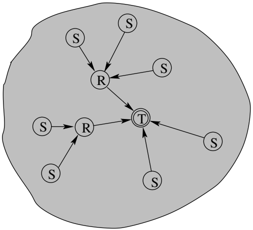
Function computation in networks: feasibility and rates

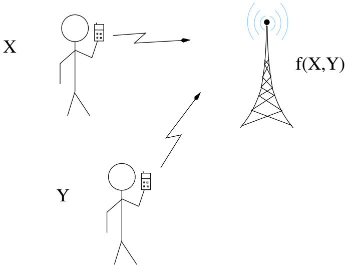
Bikash Kumar Dey

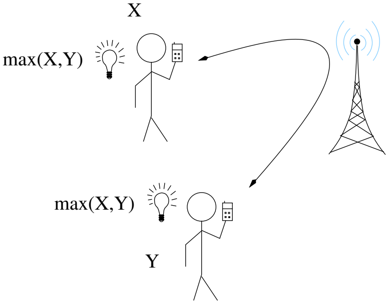
IIT Bombay

30th October, 2017

Some computation problems





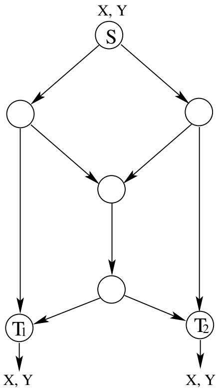


Outline of the talk

- Computing the sum function [with Brijesh Rai and Sagar Shenvi]
- Network flow for functions [with Virag Shah and D. Manjunath]

Computing the modulo-sum function

Multicasting over a network



For unicast:

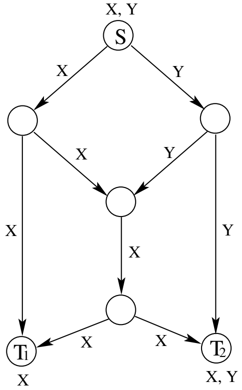
$$\text{Capacity} = \text{min} - cut\{\text{S-T}\}$$

For multicast:

$$\text{Capacity} \leq \text{min}_i\, \text{min} - cut\{S - T_i\}$$

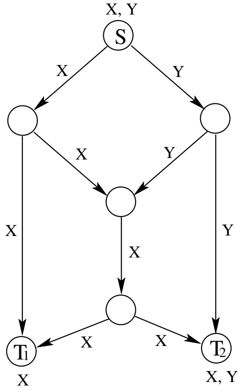
Can we multicast at rate 2?

Multicasting over a network



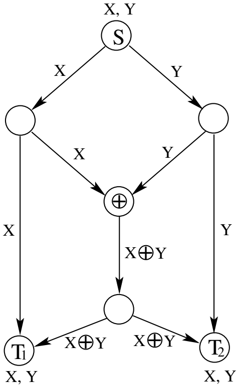
Can we multicast at rate 2?

Multicasting over a network



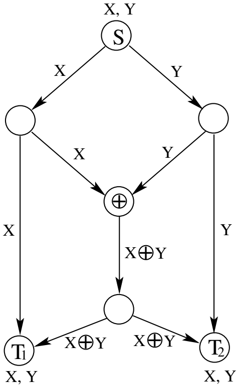
Can we multicast at rate 2?

NO!



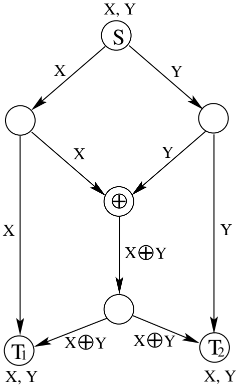
Can we multicast at rate 2?

Network coding



Can we multicast at rate 2?

YES, with network coding



Can we multicast at rate 2?

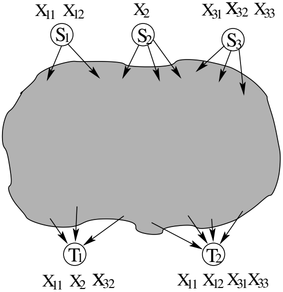
YES, with network coding

Multicast capacity $= \min_i \min -cut\{S - T_i\}$

Network coding for multicasting:

- Easy to compute capacity (unlike under routing)
- Easy to design codes
- Linear coding sufficient
- Random linear codes work

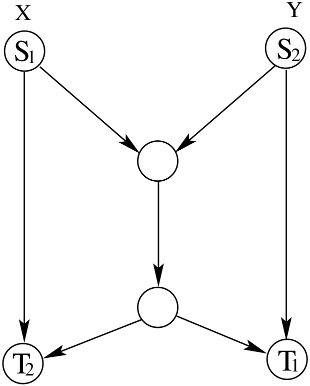
General communication networks



Capacity= $\sup\{\frac{k}{n}\}$

- Coding capacity is difficult to compute
- Linear coding is not sufficient

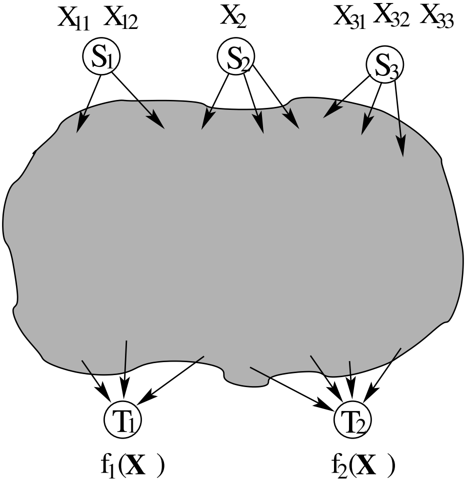
Multiple unicast networks (MUN)



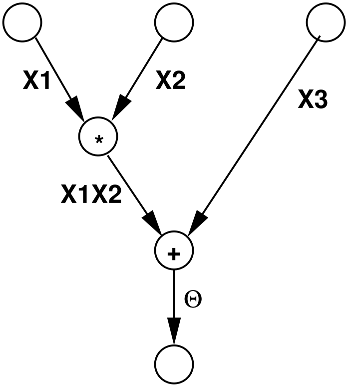
- For every network, there is an equivalent multiple unicast network.
- MUNs are as difficult as general networks.

MUNs $\Leftrightarrow$ Polynomial equations

The most general function computation problem

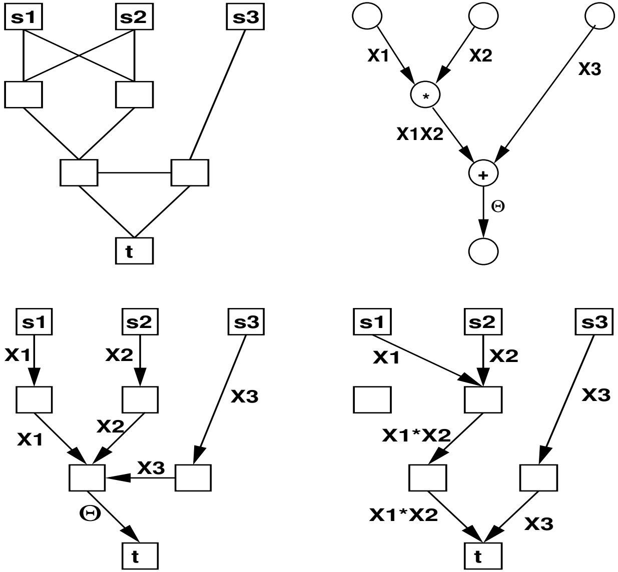


Detour: Computing function without network coding



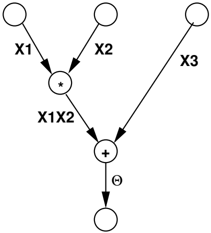
Computation tree of $\Theta(X_1, X_2, X_3) = X_1 X_2 + X_3$

Embedding


s1
s2
s3
t
X1
X2
X3
*
X1X2
+
Θ
s1
s2
s3
X1
X2
X3
X1
X2
X3
Θ
t
s1
s2
s3
X1
X2
X3
X1*X2
X1*X2
X3
t

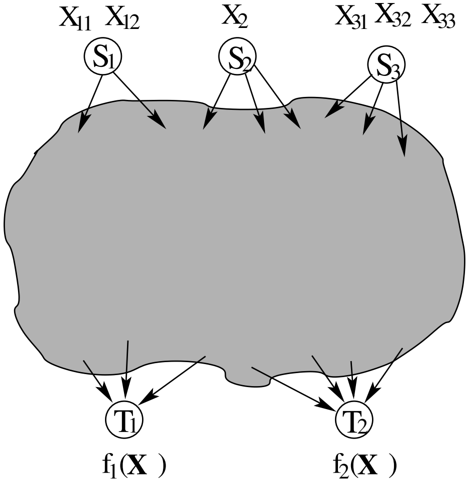
Flow for functions

- Packing of the embeddings (LP not efficient)
- Flow conservation based LP (efficient)
- More efficient algorithm using min-cost embedding

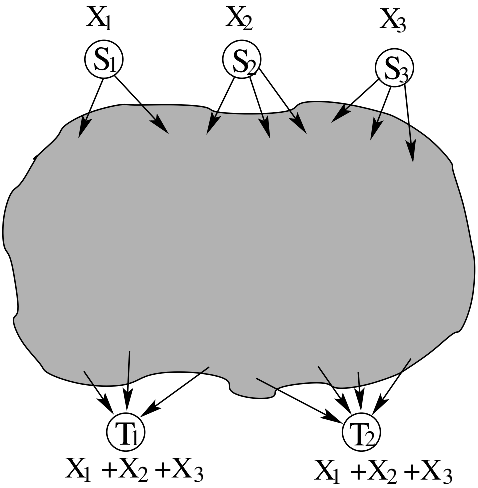


[Shah, Dey, Manjunath 2013]

The most general function computation problem



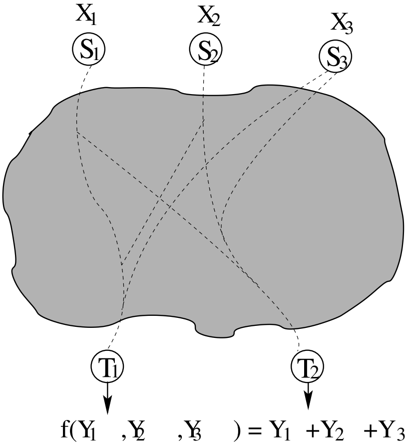
A special case: sum-networks



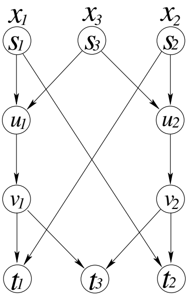
- *m* sources, *n* terminals
- Source alphabet is a field (F) or an abelian group (G)
- Linear coding over the same field

$Min\{m, n\} \leq 2$ [Ramamoorthy 2008]

The sum can be communicated if and only if every source-terminal pair is connected.

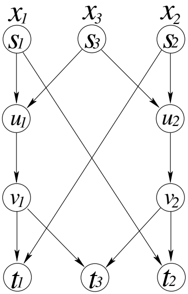


A 'connected' Network with no solution



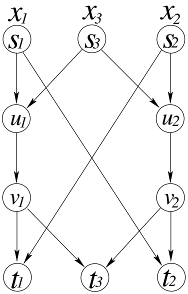
The network $\mathcal{S}_3$

Capacity of $\mathcal{S}_3 = 2/3$



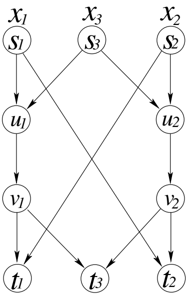
- t_3 can recover x_1, x_2, x_3.

Capacity of $\mathcal{S}_3 = 2/3$



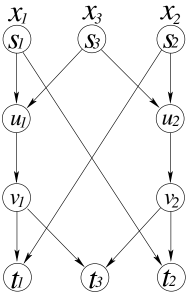
- t_3 can recover x_1, x_2, x_3.
- But $mincut(s_1, s_2, s_3; t_3) = 2$

Capacity of $\mathcal{S}_3 = 2/3$



- t_3 can recover x_1, x_2, x_3.
- But $mincut(s_1, s_2, s_3; t_3) = 2$
- So, $q^{2n} \geq q^{3k} \Rightarrow k/n \leq 2/3$.

Capacity of $\mathcal{S}_3 = 2/3$



- t_3 can recover x_1, x_2, x_3.
- But $mincut(s_1, s_2, s_3; t_3) = 2$
- So, $q^{2n} \geq q^{3k} \Rightarrow k/n \leq 2/3$.
- Achievability: ...

Capacity of $\mathcal{S}_3 = 2/3$

Achievability:

- Two sums in three slots

Capacity of $\mathcal{S}_3 = 2/3$

Achievability:

- Two sums in three slots
- Two realizations: $(X_{11}, X_{21}, X_{31}), (X_{12}, X_{22}, X_{32})$

Capacity of $\mathcal{S}_3 = 2/3$

Achievability:

- Two sums in three slots
- Two realizations: $(X_{11}, X_{21}, X_{31}), (X_{12}, X_{22}, X_{32})$
- Two sums: $S_1 = X_{11} + X_{21} + X_{31}, S_2 = X_{12} + X_{22} + X_{32}$

Capacity of $\mathcal{S}_3 = 2/3$

Achievevability:

- Two sums in three slots
- Two realizations: $(X_{11}, X_{21}, X_{31}), (X_{12}, X_{22}, X_{32})$
- Two sums: $S_1 = X_{11} + X_{21} + X_{31}, S_2 = X_{12} + X_{22} + X_{32}$
- Slot 1: $S_1 \rightarrow t_1, t_2$

Capacity of $\mathcal{S}_3 = 2/3$

Achievability:

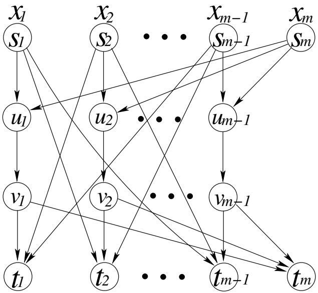
- Two sums in three slots
- Two realizations: $(X_{11}, X_{21}, X_{31}), (X_{12}, X_{22}, X_{32})$
- Two sums: $S_1 = X_{11} + X_{21} + X_{31}, S_2 = X_{12} + X_{22} + X_{32}$
- Slot 1: $S_1 \to t_1, t_2$
- Slot 2: $S_2 \to t_2, t_3$

Capacity of $\mathcal{S}_3 = 2/3$

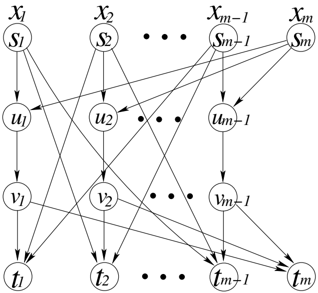
Achievability:

- Two sums in three slots
- Two realizations: $(X_{11}, X_{21}, X_{31}), (X_{12}, X_{22}, X_{32})$
- Two sums: $S_1 = X_{11} + X_{21} + X_{31}, S_2 = X_{12} + X_{22} + X_{32}$
- Slot 1: $S_1 \to t_1, t_2$
- Slot 2: $S_2 \to t_2, t_3$
- Slot 3: $S_1 + S_2 \to t_1, t_3$

Polynomial equations



Polynomial equations

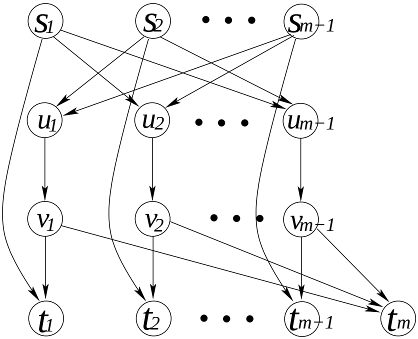


- Linearly solvable iff

$$m - 2 = 0$$

[Rai-Dey 2012]

Polynomial equations



- Linearly solvable iff

$$(m-2)X - 1 = 0 \text{ has a solution}$$

[Rai-Dey 2012]

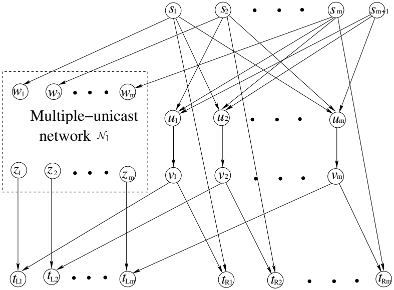
How general/rich is the class of problems?

- Difficulty of computing capacity, solvability, designing codes
- Equivalence with other classes of problems
- Sufficiency of linear codes

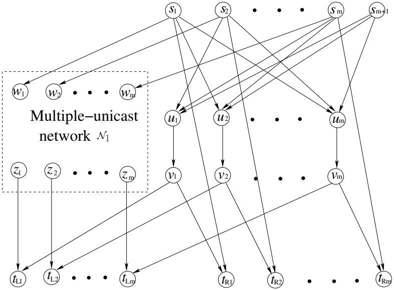
Special classes:

- Multicast networks are not general enough - no network for $f(X) = 2X - 1$.
- Multiple-unicast networks are very general $\sim$ systems of polynomials.

Equivalence with MUN [Rai-Dey 2012]

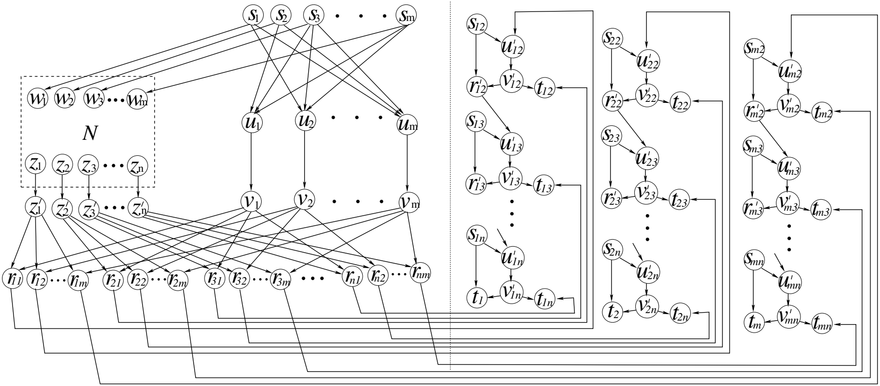


Equivalence with MUN [Rai-Dey 2012]

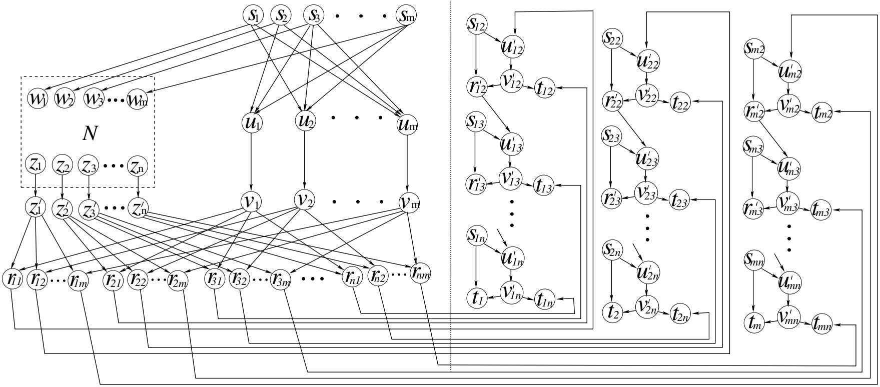


- Equivalent under linear coding
- Equivalent under non-linear coding
- Same holds for the reverse networks

Equivalence with MUN [Rai-Dey 2012]



Equivalence with MUN [Rai-Dey 2012]



- Equivalent under linear coding
- Equivalent under non-linear coding

Sum-networks as a class

- Sum-networks are as (but not more) rich a class as general communication networks
- Linear codes not sufficient
- Equivalence with polynomial equations
- Many more theoretical insights in future works [Rai and Das 2013, ...]
- Capacity can be arbitrarily less than the $min - cut$ [Tripathy and Ramamoorthy 2016] - also follows from [Rai and Dey 2012].

Thank You