Lower Bounds for Number-in-Hand Multiparty Communication Complexity

Jeff M. Phillips — Univ. of Utah

Elad Verbin, Qin Zhang — CTIC/MADALGO, Aarhus Univ.

SODA 2012, Kyoto

Jan. 17, 2012

The multiparty communication model

$x_1 = 010011$ $\quad$ $x_2 = 111011$

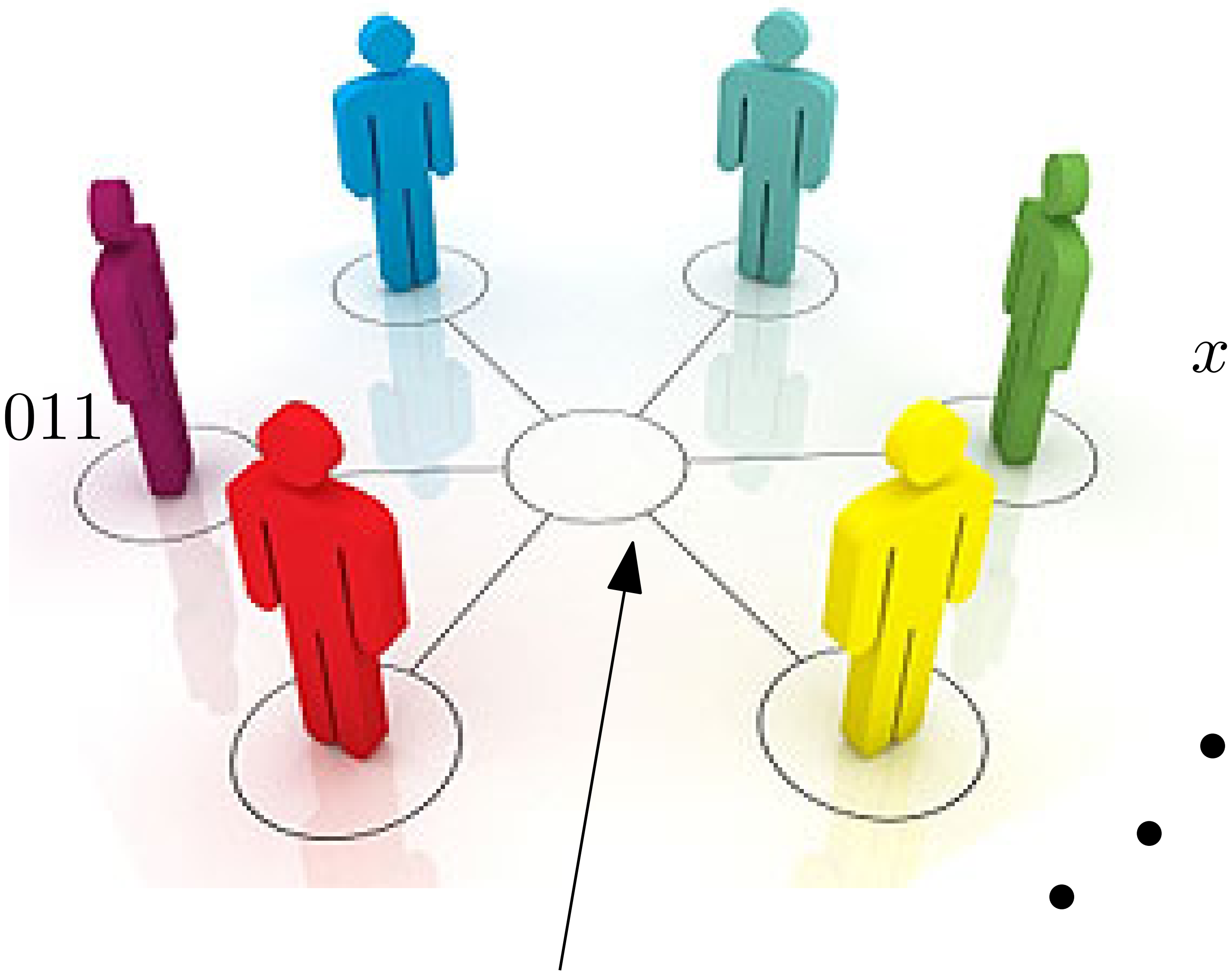
$x_k = 100011$

$x_3 = 111111$

$\vdots$

The multiparty communication model

$x_1 = 010011$ $x_2 = 111011$



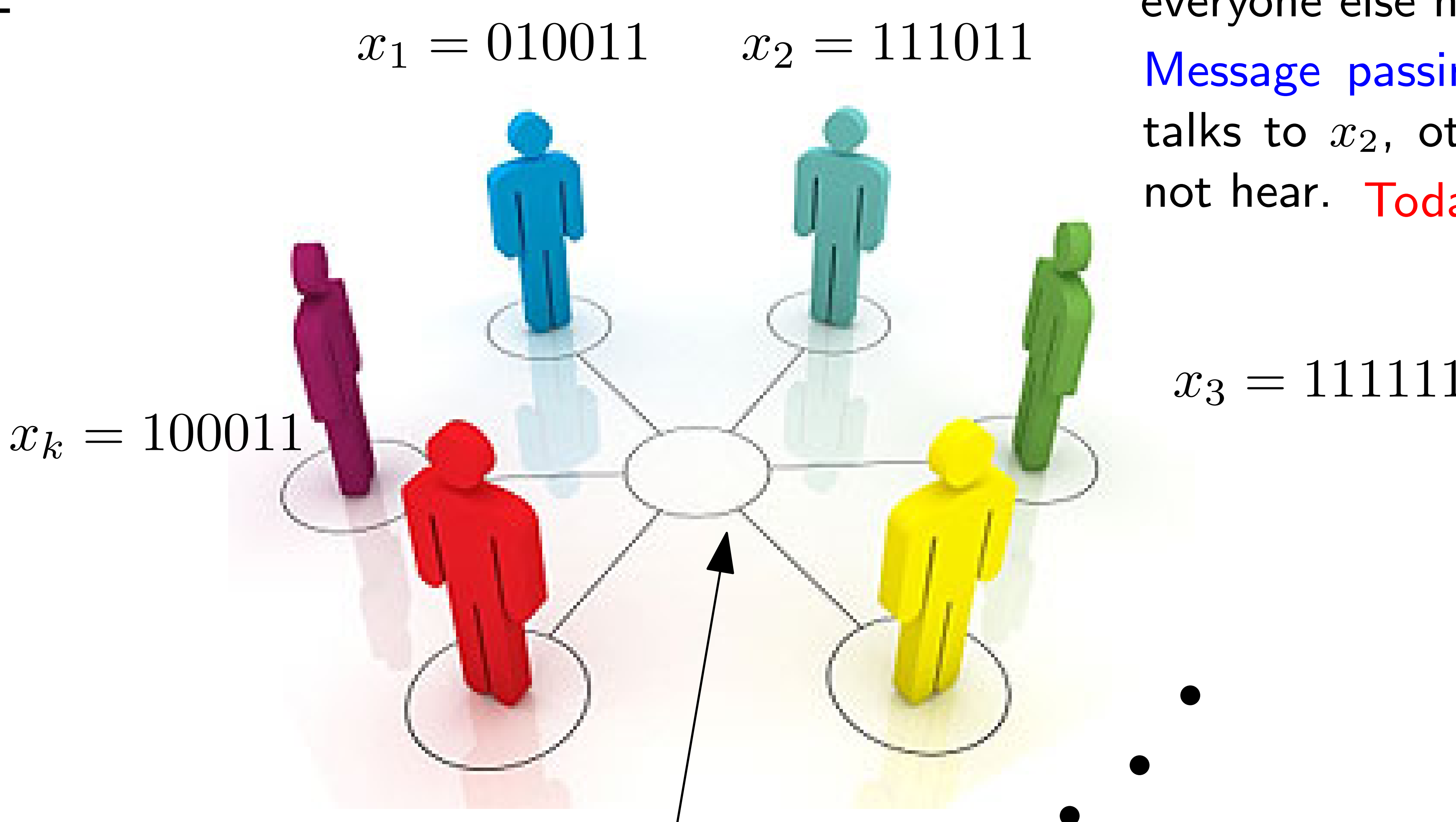
$x_3 = 111111$

$x_k = 100011$

We want to compute $f(x_1, x_2, \ldots, x_k)$

f can be bit-wise XOR, OR, AND, MAJ ...

The multiparty communication model



Blackboard: One speaks, everyone else hears.

Message passing: If x_1 talks to x_2, others cannot hear. Today's focus

We want to compute $f(x_1, x_2, \ldots, x_k)$

f can be bit-wise XOR, OR, AND, MAJ ...

Related work

So natural, must be studied?

Related work

So natural, must be studied?

- The *Blackboard* model: Quite a few works.
- The *Message-passing* model: Almost nothing.

Related work

So natural, must be studied?

- The *Blackboard* model: Quite a few works.
- The *Message-passing* model: Almost nothing.
  - Back to the "ancient" time:
    "lower bounds on the multiparty communication complexity"
    by Duris and Rolim '98.

    Gives some deterministic lower bounds.

Related work

So natural, must be studied?

- The *Blackboard* model: Quite a few works.
- The *Message-passing* model: Almost nothing.
  - Back to the "ancient" time:
    "lower bounds on the multiparty communication complexity"
    by Duris and Rolim '98.

    Gives some deterministic lower bounds.
  - Gal and Gopalan for *"longest increasing sequence"*, '07.
    and Guha and Huang for "*random order streams*", '09.

    Under "private message model" but it is different from ours.

Our results

1. $\Omega(nk)$ for the k-bitwise-XOR/OR/AND/MAJ.
2. $\Omega(n \log k)$ for k-bitwise-AND/OR in the blackboard model.
3. $\tilde{\Omega}(nk)$ for k-connectivity.

All tight, and for randomized algorithms.

Our results

1. $\Omega(nk)$ for the k-bitwise-XOR/OR/AND/MAJ.
2. $\Omega(n \log k)$ for k-bitwise-AND/OR in the black-board model.
3. $\tilde{\Omega}(nk)$ for k-connectivity.

All tight, and for randomized algorithms.

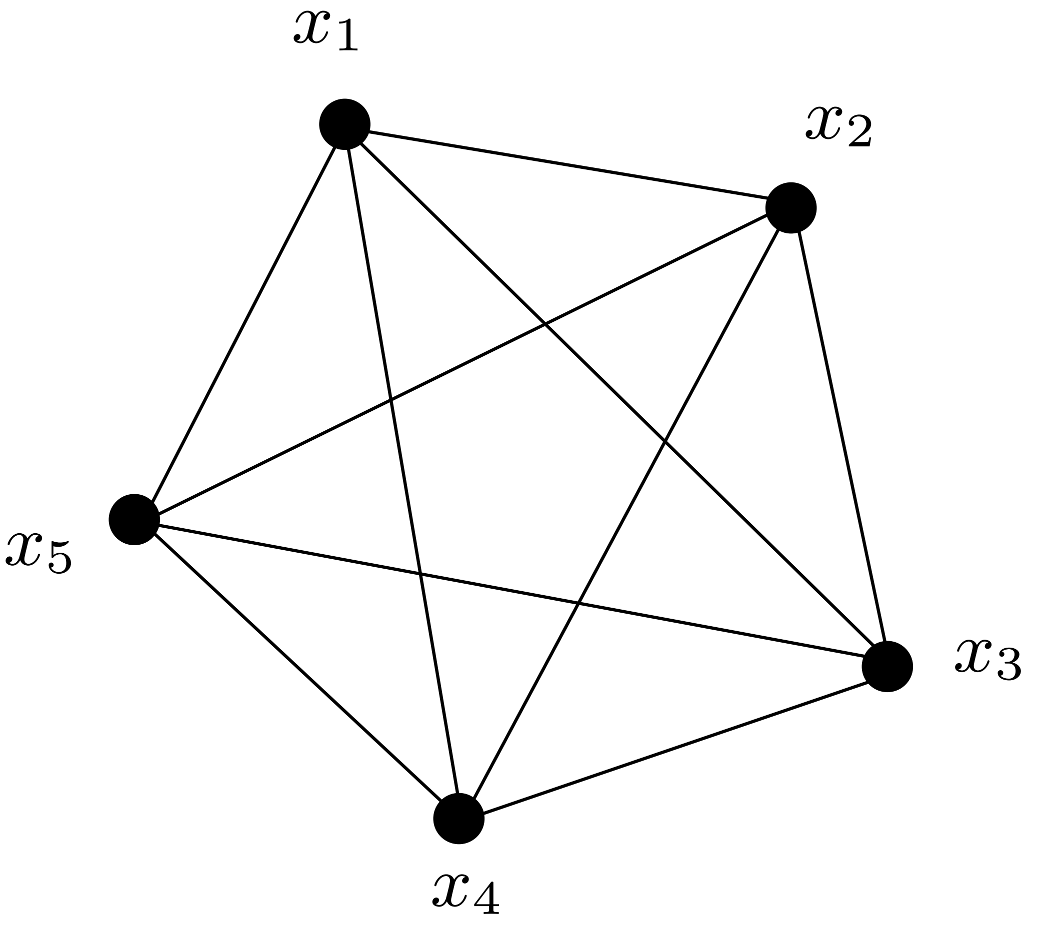
Artificial?

Well, some interesting problems can be reduced to these (later).

Warm up – k-bitwise-XOR

XOR	1	0	$\cdots$	0
S_1	$A_{1,1}$	$A_{1,2}$	$\cdots$	$A_{1,n}$
S_2	$A_{2,1}$	$A_{2,2}$	$\cdots$	$A_{2,n}$
$\vdots$				
$S_{\frac{k}{2}}$	$A_{\frac{k}{2},1}$	$A_{\frac{k}{2},2}$	$\cdots$	$A_{\frac{k}{2},n}$
$\vdots$				
S_k	$A_{k,1}$	$A_{k,2}$	$\cdots$	$A_{k,n}$

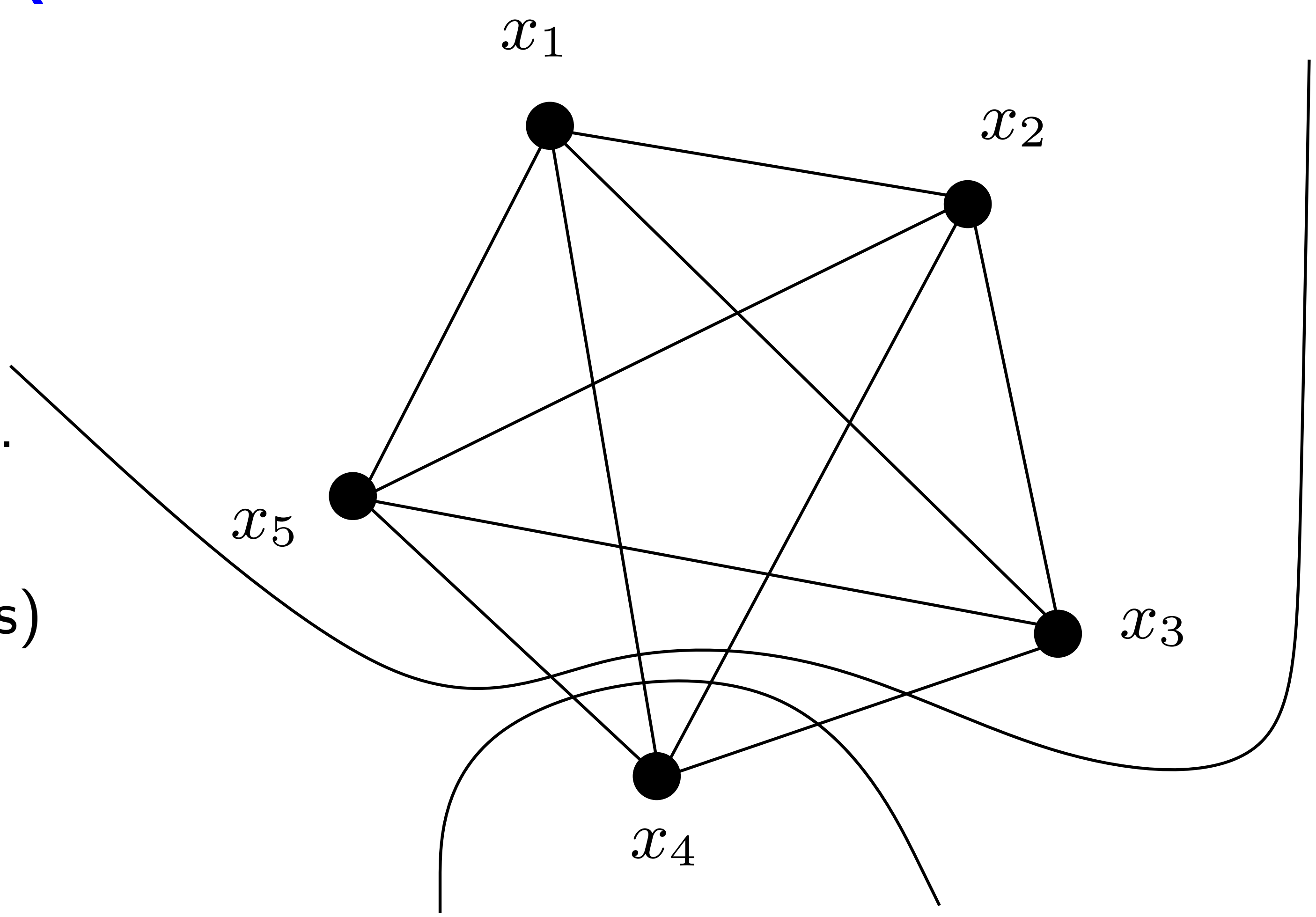
2-XOR $\Rightarrow$ k-XOR



2-XOR $\Rightarrow$ k-XOR

Pick a random guy, say x_4.

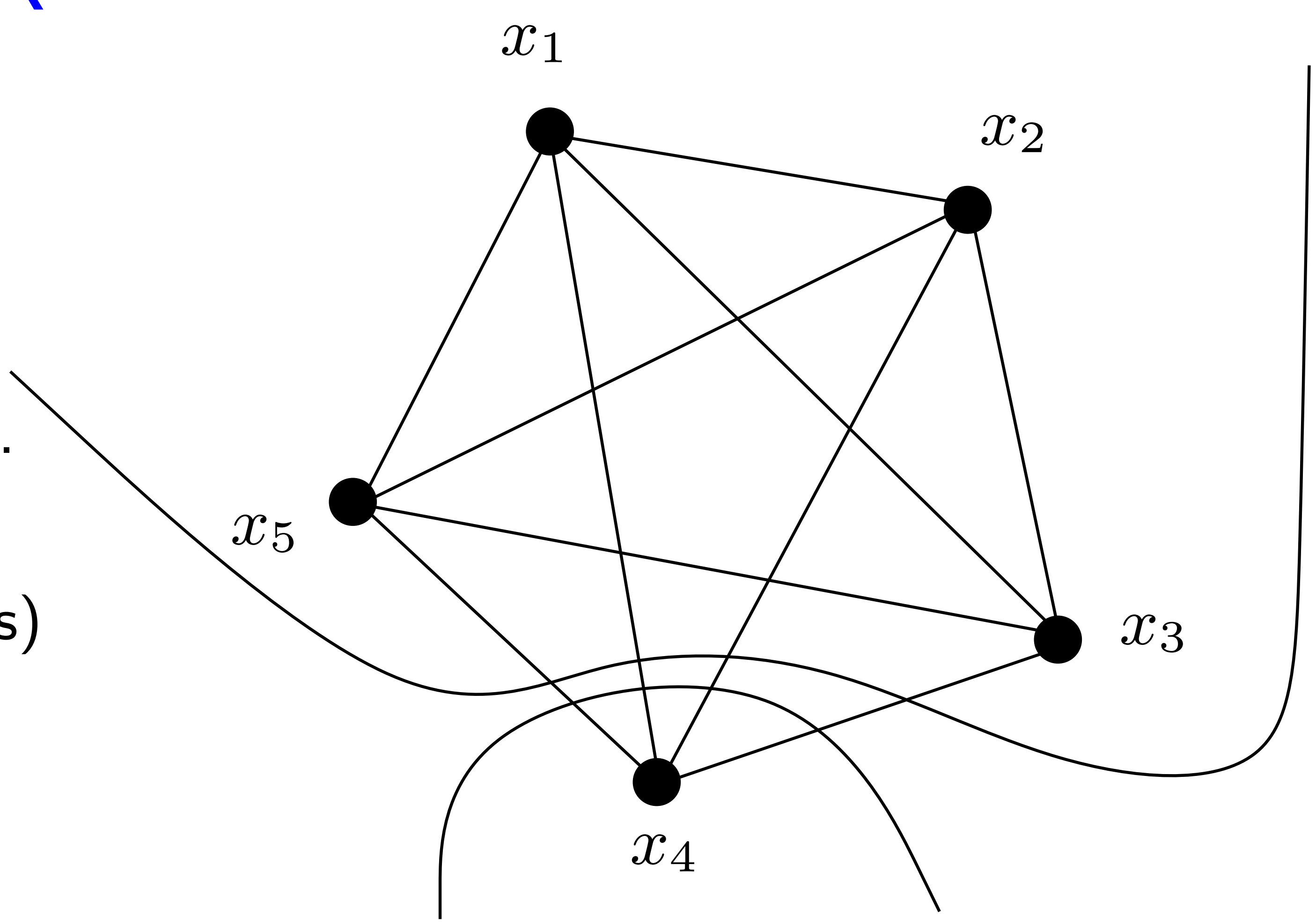
Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.



2-XOR $\Rightarrow$ k-XOR

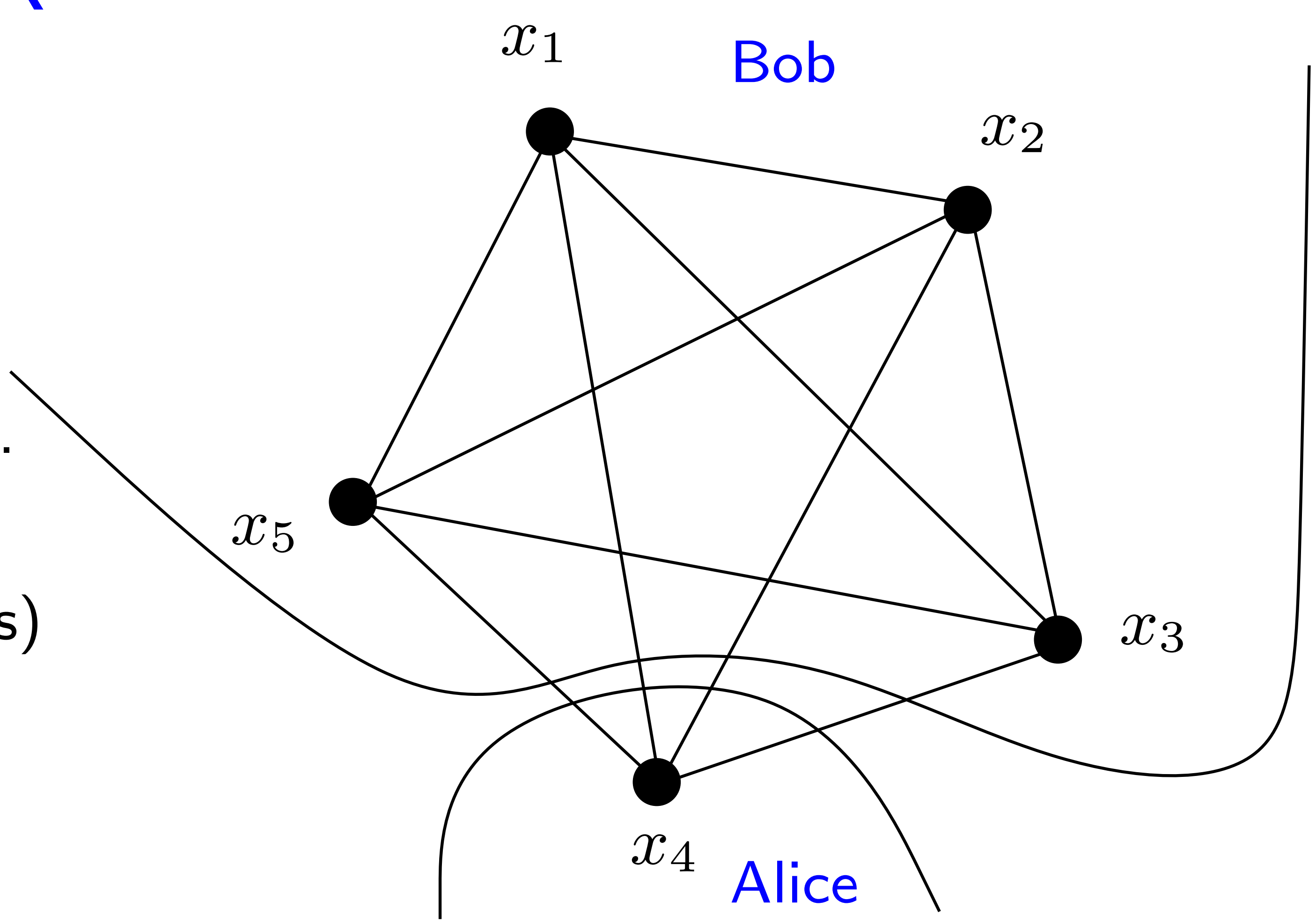
Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.



Alice and Bob want to solve the 2-XOR (the inputs are randomly from $\{0,1\}^n$)
$\Rightarrow$ running a protocol for k-XOR as follows:

2-XOR $\Rightarrow$ k-XOR



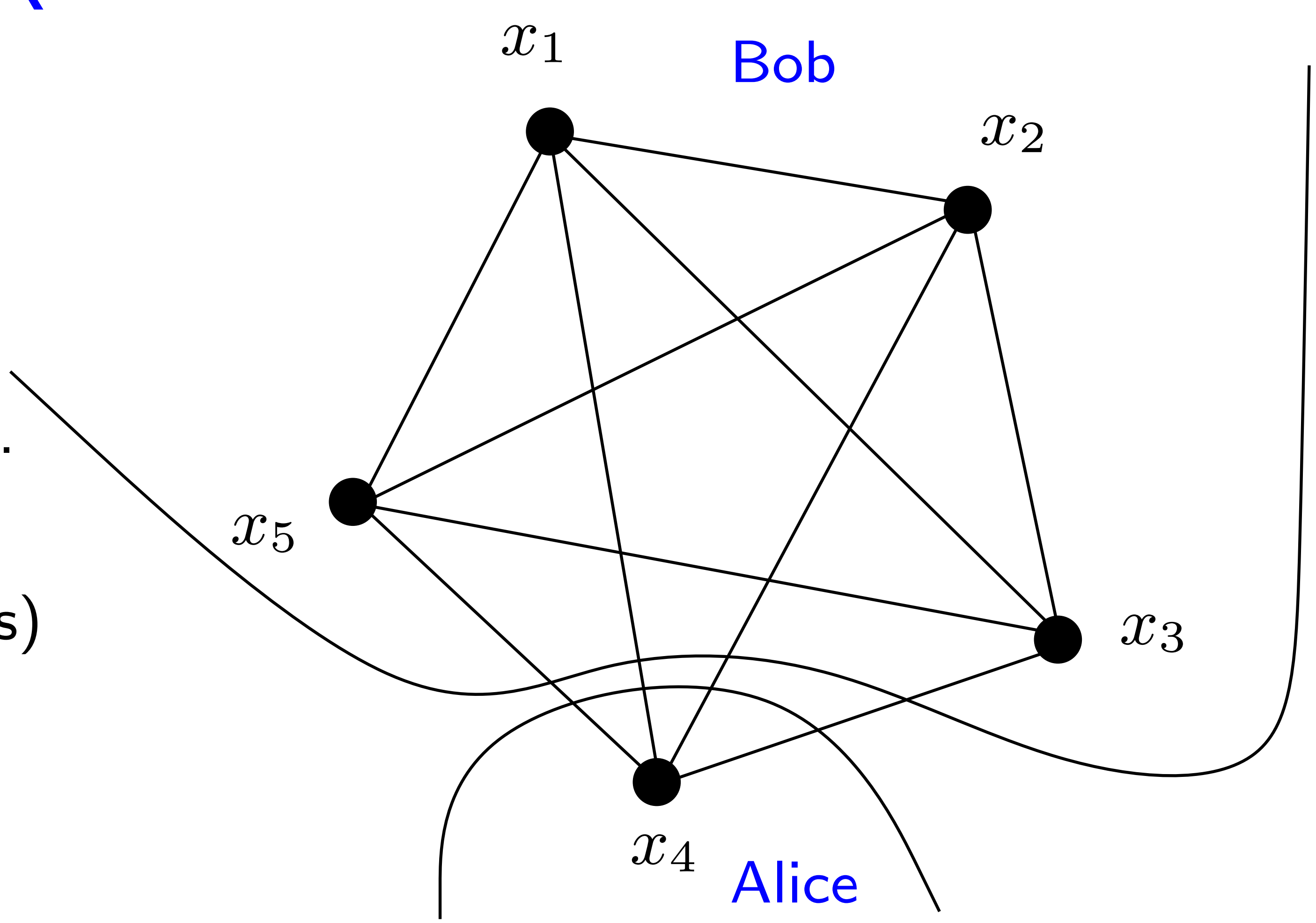
Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

Alice and Bob want to solve the 2-XOR (the inputs are randomly from $\{0,1\}^n$)
$\Rightarrow$ running a protocol for k-XOR as follows:

Alice plays a random guy with her input.
Bob plays another random guy with his input.
He also plays the other $k-2$ guys with random inputs from $\{0,1\}^n$.

2-XOR $\Rightarrow$ k-XOR



Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

Alice and Bob want to solve the 2-XOR (the inputs are randomly from $\{0,1\}^n$)
$\Rightarrow$ running a protocol for k-XOR as follows:

Alice plays a random guy with her input.
Bob plays another random guy with his input.
He also plays the other $k-2$ guys with random inputs from $\{0,1\}^n$.

Note: inputs of all k-players are symmetric.

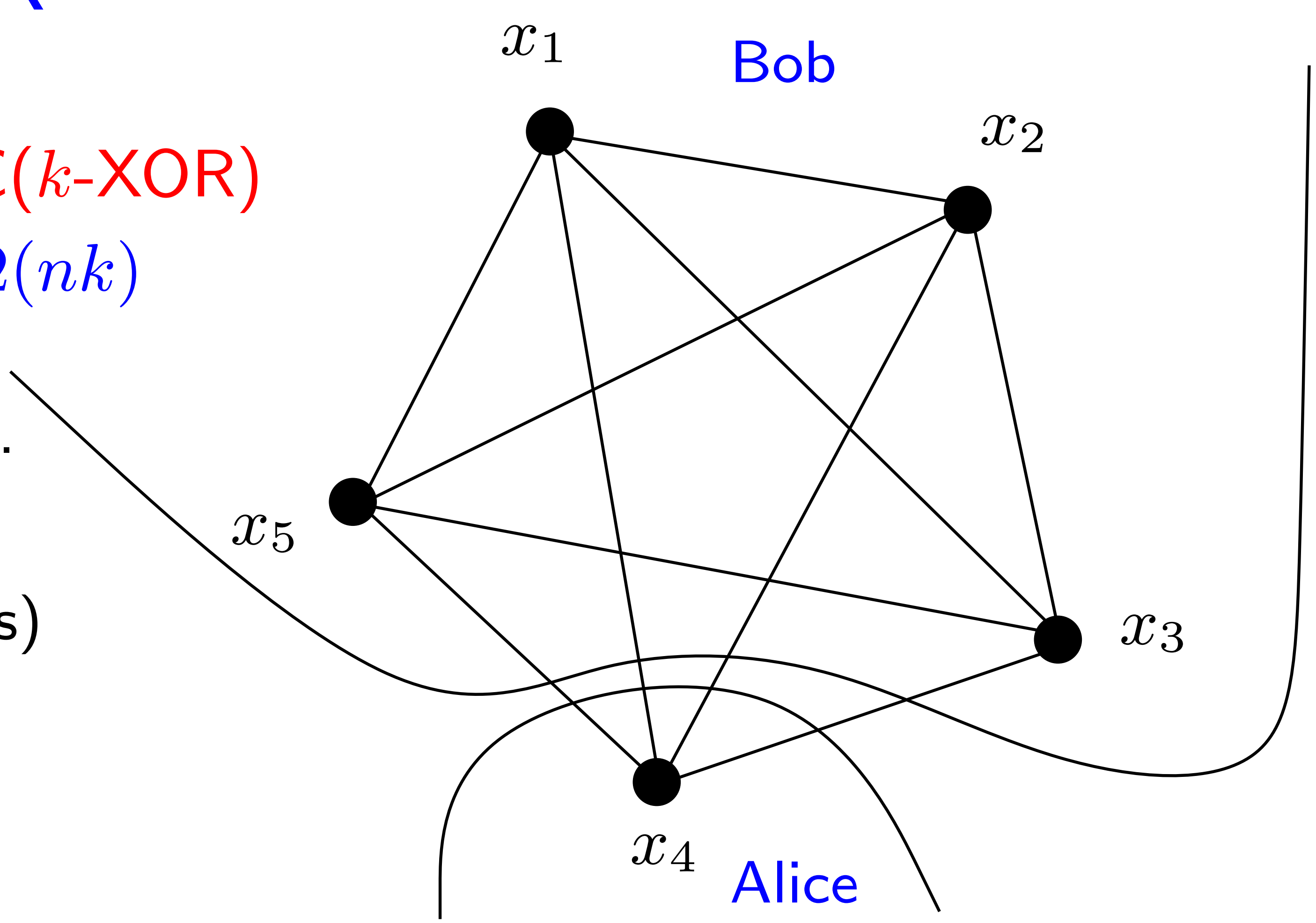
2-XOR $\Rightarrow$ k-XOR

$\mathbf{E}[\texttt{CC}(2\text{-XOR})] \leq \frac{2}{k}\ \texttt{CC}(k\text{-XOR})$

$\Omega(n)$ $\Omega(nk)$

Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

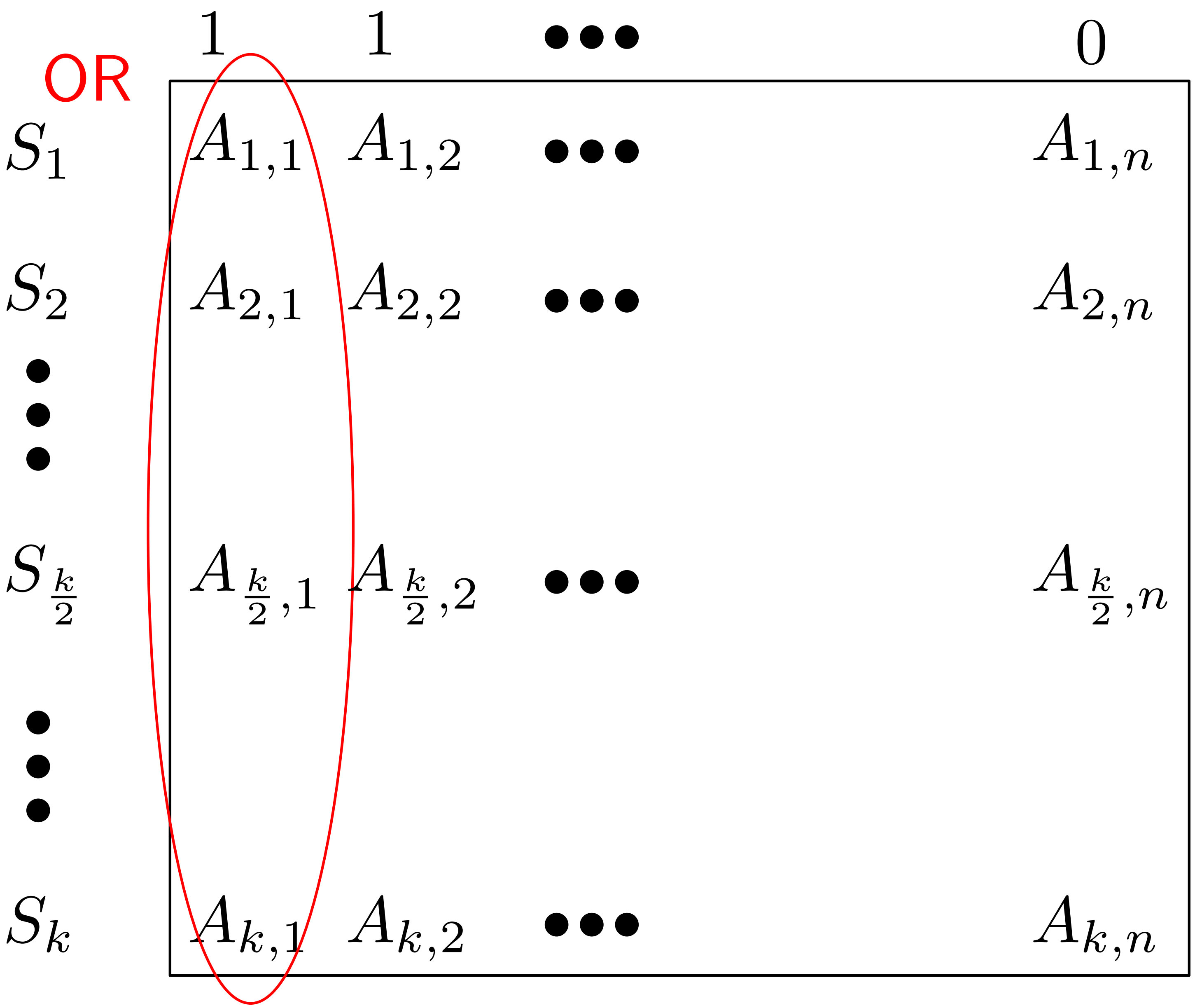


Alice and Bob want to solve the 2-XOR (the inputs are randomly from $\{0,1\}^n$)
$\Rightarrow$ running a protocol for k-XOR as follows:

Alice plays a random guy with her input.
Bob plays another random guy with his input.
He also plays the other $k-2$ guys with random inputs from $\{0,1\}^n$.

Note: inputs of all k-players are symmetric.

k-bitwise-OR



OR
1
1
0
S_1
S_2
$S_{\frac{k}{2}}$
S_k
$A_{1,1}$ $A_{1,2}$ $A_{1,n}$
$A_{2,1}$ $A_{2,2}$ $A_{2,n}$
$A_{\frac{k}{2},1}$ $A_{\frac{k}{2},2}$ $A_{\frac{k}{2},n}$
$A_{k,1}$ $A_{k,2}$ $A_{k,n}$

Intuition to reduce from 2-DISJ

As always, first, try to find the hard distance for k-OR!

- First attempt: each coordinate is 1 w.p. $1/k$.

Intuition to reduce from 2-DISJ

As always, first, try to find the hard distance for k-OR!

- First attempt: each coordinate is 1 w.p. $1/k$.

  Hard for $k = 2$ but not for general k.

Intuition to reduce from 2-DISJ

As always, first, try to find the hard distance for k-OR!

- First attempt: each coordinate is 1 w.p. $1/k$.

  Hard for $k = 2$ but not for general k.

- Second attempt: random partition n coordinates to two equal-sized sets.
  - *important set*: each entry is 1 w.p. $1/k$.
  - *balancing set*: all entries are 1.

Intuition to reduce from 2-DISJ

As always, first, try to find the hard distance for k-OR!

- First attempt: each coordinate is 1 w.p. $1/k$.

  Hard for $k = 2$ but not for general k.

- Second attempt: random partition n coordinates to two equal-sized sets.
  - *important set*: each entry is 1 w.p. $1/k$.
  - *balancing set*: all entries are 1.

  Seems hard but, wait! The *Slepian-Wolf* coding.

Intuition to reduce from 2-DISJ

As always, first, try to find the hard distance for k-OR!

- First attempt: each coordinate is 1 w.p. $1/k$.

  Hard for $k = 2$ but not for general k.

- Second attempt: random partition n coordinates to two equal-sized sets.
  - *important set*: each entry is 1 w.p. $1/k$.
  - *balancing set*: all entries are 1.

  Seems hard but, wait! The *Slepian-Wolf* coding.

- Third attempt: same as the second. Except the balancing set: each entry is 1 w.p. 1/2.

Intuition to reduce from 2-DISJ

As always, first, try to find the hard distance for k-OR!

- First attempt: each coordinate is 1 w.p. $1/k$.

  Hard for $k = 2$ but not for general k.

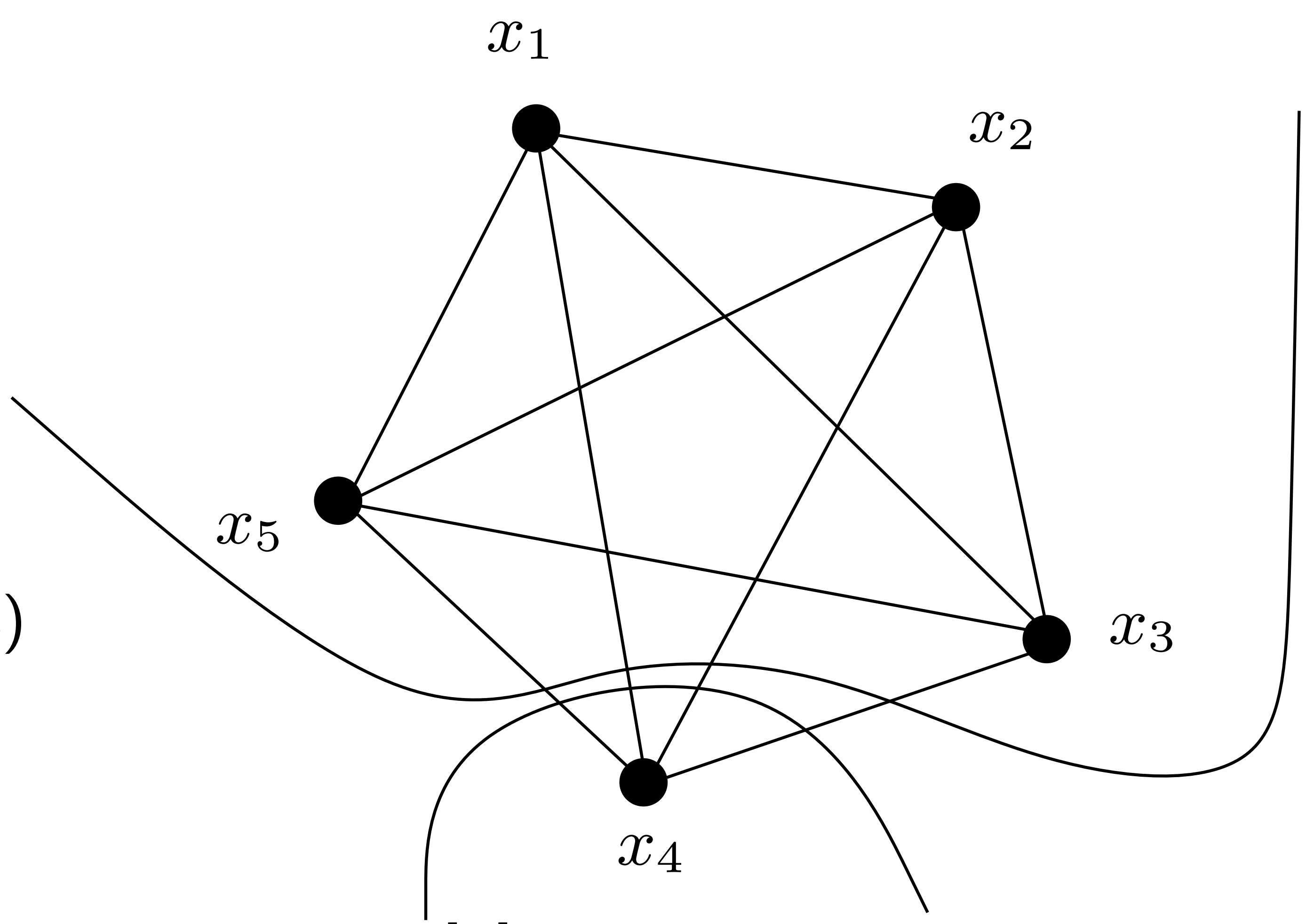
- Second attempt: random partition n coordinates to two equal-sized sets.
  - *important set*: each entry is 1 w.p. $1/k$.
  - *balancing set*: all entries are 1.

  Seems hard but, wait! The *Slepian-Wolf* coding.

- Third attempt: same as the second. Except the balancing set: each entry is 1 w.p. $1/2$. It works!

  Now Alice takes one vector. Bob takes the other $k-1$ vectors and OR them together, and then takes the complement.
  Looks like 2-DISJ.

2-DISJ $\Rightarrow$ k-OR



Pick a random guy, say x_4.

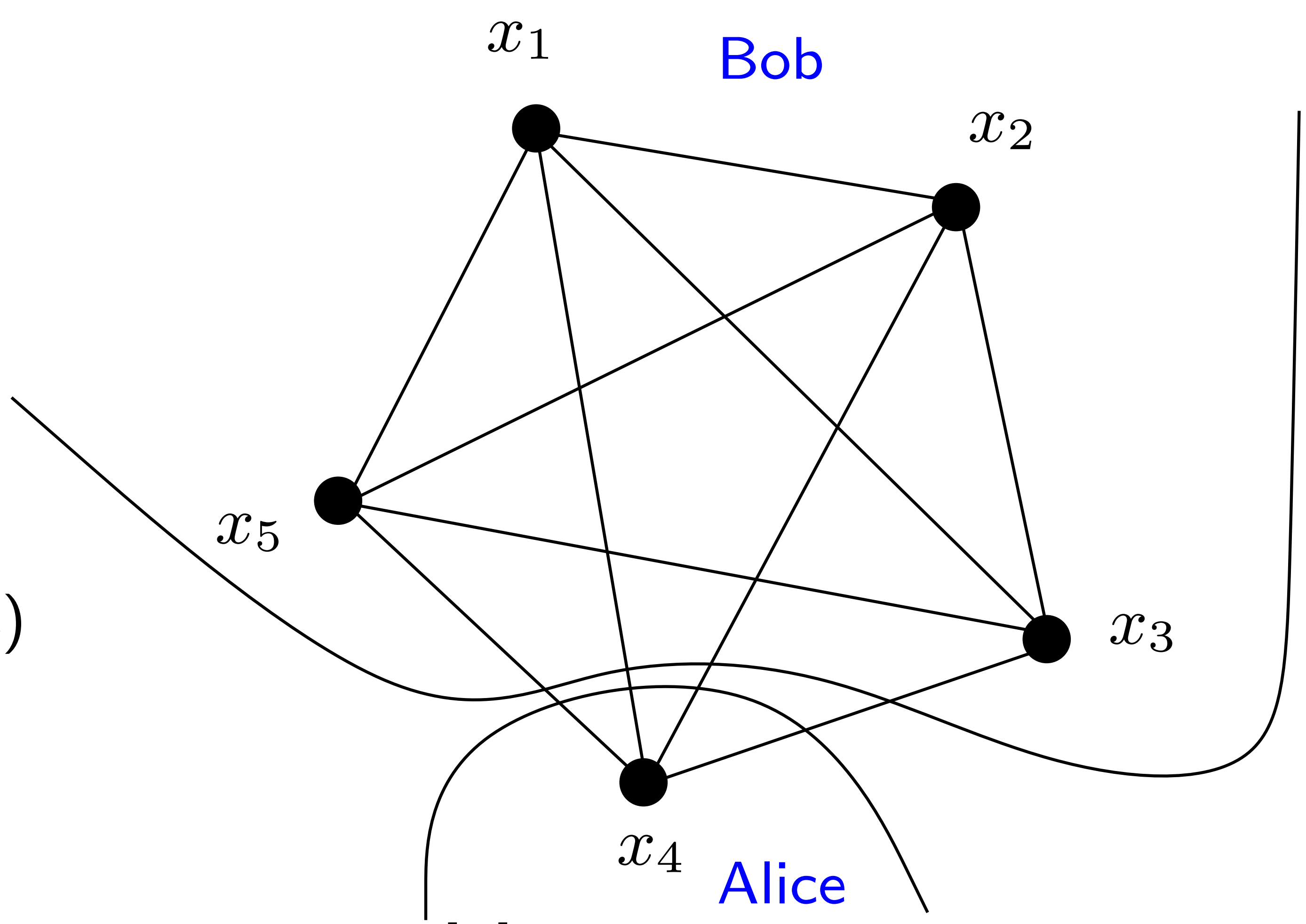
Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

2-DISJ: Alice has $x \in [n]$ and Bob has $y \in [n]$.

W.p. $1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $|x \cap y| = 1$.

And w.p. $1 - 1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $x \cap y = \emptyset$.

2-DISJ $\Rightarrow$ k-OR



Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

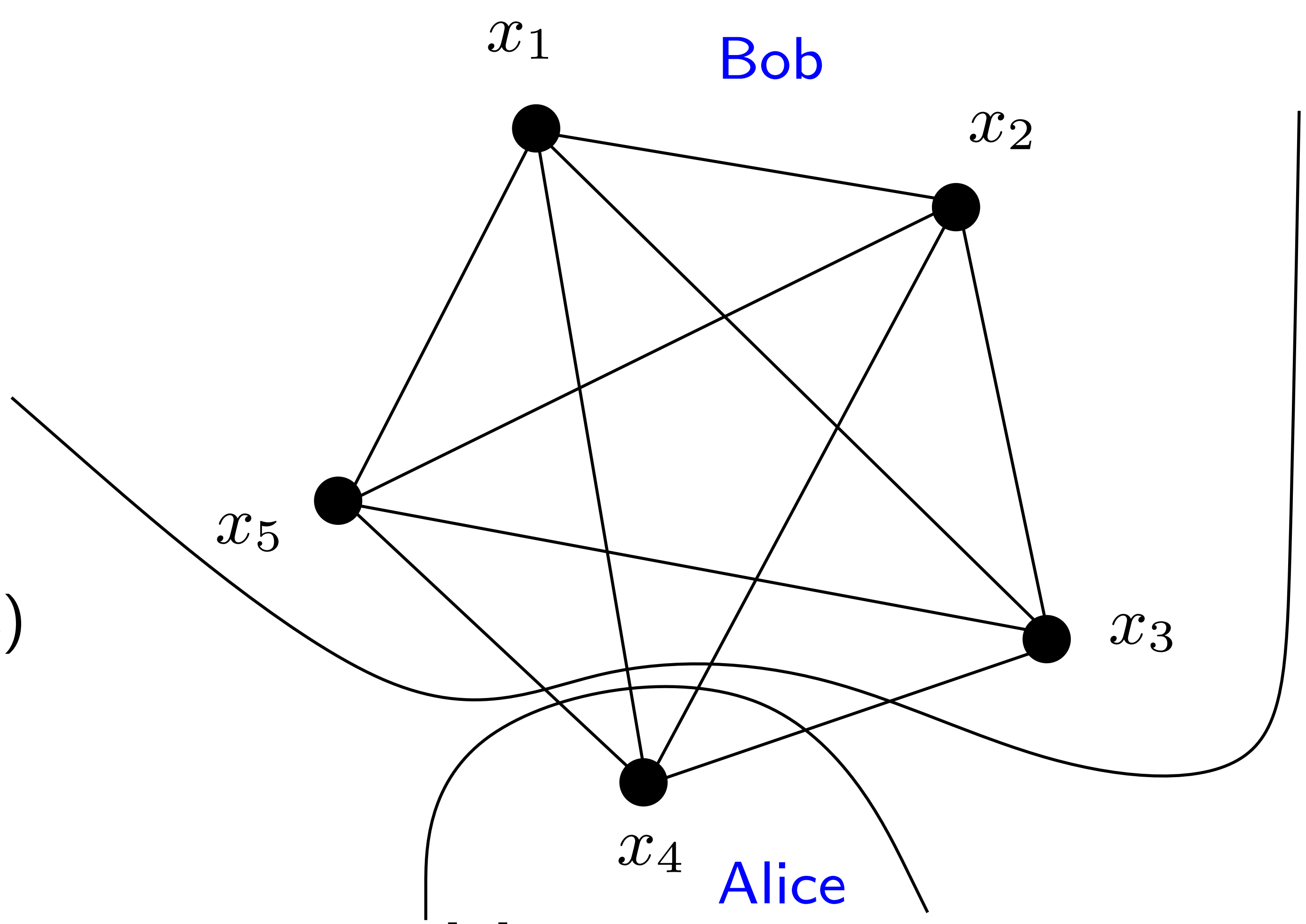
2-DISJ: Alice has $x \in [n]$ and Bob has $y \in [n]$.

W.p. $1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $|x \cap y| = 1$.

And w.p. $1 - 1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $x \cap y = \emptyset$.

Alice plays a random guy with her input x.
Bob plays the other $k-1$ guys with his input y.

2-DISJ $\Rightarrow$ k-OR



Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

2-DISJ: Alice has $x \in [n]$ and Bob has $y \in [n]$.

W.p. $1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $|x \cap y| = 1$.

And w.p. $1 - 1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $x \cap y = \emptyset$.

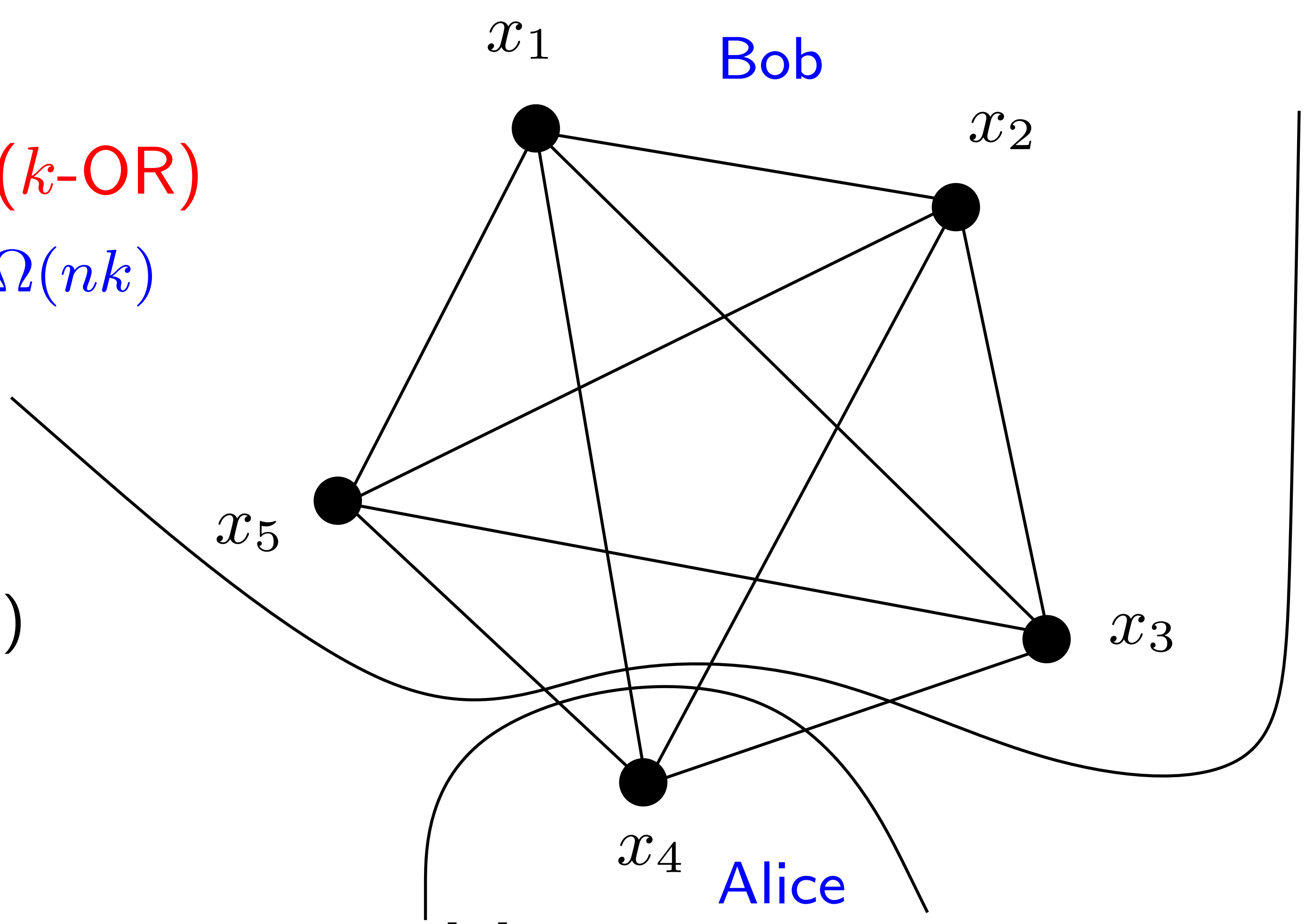
Alice plays a random guy with her input x.
Bob plays the other $k - 1$ guys with his input y.

Again: inputs of all k-players are symmetric.

2-DISJ $\Rightarrow$ k-OR

$\mathbf{E}$[CC(2-DISJ)] $\leq \frac{2}{k}$ CC(k-OR)

Razborov[90]: $\Omega(n)$. $\Omega(nk)$



Pick a random guy, say x_4.

Total CC is C $\Rightarrow$
the expected CC(x_4 : others)
is at most $2C/k$.

2-DISJ: Alice has $x \in [n]$ and Bob has $y \in [n]$.

W.p. $1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $|x \cap y| = 1$.

And w.p. $1 - 1/4$, x and y are random subsets of $[n]$ of size $n/4$ and $x \cap y = \emptyset$.

Alice plays a random guy with her input x.
Bob plays the other $k-1$ guys with his input y.

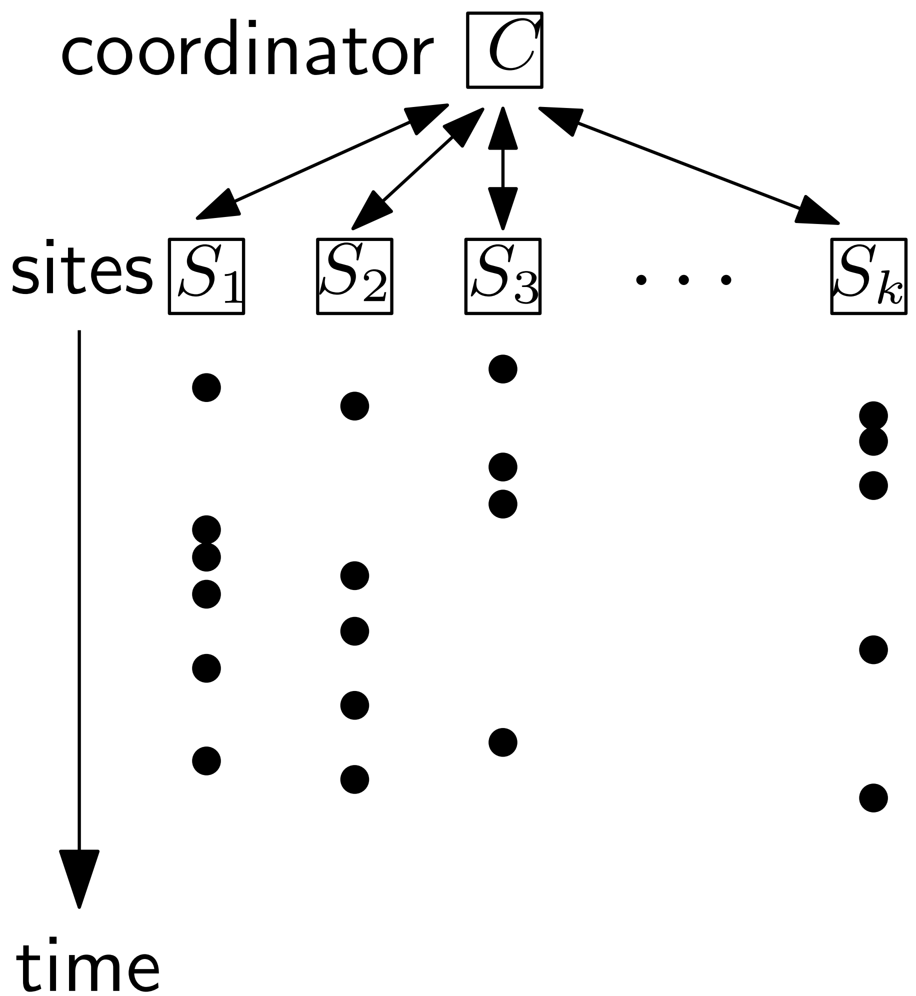
Again: inputs of all k-players are symmetric.

Summary of other reults

1. $\Omega(nk)$ for the MAJ.
2. $\Omega(n \log k)$ for AND and OR in the blackboard model.
3. $\tilde{\Omega}(nk)$ for k-connectivity.

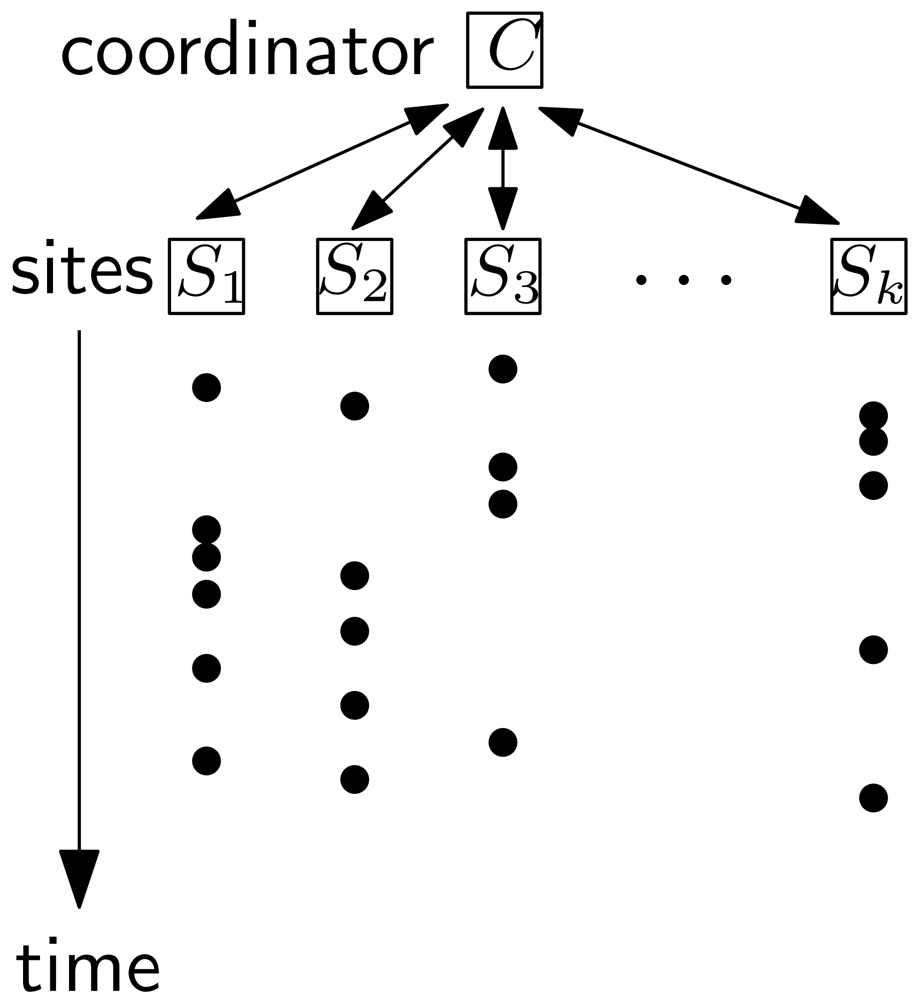
   (one of main technical contributions)
4. Some direct sum results.
5. Some applications, e.g. the heavy hitter problem and the ϵ-kernels in the site-server model (next page).

Motivation



The Distributed Streaming Model

Motivation



The Distributed Streaming Model

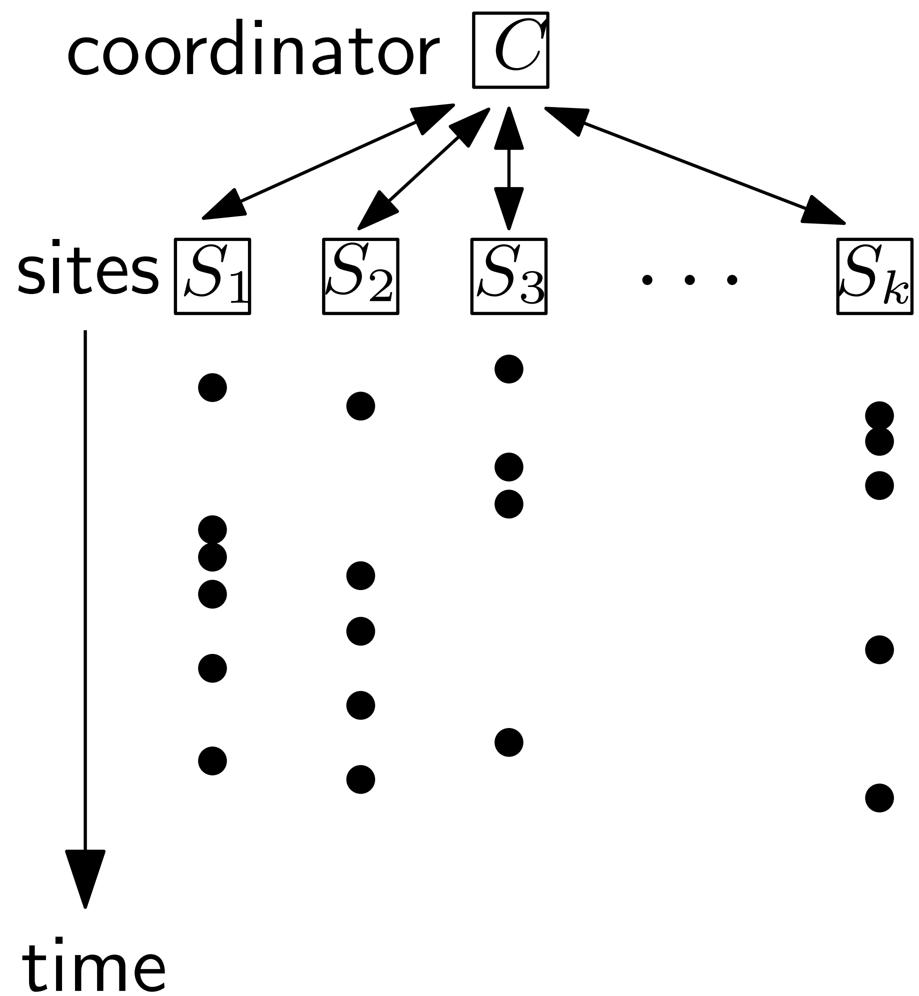
Static case (or the site-server model, exactly our model)

- Top-k (Can and Wang '04, Michel et. al. '05, Patt-Shamir and Shafrir '08)
- Heavy-hitter (Zhao et. al. '06, Huang et. al. '11)

Dynamic case

- Samplings (Cormode et. al. '10)
- Frequent moments (. . .)
- Heavy-hitter (. . .)
- Quantile (. . .)
- Entropy (. . .)
- Various sketches (. . .)
- Non-linear functions (. . .)

Motivation



The Distributed Streaming Model

A large number of upper bounds, but very few lower bounds.

Static case (or the site-server model, exactly our model)

- Top-k (Can and Wang '04, Michel et. al. '05, Patt-Shamir and Shafrir '08)
- Heavy-hitter (Zhao et. al. '06, Huang et. al. '11)

Dynamic case

- Samplings (Cormode et. al. '10)
- Frequent moments (. . .)
- Heavy-hitter (. . .)
- Quantile (. . .)
- Entropy (. . .)
- Various sketches (. . .)
- Non-linear functions (. . .)

Motivation (Cont.)

- Secure Multiparty Computation:
  Players who do not trust each other, but want to compute a joint function of their inputs.

Motivation (Cont.)

- Secure Multiparty Computation:
  Players who do not trust each other, but want to compute a joint function of their inputs.

- Streaming:
  A stream of data that can only be scanned from left to right. The goal is to compute some function of the stream, and minimize the space usage.

Motivation (Cont.)

- Secure Multiparty Computation:
  Players who do not trust each other, but want to compute a joint function of their inputs.

- Streaming:
  A stream of data that can only be scanned from left to right. The goal is to compute some function of the stream, and minimize the space usage.

  - Some nice lower bounds given, e.g., by Bar-Yossef el. al. '04 for frequent moments, but in blackboard model or the "one way" private message model.

Discussions

- There are problems that might be impossible to lower bound using symmetrization.
  - E.g. k-DISJ ...

Discussions

- There are problems that might be impossible to lower bound using symmetrization.
  - E.g. k-DISJ ...
- Require proving distributional lower bounds for 2-player problems, often over somewhat-convoluted distributions.
  - Can we avoid this?

Discussions

- There are problems that might be impossible to lower bound using symmetrization.
  - E.g. k-DISJ ...
- Require proving distributional lower bounds for 2-player problems, often over somewhat-convoluted distributions.
  - Can we avoid this?
- In order to use symmetrization, one needs to find a hard distribution for the k-player problem which is symmetric.
  - Can we relax or generalize this?

List of other problems

- Coordinate-wise problems: Each player gets a vector of length n. Some symmetric coordinate-wise function $g : \{0,1\}^k \to \{0,1\}$ is applied, resulting in a length n vector. Then a "combining function" $h : \{0,1\}^n \to Z$ is applied to the bits of the result.

- Equality: Each player gets a vector of length n, and the goal is to decide whether all players have received the same vector.

- Graph problems

- Pointer Chasing

- ...

The End

THANK YOU

Q and A