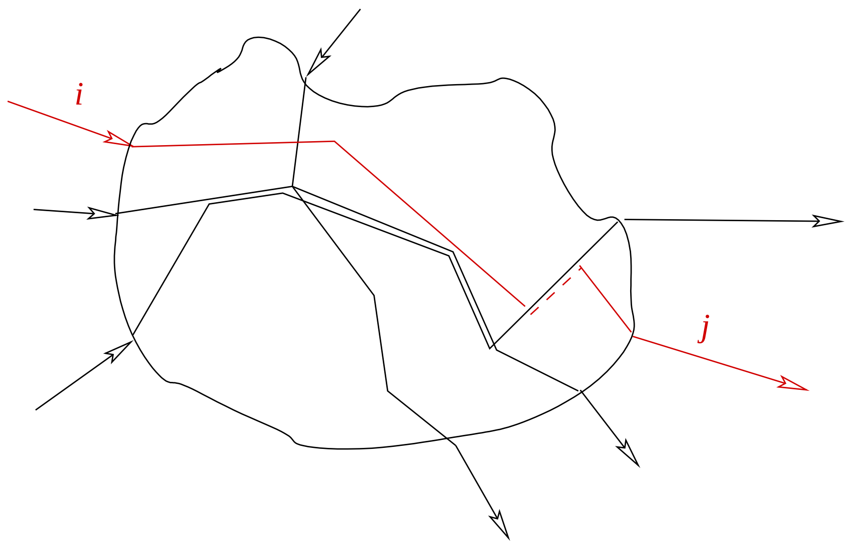


Scalability

Incremental/decremental solutions

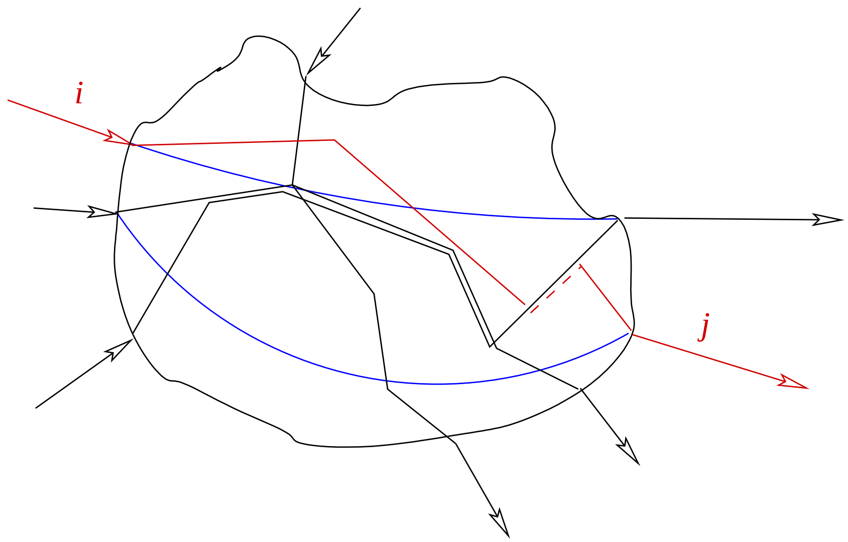
Robustness/random operation

Phasing in a new user



A new user and demand from i to j can only be accomodated if er can re-use a link in the network.

Phasing in a new user



A new user and demand from i to j can only be accomodated if er can re-use a link in the network.

The blue links provide the remedy we need to re-use a link.

Some observations and statements

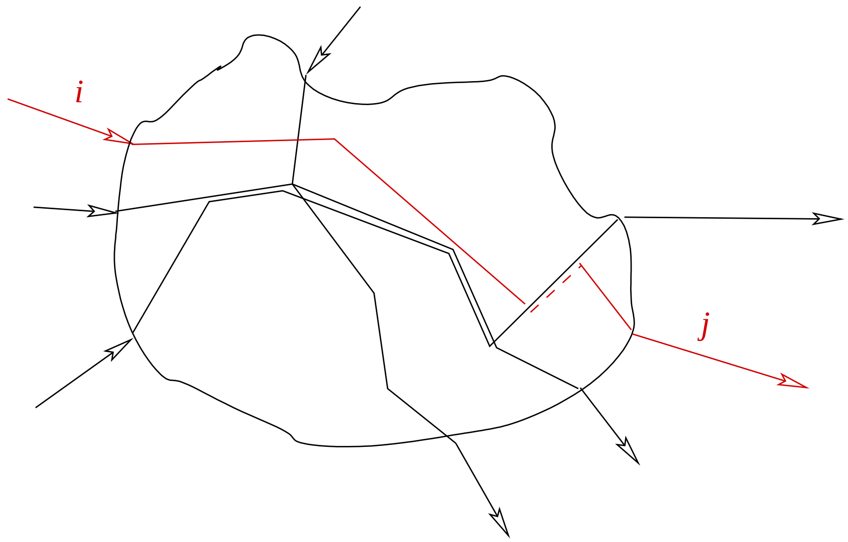
Network coding is a way to trade excess capacity in parts of the network for bottleneck capacity somewhere else.

Using an already used link comes at the price of providing other seemingly uncorrelated connections.

Network coding structures can be decomposed into these re-use and remedy patterns (with increasing levels of complexity)

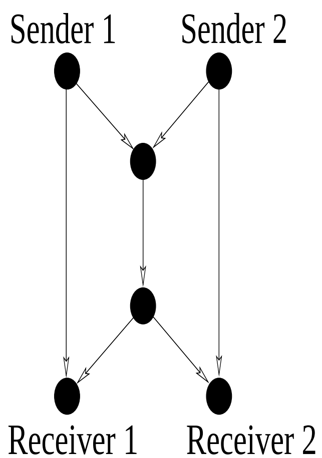
Observations for packet switched networks

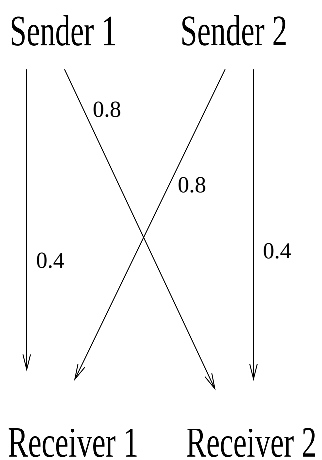
Queuing vs. Network Coding

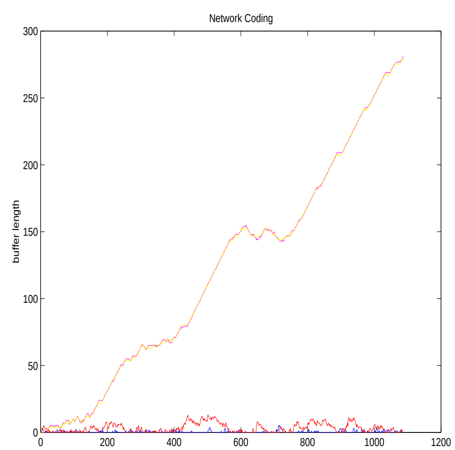


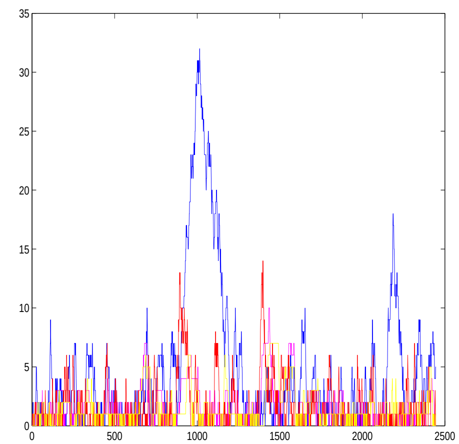
Should we queue packets or should we network-code them?

Queuing vs. Network Coding

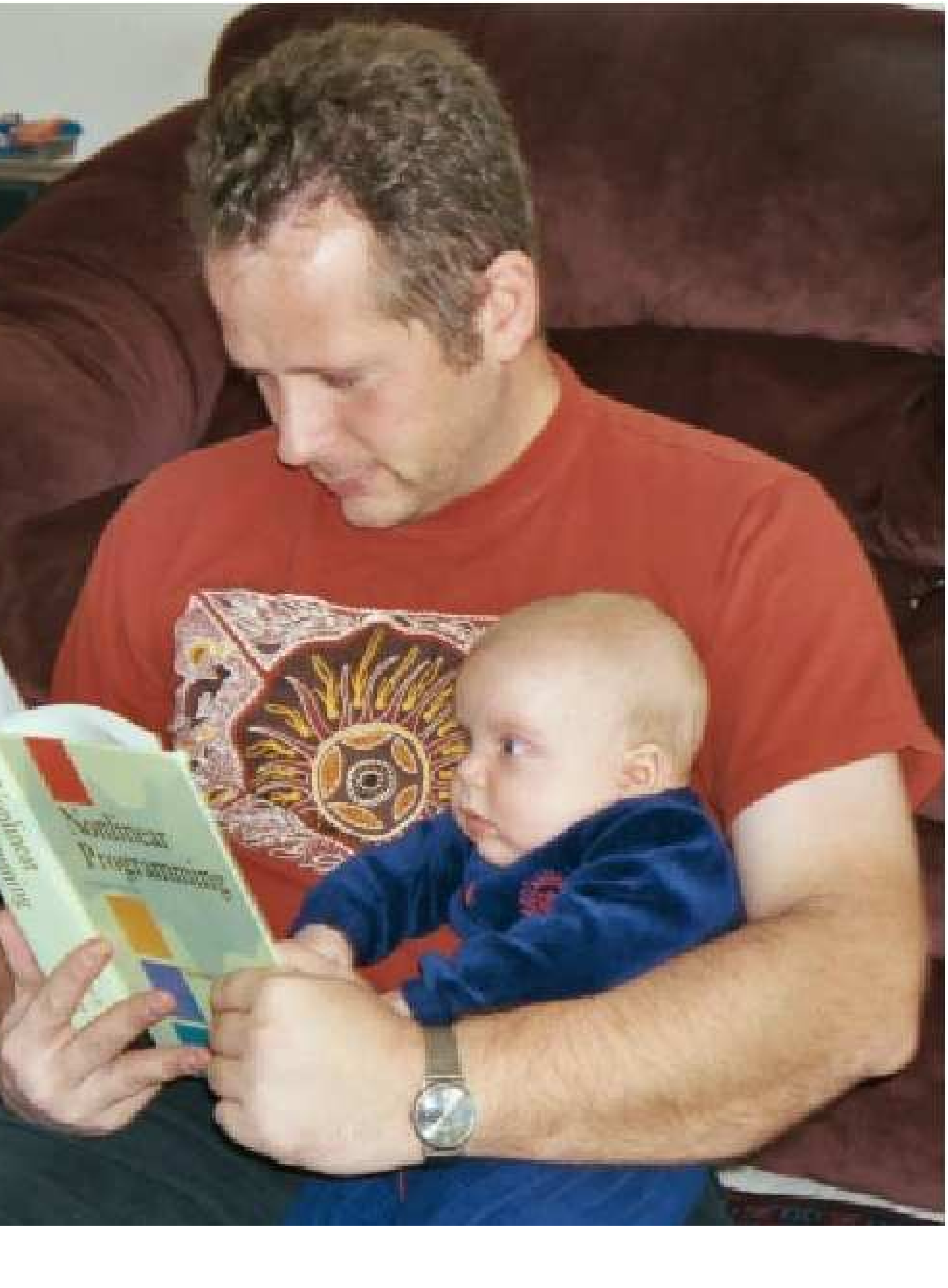








Young optimization students


Nonlinear Programming

Non-multicast connections -use of cost criterion

- We propose a linear optimization problem whose minimum cost is no greater than the minimum cost of any routing solution

- Moreover, feasible solutions correspond to network codes that perform linear operations on vectors created from the source processes

- Main idea: create a set partition of $\{1, \ldots, M\}$ that represents the sources that can be mixed (combined linearly) on links going into i.

- Code construction steps through the nodes in topological order, examining the outgoing links and defining global coding vectors on them.

Non-multicast connections -use of cost criterion

- For any node i, let $T(i)$ denote the sinks that are accessible from i
- Let $\mathcal{C}(i)$ be a set partition of $\{1, \ldots, M\}$ that represents the sources that can be mixed (combined linearly) on links going into i. For a given $C \in \mathcal{C}(i)$, the sinks that receive a source process in C by way of link (j, i) in A (set of arcs) either receive all the source processes in C or none at all.

Non-multicast connections -use of cost criterion

$$
\begin{array}{rll}
\text{minimize} & \sum_{(i,j)\in A} a_{ij} z_{ij} & \\
\text{subject to} & c_{ij} \geq z_{ij} = \sum_{C\in\mathcal{C}(j)} y_{ij}^{(C)}, & \forall\ (i,j)\in A, \\
& y_{ij}^{(C)} \geq \sum_{m\in C} x_{ij}^{(t,m)}, & \forall\ (i,j)\in A,\ t\in T,\ C\in\mathcal{C}(j), \\
& x_{ij}^{(t,m)} \geq 0, & \forall\ (i,j)\in A,\ t\in T,\ m=1,\ldots,M,
\end{array}
$$

$$
\sum_{\{j|(i,j)\in A\}} x_{ij}^{(t,m)} - \sum_{\{j|(j,i)\in A\}} x_{ji}^{(t,m)} = \begin{cases} R_m & \text{if } v = s_m \text{ and } m\in D(t), \\ -R_m & \text{if } m\in D(i), \\ 0 & \text{otherwise,} \end{cases}
$$

$$
\forall\ i\in A,\ t\in T,\ m=1,\ldots,M, \quad (1)
$$

where we define $D(i) := \emptyset$ for i in $N \setminus T$. Again, the optimization problem can be easily modified to accommodate convex cost functions.

Is the non-multicast case interesting?

Summary:

The non multicast scenario exhibits far more subtleties than the multicast setup. This is due to the fact that cancellations now need to be carefully arranged.

There are some generalizations to vector solutions which can be incorporated into the algebraic framework.

Not even the principle problem of linearity vs. nonlinear operation is entirely clear.

From a practical point of view a non interacting arrangement of multicast is most interesting and robust.