Optimizing Throughput with Network Coding

Zongpeng Li, Baochun Li

Department of Electrical and Computer Engineering
University of Toronto

DIMACS Working Group on Network Coding
Rutgers University
January 28, 2005

URL: google "Baochun"

Outline

- From practice to theory
  - The problem of optimizing throughput
  - A matrix of problems
  - Does network coding really help?
- From theory to practice

The problem of optimizing throughput

(from practice to theory)

Maximizing throughput

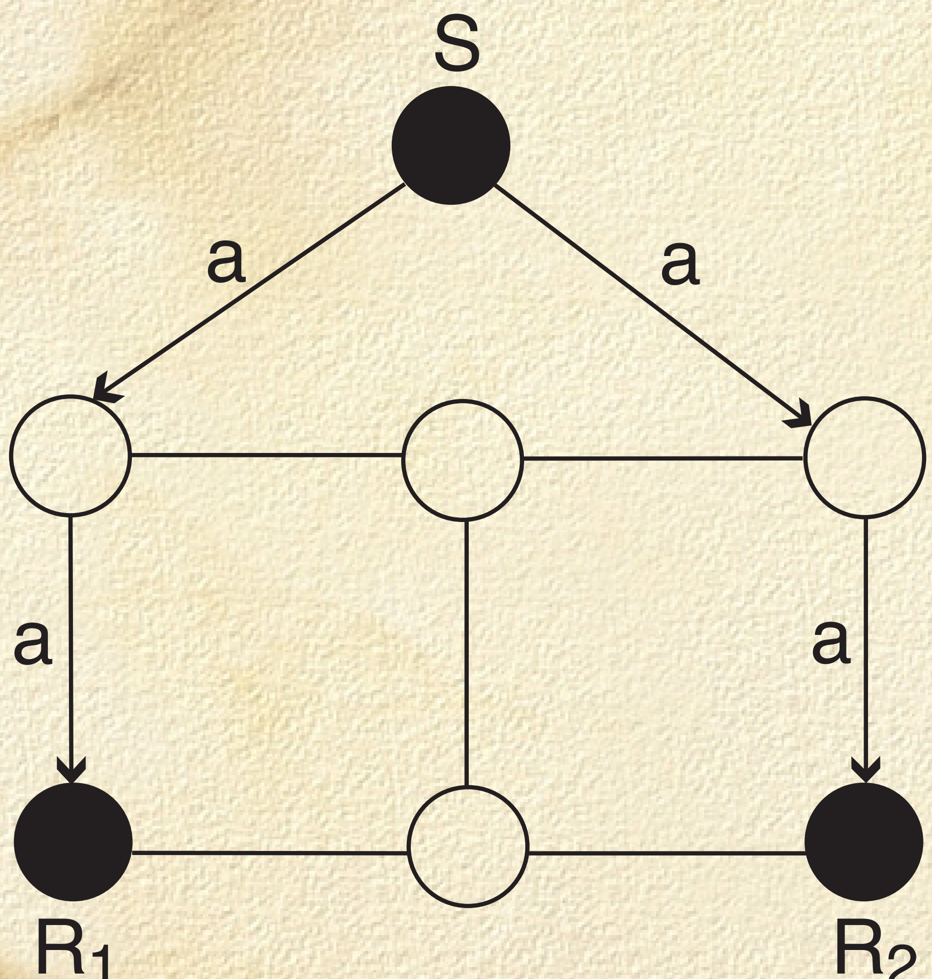
- Given an existing network topology and capacities, how to maximize throughput between the source and the receivers?
  - Past work from systems research:
    - *Digital Fountain (SIGCOMM 98 and 02):* uses fountain codes to improve throughput
    - *SplitStream (SOSP 03):* uses multiple multicast trees
    - *BitTorrent:* responsible for a fair amount of Internet traffic (rumor: 30%)

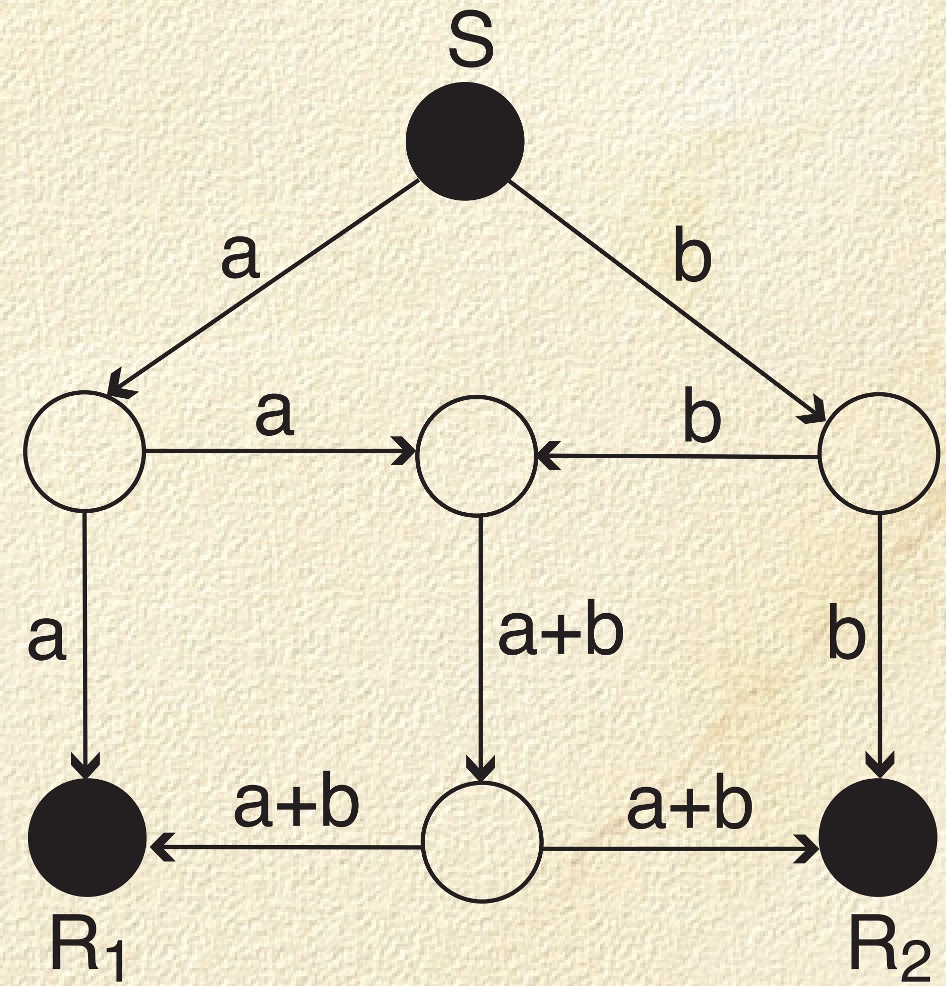
Network Flows

- The problem of maximizing throughput naturally corresponds to the problem of finding maximum flow rates in a capacitied network:
  - Single unicast session: unicommodity flows
  - Multiple unicast sessions: multicommodity flows
- But a node in a realistic network can do more than simply forwarding data
  - Replicating data for multiple downstream nodes: *multicast*
  - Encoding and decoding data: *network coding*

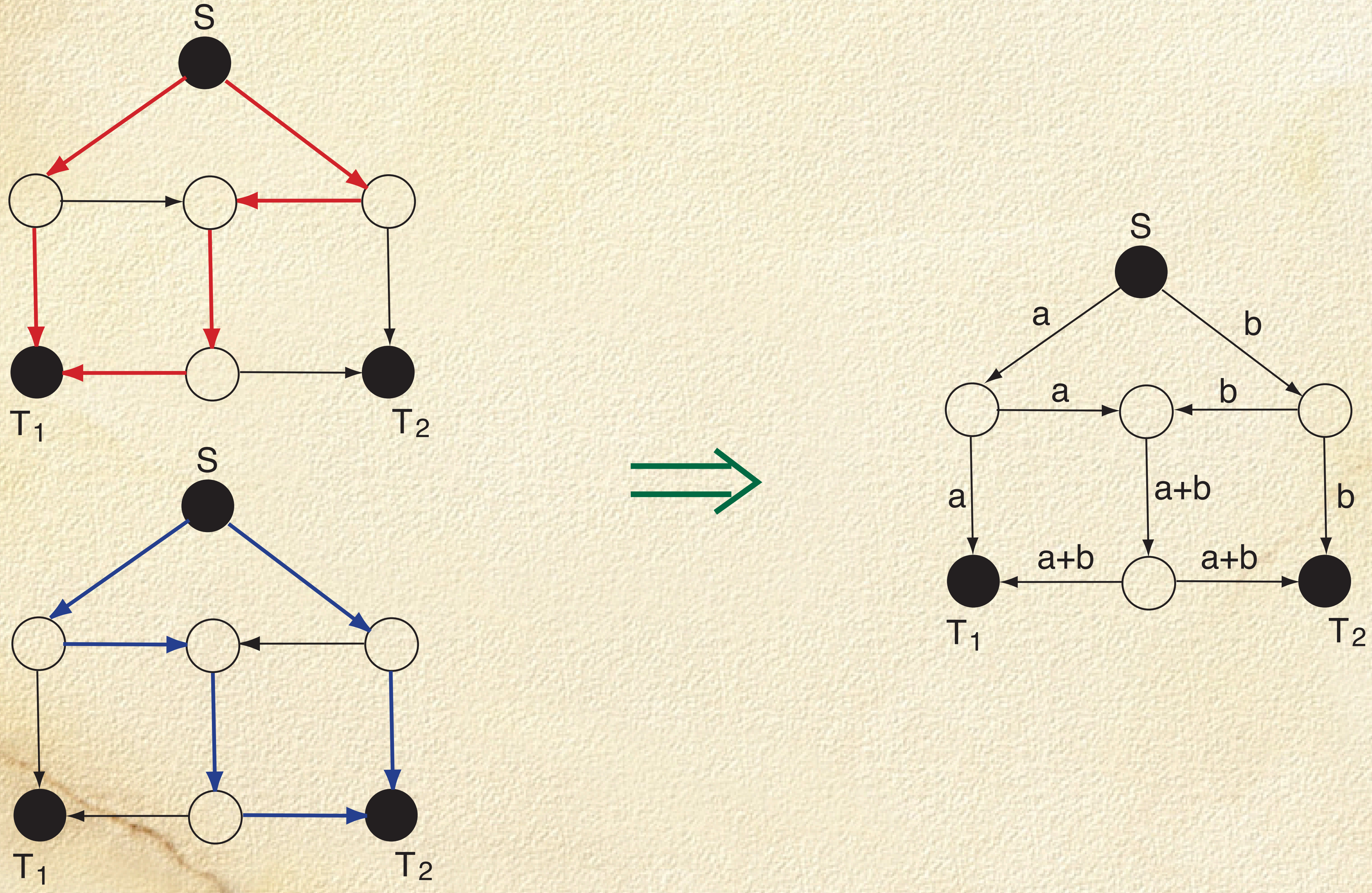
	topology confinement	respect link cap.	replicable	encodable
info. flow	yes	yes	yes	yes
fluid flow	yes	yes	no	no

Given a source and a group of receivers, what is the maximum throughput one can achieve in a network topology with known link capacities?



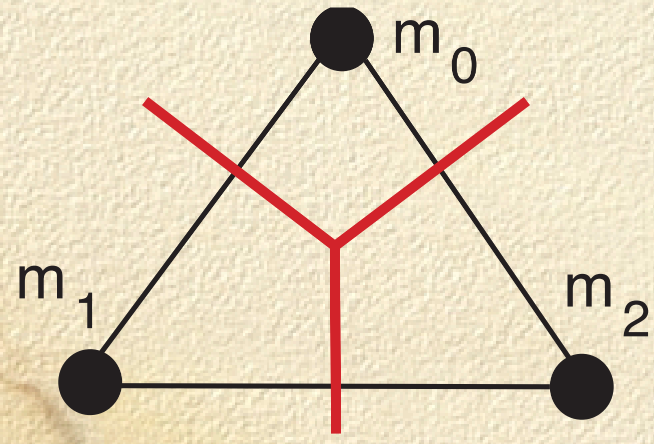


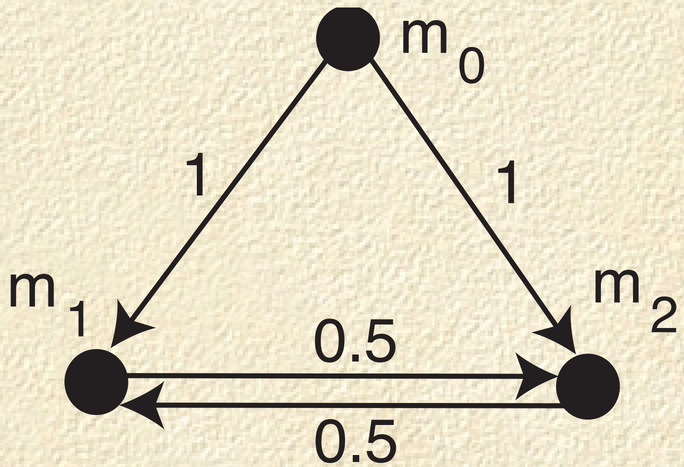
- Ahlswede *et al.* and Koetter *et al.*: for a multicast communication session in a *directed* network, if a rate x can be achieved to each receiver independently, it can also be achieved for the entire session.



Directed vs. Undirected Networks

- Bidirectional links: $f(\overrightarrow{AB}) + f(\overrightarrow{BA}) \leq Cap(AB)$.
- A more general, and "harder" model than directed networks
- Results in directed settings no longer hold:

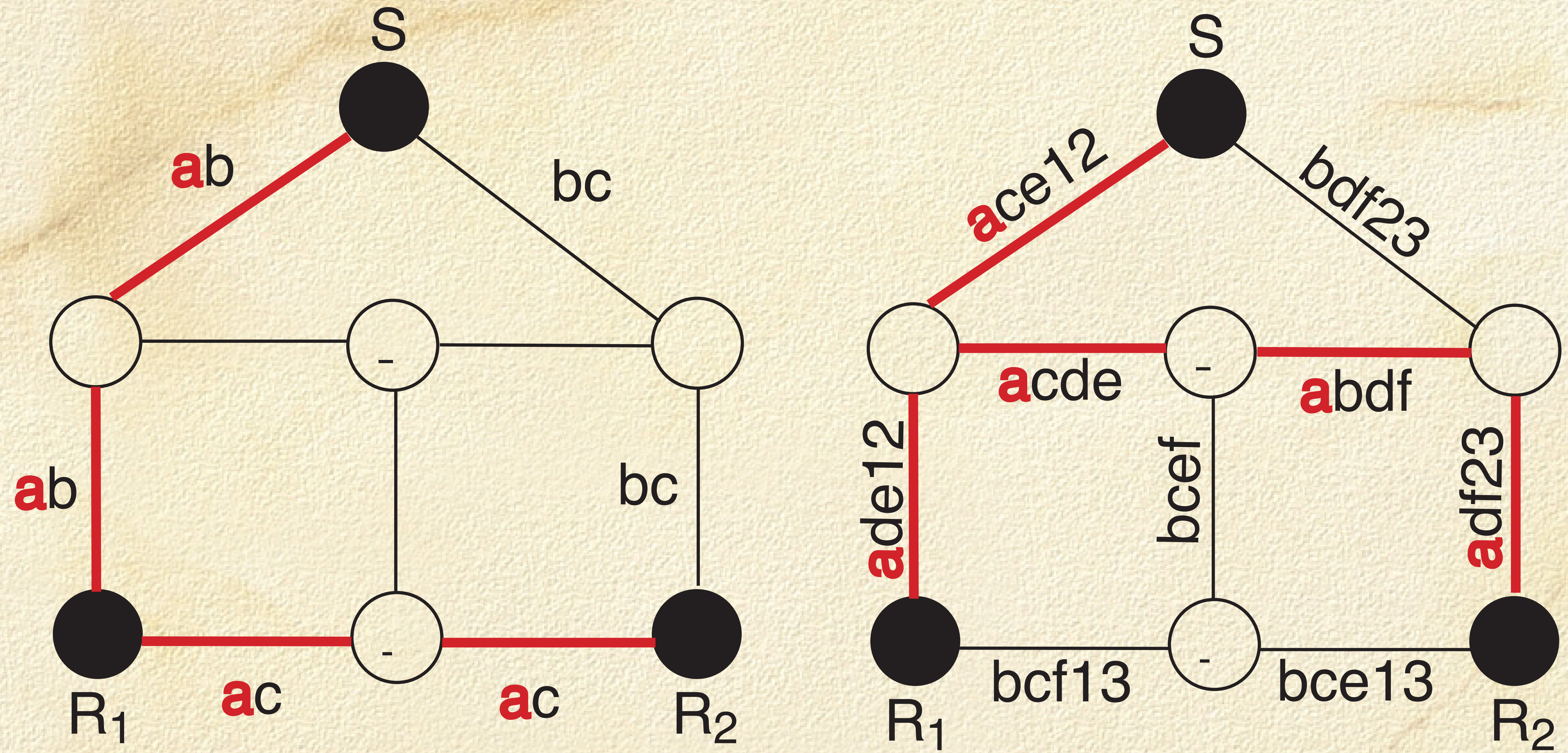




Fractional vs. integral routing

- Fractional routing: link capacities can be shared fractionally, and flows can be split and merged in *arbitrarily fine scales*

- Integral routing: all link capacities and flow rates have integer values

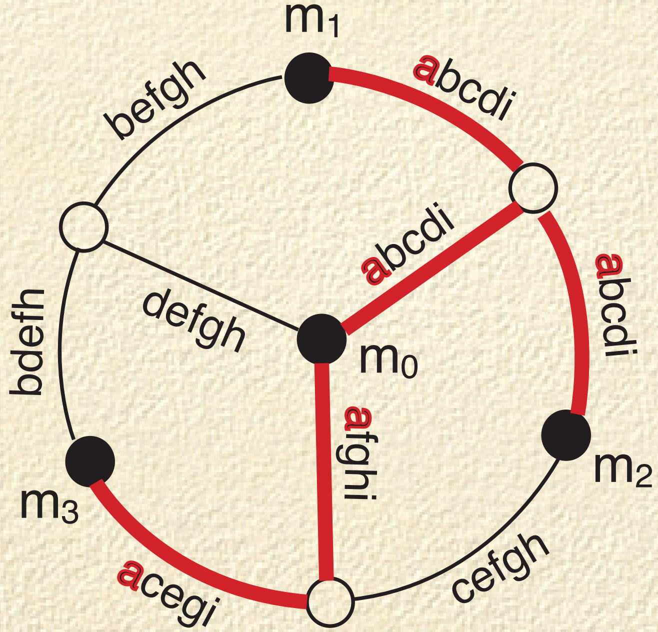


(a) Half-integer routing, optimal throughput = 1.5.

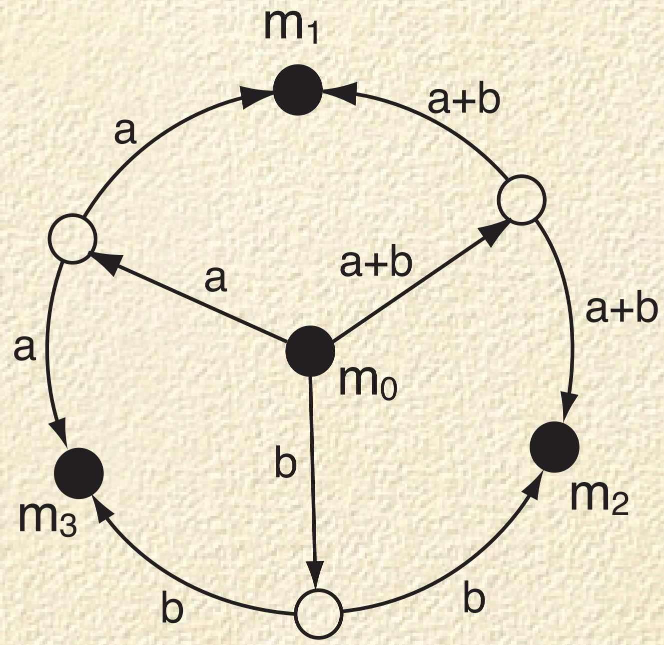
(b) Arbitrary fractional routing, optimal throughput = 1.875.

Steiner tree packing

- *Steiner tree packing:* decompose the network into weighted steiner trees, such that the total tree weight is maximized, and the capacity constraints are not violated.



(a) steiner tree packing and multicast without coding.



(b) multicast with network coding.

The achievable optimal throughput is 1.8 without coding, and 2 with coding.

Steiner tree packing

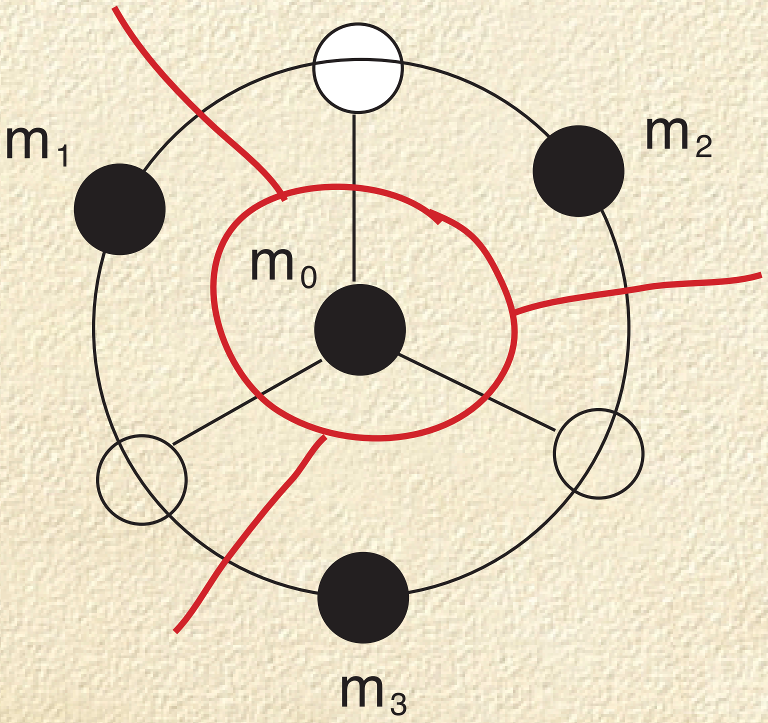
- For fractional routing, Steiner tree packing is NP-complete, with the best polynomial time approximation ratio of ~ 1.55 (Robins et al., SODA 2000).
- Even worse for integral routing: 26 (Lau, FOCS 2004, *unknown* before this paper)
- In practice, it can not be used to maximize throughput

The *coding advantage:* the ratio of achievable throughput with network coding and that without coding (steiner tree packing)

- We have proved that [CISS 2004], in fractional or half-integral routing:
  - The coding advantage of *a single unicast session* and of *a single broadcast session* is *one*.
  - The coding advantage of *a single multicast session* is upper bounded by 2.
- We have conjectured that [Allerton 2004]:
  - The coding advantage of *multiple unicast sessions* (allowing inter-session coding) is also *one*.

Steiner Strength

- *Partition of the network:* there exists at least one source or receiver node in each component of the partition.
- *Steiner strength:* $\min_{p \in P} |E_c|/(|p|-1)$
  - P: set of all partitions
  - $|E_c|$: total inter-component link capacity



Maximum throughput

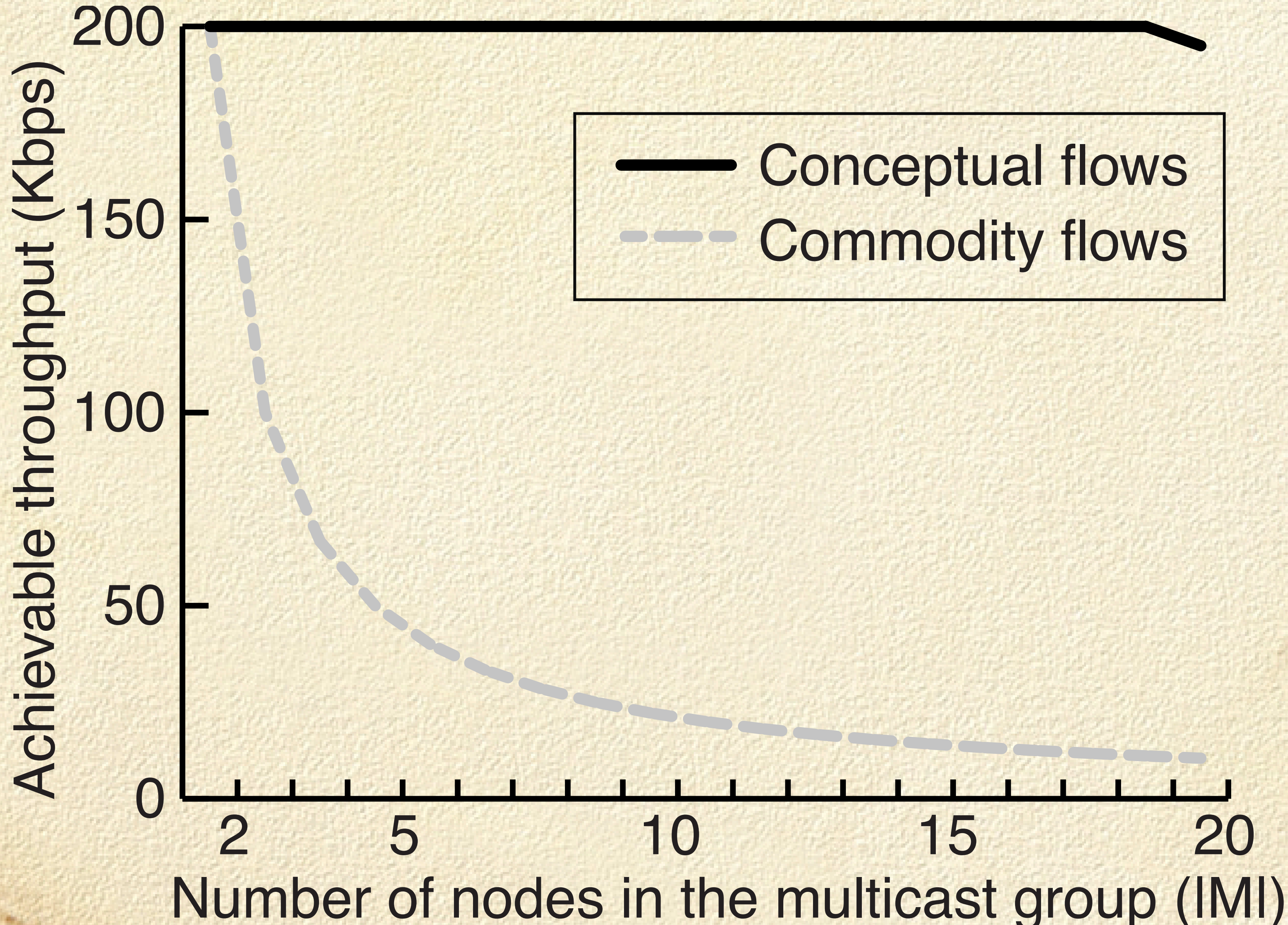
- *Maximum throughput:* maximum information flow rate from a source to a group of receivers concurrently, with fractional routing
- We have proved that [CISS 2004]:
  - achievable throughput with steiner tree packing $\leq$ *maximum throughput* $\leq$ steiner strength
  - Both steiner tree packing and steiner strength is NP-hard
  - Maximum throughput $\in$ P [INFOCOM 2005]

How to design an efficient algorithm to compute the *maximum throughput?*

Conceptual Flows

- The power of network coding resides in its ability to resolve competition for link capacities
- Rather than considering a multicast flow, we consider unicast flows from source to each of the receivers
- *Conceptual Flows:* network flows that co-exist in the network without contending for link capacities

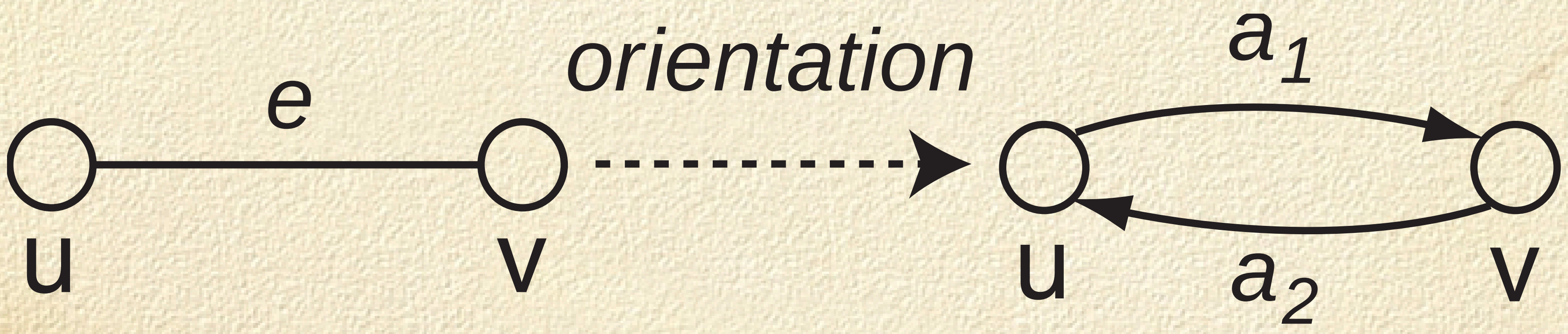
Conceptual Flows vs. *Commodity Flows*



Orientation of a network

- *Orientation* of an undirected network is a strategy of replacing each undirected link with two directed arcs, without violating the capacity constraint:

$$C(e) = C(a_1) + C(a_2), \forall e \in E$$



The *cFlow* LP

Maximize: f^*

Subject to:

Orientation constraints:

$$\begin{cases} 0 & \leq & C(a) & \forall a \in D \\ C(a_1) + C(a_2) & = & C(e) & \forall e \in E \end{cases}$$

Independent network flow constraints for each conceptual flow:

$$\begin{cases} 0 & \leq & f^i(a) & \forall i \in [1..k], \forall a \in D \\ f^i(a) & \leq & C(a) & \forall i \in [1..k], \forall a \in D \\ f^i_{in}(v) & = & f^i_{out}(v) & \forall i \in [1..k], \forall v \in V - \{m_0, m_i\} \\ f^i_{in}(m_0) & = & 0 & \forall i \in [1..k] \\ f^i_{out}(m_i) & = & 0 & \forall i \in [1..k] \end{cases}$$

Equal rate constraints:

$$f^* \quad = \quad f^i_{in}(m_i) \quad \forall i \in [1..k]$$

The Complete Solution

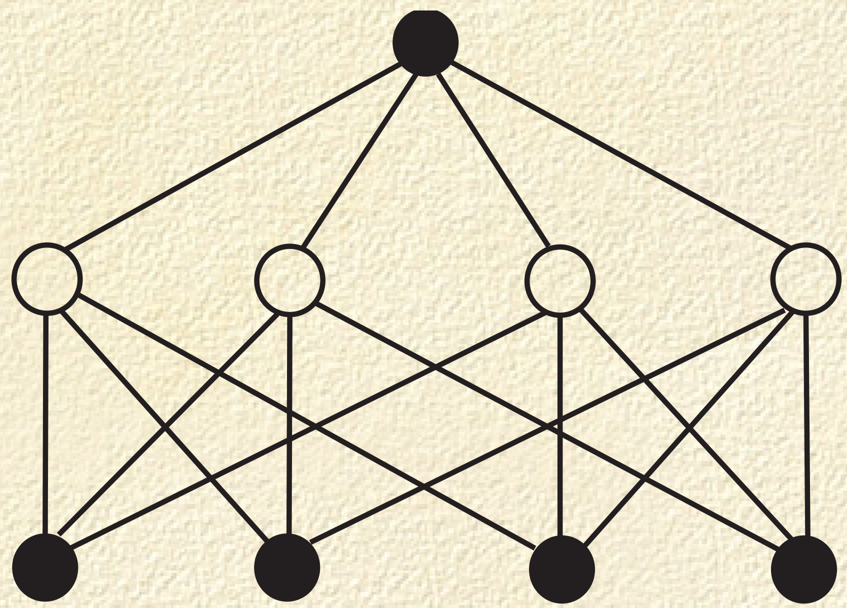
- Computing the coding strategies: *polynomial time algorithm* of code assignment [Sanders *et al.*, SPAA 2003]
- The complete solution that achieves maximum throughput in undirected networks with a single multicast session can be computed in polynomial time, including both *routing* and *coding* strategies.

Let us put it to good use

- *Uniform Bipartite Networks:* conjectured to be good candidates to show the power of coding on improving throughput

  - *C(n, k)*: consists of the source and two layers, one with *n* relay nodes and the other with $\binom{n}{k}$ receivers. Each relay node is connected to the sender, and each receiver is connected to a different group of *k* relay nodes. All link capacities are *one*.

Uniform Bipartite Networks



The uniform bipartite network $\mathsf{C}(4,3)$.

cFlow LP vs. *steiner tree packing*: a comparison

Network	$\|V\|$	$\|M\|$	$\|E\|$	$\chi(N)$	$\pi(N)$	$\frac{\chi(N)}{\pi(N)}$	# of trees
Fig. 1	7	3	9	2	1.875	1.067	17
$C(3,2)$	7	4	9	2	1.8	1.111	26
$C(4,3)$	9	5	16	3	2.667	1.125	1,113
$C(4,2)$	11	7	16	2	1.778	1.125	1,128
$C(5,4)$	11	6	25	4	3.571	1.12	75,524
$C(5,2)$	16	11	25	2	1.786	1.12	119,104
$C(5,3)$	16	11	35	3	–	–	49,956,624

Good and bad news

- Good news: the polytime *cFlow* LP is much more computationally feasible;
- Bad news: the coding advantage with a single multicast session in thousands of randomly chosen network topologies is *one*.
  - Except for the cases of *uniform bipartite networks*.

Does network coding really help?

- It does not help much with respect to improving the maximum achievable throughput
- It does help to reduce the complexity of computing routing strategies

Complexity: multicast with fractional routing

- Problem: computing maximum multicast rate in an undirected network, with fractional routing
- Without coding: fractional steiner tree packing, NP-complete.
- With coding: transform into a LP problem, P.

Complexity: multicast with integral routing

- Problem: computing maximum multicast rate with in an undirected network, with integral routing
- Without coding: integral steiner tree packing, only polytime approximation known: 26-approximation [Lau, FOCS 2004]
- With network coding: 2-approximation [CISS 2004]

Towards efficient and distributed computation of the *cFlow LP*

- Performance of general LP solvers is not good, as the *cFlow* LP has *O(km)* number of variables and *O(km)* number of constraints, k is the number of receivers, and m is the number of links
- Experimented with:
  - Interior Point method: can handle $m = 1000$, and $k = 10$
  - Simplex method: scalability much worse

Primal *cFlow* LP: revisited

Maximize χ

Subject to:

$$\begin{cases} \chi \le f_i(\overrightarrow{T_i S}) & \forall i & (1) \\ f_i(\overrightarrow{uv}) \le c(\overrightarrow{uv}) & \forall i, \forall\ \overrightarrow{uv} \neq \overrightarrow{T_i S} & (2) \\ \sum_{v \in N(u)} f_i(\overrightarrow{uv}) = \sum_{v \in N(u)} f_i(\overrightarrow{vu}) & \forall i, \forall u & (3) \\ c(\overrightarrow{uv}) + c(\overrightarrow{vu}) \le C(uv) & \forall uv \neq T_i S & (4) \end{cases}$$

$$c(\overrightarrow{uv}), f_i(\overrightarrow{uv}), \chi \ge 0, \qquad \forall i, \forall\ \overrightarrow{uv}$$

Dual *cFlow* LP

Minimize $\sum_{uv} C(uv)x(uv)$

Subject to:

$$\begin{cases} x(uv) \geq \sum_i y_i(\overrightarrow{uv}) & \forall uv \neq T_iS & (5) \\ y_i(\overrightarrow{uv}) + p_i(v) \geq p_i(u) & \forall i, \forall\, \overrightarrow{uv} \neq \overrightarrow{T_iS} & (6) \\ p_i(T_i) - p_i(S) \geq z_i & \forall i & (7) \\ \sum_i z_i \geq 1 & & (8) \end{cases}$$

$$x(uv), y_i(\overrightarrow{uv}), z_i \geq 0 \qquad \forall i, \forall\, \overrightarrow{uv}$$

primal	(1)	(2)	(3)	(4)	c	$f(\overrightarrow{uv})$	$f(\overrightarrow{T_iS})$	χ
dual	z	y	p	x	(5)	(6)	(7)	(8)

Subgradient algorithm: dualization strategy

- Applying Lagrangian relaxation on the constraint (5) in the dual program (primal subgradient)
  - Decomposes the entire problem into a sequence of max-flow/min-cut computations
  - Allows a decentralized implementation

(1) *Choose initial orientation* (e.g., *balanced orientation)*

(2) Repeat

Compute $S \rightarrow T_i$ *max-flow*, $\forall i$

Refine orientation:

increase bandwidth share for saturated links

decrease bandwidth share for under-utilized links

Until convergence

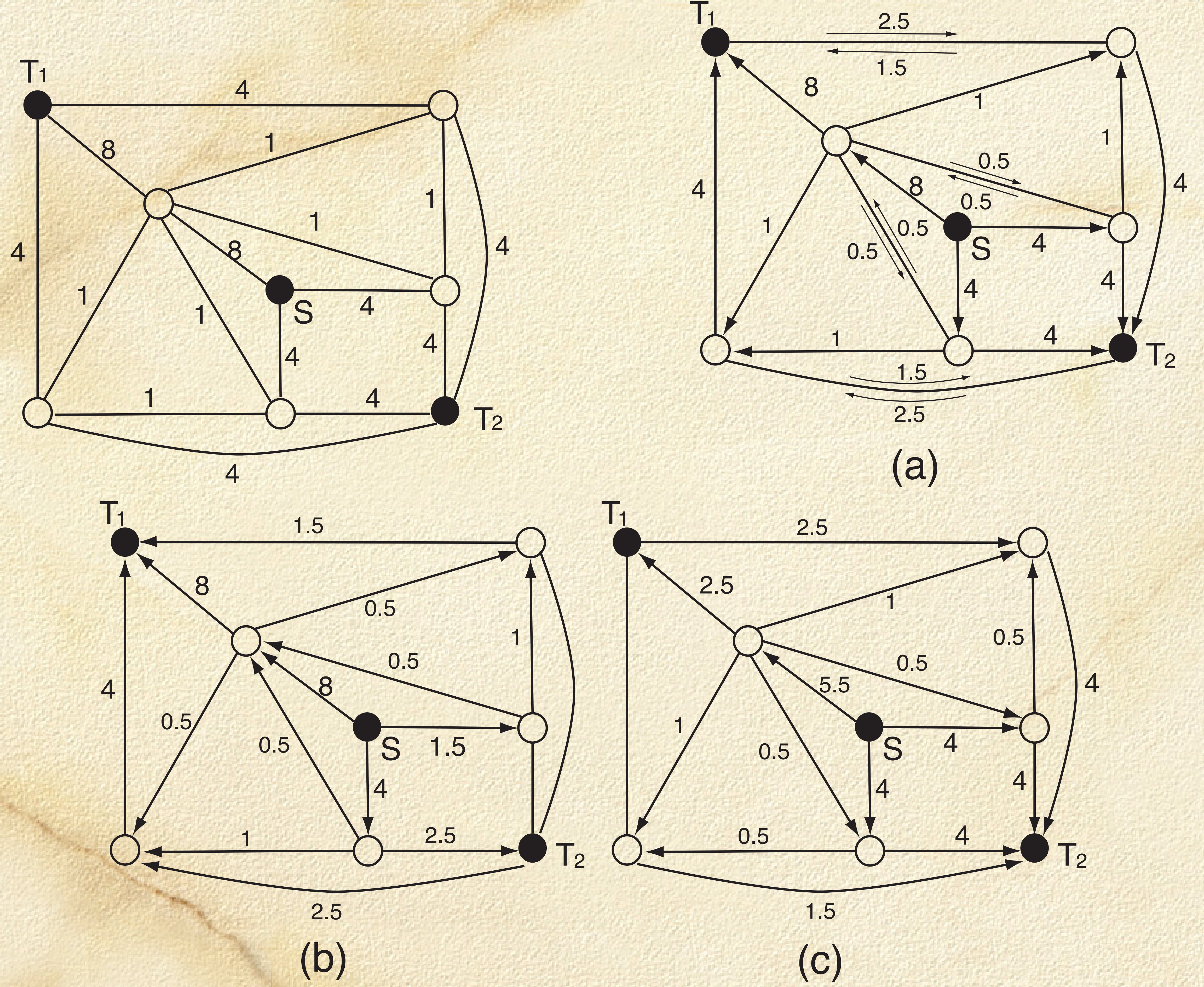
$\rightarrow$ optimal orientation obtained

(3) *Compute* $S \rightarrow T_i$ *max-flow*, $\forall i$

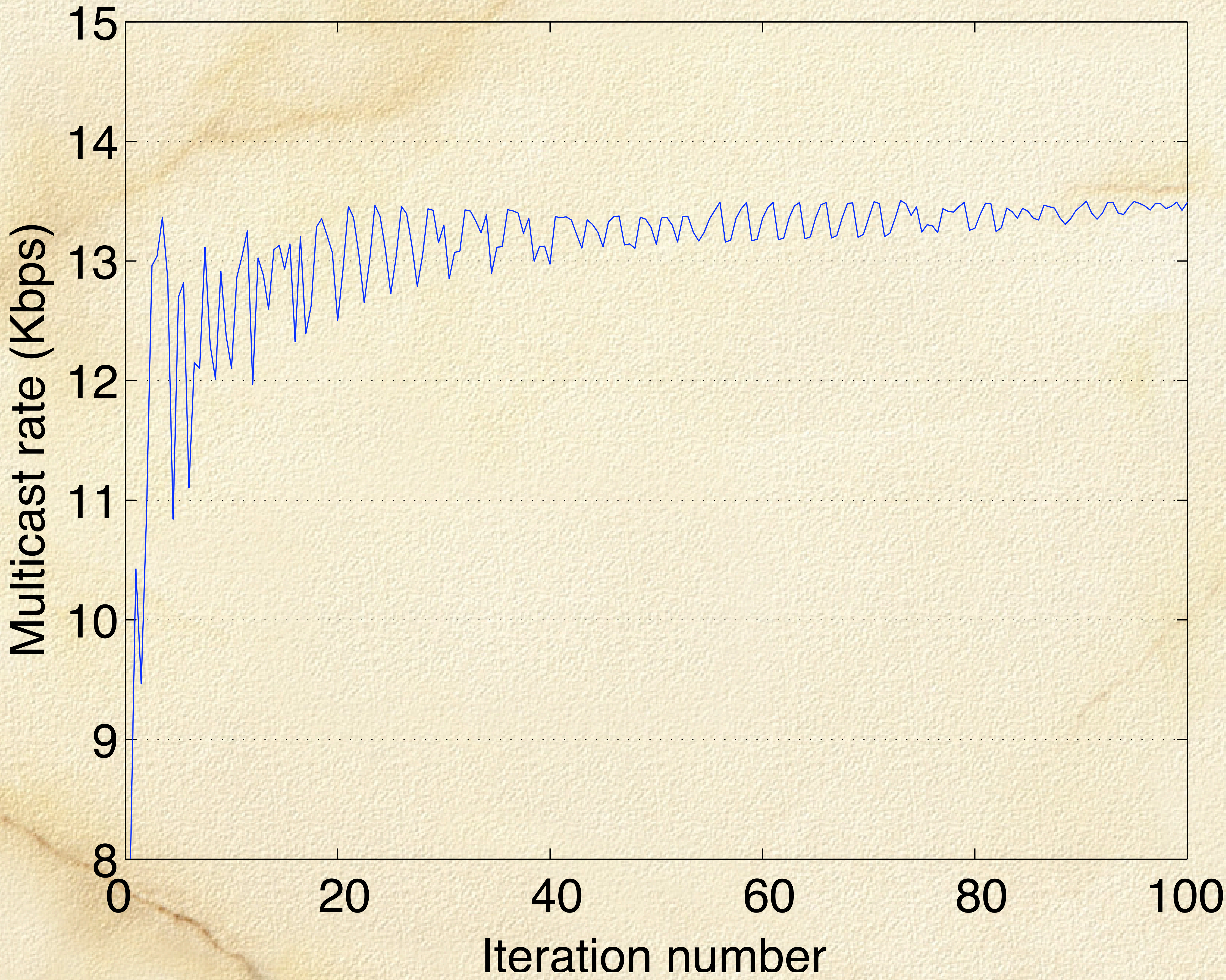
$\rightarrow$ optimal multicast rate and routing strategy obtained

(4) *Randomized code assignment*

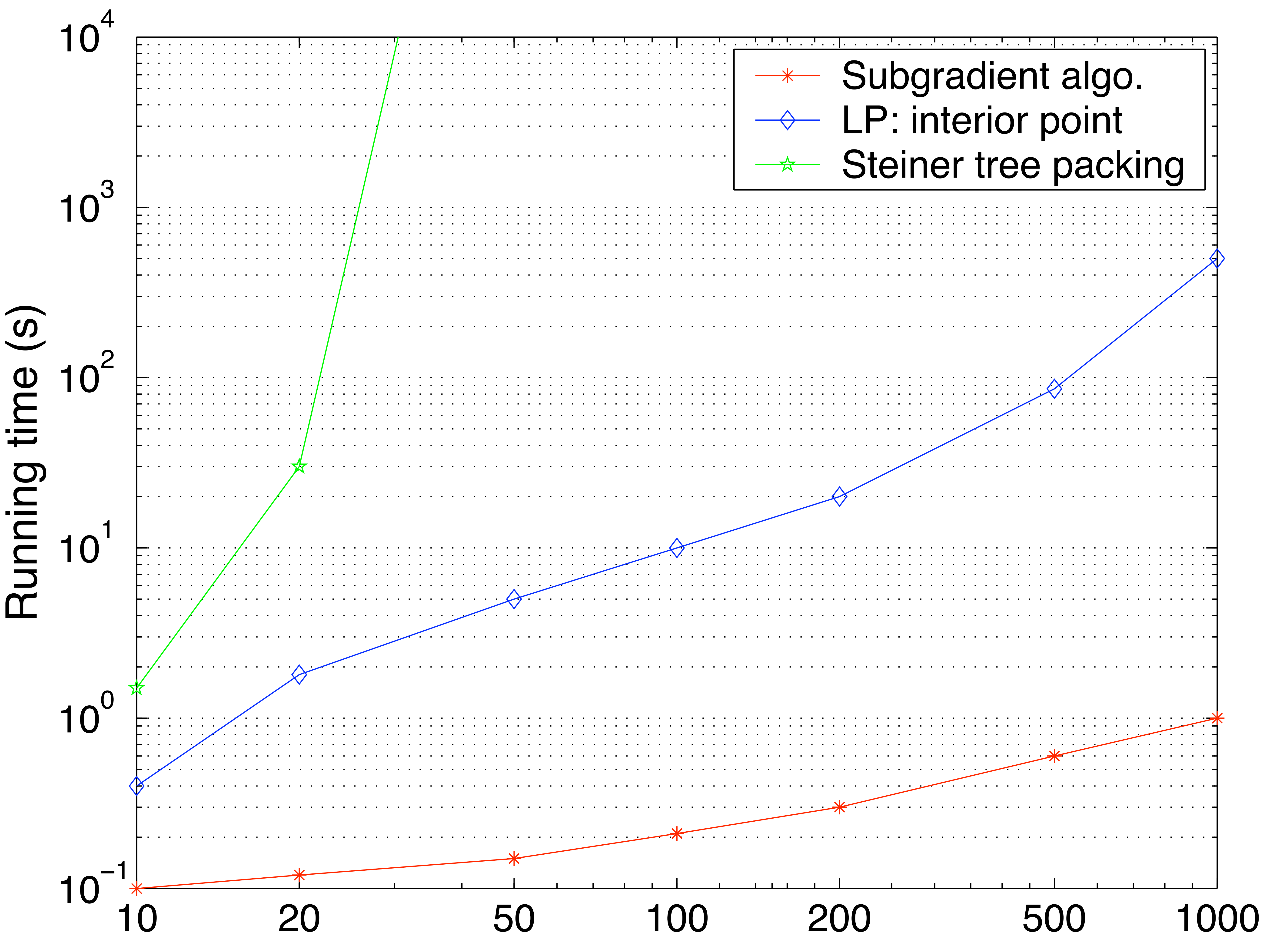
$\rightarrow$ complete transmission strategy obtained



T1
S
T2
(a)
(b)
(c)



15
14
13
12
11
10
9
8
0
20
40
60
80
100
Multicast rate (Kbps)
Iteration number



Running time (s)
10^4
10^3
10^2
10^1
10^0
10^{-1}
10
20
50
100
200
500
1000
Subgradient algo.
LP: interior point
Steiner tree packing

Extensions

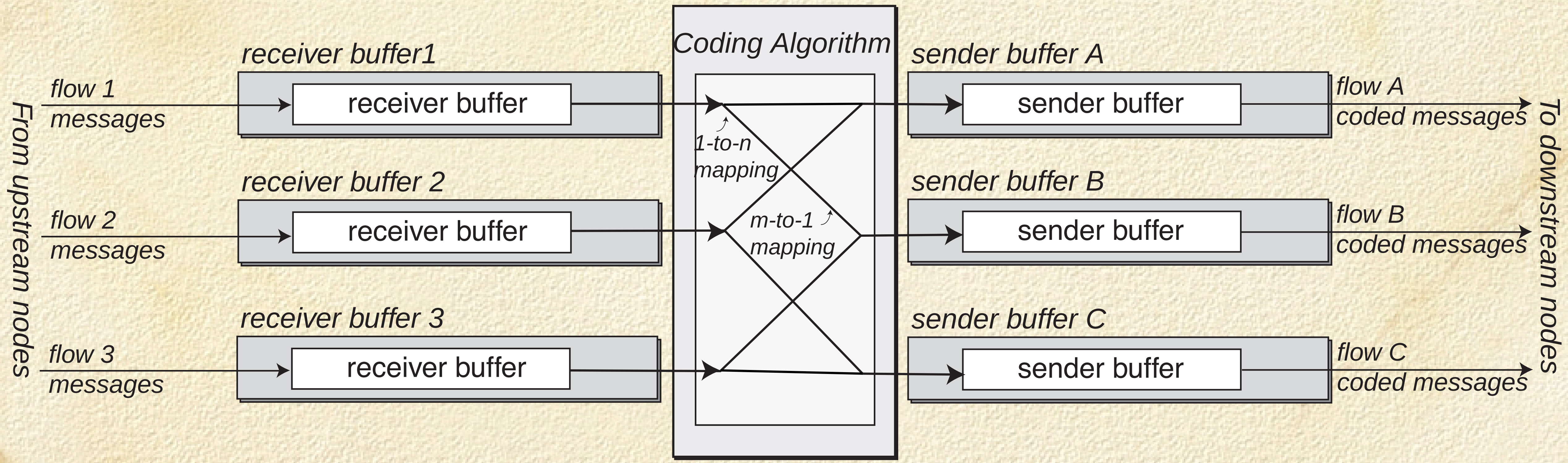
- The case of multiple multicast sessions (without inter-session coding)
- The case of overlay networks: only a subset of the nodes (the end hosts) may be able to replicate and code data
- In both cases, the corresponding problem can be formulated as LP problem, with a polynomial number of variables and constraints

From theory to practice

From theory to reality

- Coding in GF(256)
- Random code assignment
- Start with a high quality mesh
- Distributed computation of flow routing strategy (with network coding)

Application-layer message switch



A long way to go

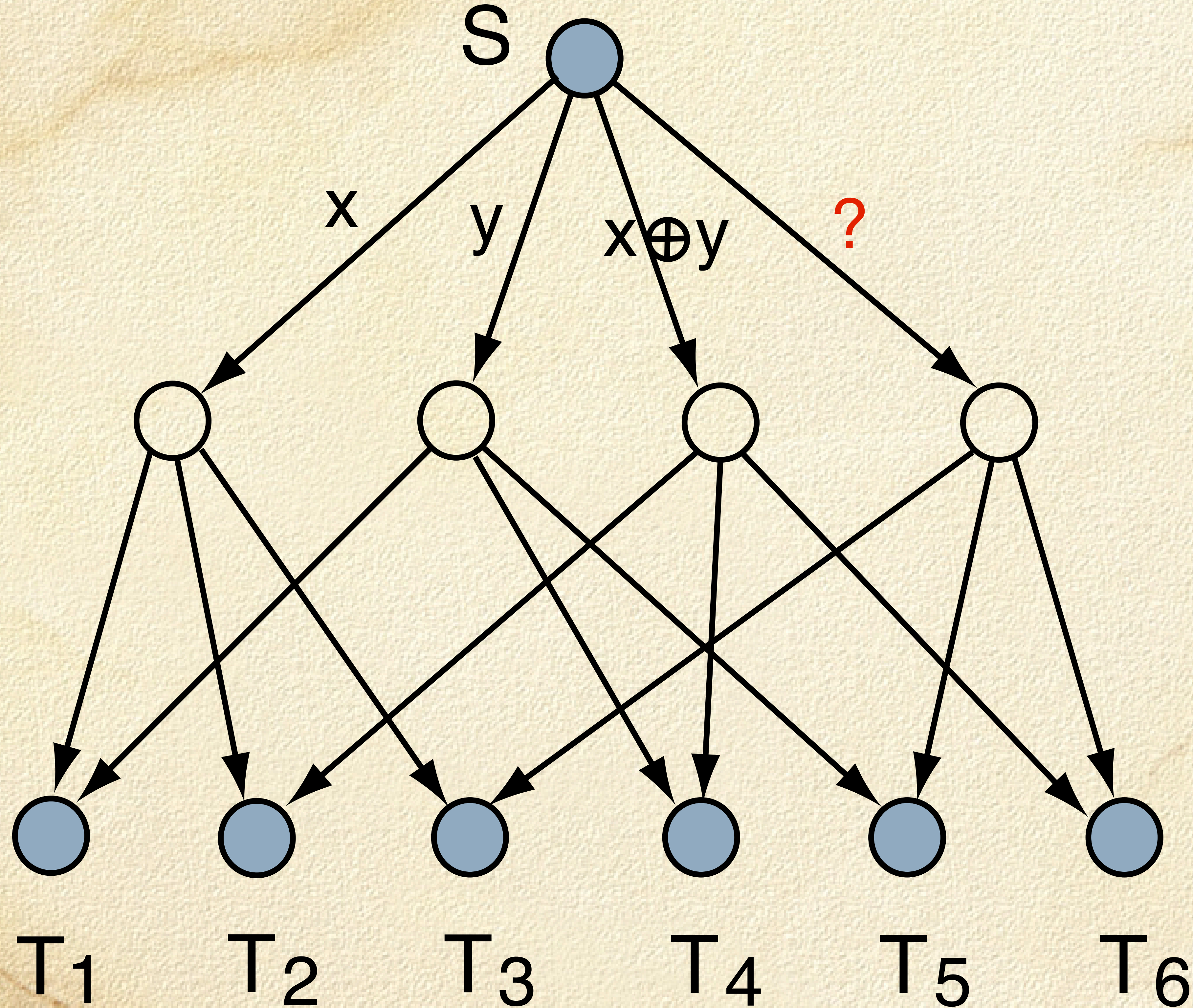
- Penalty of synchrony: flows have to wait for other incoming flows to be encoded or decoded
- Link capacities often unknown
- Decoding efficiency is a concern, if we use network coding on large volume of data

Towards better decoding efficiency

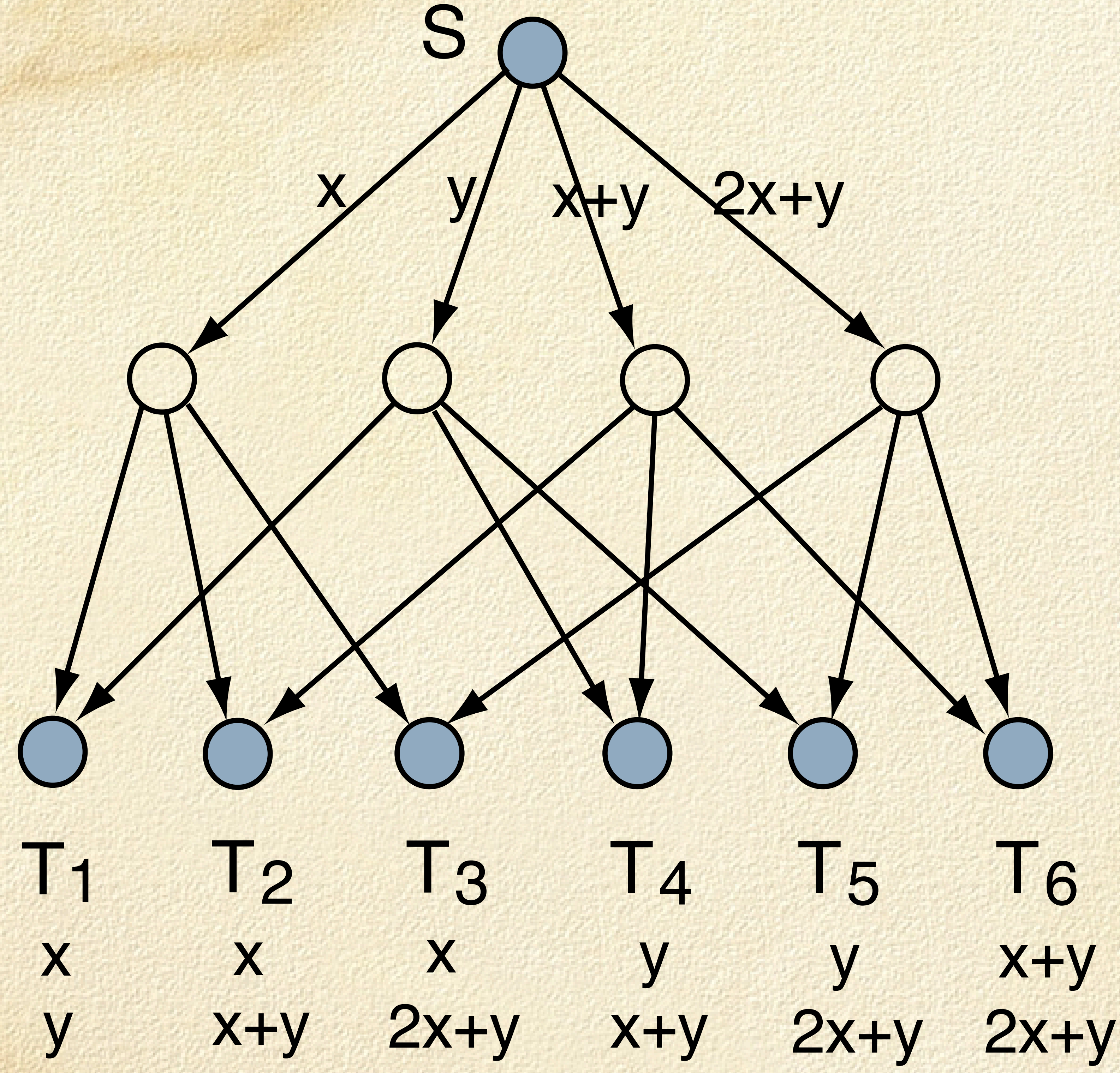
	single session	*multiple sessions (w/ intersession coding)*
integral routing	linear coding is sufficient	nonlinear coding required
fractional routing	*XOR only coding is sufficient* [*]	linear coding is sufficient [?]

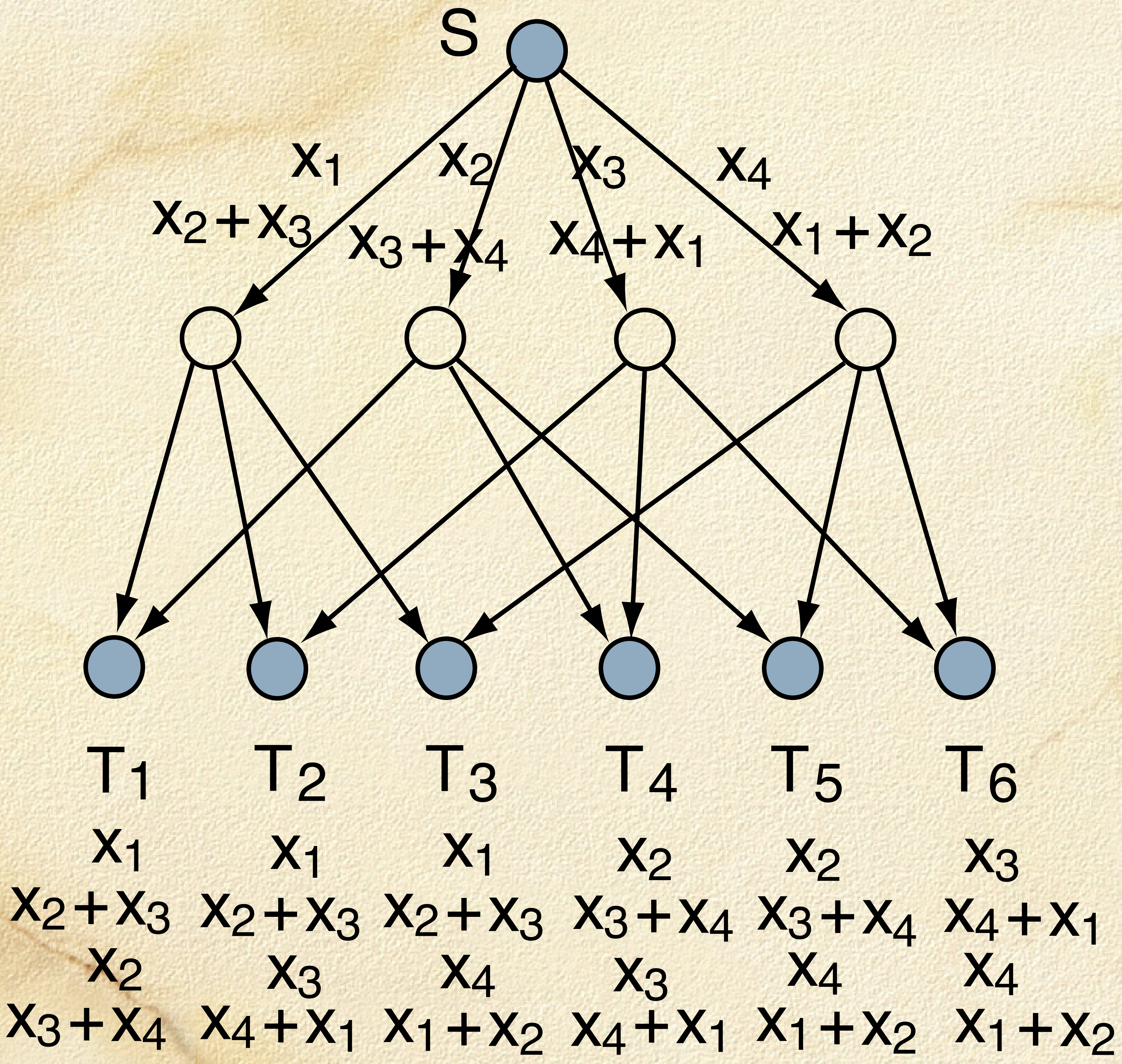
[?] Conjecture, Sec. 3, Medard, Effros, Karger, Ho, *"On Coding for Non-multicast Networks,"* Allerton 2003.

[*] Zongpeng Li, Baochun Li, *untitled*, tech. report in preparation.



S
x
y
x⊕y
?
T_1
T_2
T_3
T_4
T_5
T_6


S
x
y
x+y
2x+y
T1
x
y
T2
x
x+y
T3
x
2x+y
T4
y
x+y
T5
y
2x+y
T6
x+y
2x+y


S
x_1
x_2+x_3
x_2
x_3+x_4
x_3
x_4+x_1
x_4
x_1+x_2
T_1
T_2
T_3
T_4
T_5
T_6
x_1, x_2+x_3, x_2, x_3+x_4
x_1, x_2+x_3, x_3, x_4+x_1
x_1, x_2+x_3, x_4, x_1+x_2
x_2, x_3+x_4, x_3, x_4+x_1
x_2, x_3+x_4, x_4, x_1+x_2
x_3, x_4+x_1, x_4, x_1+x_2

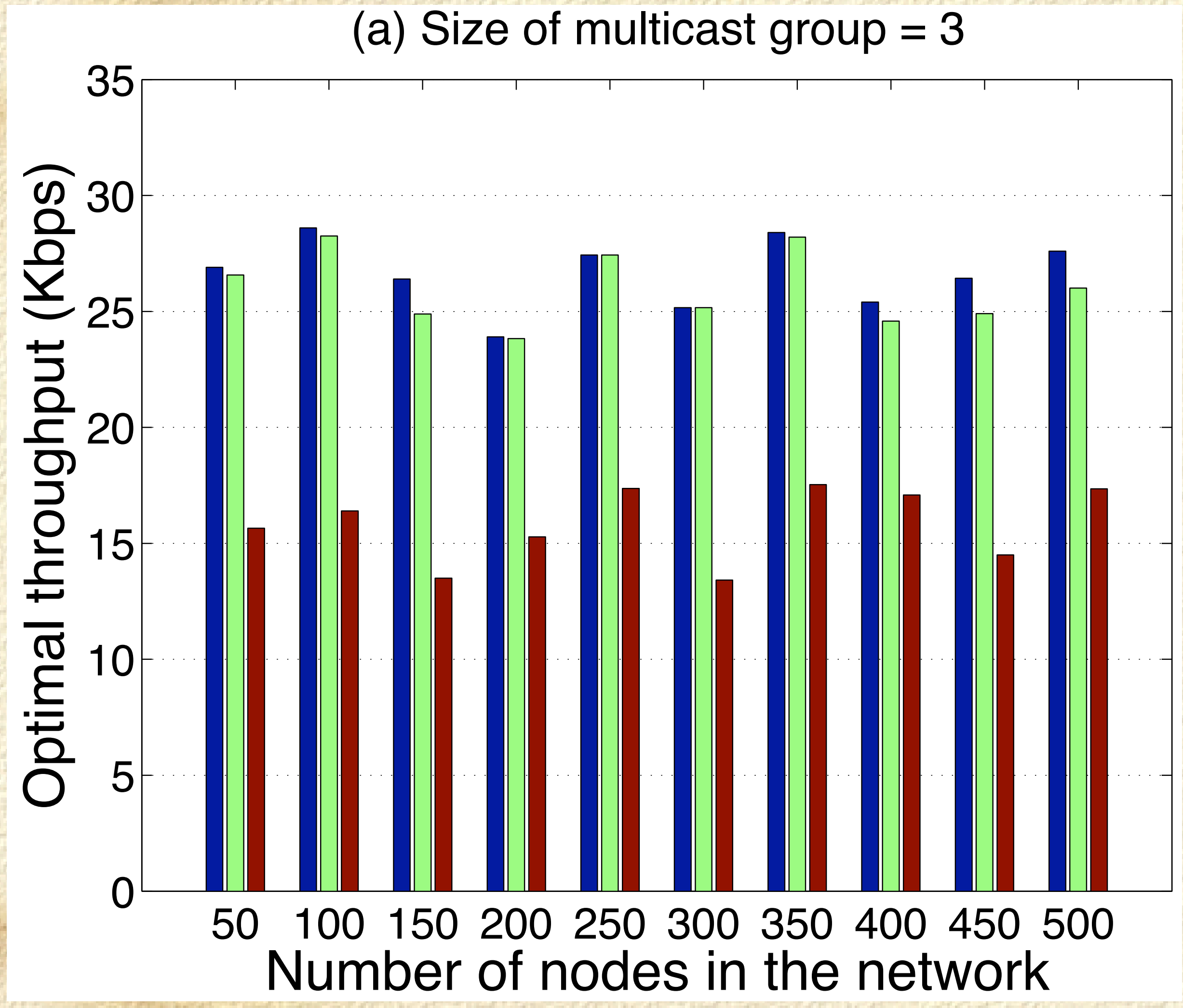
Papers that this talk is based on

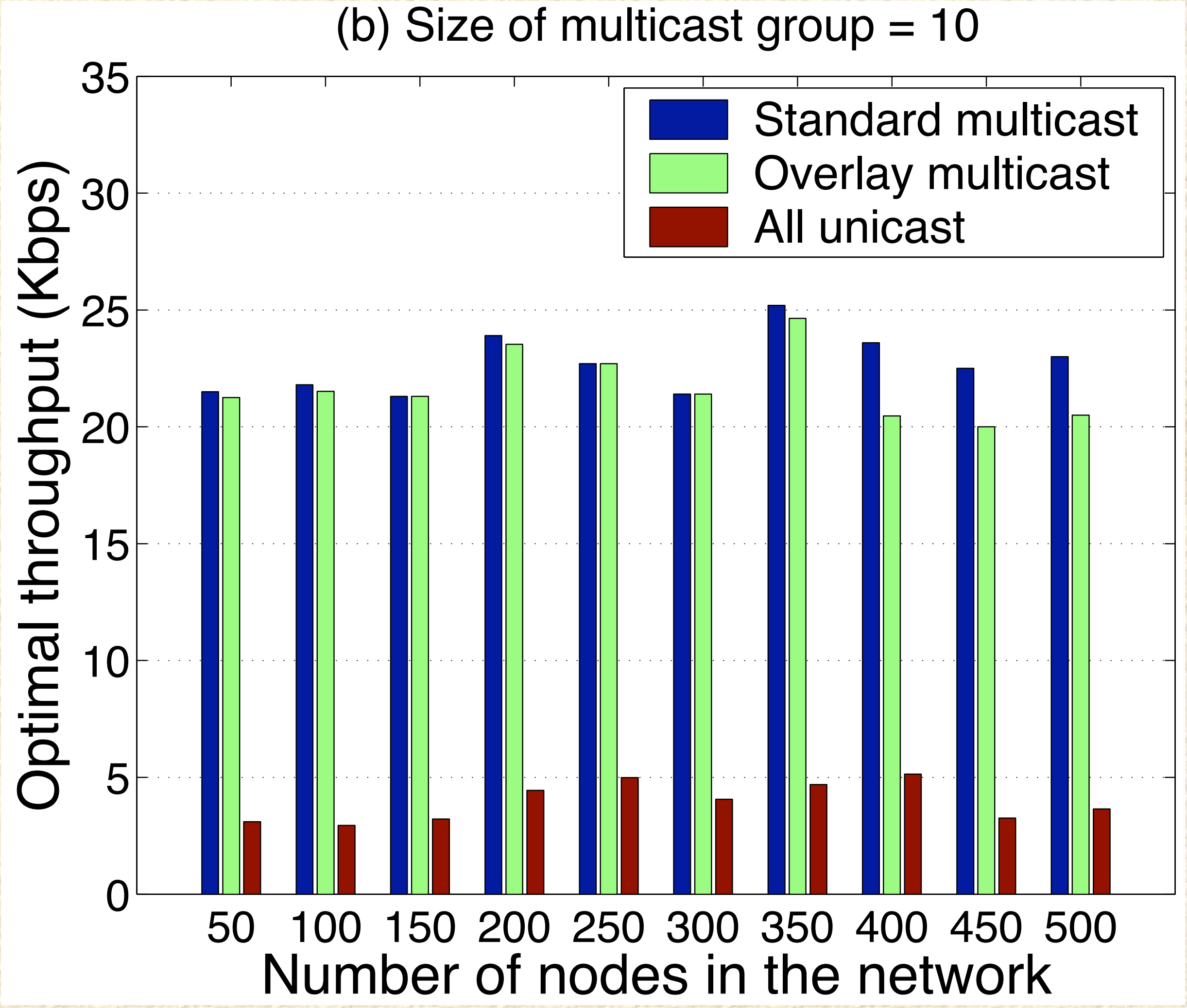
- Zongpeng Li, Baochun Li, Dan Jiang, Lap Chi Lau. *"On Achieving Optimal Throughput with Network Coding,"* INFOCOM 2005.
- Zongpeng Li, Baochun Li. *"Efficient and Distributed Computation of Maximum Multicast Rates,"* INFOCOM 2005.
- Zongpeng Li, Baochun Li. *"Network Coding in Undirected Networks,"* CISS 2004.
- Mea Wang, Baochun Li, Zongpeng Li. *"Implementing Network Coded Flows,"* in preparation for submission.
- Zongpeng Li, Baochun Li. *"Network Coding: The Case of Multiple Unicast Sessions,"* Allerton 2004.
- Zongpeng Li, Baochun Li. *untitled,* in preparation for submission.

Zongpeng Li, Baochun Li
ECE, University of Toronto
google "Baochun"

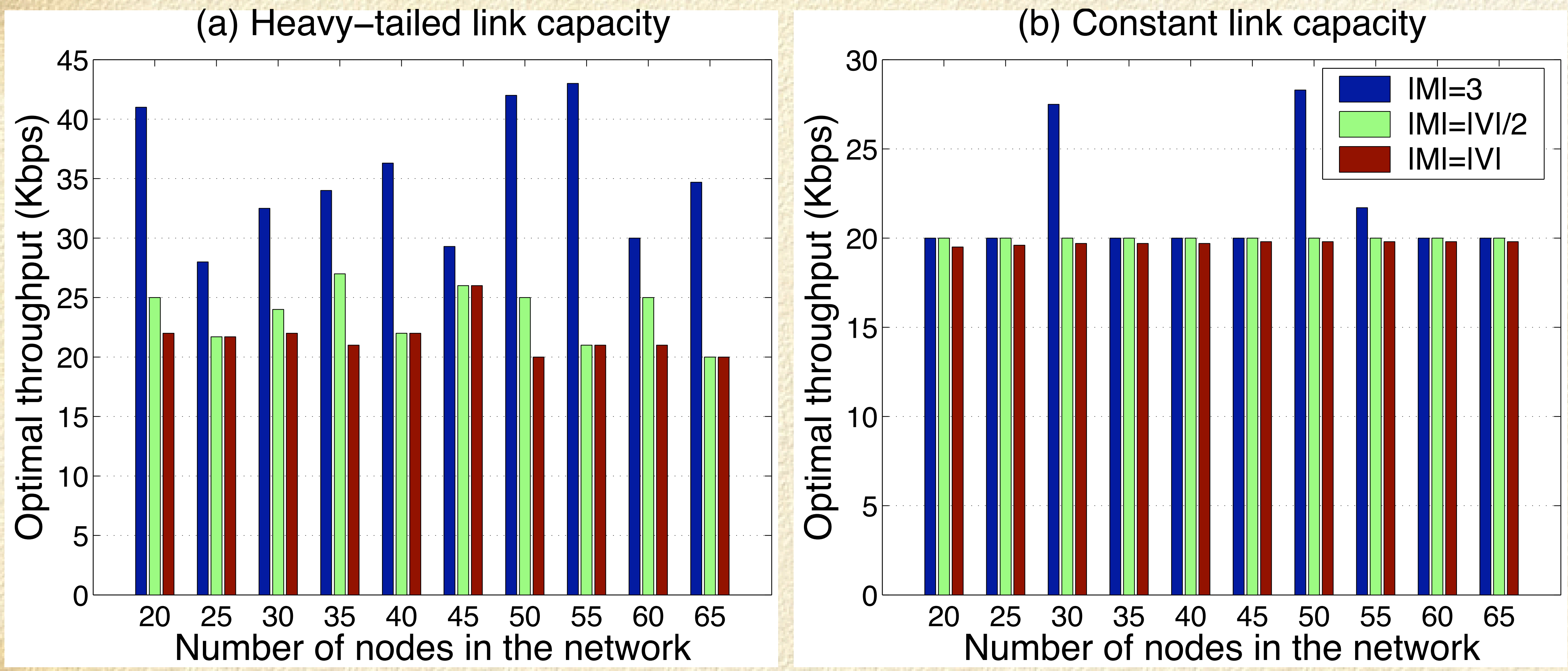
Empirical Studies

Unicast, standard multicast and overlay multicast

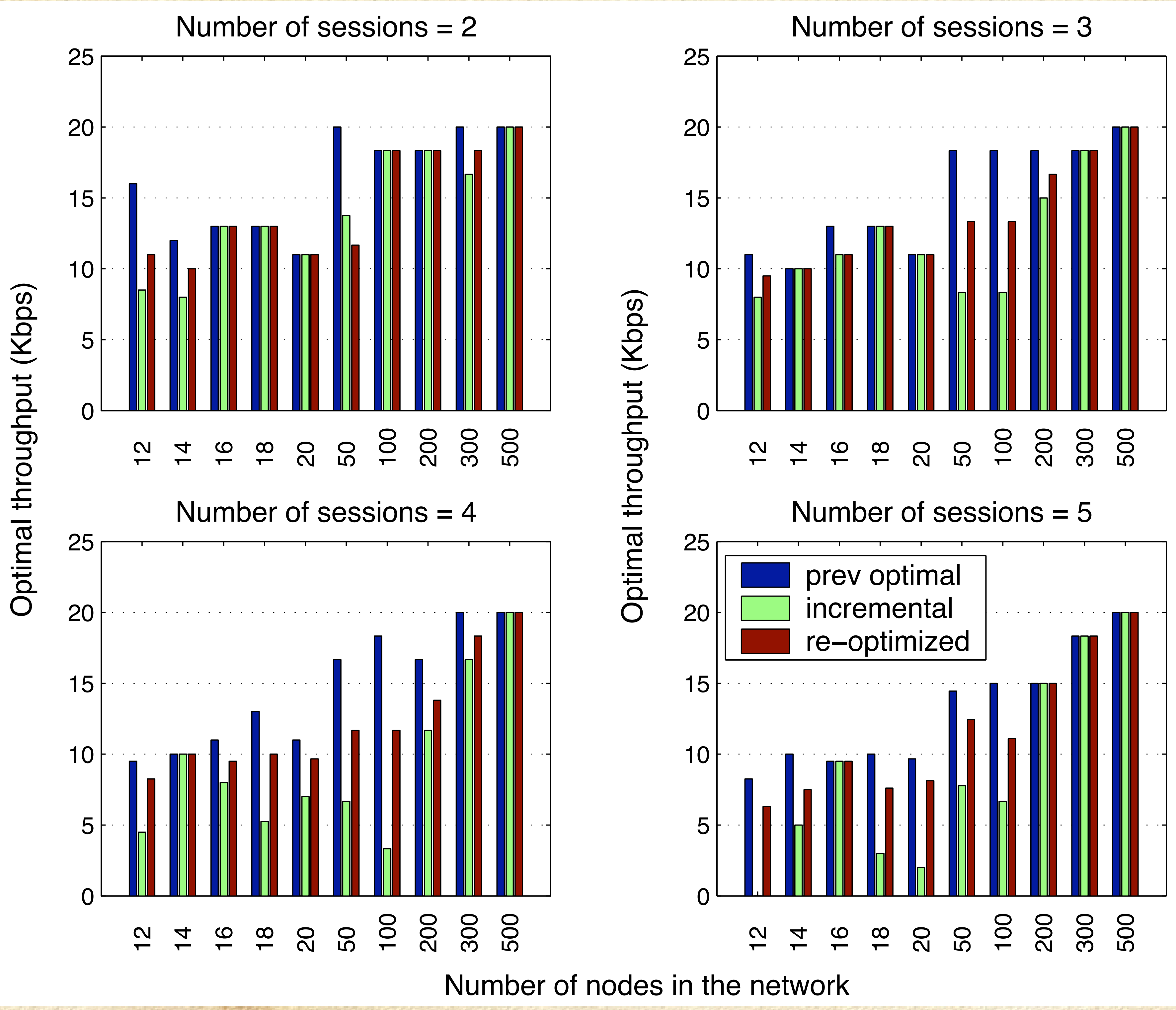




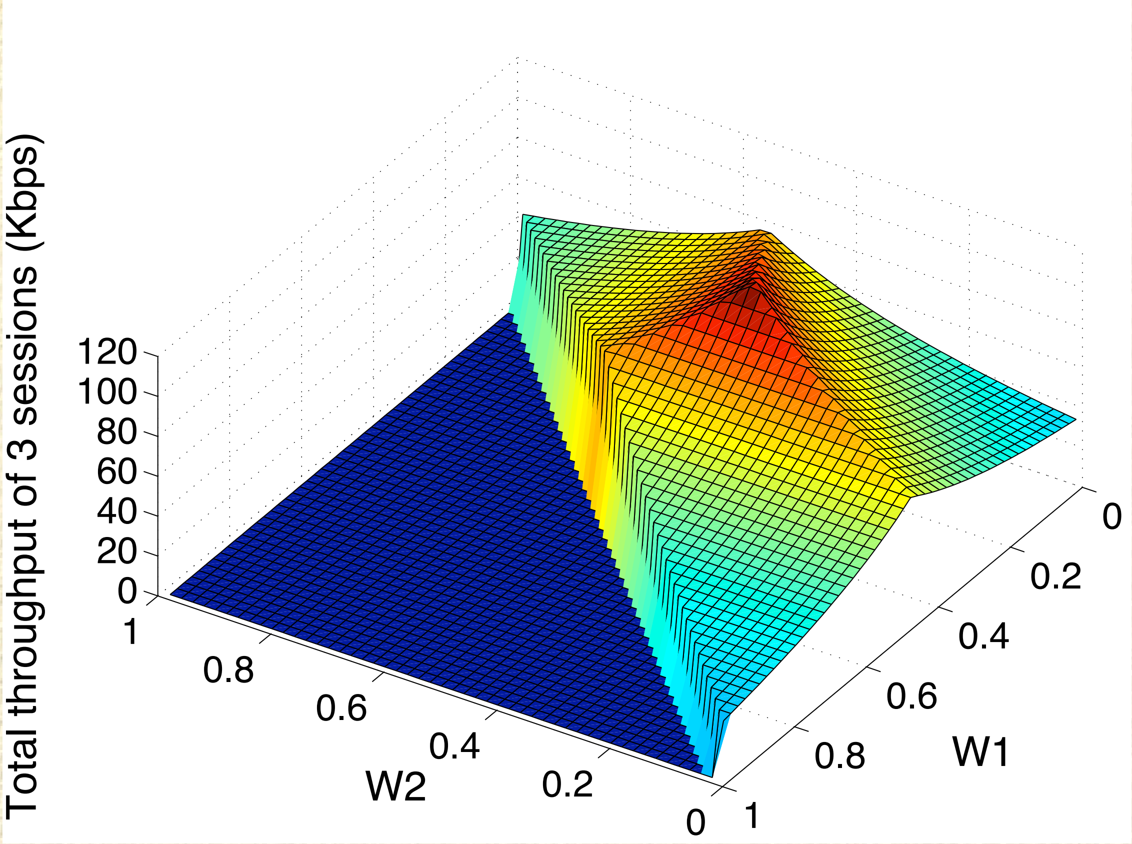
How sensitive is optimal throughput to node joins?



How sensitive is optimal throughput to new sessions?



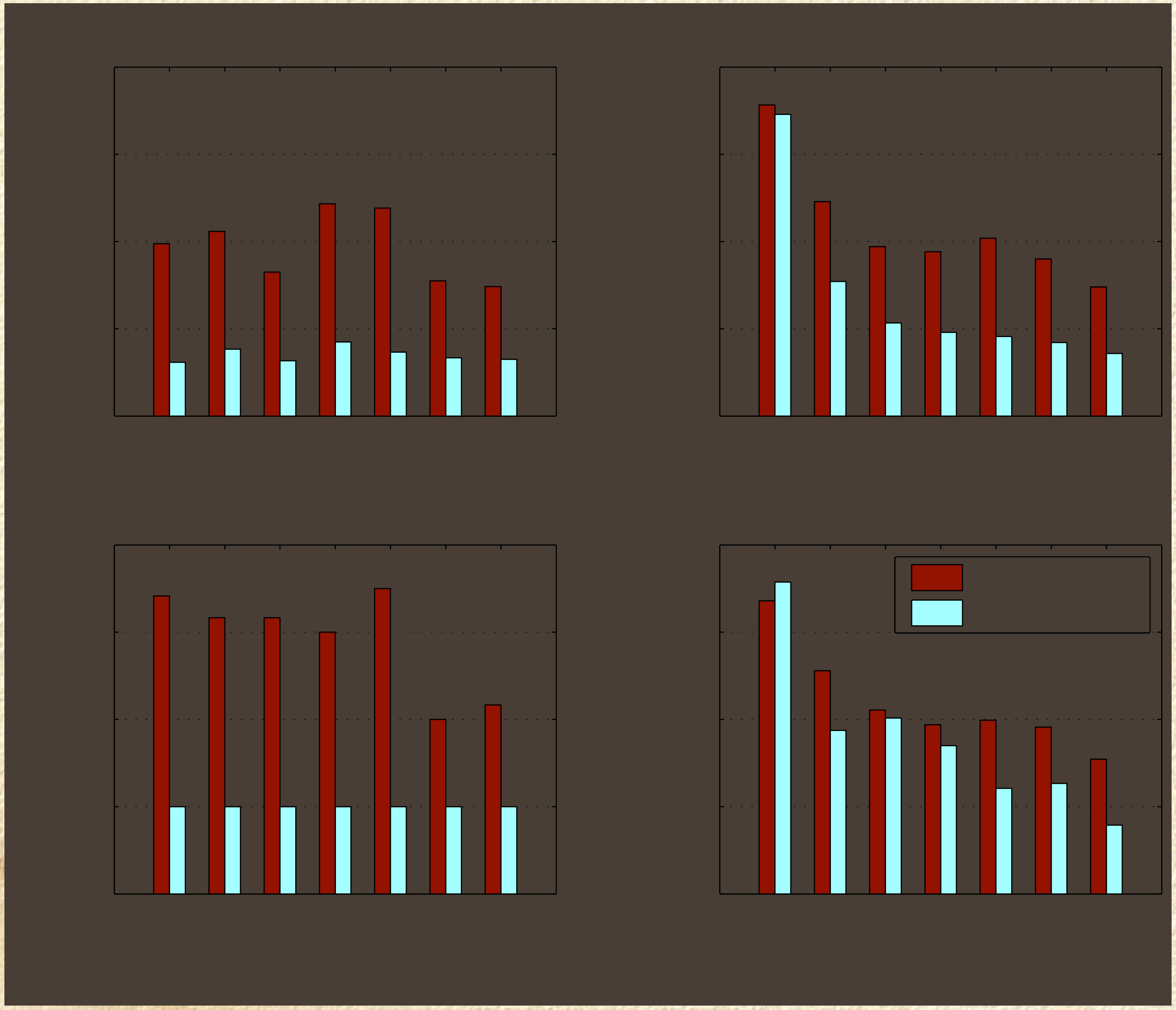
How sensitive is optimal throughput to fairness?



network size	10	50	100	150	250	350
max-min (Kbps)	120.0	173.3	160.0	146.7	146.7	183.3
optimal (Kbps)	126.1	173.3	160.0	146.7	146.7	183.3

Does optimal throughput lead to low bandwidth efficiency?

Bandwidth efficiency: total receiving rate at all receivers divided by total bandwidth consumption



Prototype implementation