Multiparty Equality Function Computation in Networks with Point-to-Point Links

Nitin Vaidya
University of Illinois at Urbana-Champaign

Joint work with Guanfeng Liang

Research supported in part by
National Science Foundation
and Army Research Office

Background

Equality Function

Determine whether the two inputs are identical

- Communication cost of an algorithm:

  # bits of communication required
  in the worst case (over all possible inputs)

- Communication cost of an algorithm:

  # bits of communication required
  in the worst case (over all possible inputs)

- Communication complexity of a problem:

  Minimum communication cost
  over all algorithms to solve the problem

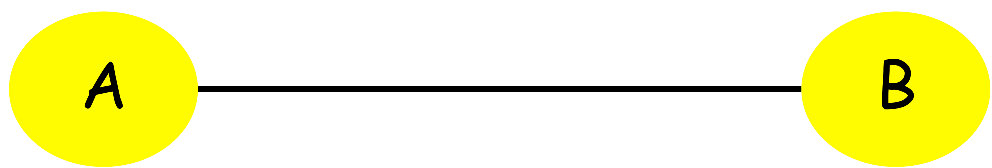
[Andrew Yao, STOC 1979]

Equality Function

K-valued input

K-valued input

Equality Function



K-valued input

K-valued input

Who knows the outcome?

Suffices for one node to know
One more bit to inform the other

Upper Bound

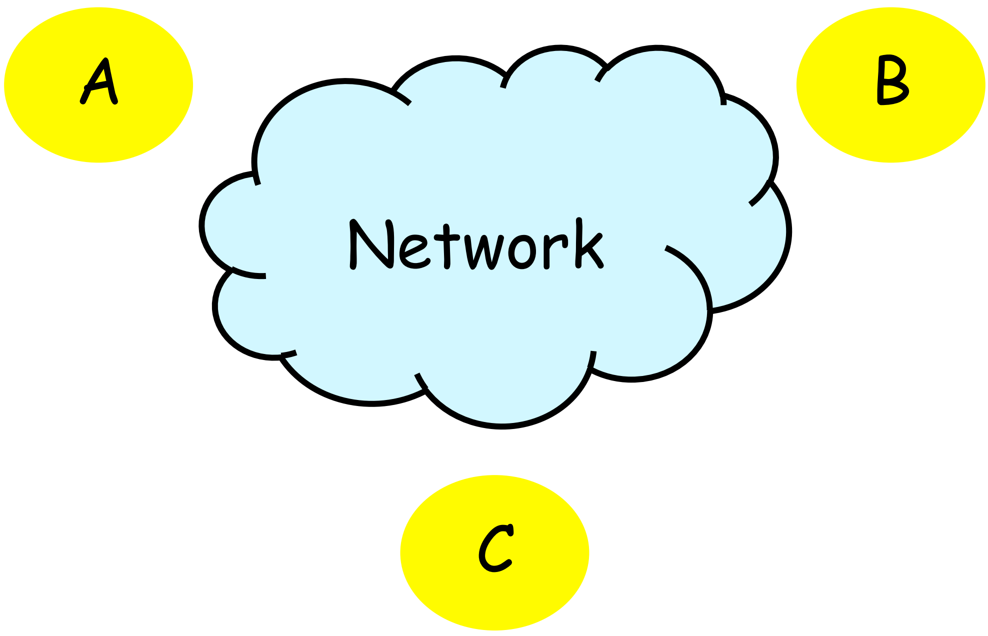
Proof by construction

Lower Bound

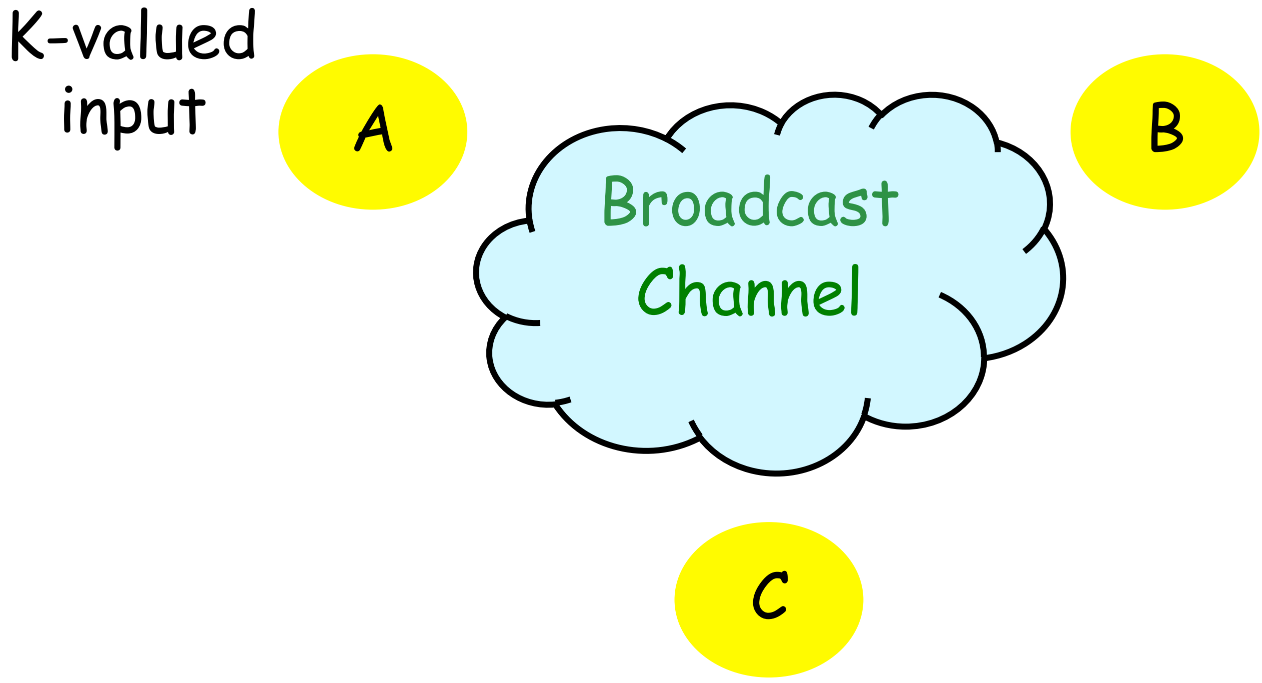
Proof by fooling set argument

Generalization to n parties

n-Node Equality Problem



Number-in-Hand Model



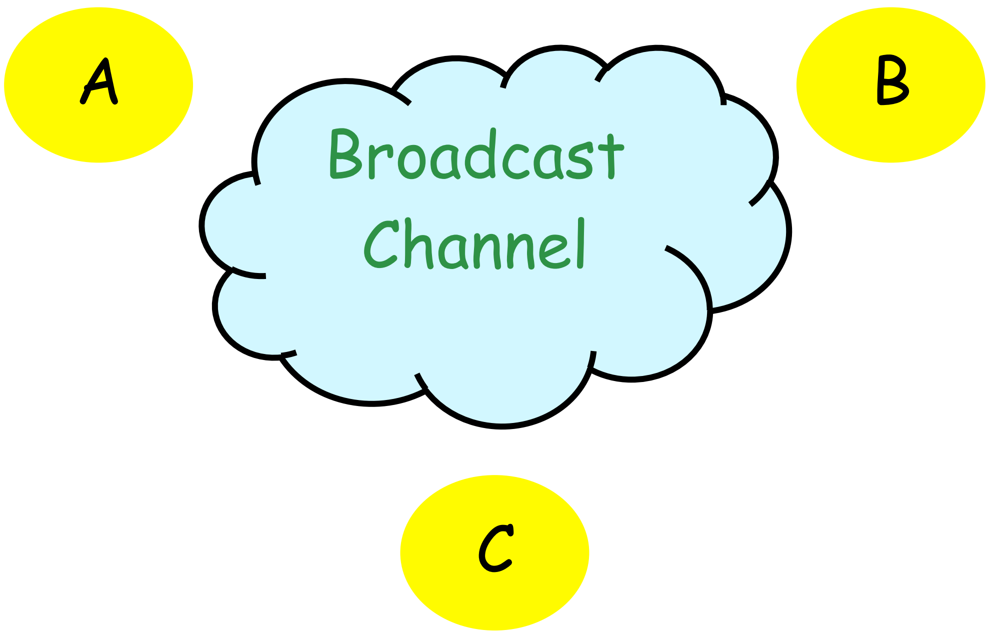
Node i initially knows Xi

n-Party Equality : Complexity

- Broadcast channel + Number-in-hand model

log K bits

Number-on-Forehead Model



Node i initially knows everything except Xi

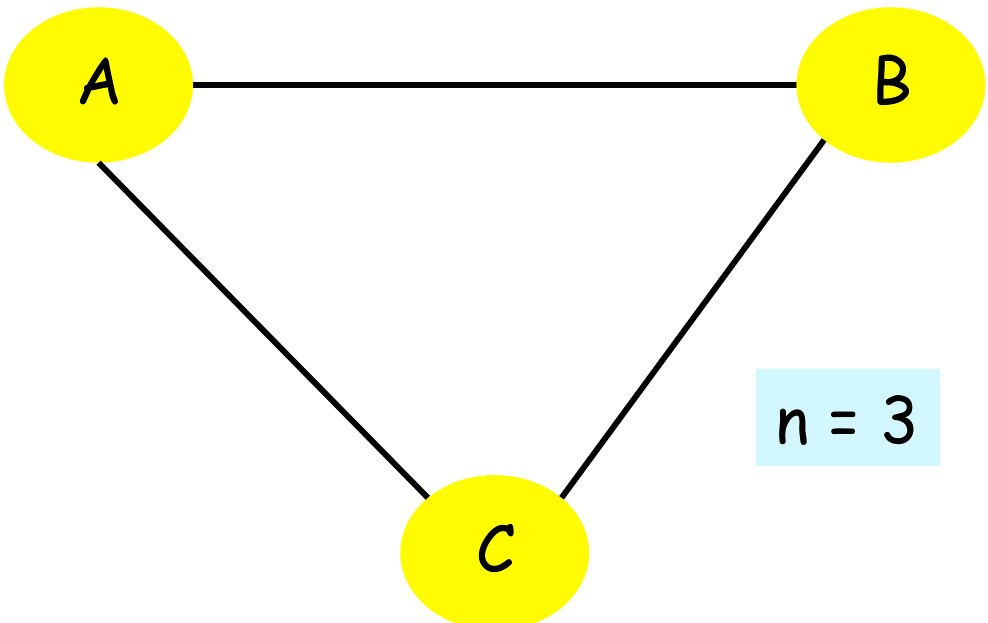
n-Party Equality : Complexity

- Broadcast channel + Number-on-forehead model

  2 bits

Point-to-Point Networks

Private channels & number-in-hand

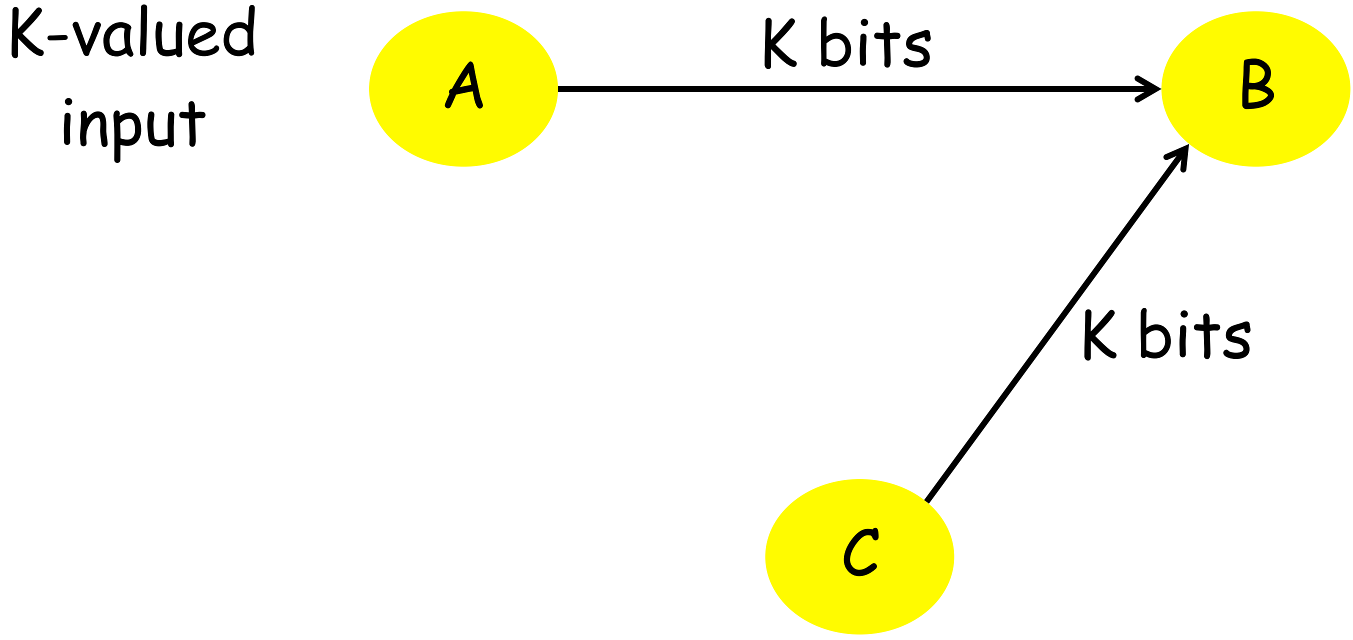


Upper Bound

Emulate broadcast channel using p2p links

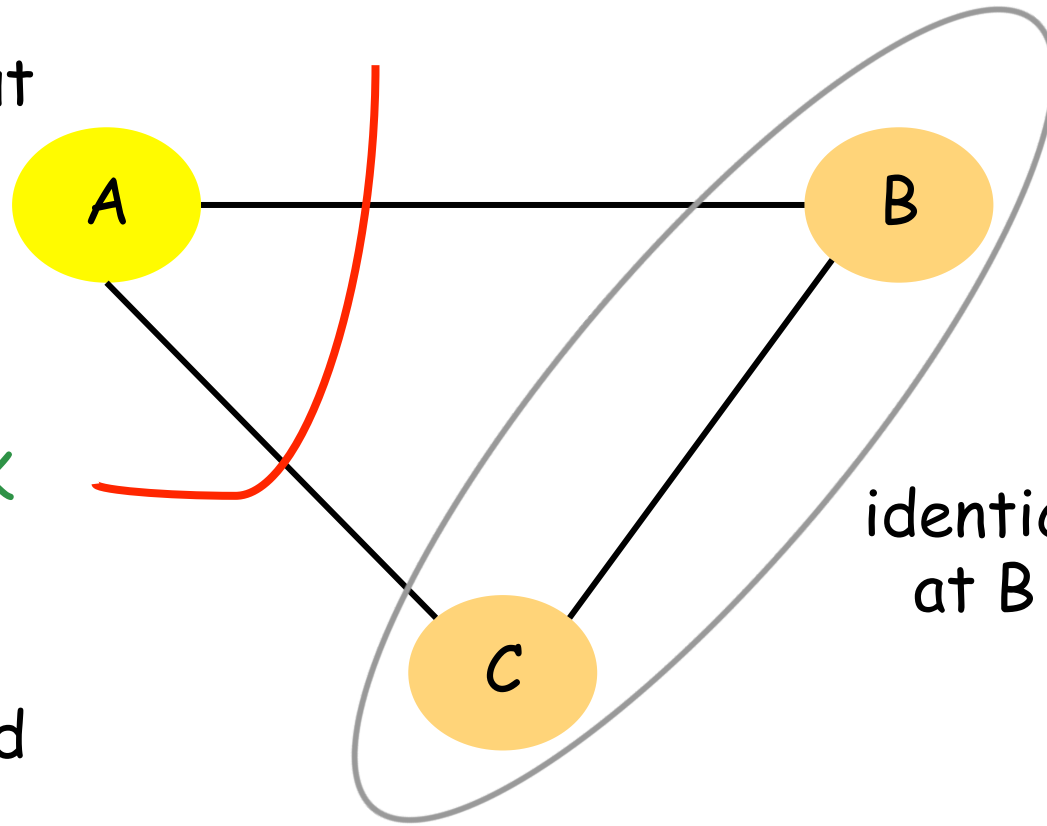
➔ (n-1) * complexity with broadcast channel

Upper Bound = 2 log K



Lower Bound $= \frac{3}{2} \log K$

K-valued input

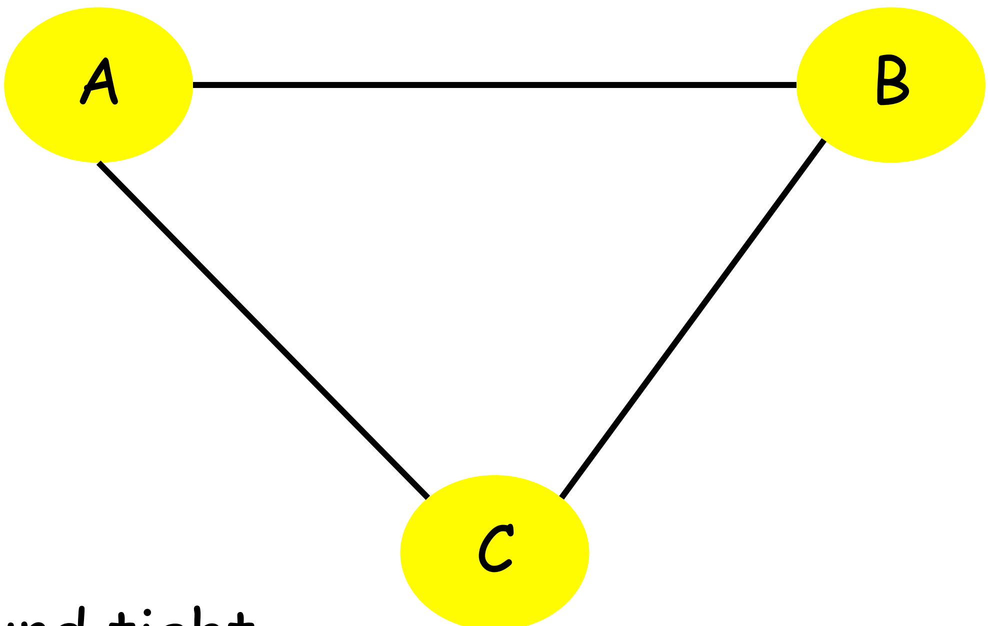


cut $\geq \log K$

identical value at B and C

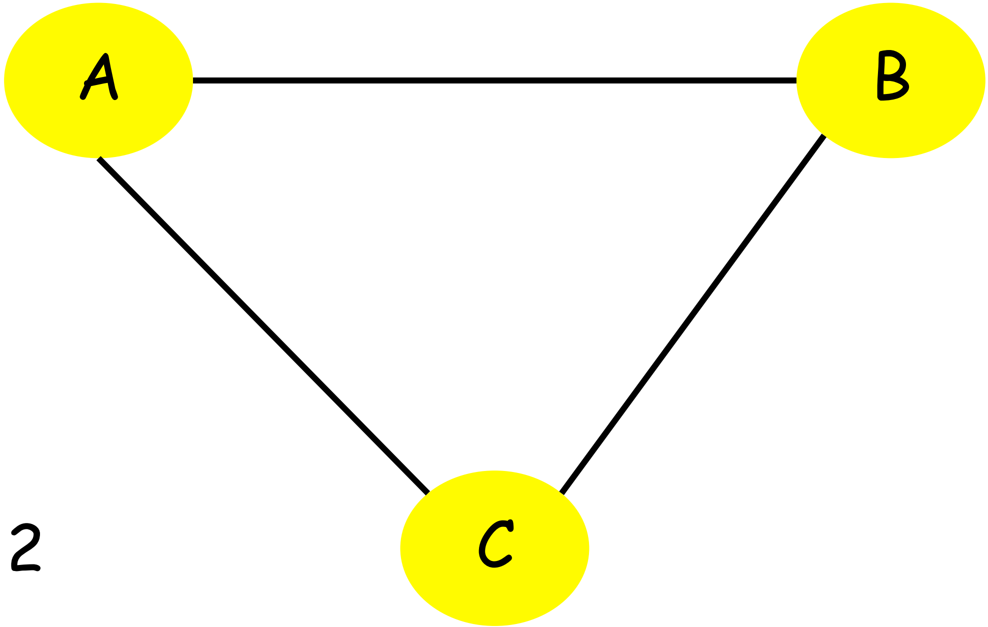
by 2-node lower bound

$1.5 \log K \leq \text{Complexity} \leq 2 \log K$



Neither bound tight
in general

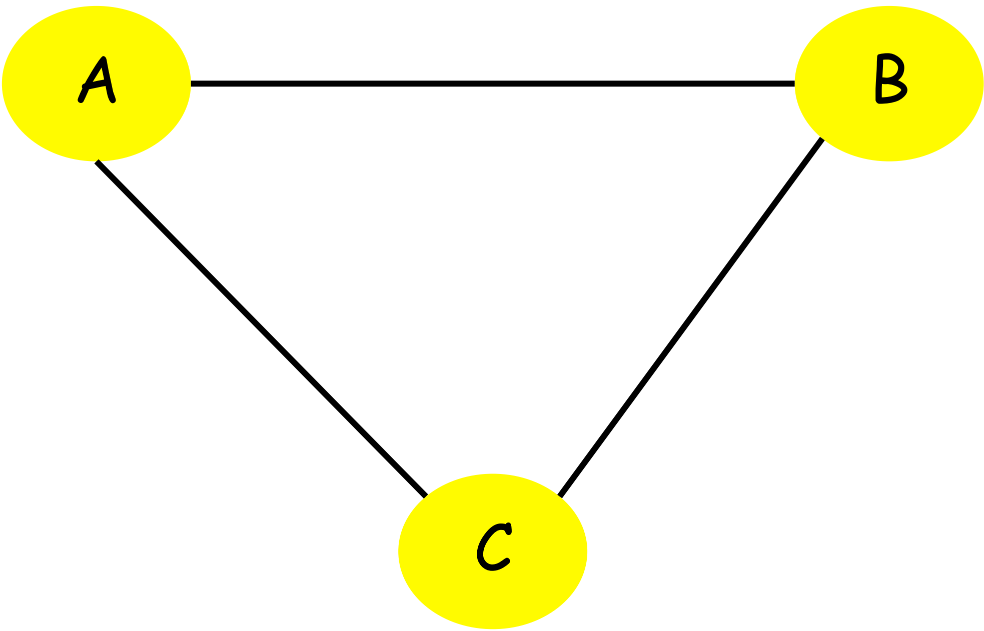
1.5 log K   Not Tight



K = 2

Requires at least 2 bits

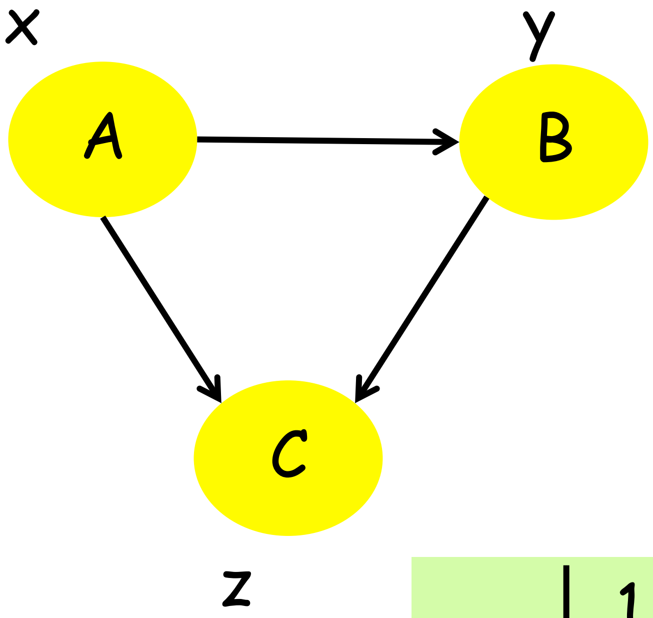
2 log K Not Tight



Proof by construction
for K = 6

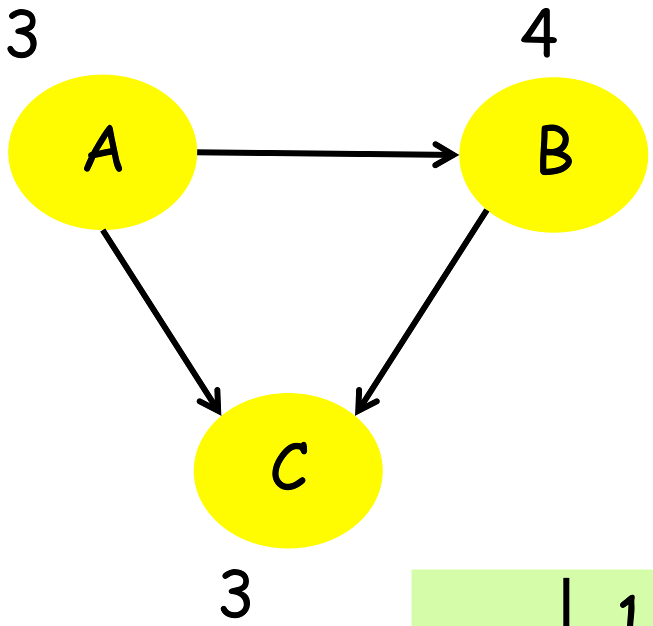
⇨ 2 log K = log 36

K = 6



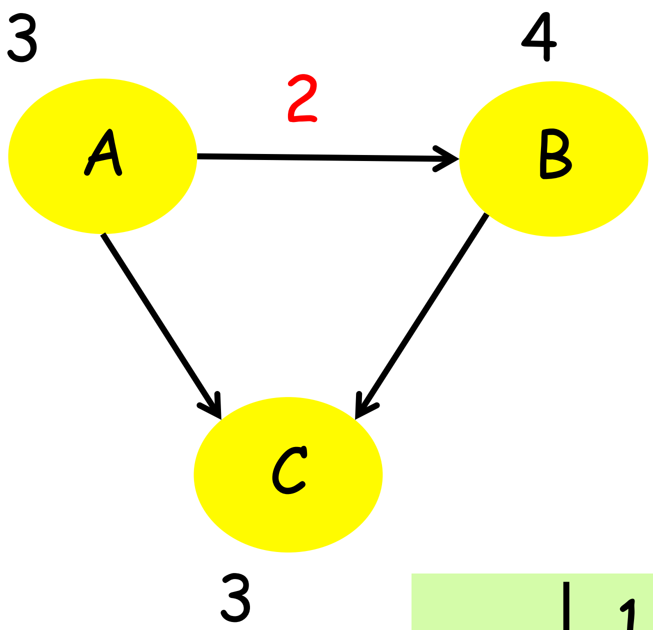
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Example



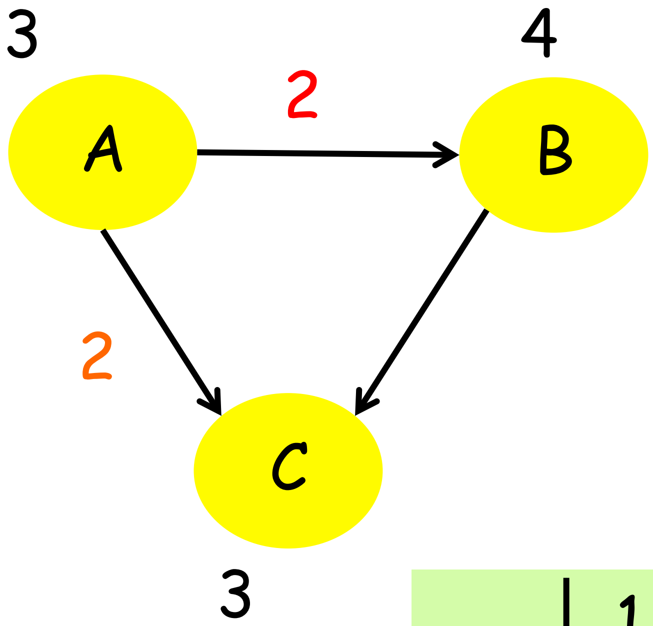
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Example



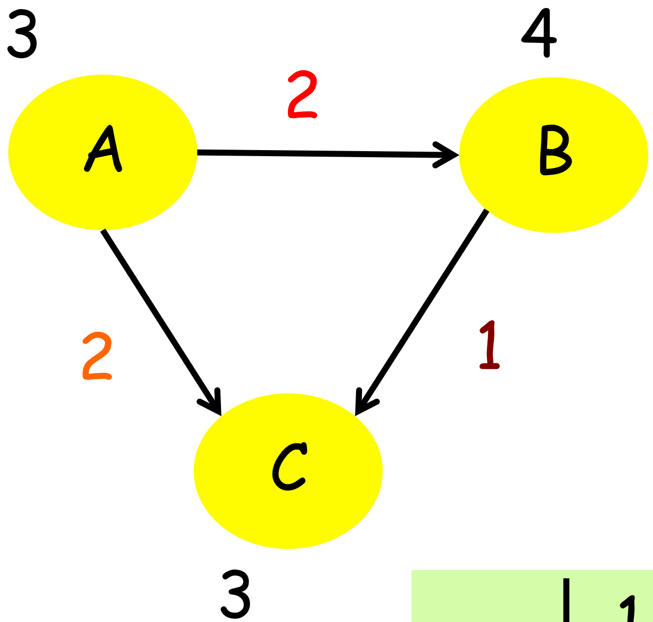
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Example



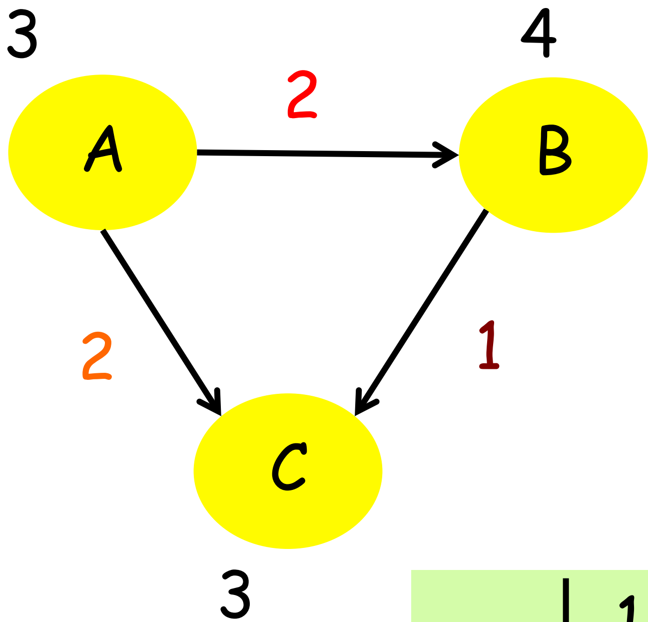
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Example



	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Example



AB(4) = 2

AC(2) = 3

BC(3) ≠ 1 ←

	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Communication Cost

$3 \log 3 = \log 27 < \log 36 = 2 \log K$

Can be generalized to large K and n to yield communication cost approximately

$$0.92\,(n-1) \log K$$

Communication Cost

$3 \log 3 = \log 27 \quad < \quad \log 36 = 2 \log K$

Can be generalized to large K and n to yield communication cost approximately

$$0.92\,(n-1)\log K$$

Cost of informing outcome to each other negligible for large K

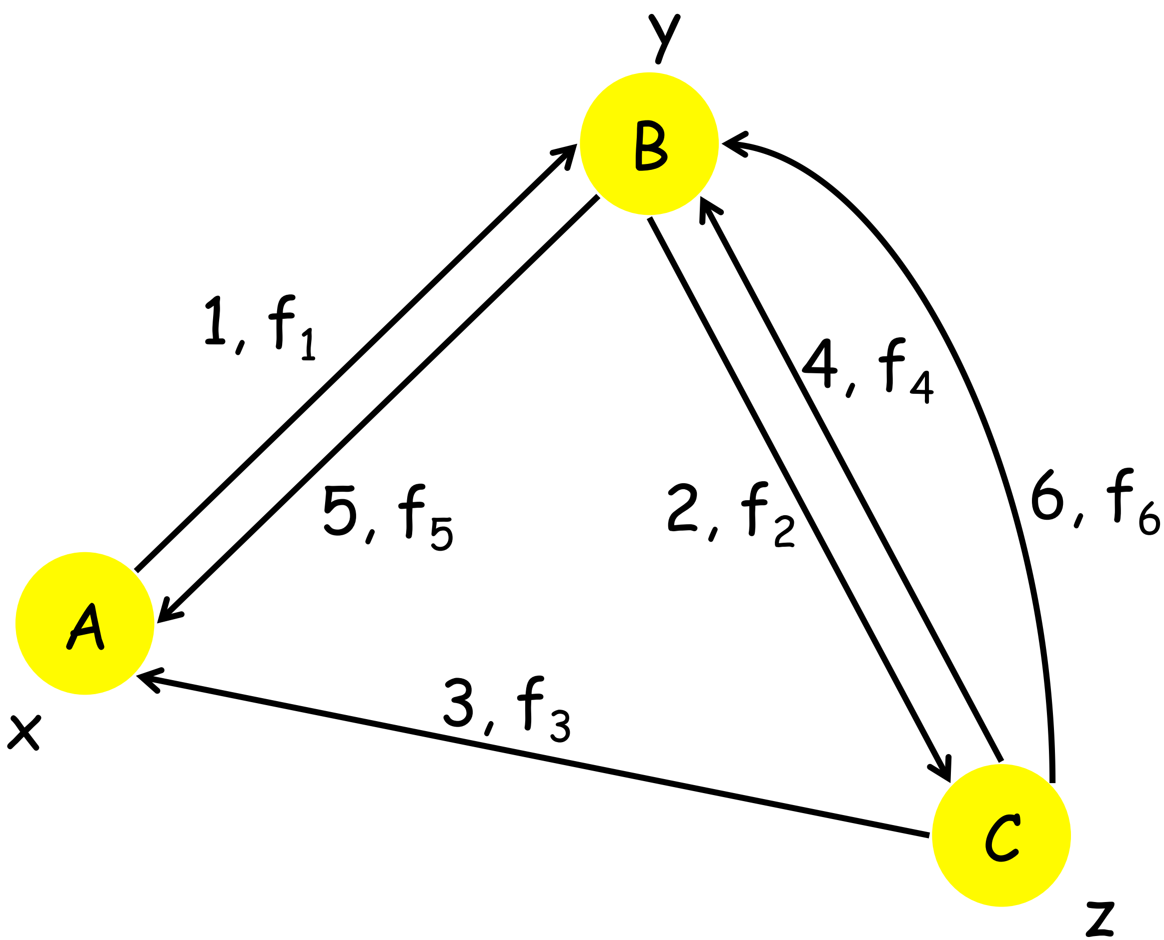
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

Reduce Search Space

"Static" Algorithms

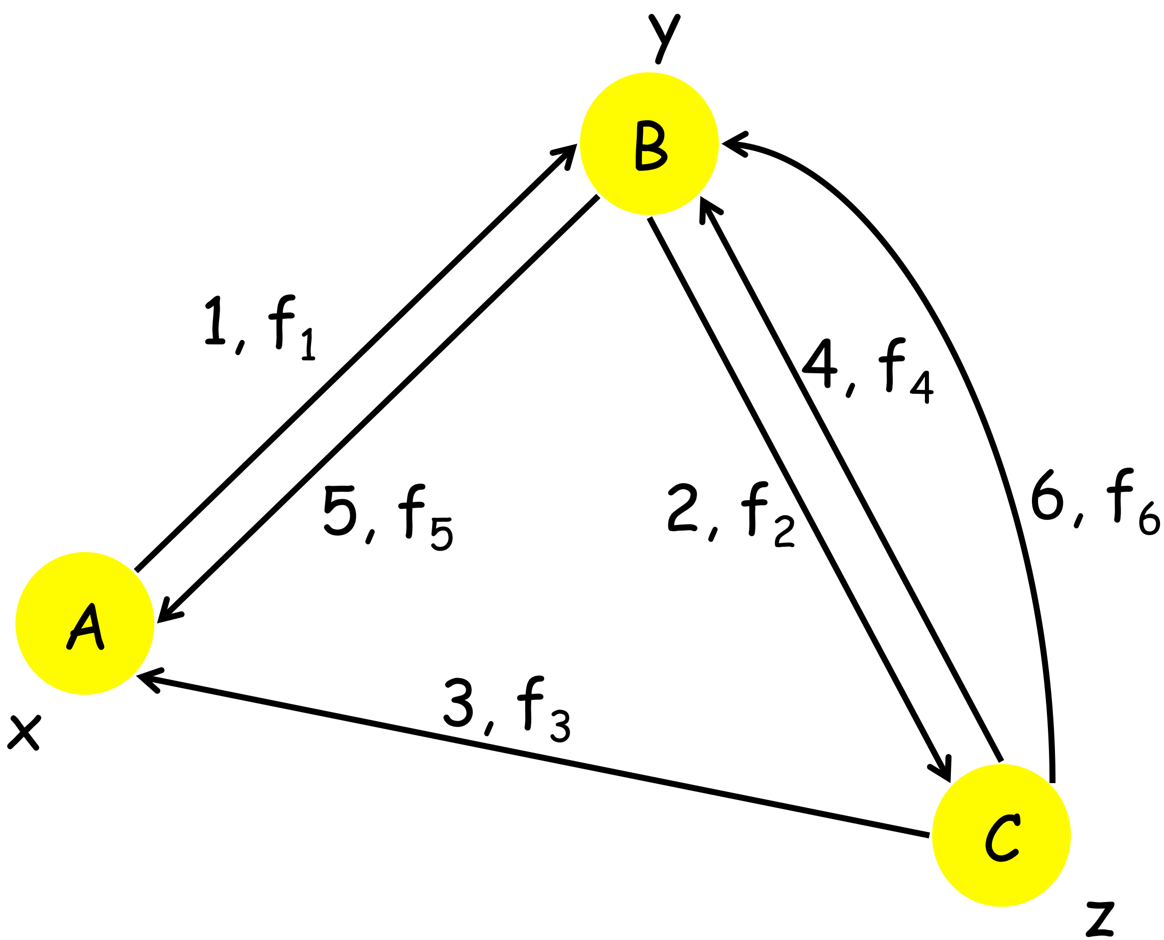
- Node transmitting in round R &
  its output function in round R
  pre-determined
  - Output … function of initial input, and history

Fixed Algorithm: Directed Graph Representation



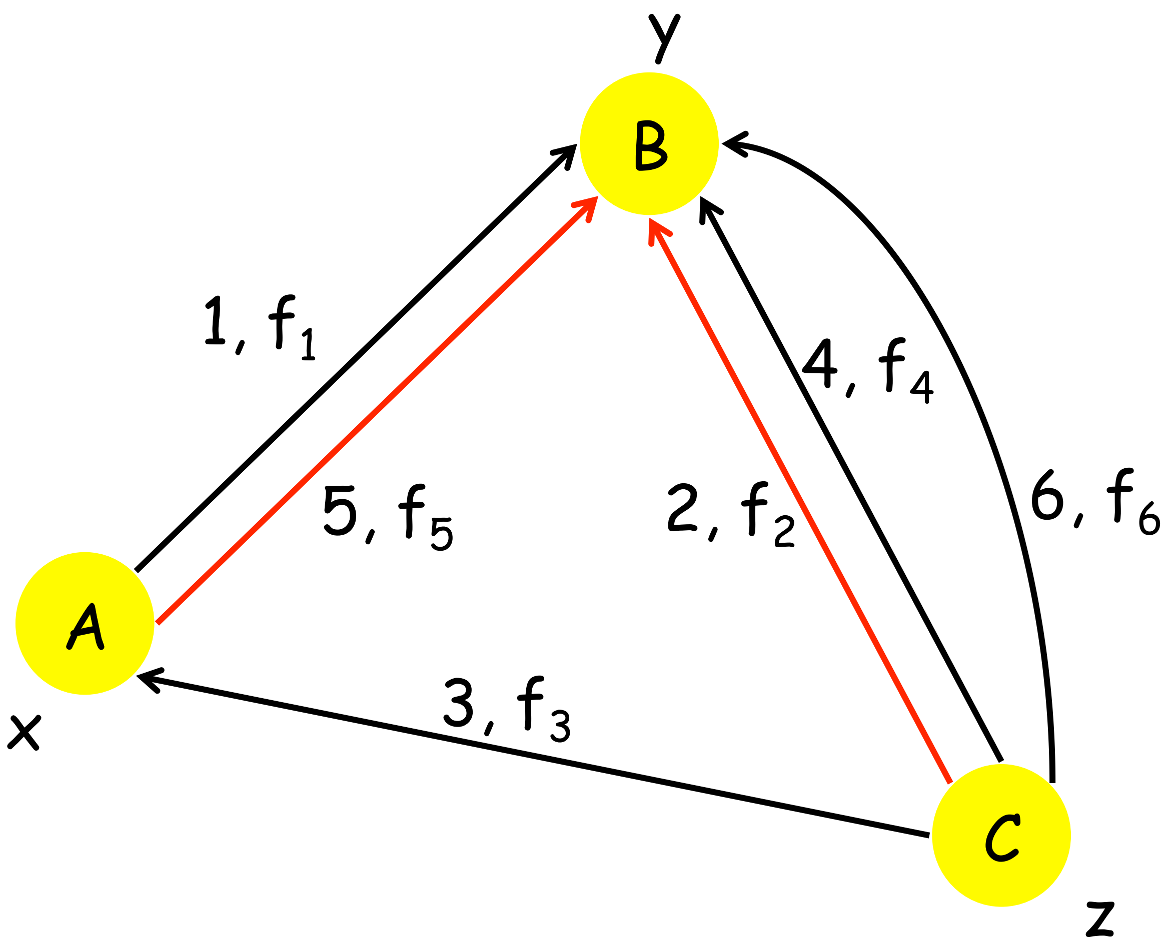
Round number R , function f used in round R

Fixed Algorithm: Directed Graph Representation



Round number R , function f used in round R

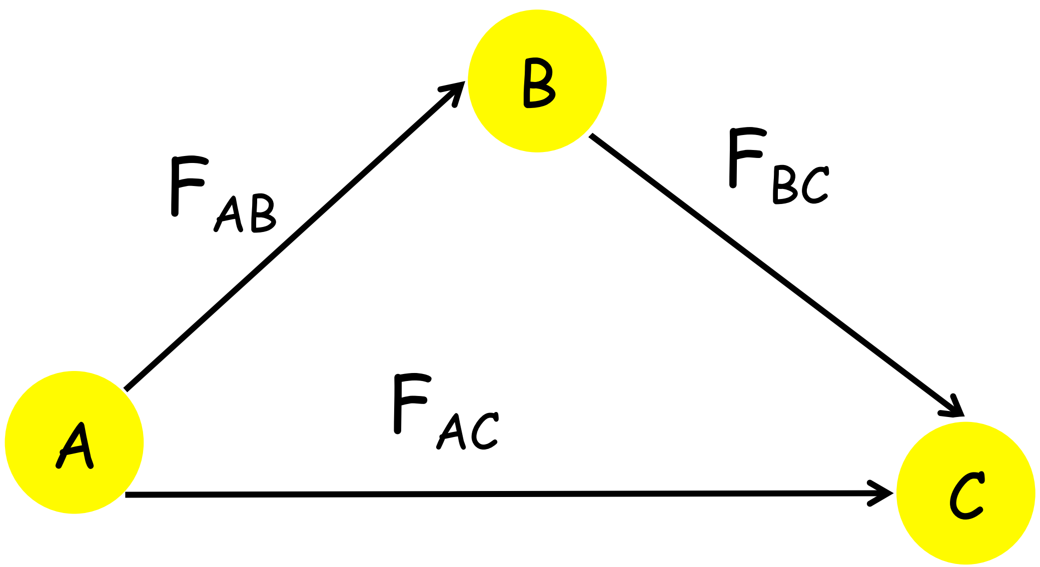
Equivalent Algorithm: Directed Acyclic Graph



Round number R , function f used in round R

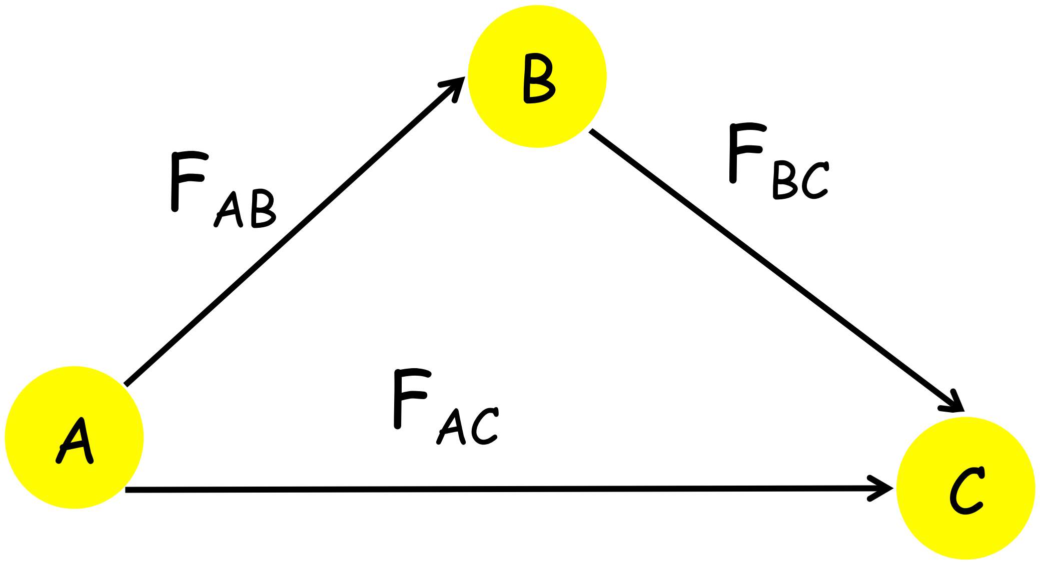
Equivalent Algorithm

- Acyclic graph
- Output depends only on initial input



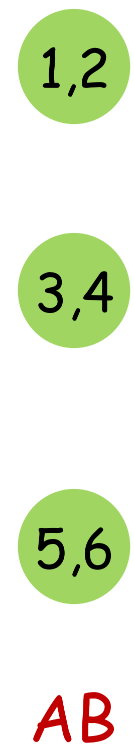
Mapping to a Bipartite Graph

- Each such algorithm can be mapped to a bipartite graph representation



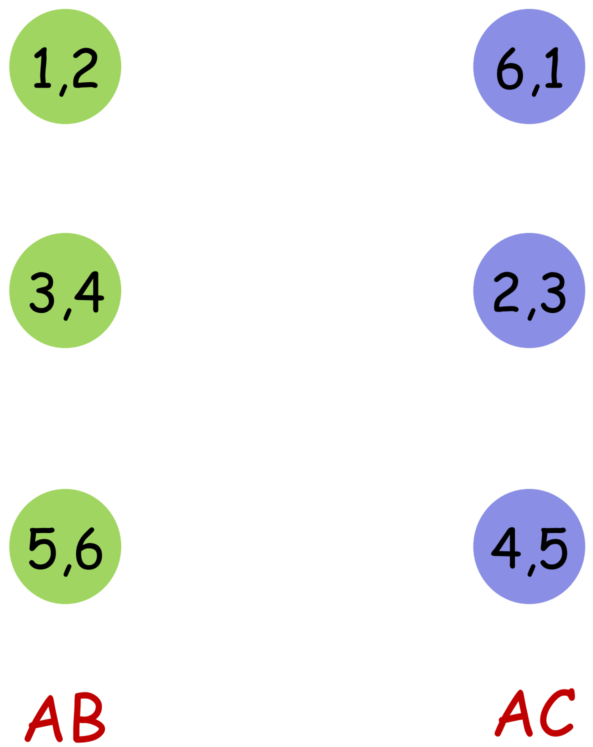
Example

	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3



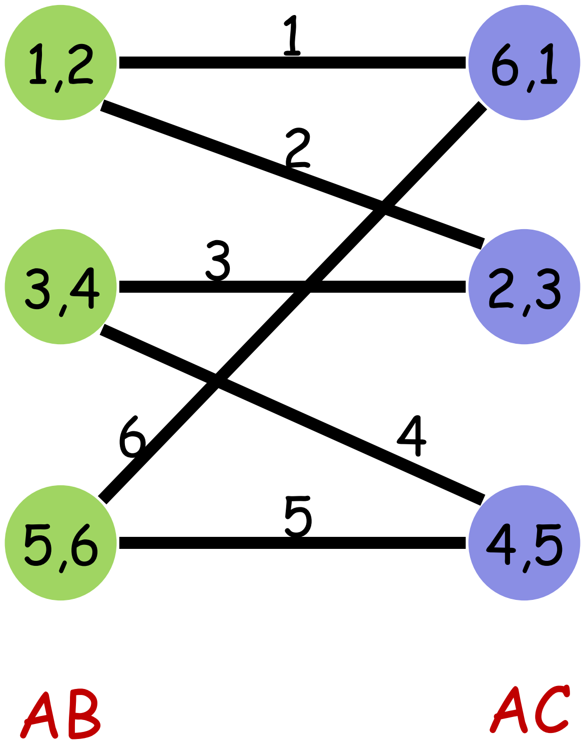
Example

	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3



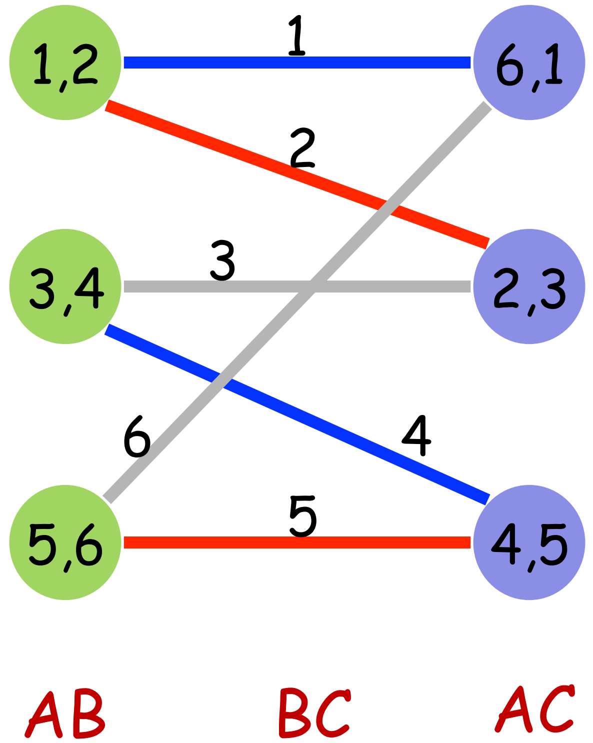
Example

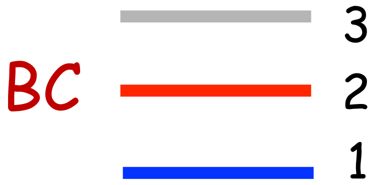
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3



Example

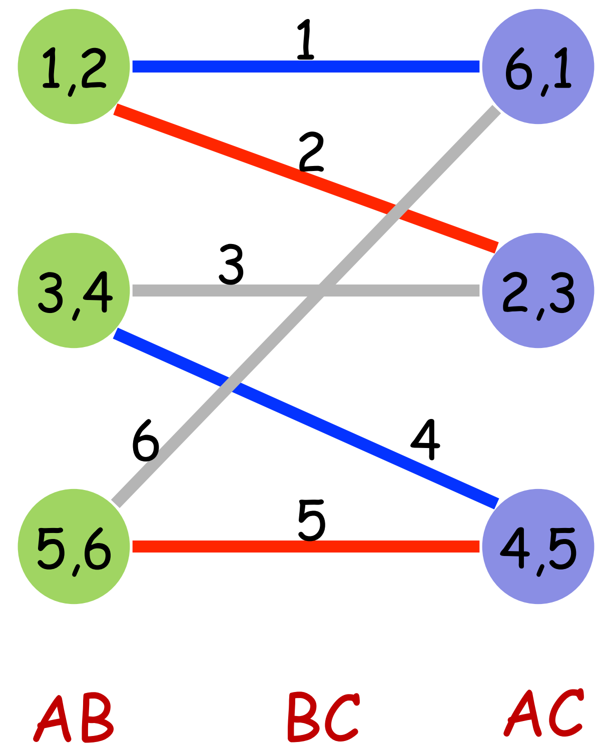
	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3

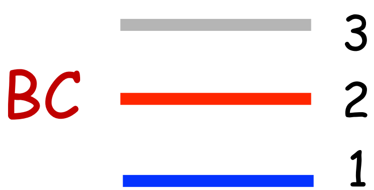




BC assigns colors to edges

	1	2	3	4	5	6
AB	1	1	2	2	3	3
AC	1	2	2	3	3	1
BC	1	2	3	1	2	3



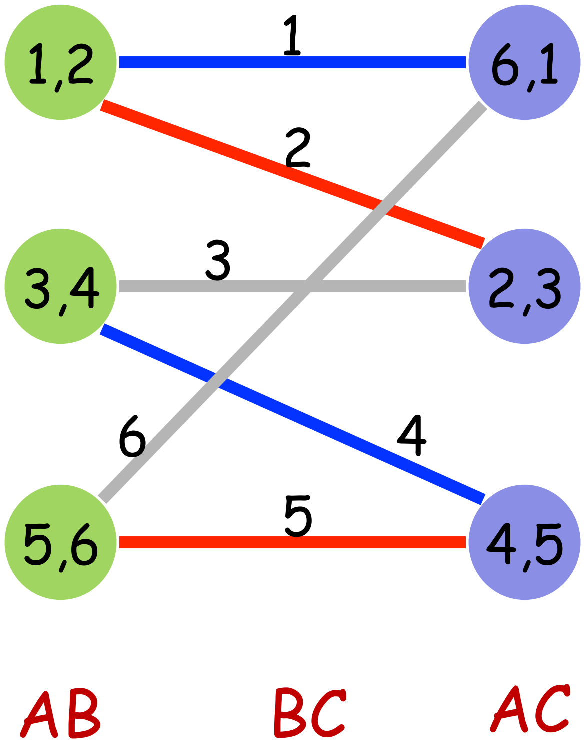


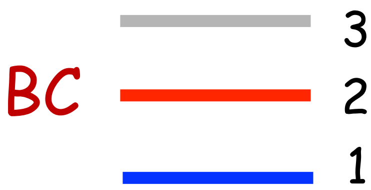
U : # nodes on left

V : # colors

W : # nodes on right

U	V	W
3	3	3





Equality ←→ Bipartite Graph

A colored bipartite graph corresponds to a fixed algorithm for 3-node equality with cost log UVW

if and only if

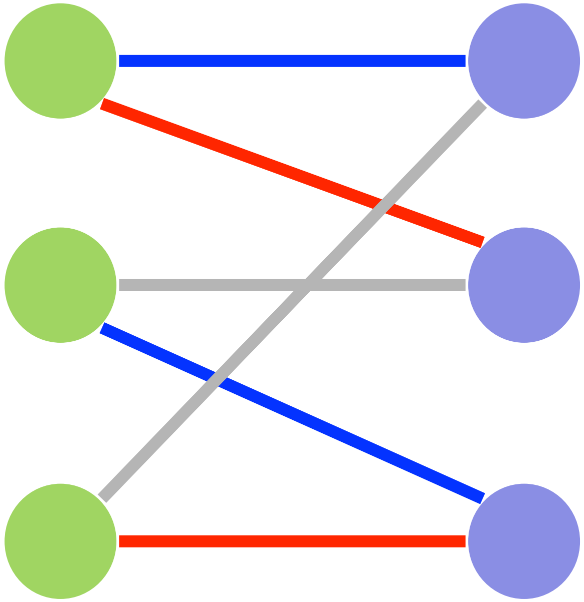
(a) distance-2 colored (strong edge coloring)
(b) Number of edges = K
(c) $U \times V \geq K$
(d) $U \times W \geq K$
(e) $V \times W \geq K$

Fixed Algorithm Design

- Find a suitable bipartite graph

- Our algorithm

    ➔ 6-cycle



Lower Bounds

- The mapping can be used to prove lower bounds for small K

For K = 6

- Least cost over all fixed algorithms is
  log 27

Detour … an open conjecture

A bipartite graph with

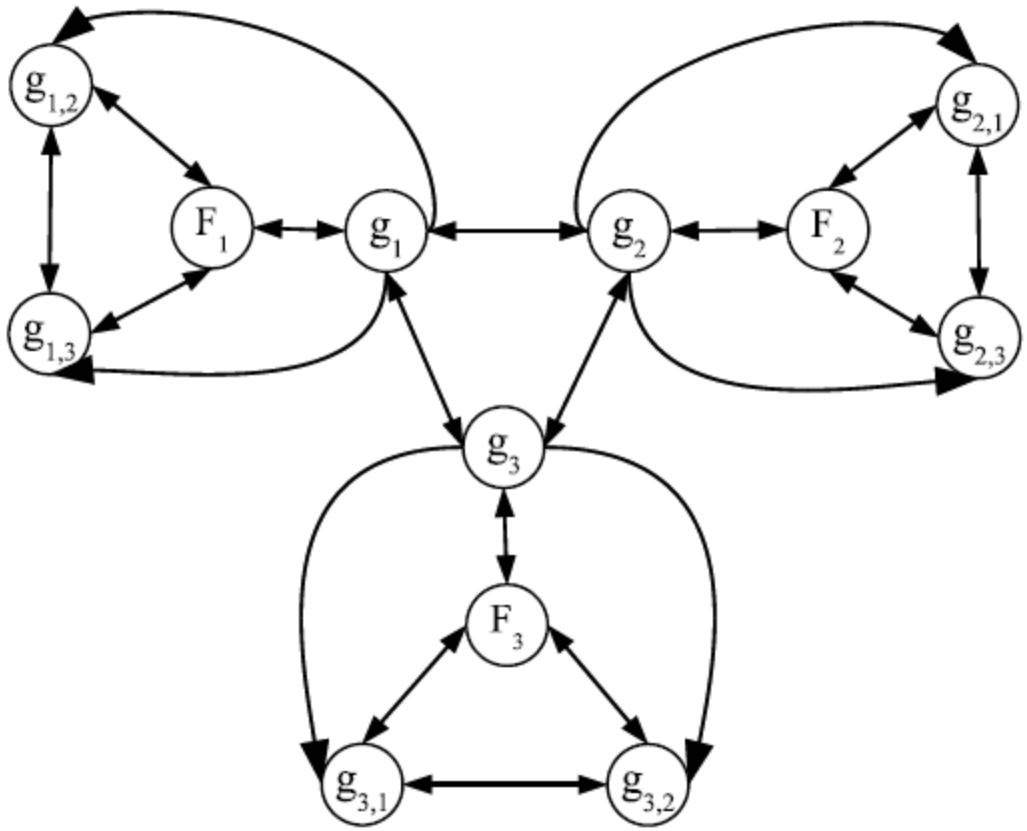
- D1 = maximum degree on left
- D2 = maximum degree on right

can be distance-2 colored with D1 * D2 colors

Why is equality interesting ?

Lower Bound on Consensus

- Mapping between Byzantine broadcast and multiple instances of equality



- g_1 – g_3 are good peers
- F_1 – F_3 are virtual bad sources acting with different inputs
- $g_{i,j}$ are virtual good peer of node j to node i

Byzantine Broadcast: n nodes, f faults

- Broadcast algorithm solves Equality (MEQ-AD) problem for each subset of (n-f) nodes

- n-choose-(n-f) such subsets
- Each link belongs to (n-2)-choose-(n-f-2) such subsets

➔ Complexity of broadcast lower bounded by

EQ * n-choose-(n-f) / (n-2)-choose-(n-f-2)

- EQ ≥ (n-f) L /2 bits for equality of L bits among n-f nodes

→ Broadcast of L bits requires at least

$$L * n(n-1)/2(n-f-1)$$

Our algorithms for broadcast/consensus:

within factor of 2 of above lower bound

Open Problems

Open Problems

- Characterizations of communication complexity for point-to-point networks
  - Alternatives to Yao model seem more appropriate
- Equality for larger networks
- Lower bounds on
  - Equality
  - Byzantine consensus
  - Byzantine broadcast …

Thanks !

Thanks !

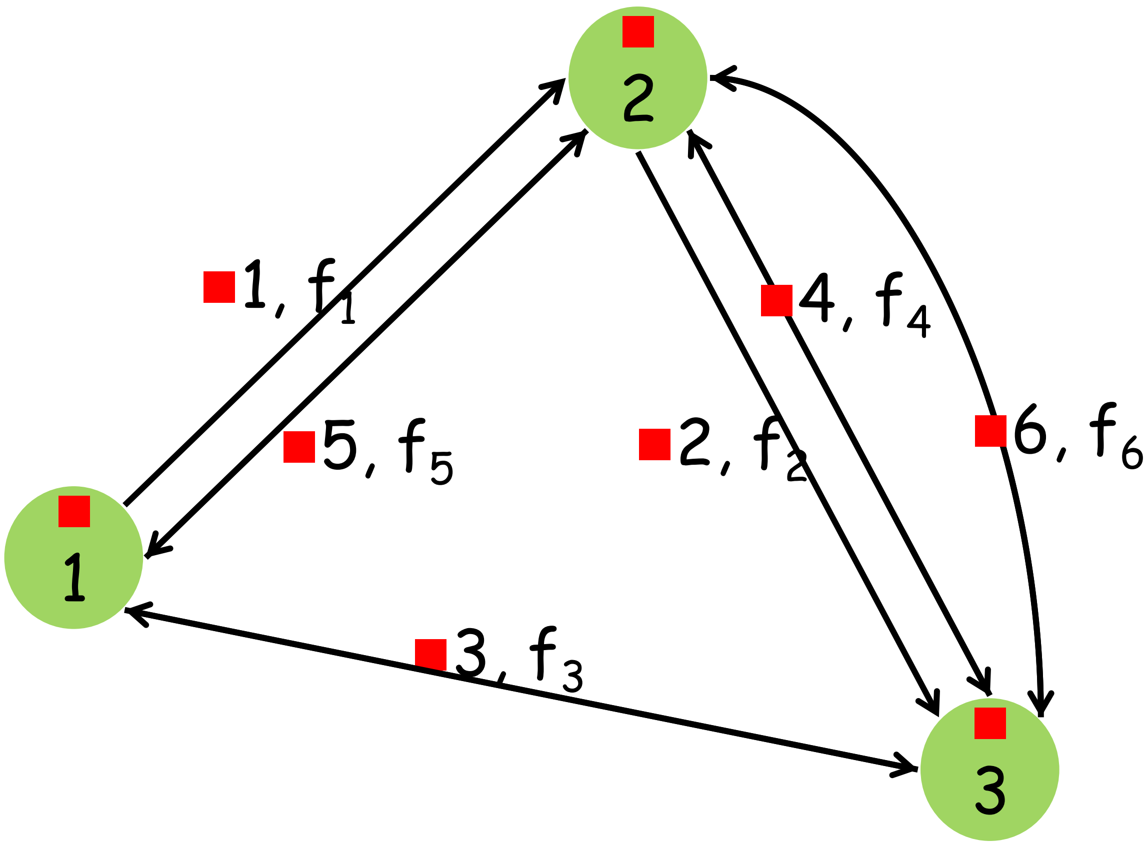
The MEQ(n,M) Problem

- n nodes each given x_i from {1,…,M}, to check if all x_i are equal
- Each node computes EQ_i

$$EQ(x_1,\cdots,x_n)=\begin{cases}0 & \text{if } x_1=\cdots=x_n\\ 1 & \text{otherwise}\end{cases}$$

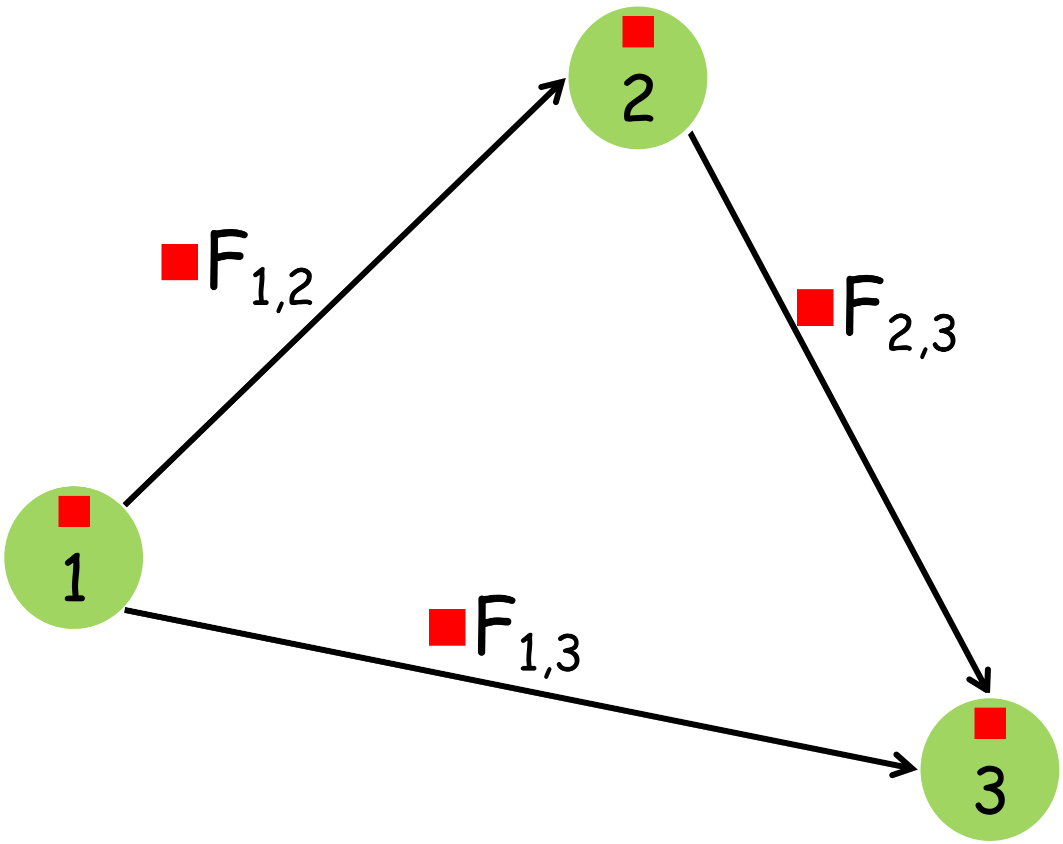
$$\exists i, EQ_i = 1 \Leftrightarrow EQ(x_1,\cdots,x_n)=1$$

Graph Representation an Algorithm



- Transform to a partially ordered DAG

Graph Representation an Algorithm



- $F_{i,j}$ depends on x_i only

Definition of Complexity

- Complexity of an algorithm

$$C(P) = \sum_{i>j} \log_2 \left| F_{i,j} \right|$$

- Complexity of MEQ(n,M)

$$C_{MEQ}(n,M) = \min_{P \text{ solves } MEQ(n,M)} C(P)$$

Upper Bound by Construction

- Send $x_1, \ldots, x_{n-1}$ to node n
- Set $EQ_1 = \ldots = EQ_{n-1} = 0$
- Compute $EQ_n = EQ(x_1, \ldots, x_n)$

- → $$C_{MEQ}(n, M) \le (n-1)\log_2 M$$

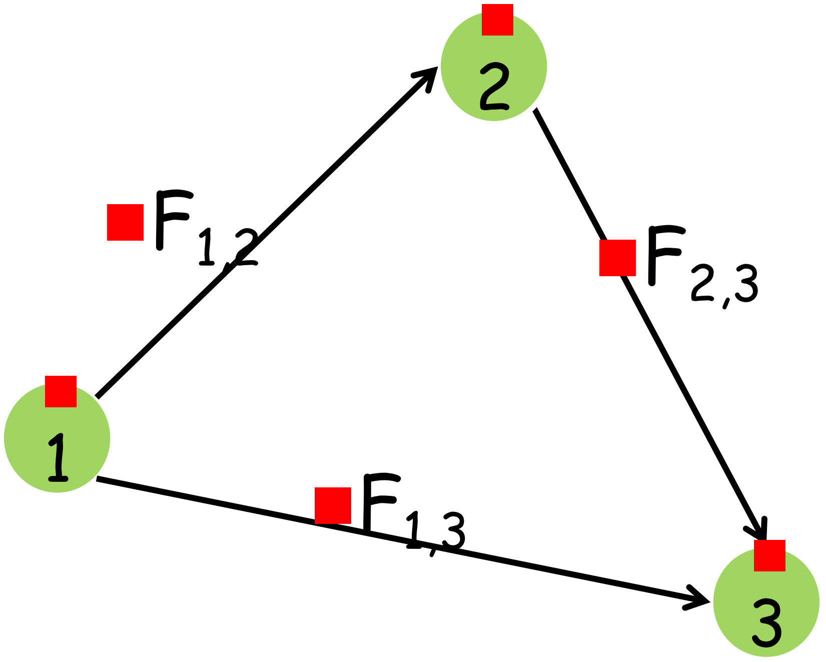
Cut-Set Lower Bound

- Fooling Set argument
  - Every node must send + receive $\geq \log_2 M$

- ➔ $$C_{MEQ}(n, M) \geq \frac{n}{2} \log_2 M$$

Neither bound is tight

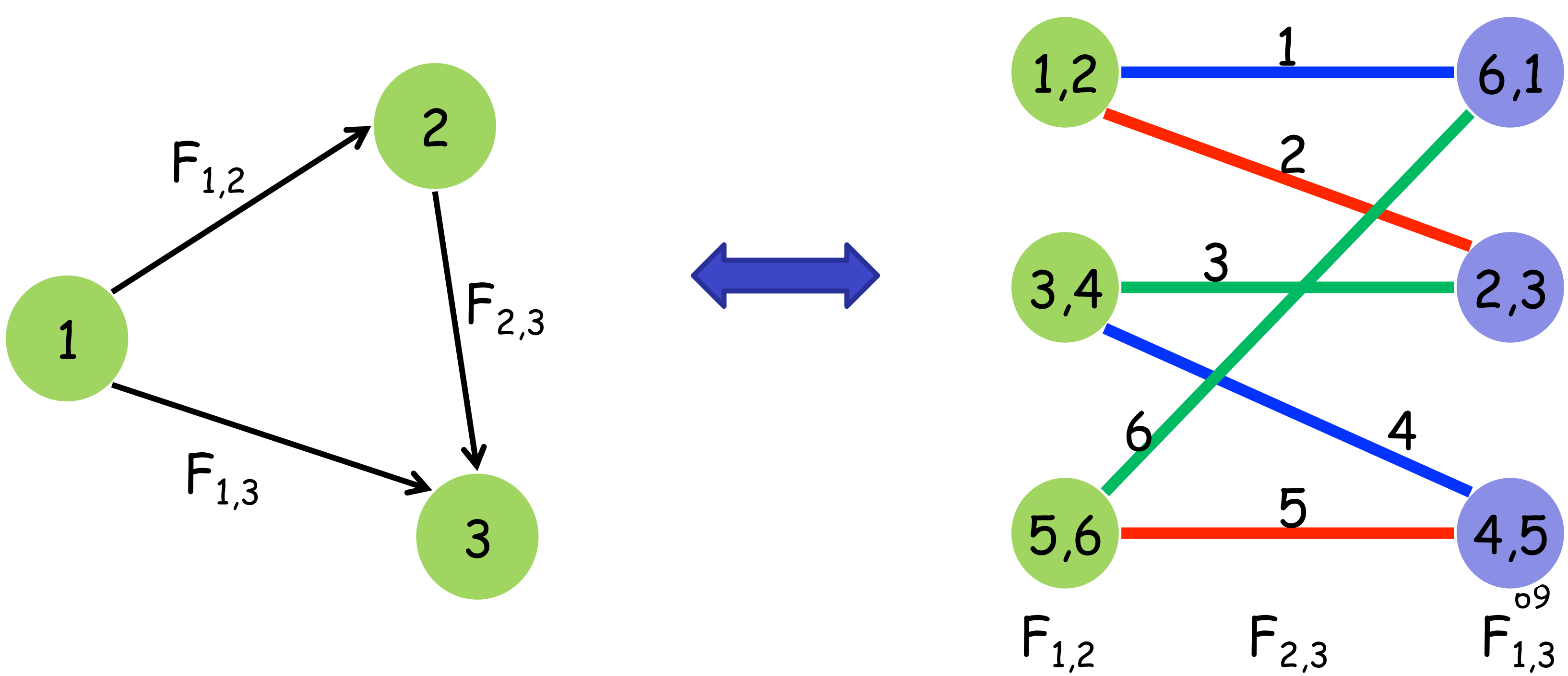
MEQ(3,6)



$$\frac{n}{2}\log_2 6 < C_{MEQ}(3,6) = 3\log_2 3 < 2\log_2 6$$

Proof by Strong Edge Coloring

MEQ(3,M) algorithm
= bipartite graph with M edges
+ distance-2 edge coloring scheme

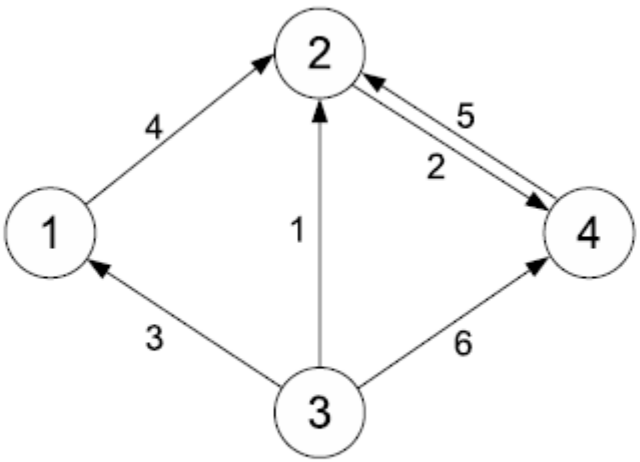


Summary

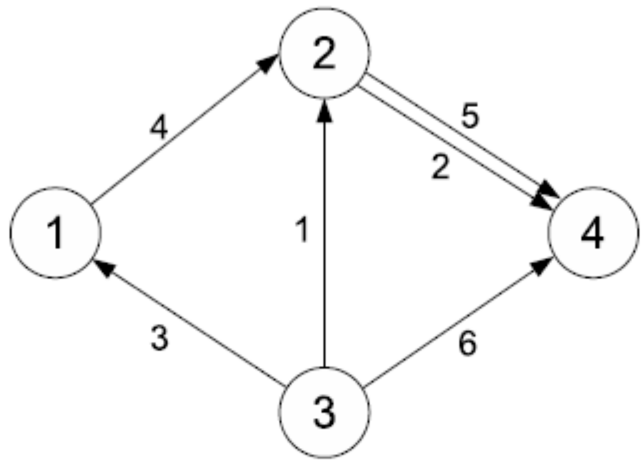
- Introduce the MEQ problem
- Existing techniques give loose bounds
- New technique to reduce space
- Connection among distributed source coding, distributed algorithm, and graph coloring

Future Work

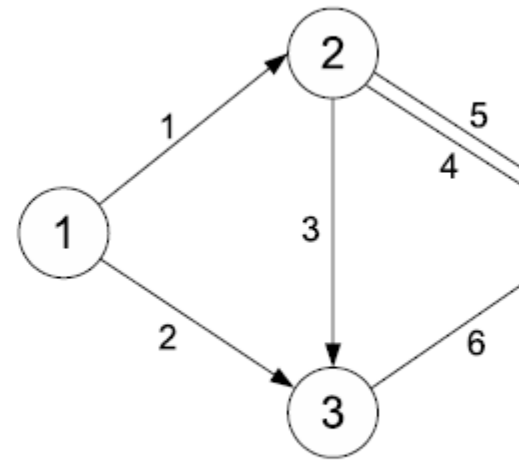
- MEQ(3,M) is open
  - Optimize over $F_{i,j}$ = find an optimal bipartite graph + strong coloring
- Even given $|F_{i,j}|$ is open
- Looking for new techniques



(a) Graph representation of P



(b) An equivalent protocol of P with Step 5 flipped



(c) An iid partially ordered ec
protocol of P

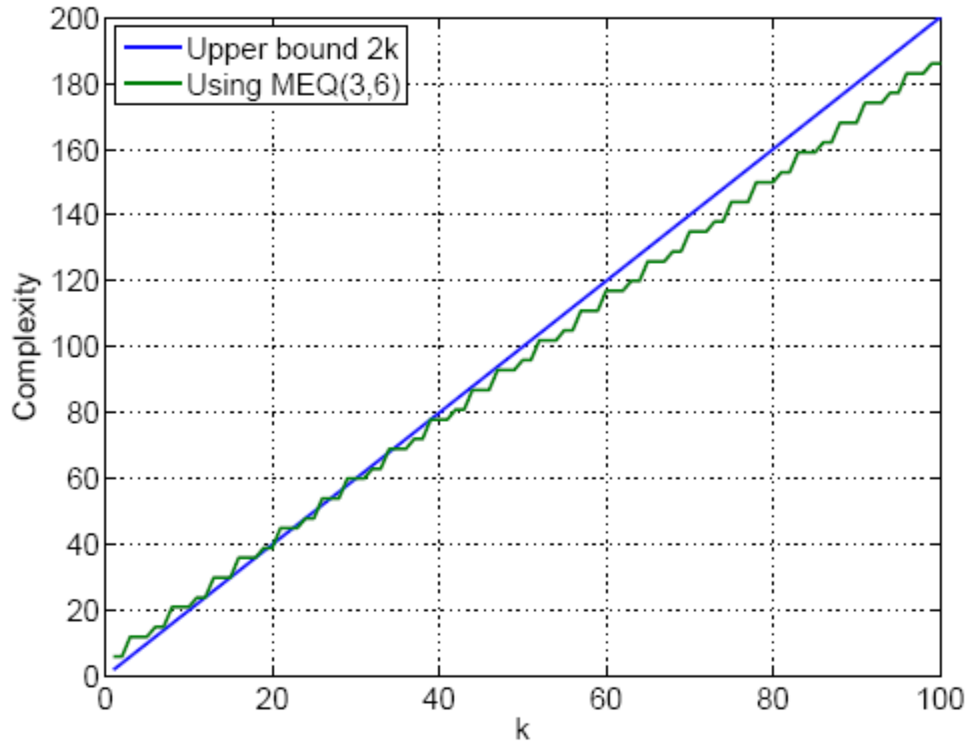


Figure 3: Complexity of the proposed protocol v.s. upper bound 2k

x	1	2	3	4	5	6
s_{AB}	1	1	2	2	3	3
s_{AC}	1	2	2	3	3	1
s_{BC}	1	2	3	1	2	3

Table 1: A protocol for MEQ-AD(3,6)