Lower Bounds for Distributed Sketching

Sepehr Assadi

October 2022

RUTGERS
UNIVERSITY | NEW BRUNSWICK

ADGA 2022
11th Workshop on Advances in Distributed Graph Algorithms

This Talk

- Distributed Sketching Model
- Is it an interesting model theoretically?
- Some lower bounds
- Open problems

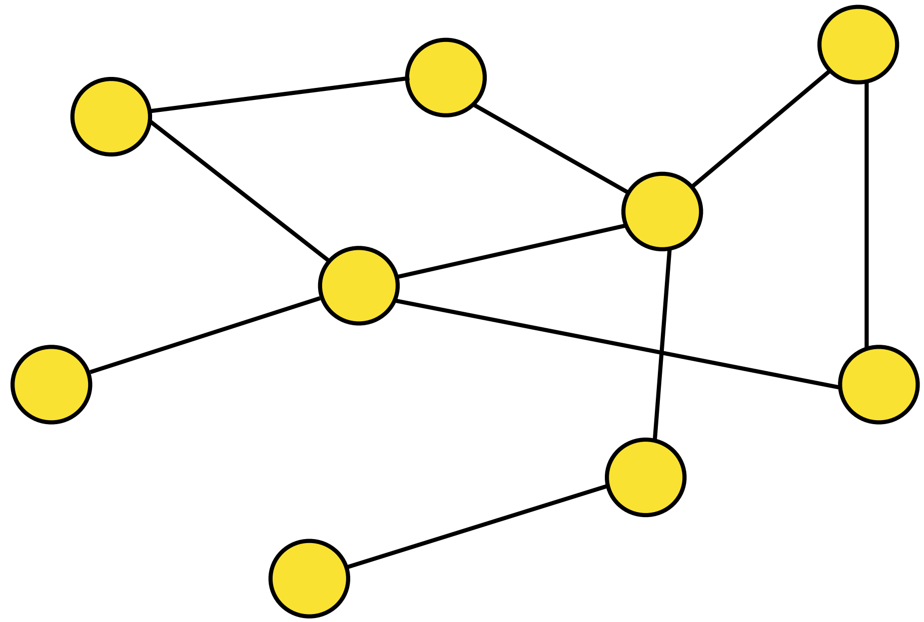
This Talk

- Distributed Sketching Model
- Is it an interesting model <u>theoretically</u>?
- Some lower bounds
- Open problems

- Based on joint works with
  - Gillat Kol and Rotem Oshman (PODC 2020)
  - Gillat Kol and Zhijun Zhang (FOCS 2022)

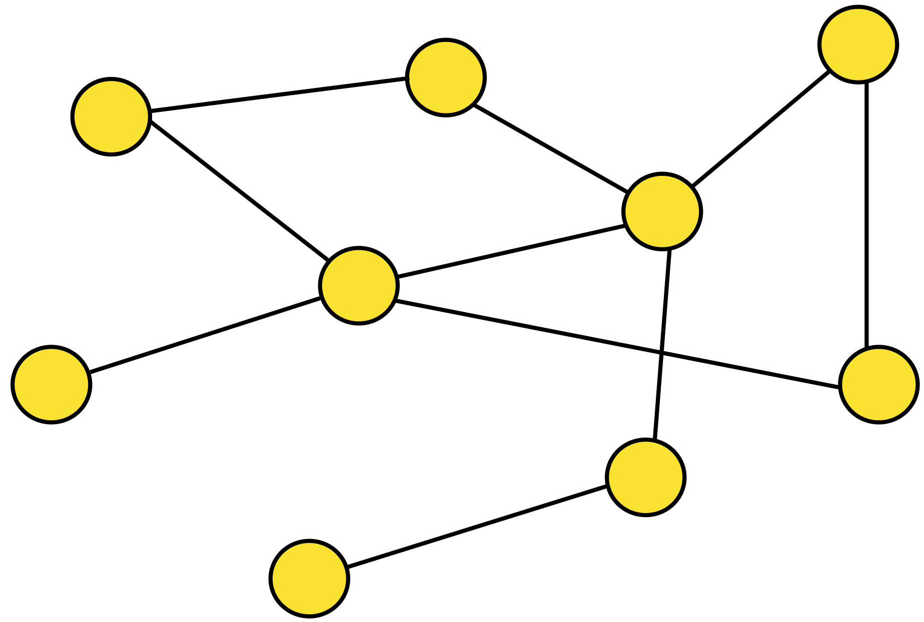
Distributed Sketching Model

- A Graph $G = (V, E)$



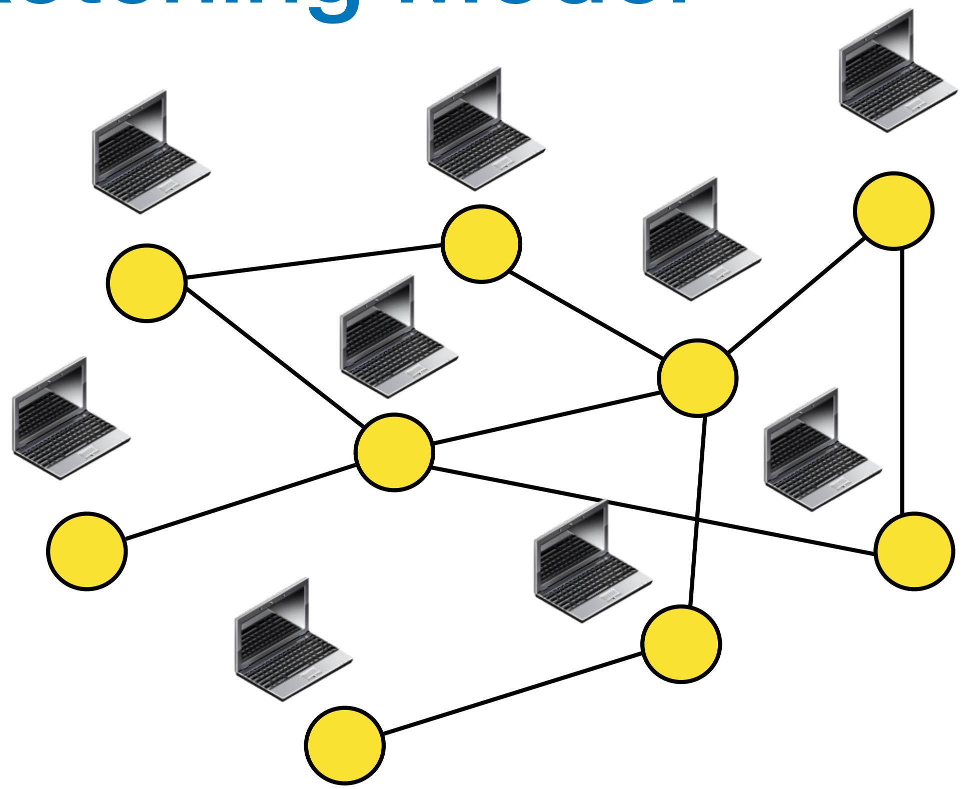
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex



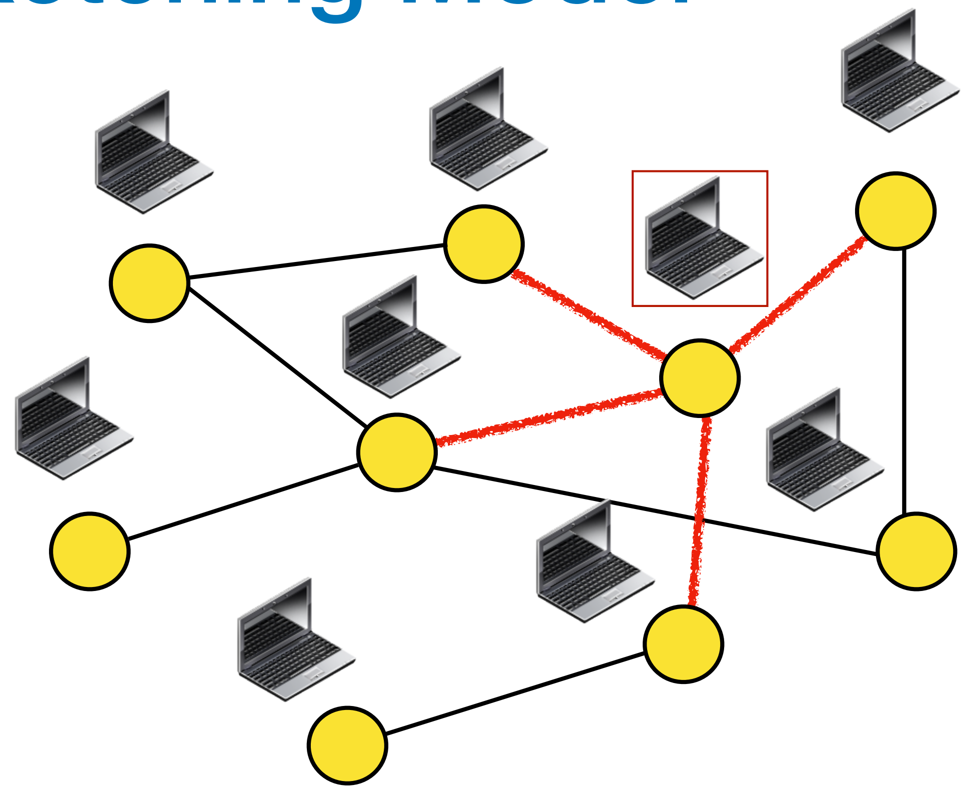
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex



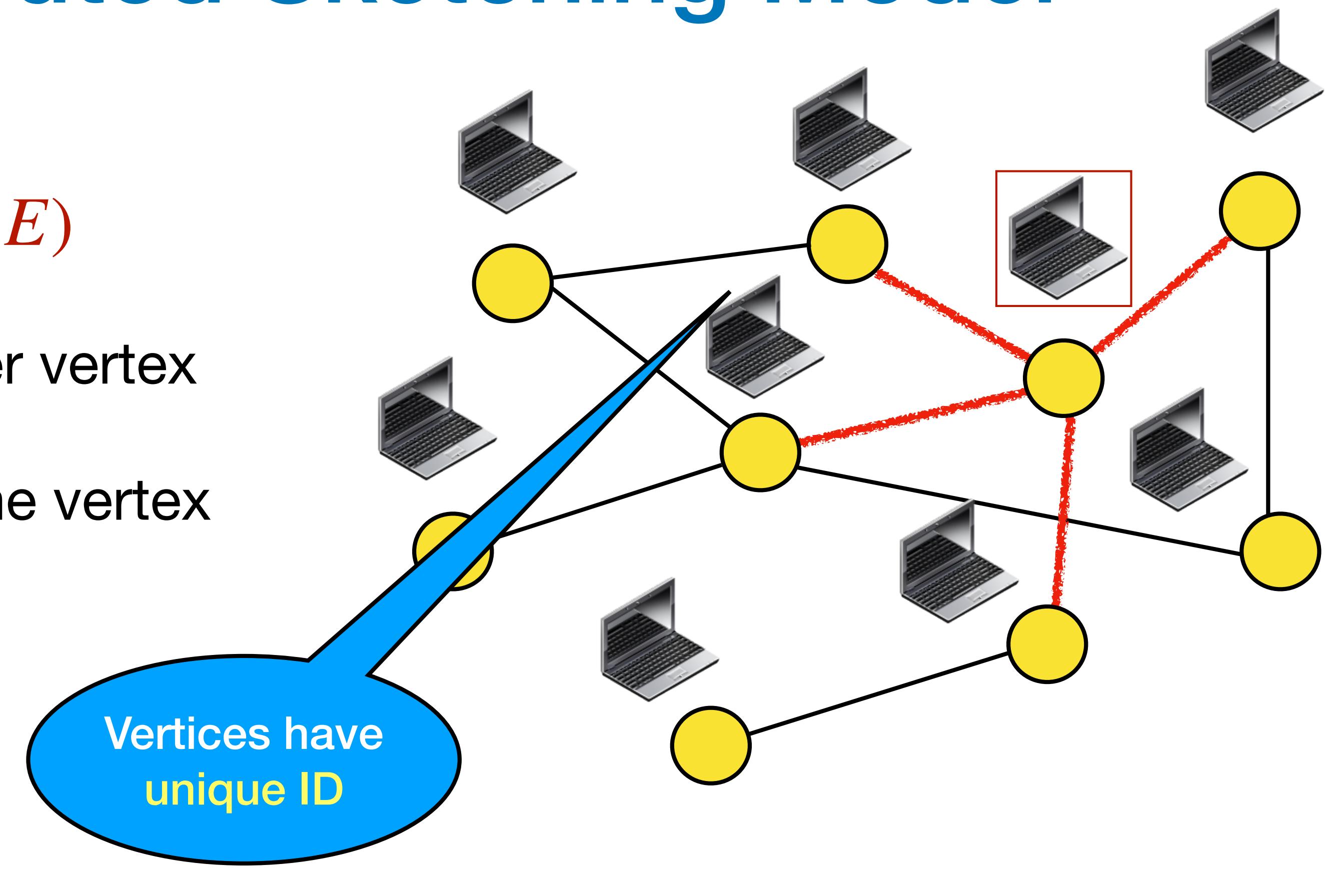
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex



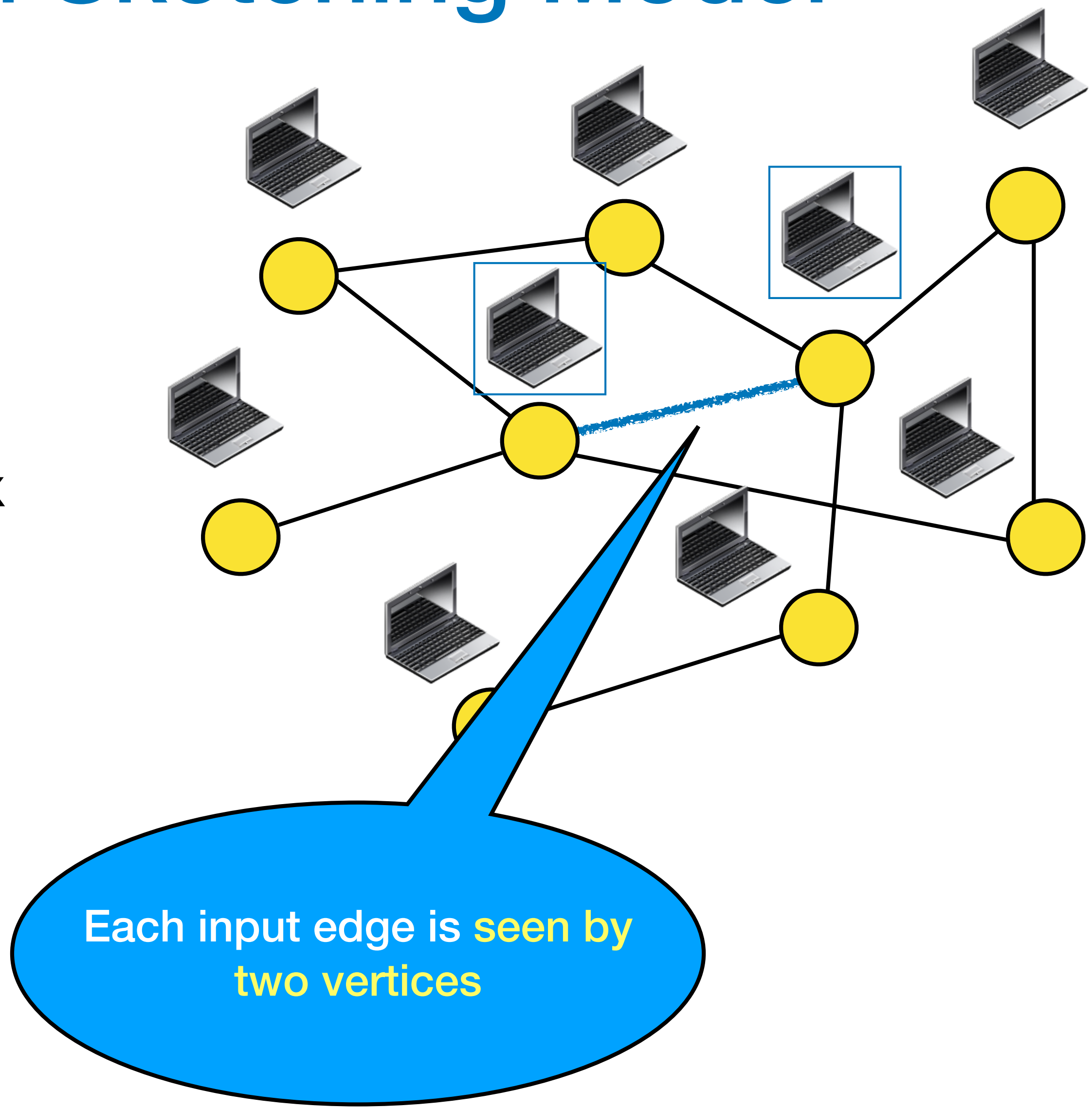
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex



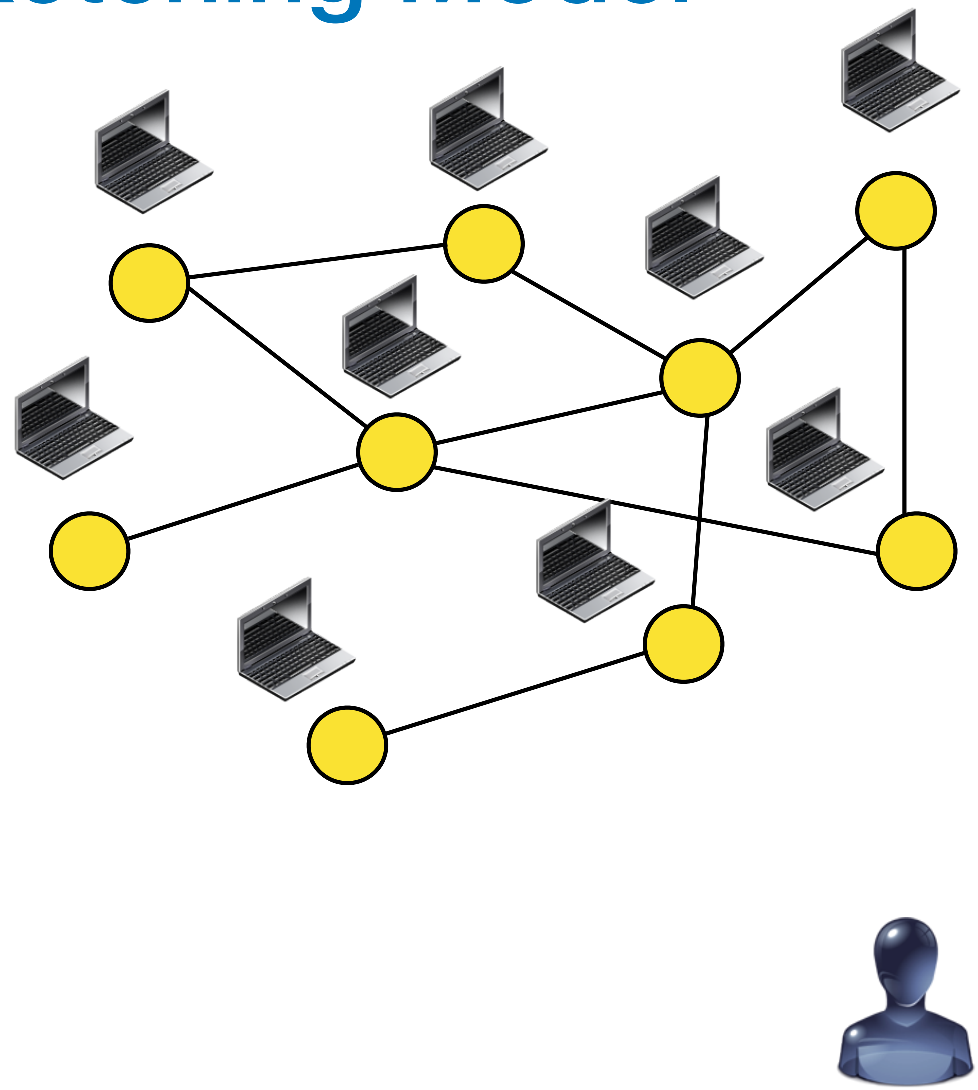
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex



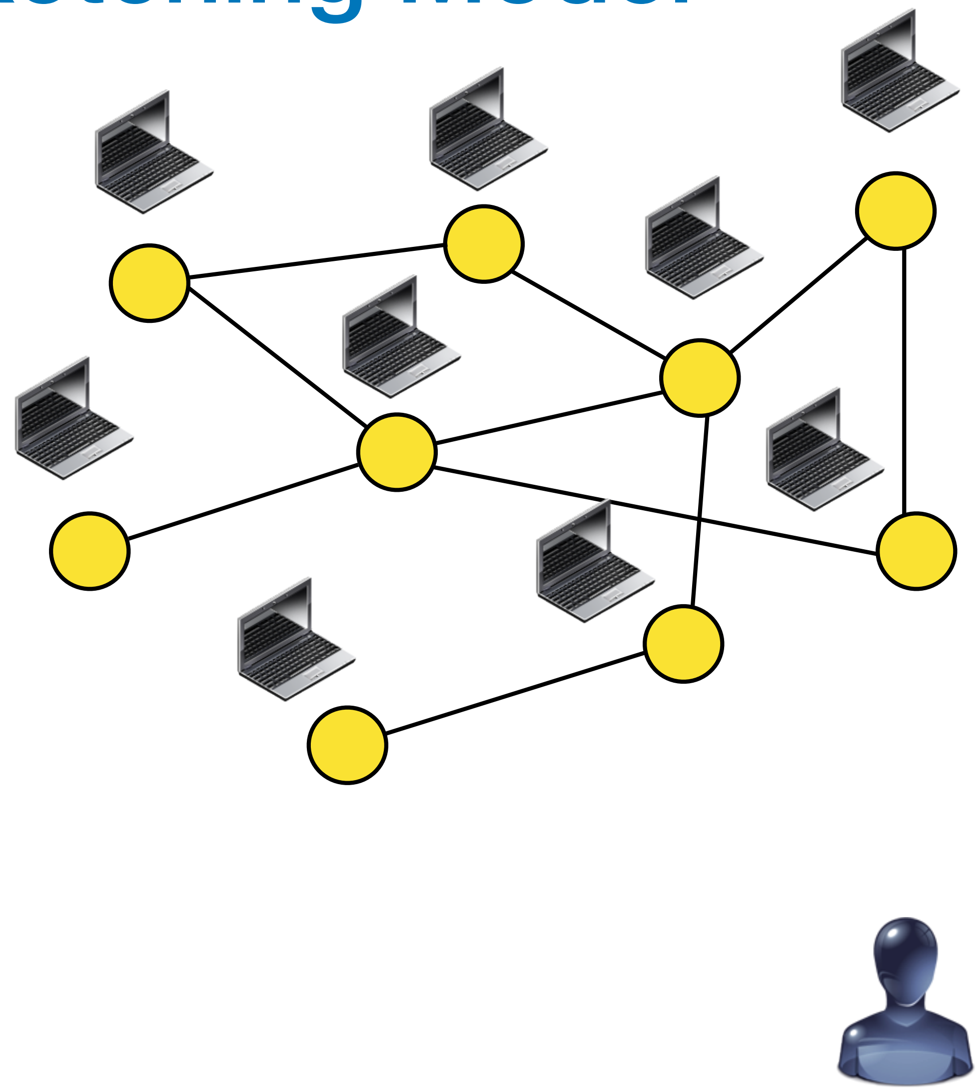
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex
- A referee with no input



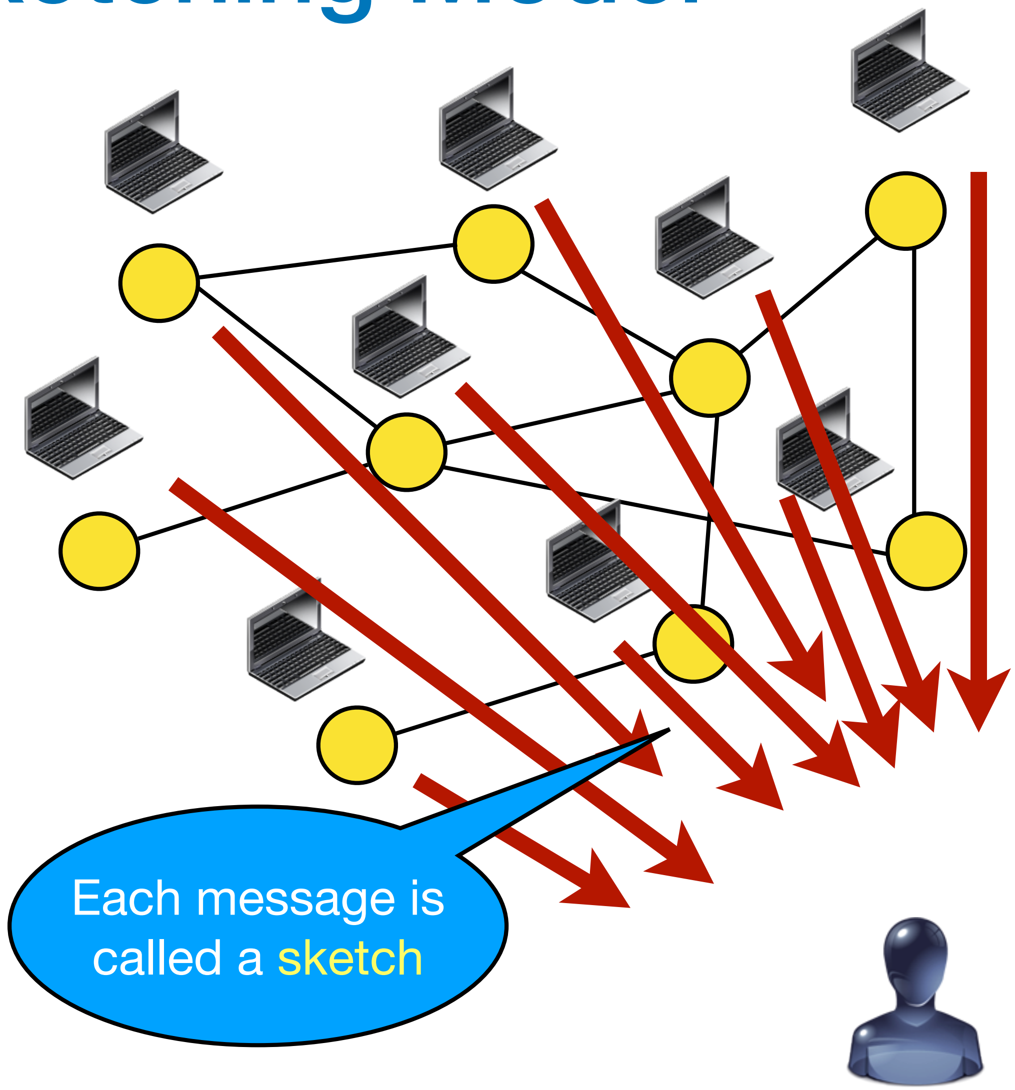
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex
- A referee with no input
- Processors simultaneously send a message to the referee



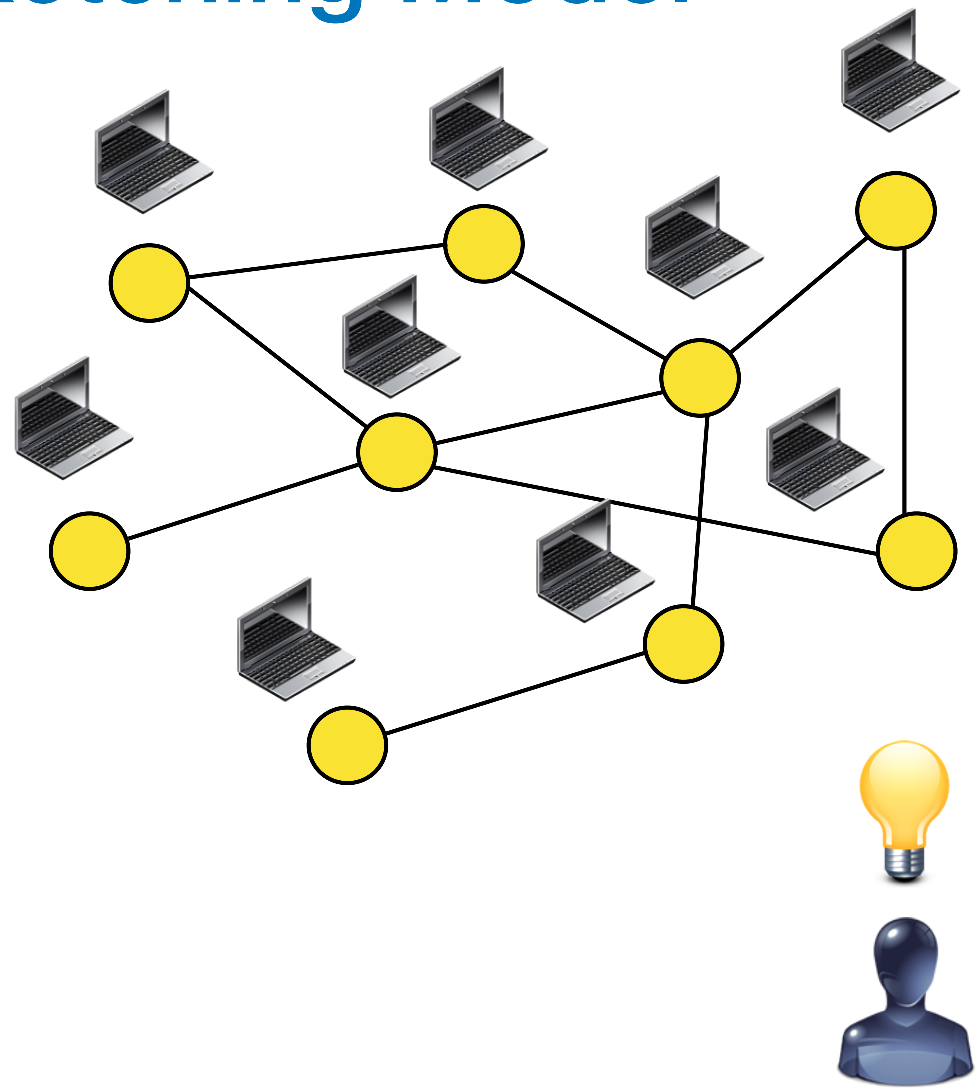
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex
- A referee with no input
- Processors simultaneously send a message to the referee



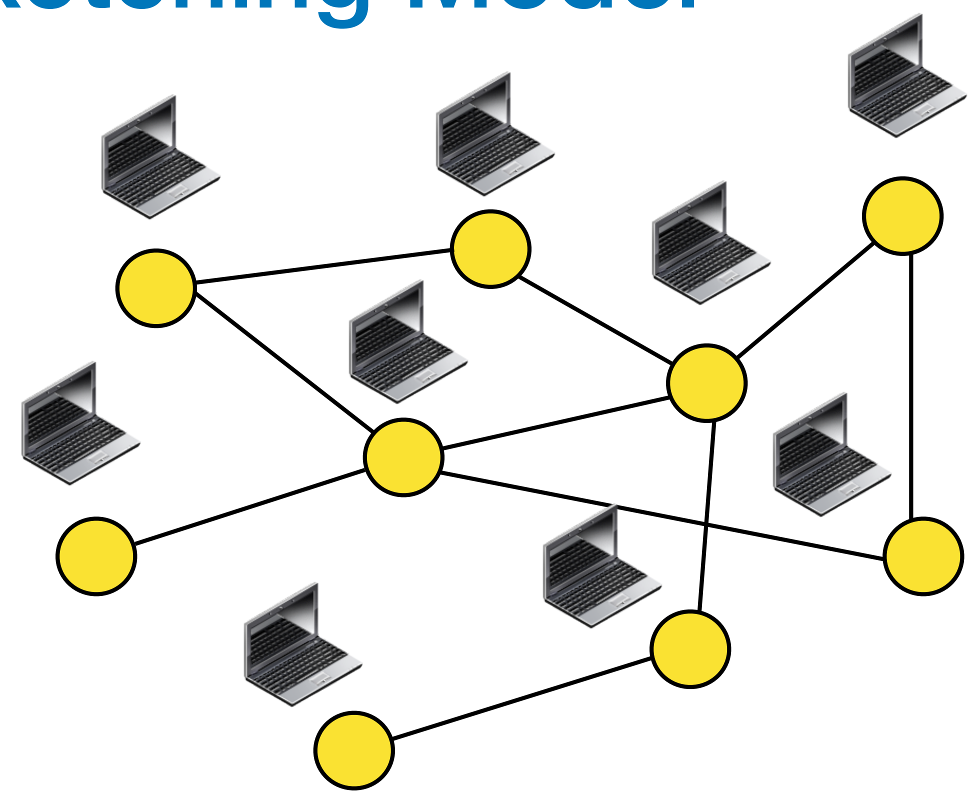
Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex
- A referee with no input
- Processors simultaneously send a message to the referee
- Referee outputs the solution



Distributed Sketching Model

- A Graph $G = (V, E)$
- One processor per vertex
- **Input:** edges of the vertex
- A referee with no input
- Processors simultaneously send a message to the referee
- Referee outputs the solution
- Access to shared randomness

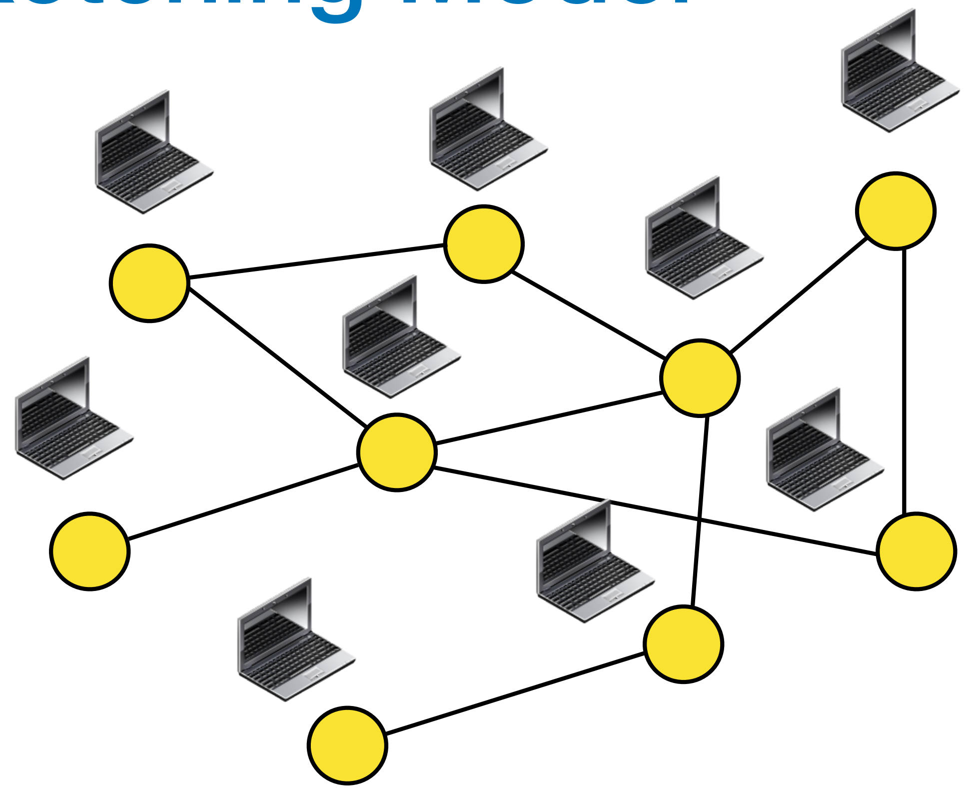


Distributed Sketching Model

- A

This model is also called
Broadcast Congested Clique
(one-round algorithms)

eree



- Referee outputs the solution

- Access to shared randomness

Distributed Sketching Model

- A C…

This model is also called **Broadcast Congested Clique** (one-round algorithms)

It is also closely related to **Dynamic Graph Streaming** (single-pass algorithms)

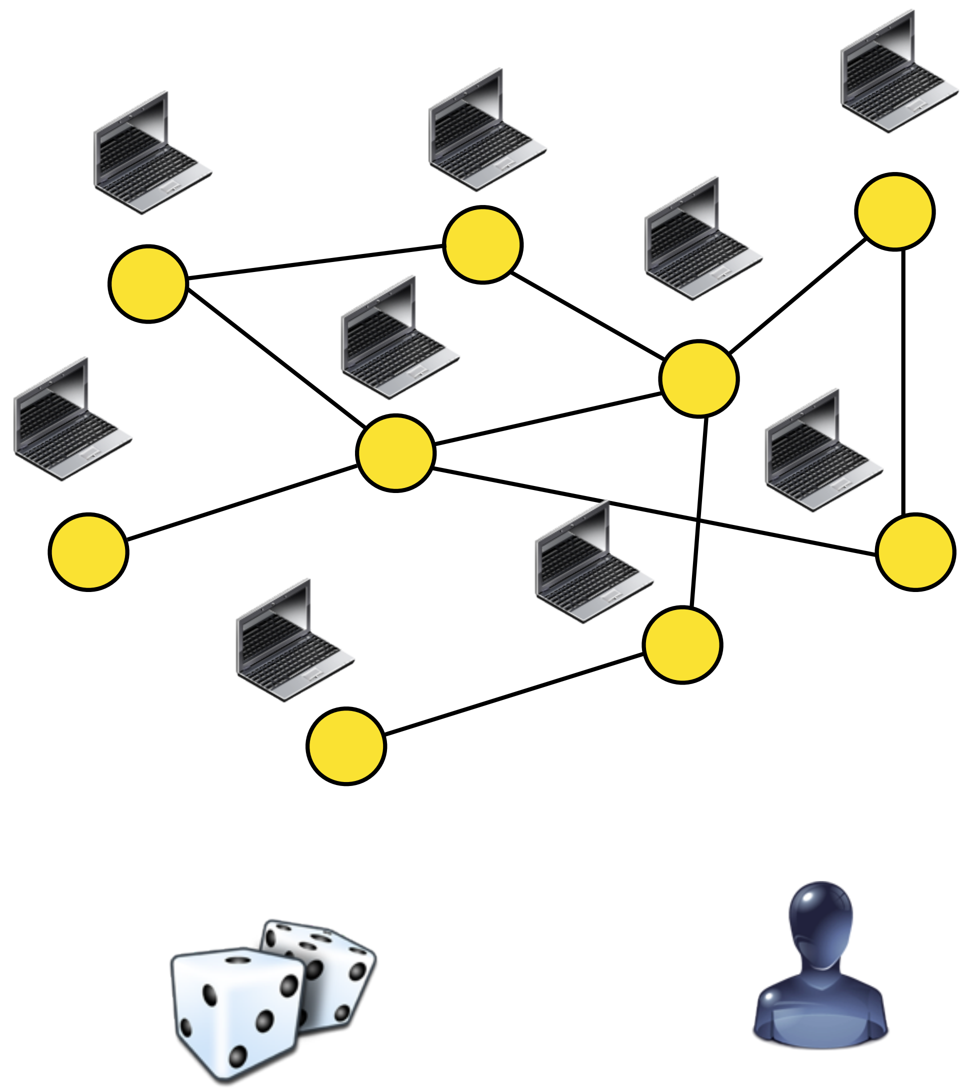
…eree

- Referee outputs the solution

- Access to shared randomness

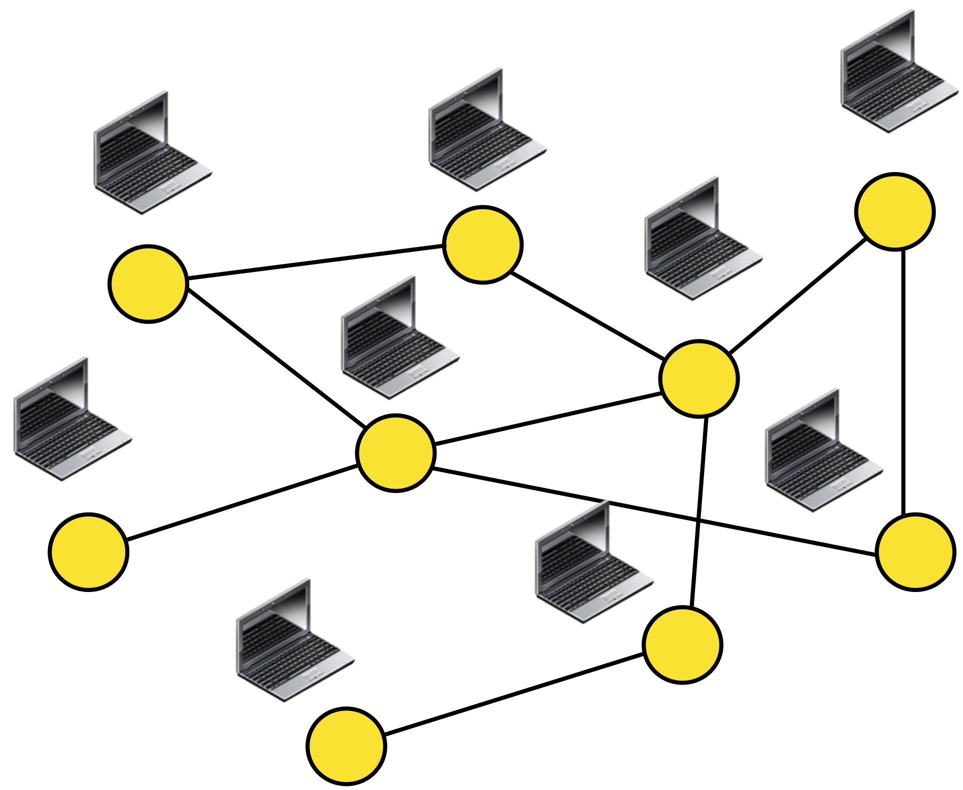
Multi-Round
Distributed Sketching Model

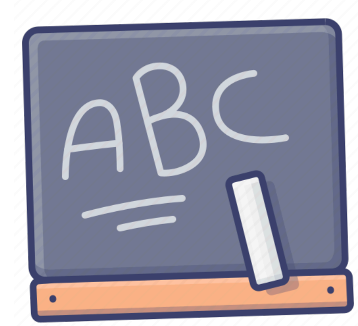
- Everything same as before
- Blackboard instead of a referee



Multi-Round
Distributed Sketching Model

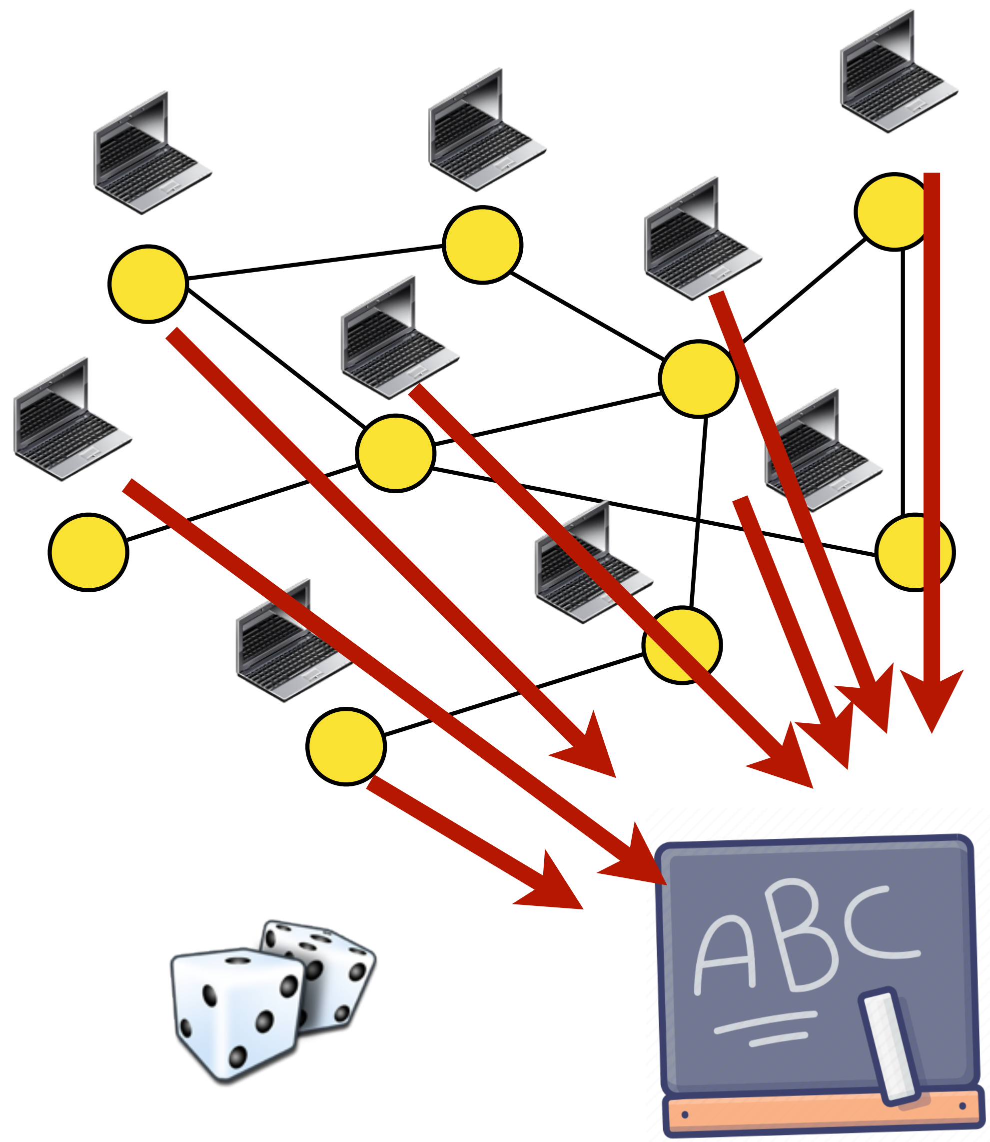
- Everything same as before
- Blackboard instead of a referee





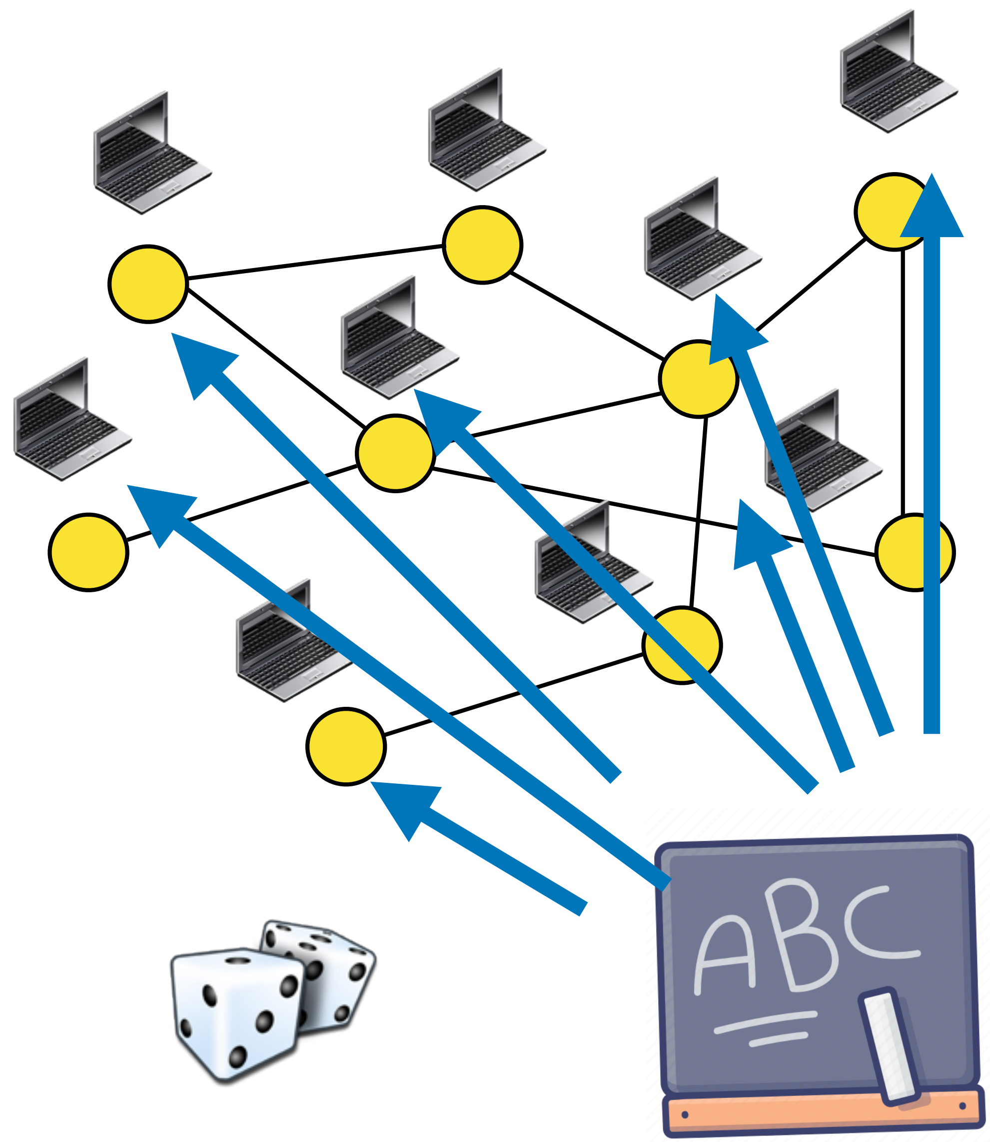
Multi-Round
Distributed Sketching Model

- Everything same as before
- Blackboard instead of a referee



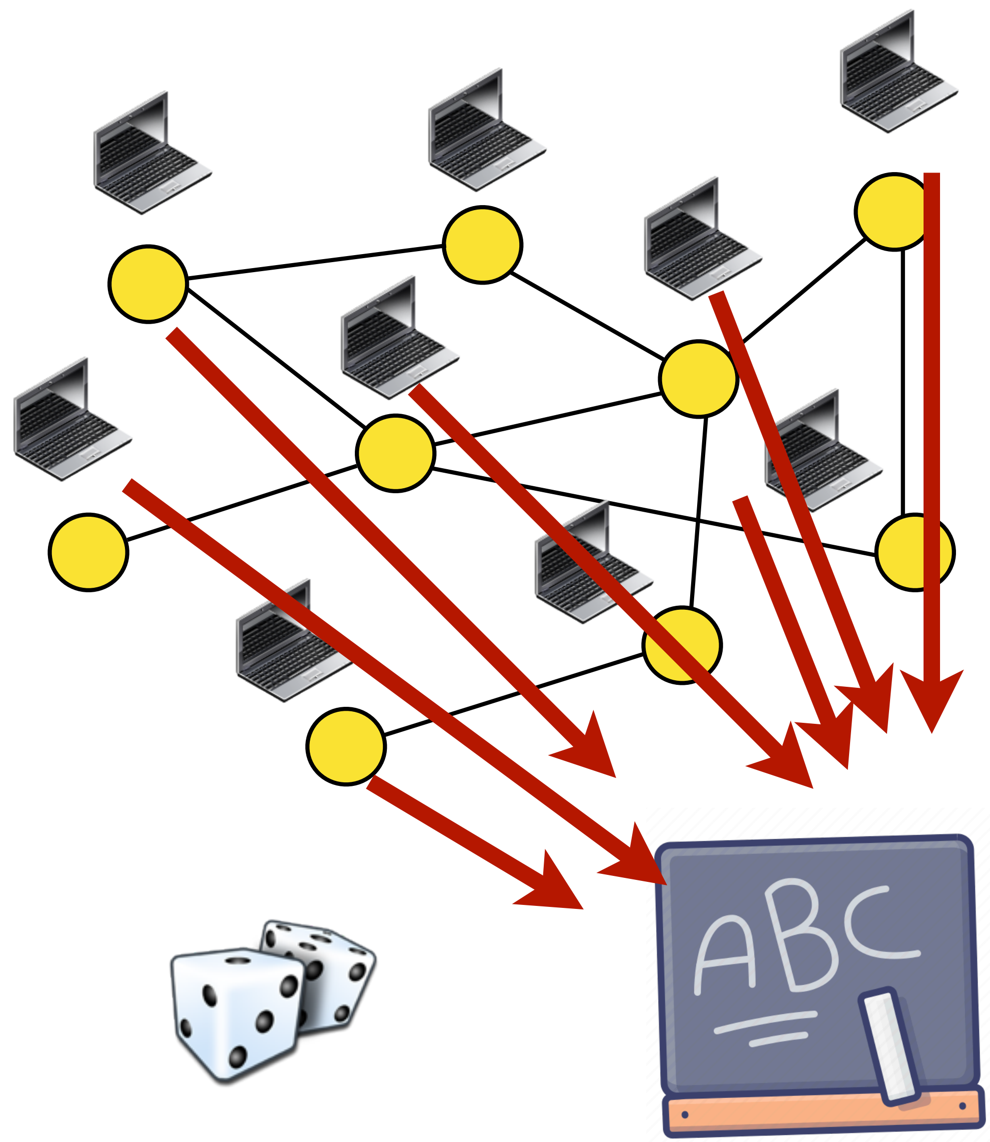
Multi-Round
Distributed Sketching Model

- Everything same as before
- Blackboard instead of a referee
- Read the blackboard at the end of the round



Multi-Round
Distributed Sketching Model

- Everything same as before
- Blackboard instead of a referee
- Read the blackboard at the end of the round
- Communicate based on the new information in the next round



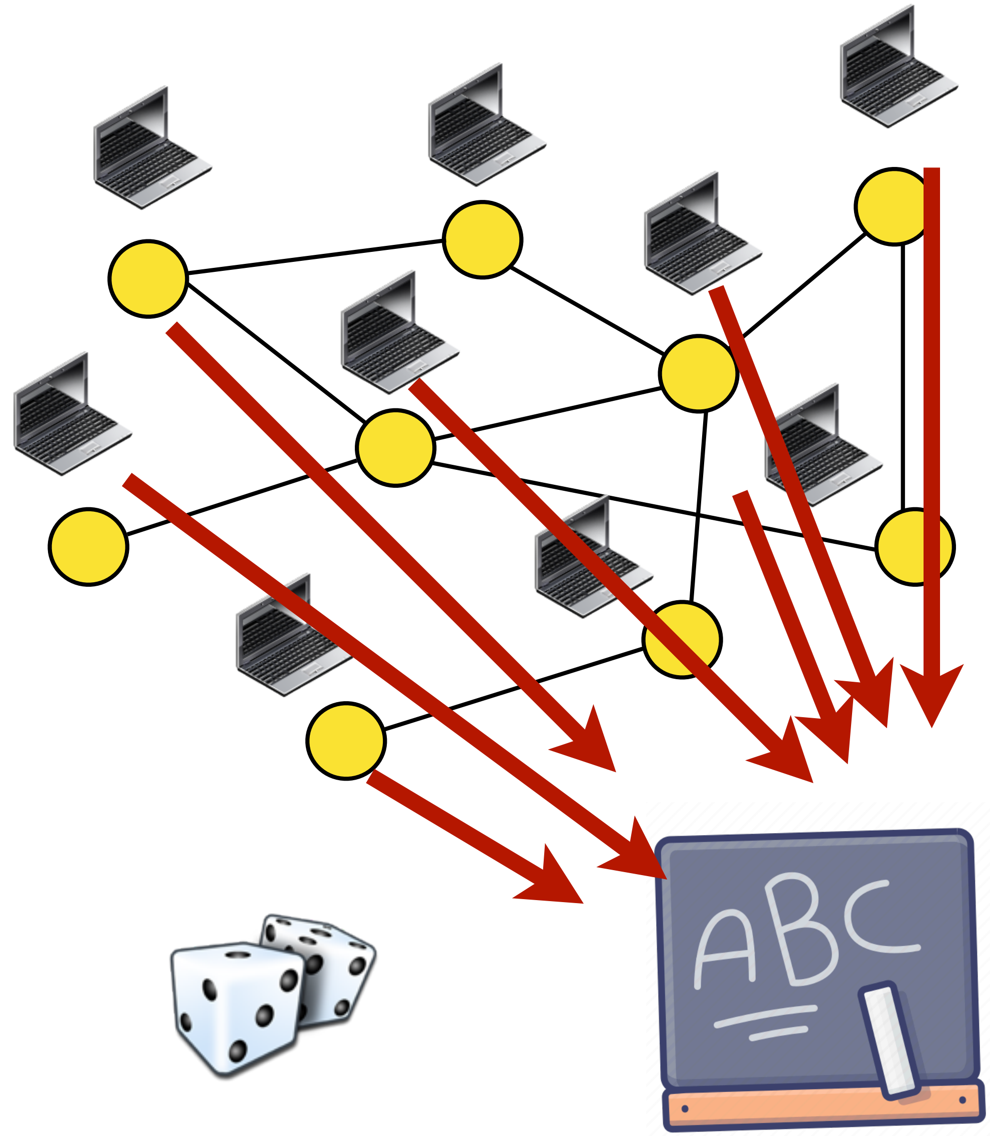
Multi-Round
Distributed Sketching Model

- Ev before

xt

rou

This model is also called
Broadcast Congested Clique



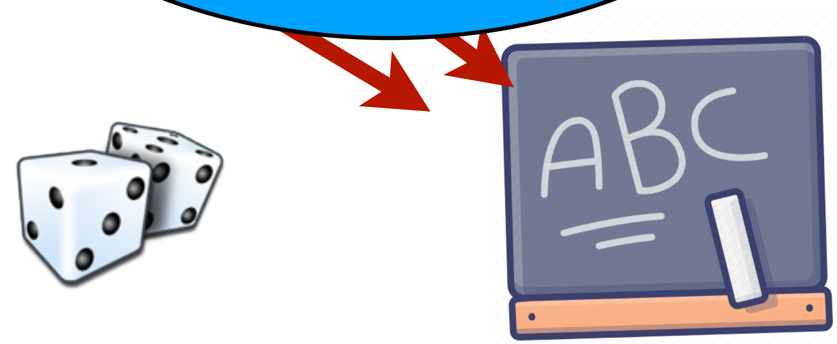
Multi-Round
Distributed Sketching Model

- Ev… before

…xt
rou…

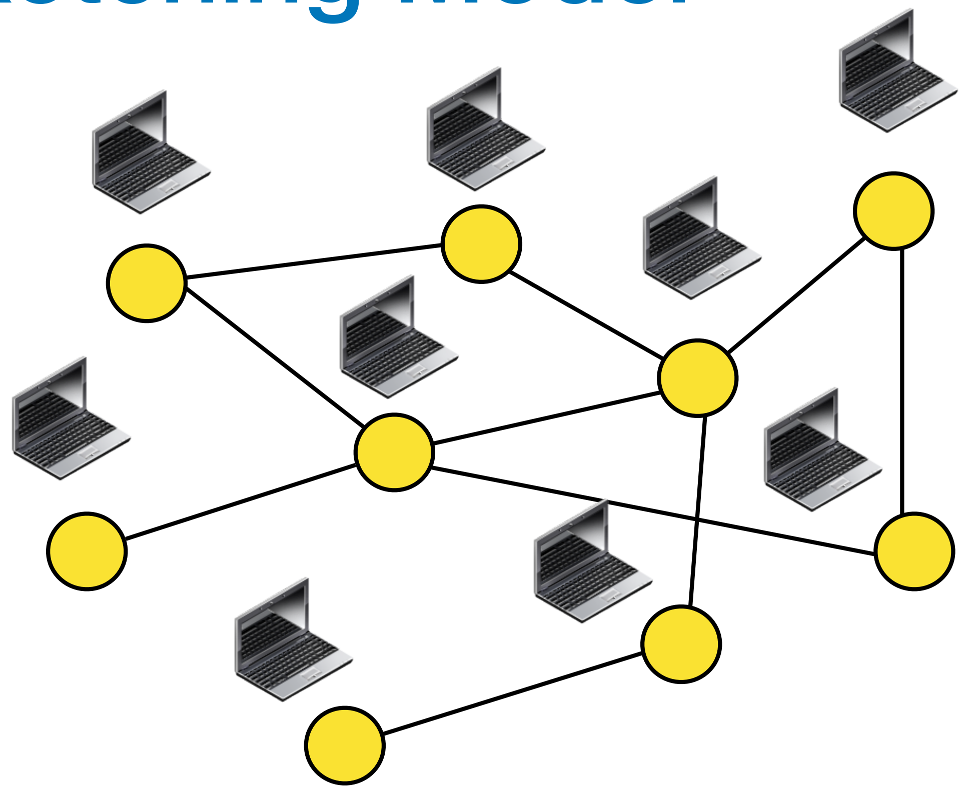
This model is also called
Broadcast Congested Clique

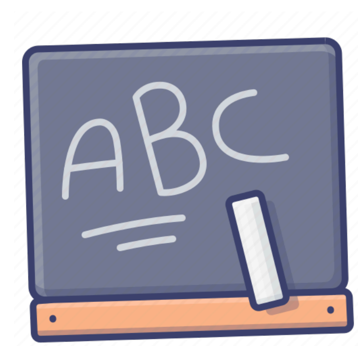
It is also closely related to
Dynamic Graph Streaming
(multi-pass algorithms)



Distributed Sketching Model

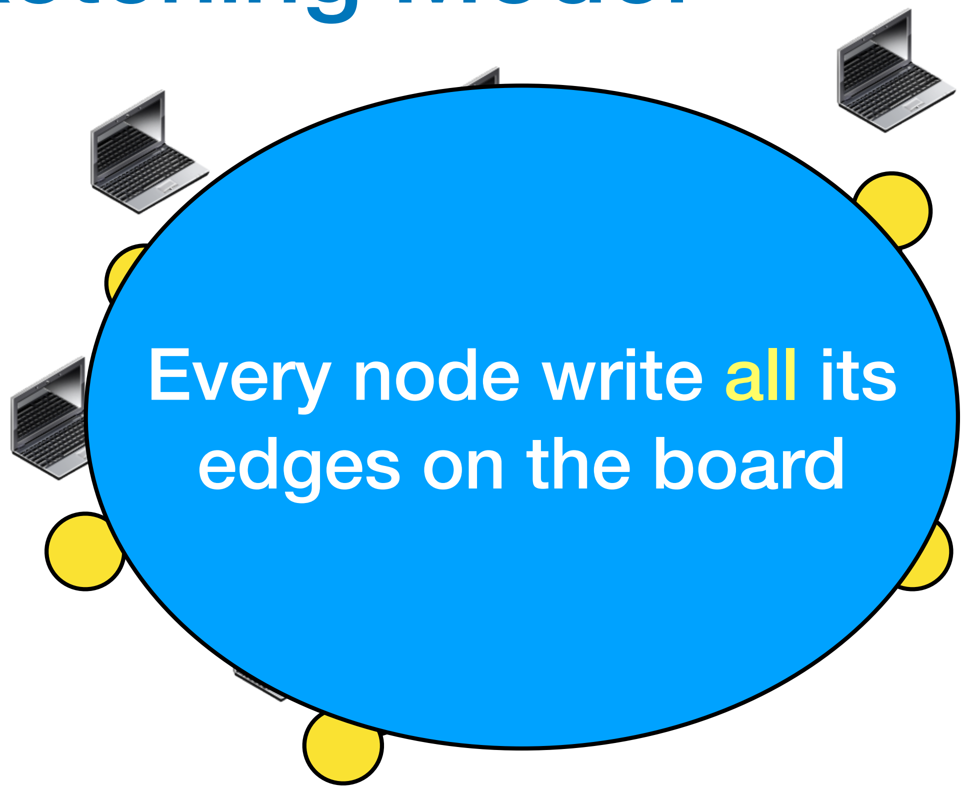
- Every problem can be solved with $O(n)$ size messages in a single round

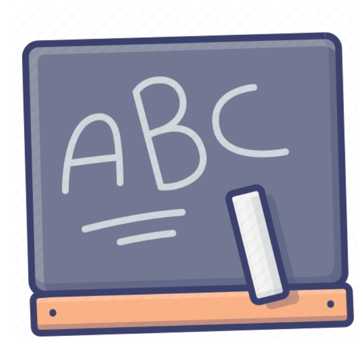




Distributed Sketching Model

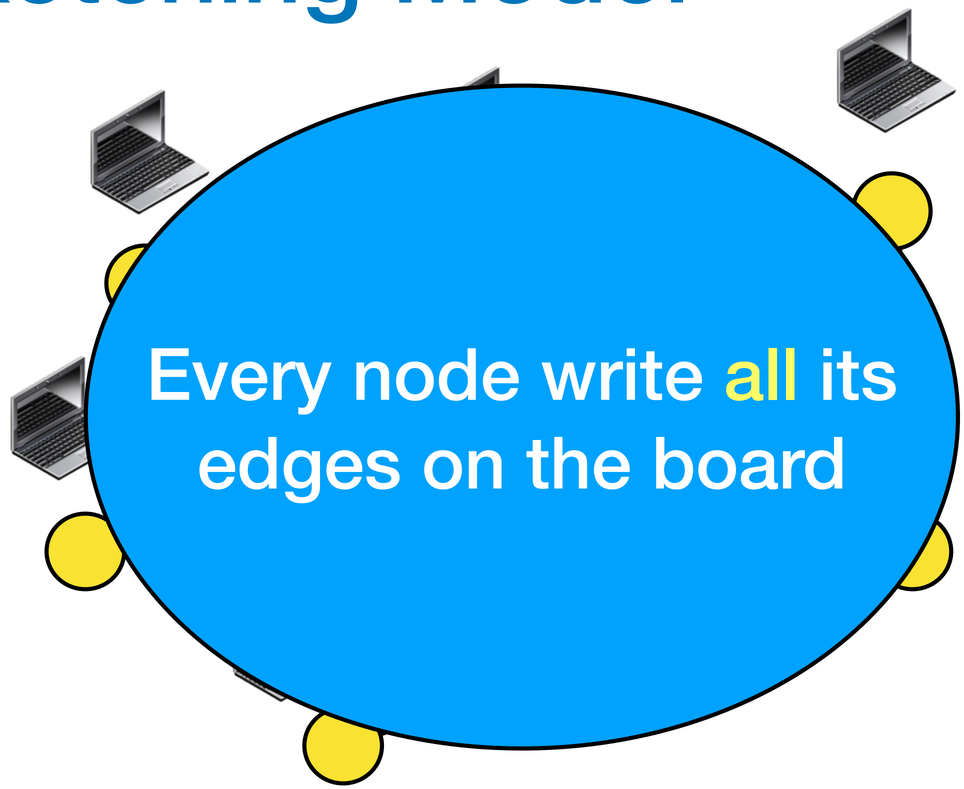
- Every problem can be solved with $O(n)$ size messages in a single round

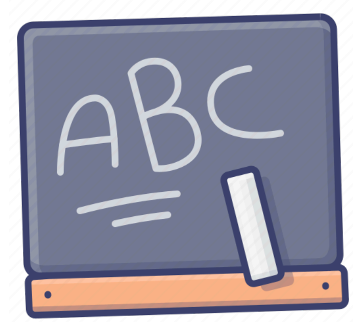




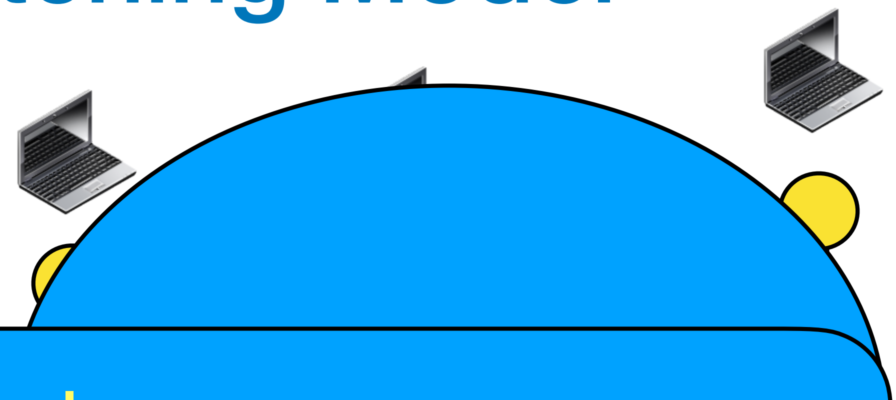
Distributed Sketching Model

- Every problem can be solved with $O(n)$ size messages in a single round
- Too much communication!





Distributed Sketching Model

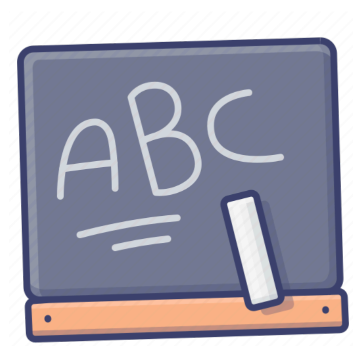


- Every problem can be solved with $O(n)$ size messages in a si
- T

Goal:

Solve problems with ideally:

1. polylog(n) communication
2. Small number of rounds (one or $O(1)$ rounds)

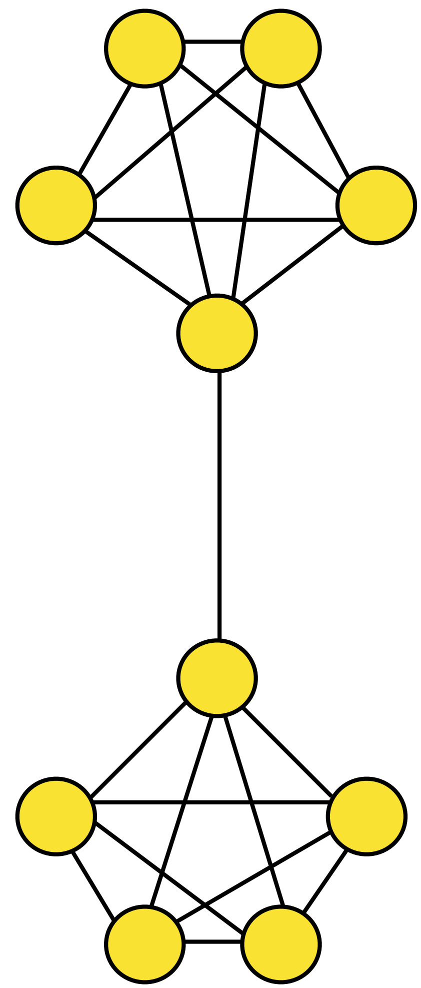


Are There Even "Non-Trivial" Algorithms in this Model?

“Non-Trivial” Algorithms?

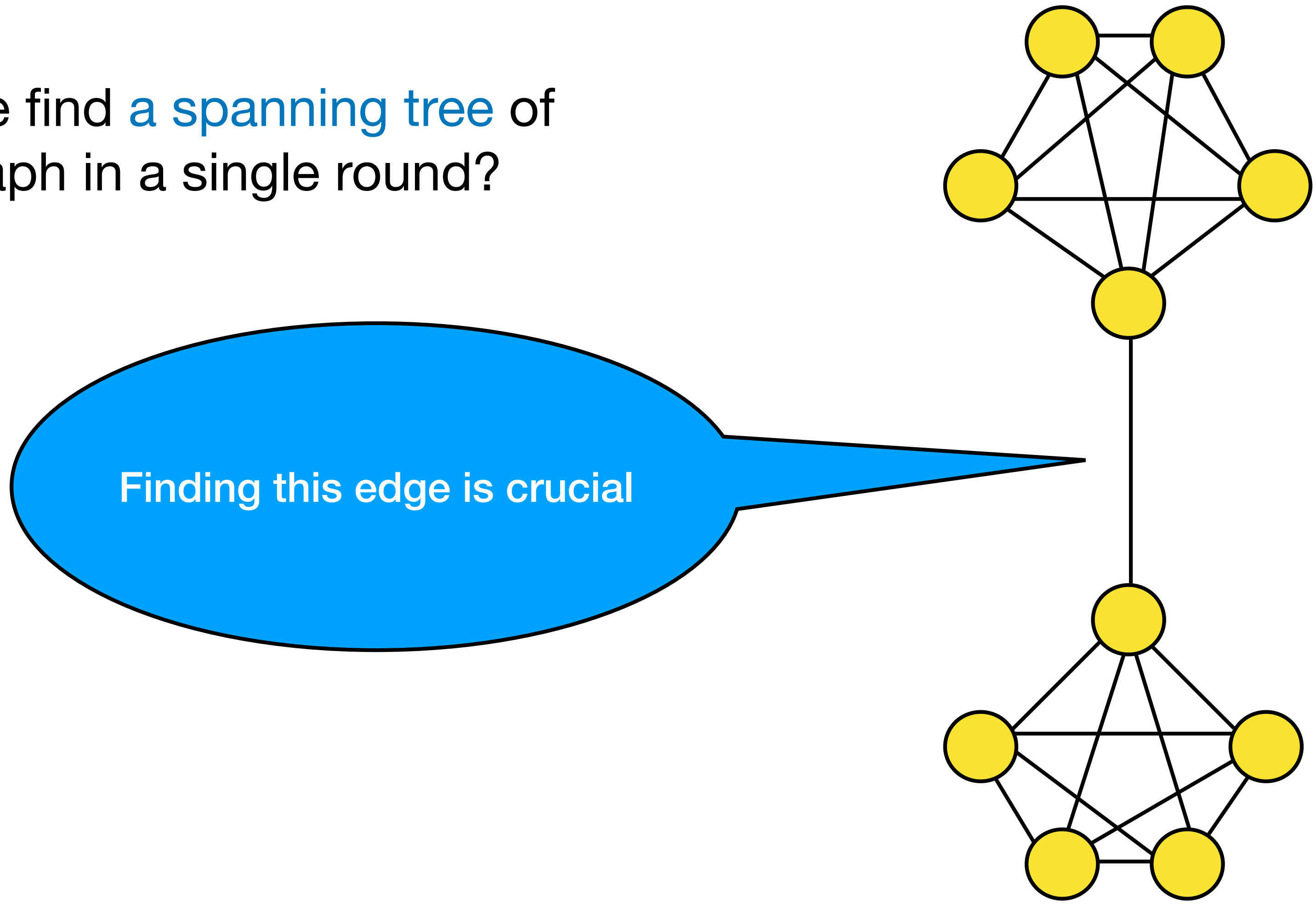
"Non-Trivial" Algorithms?

- Can we find a spanning tree of this graph in a single round?



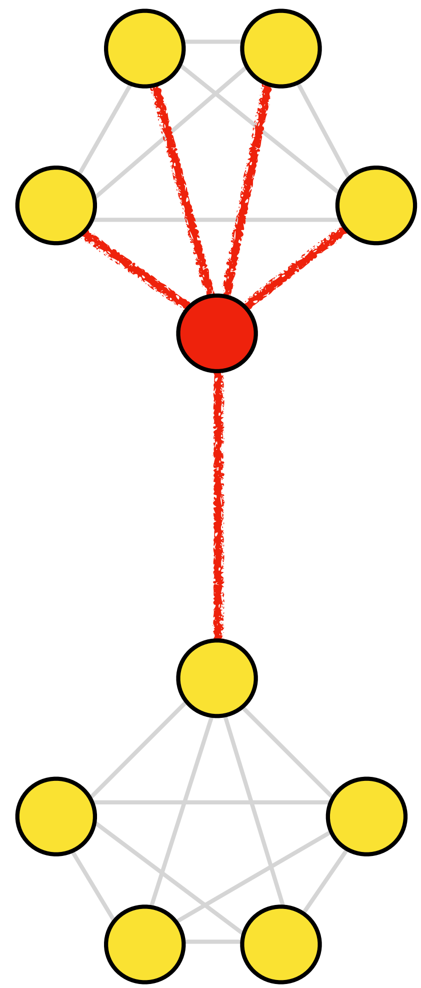
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?



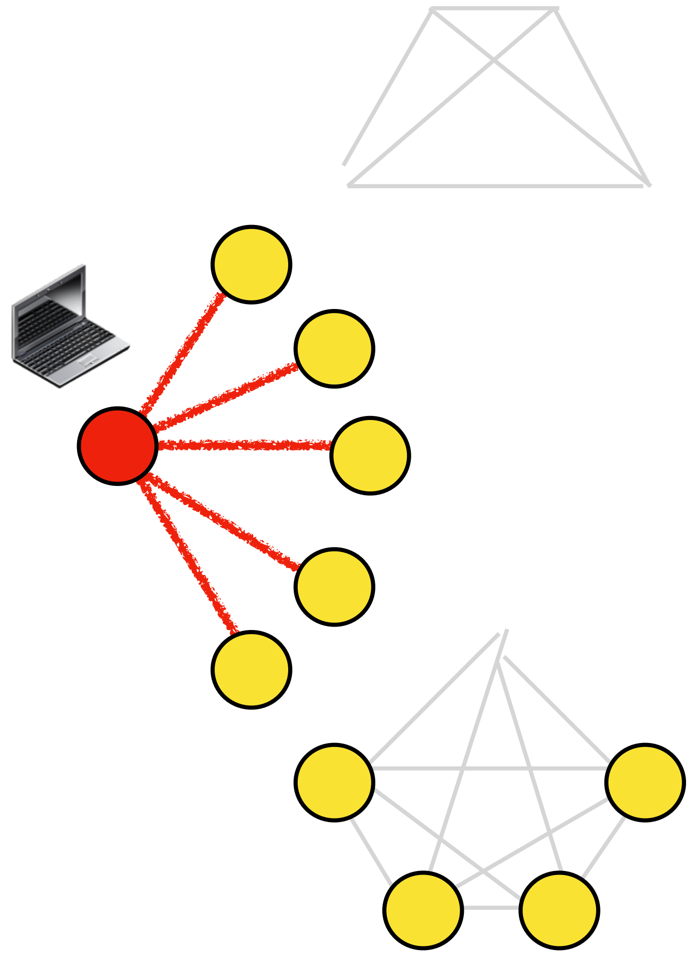
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?



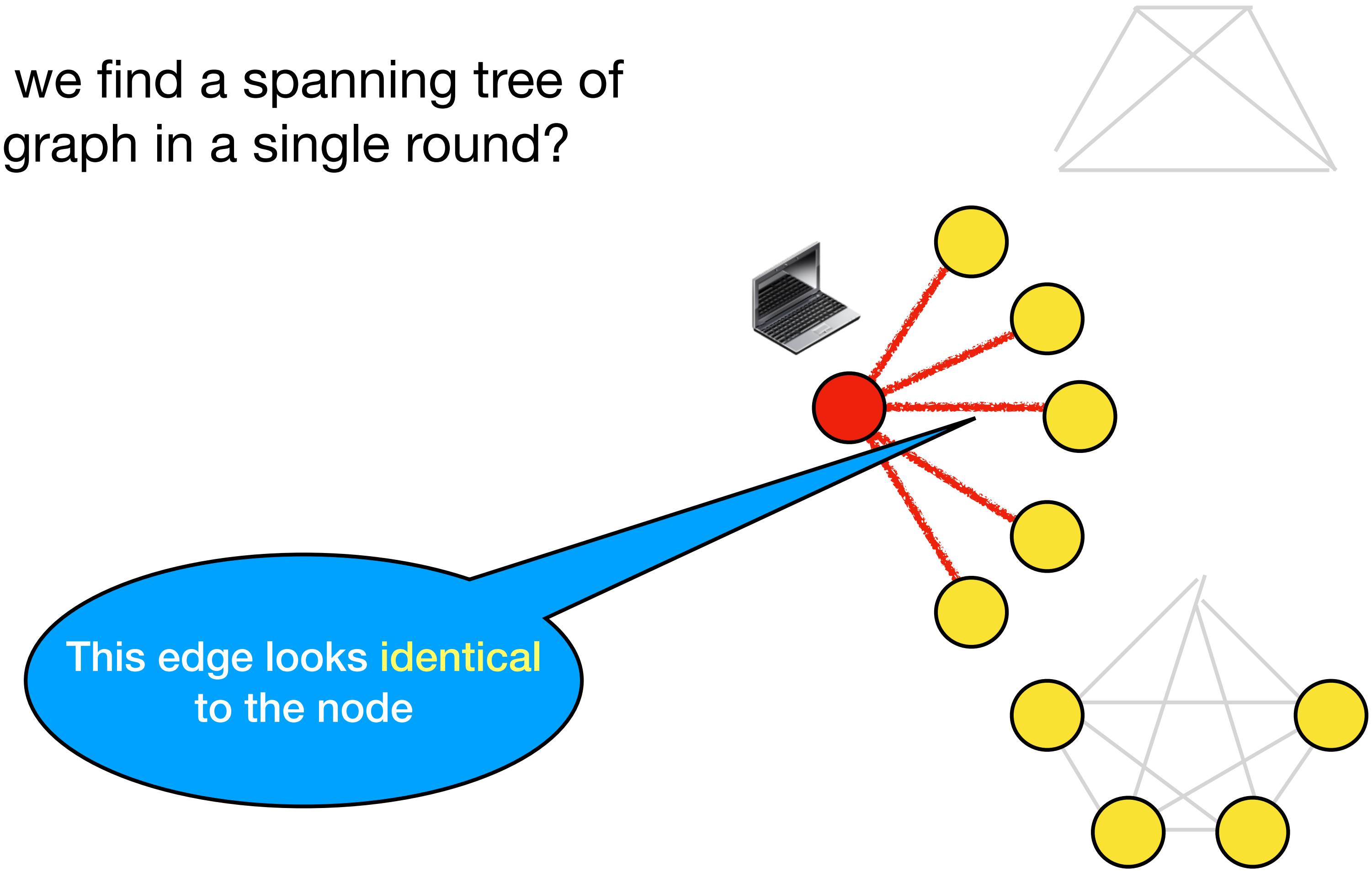
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?



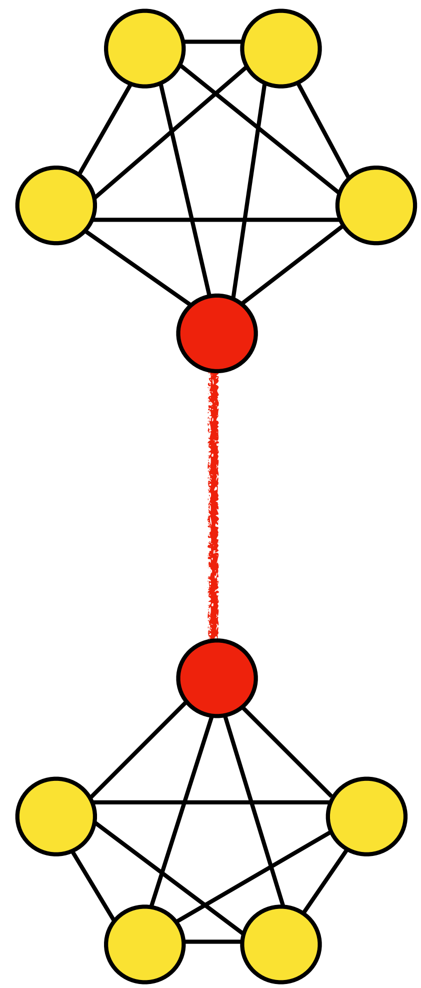
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?



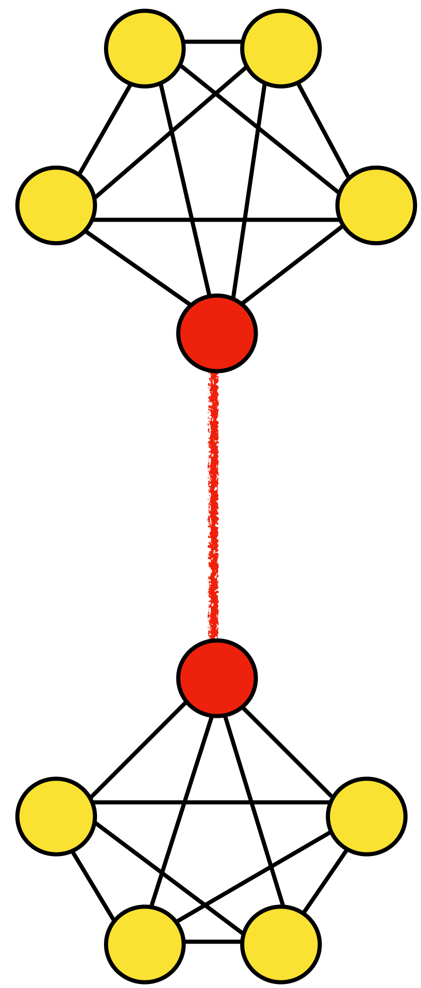
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?
- How can the two endpoints inform the referee about the crucial edge?



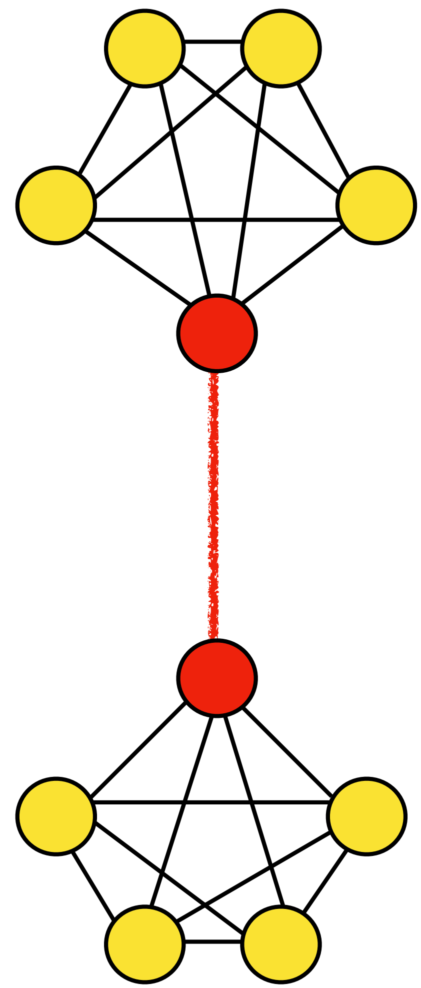
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?
- How can the two endpoints inform the referee about the crucial edge?
- **Important observation:** Edges are shared by both endpoints



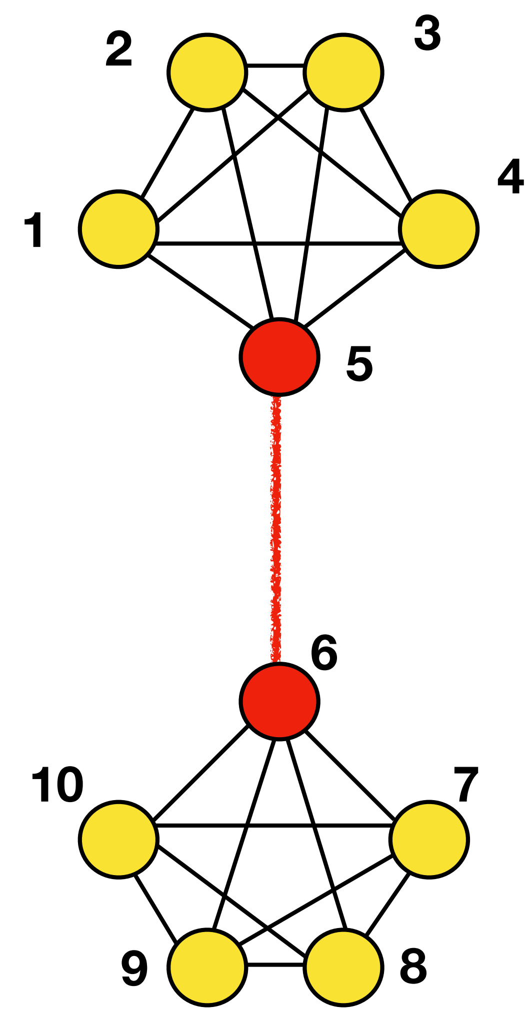
“Non-Trivial” Algorithms?

- Can we find a spanning tree of this graph in a single round?
- How can the two endpoints inform the referee about the crucial edge?
- **Important observation:** Edges are shared by both endpoints
- Other nodes can also inform the referee about this edge!



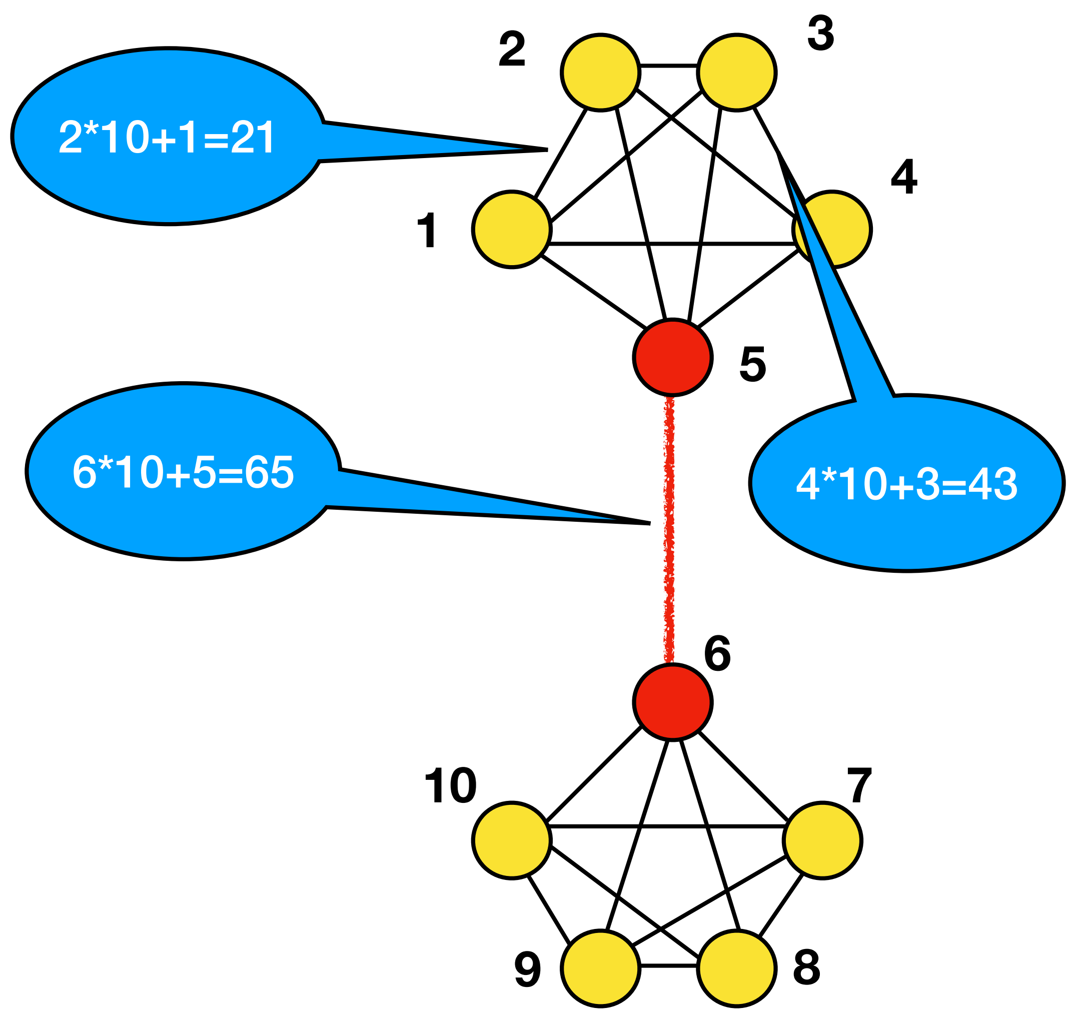
A Concrete Solution

- Assign a unique number to each edge



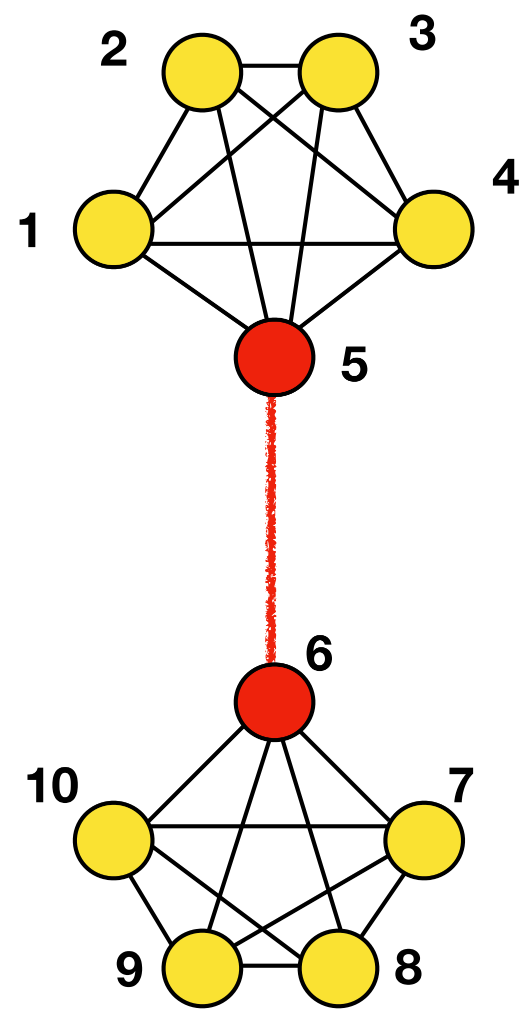
A Concrete Solution

- Assign a unique number to each edge



A Concrete Solution

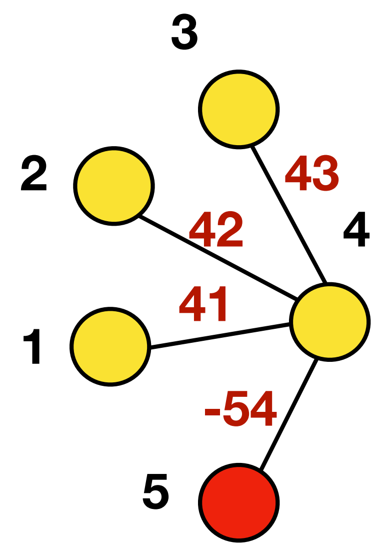
- Assign a unique number to each edge
- Direct edges and write the number or its negation on each edge based on direction



A Concrete Solution

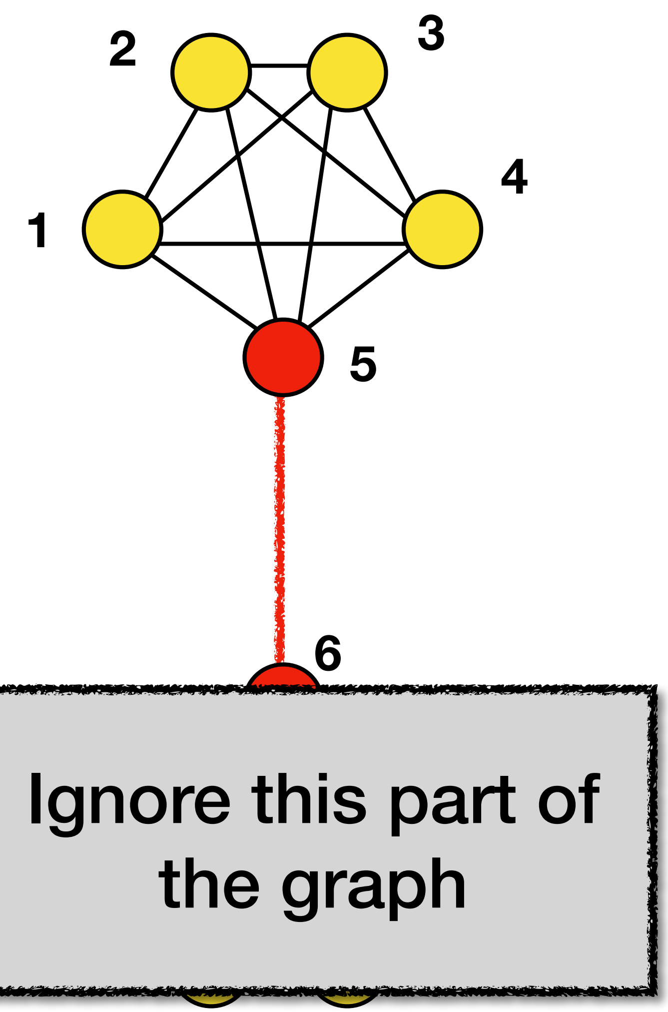
- Assign a unique number to each edge

- Direct edges and write the number or its negation on each edge based on direction



A Concrete Solution

- Assign a unique number to each edge
- Direct edges and write the number or its negation on each edge based on direction
- Each node sends sum of the numbers on its edges

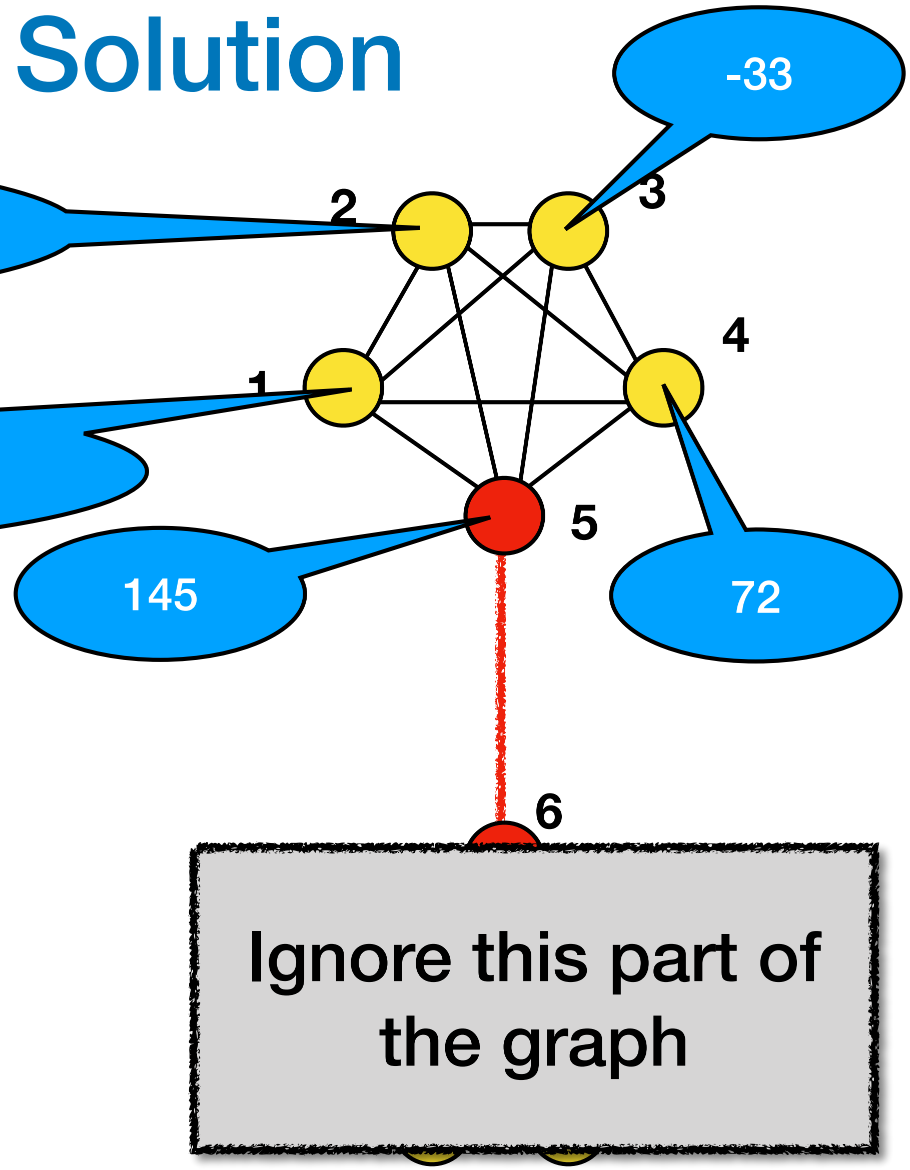


A Concrete Solution

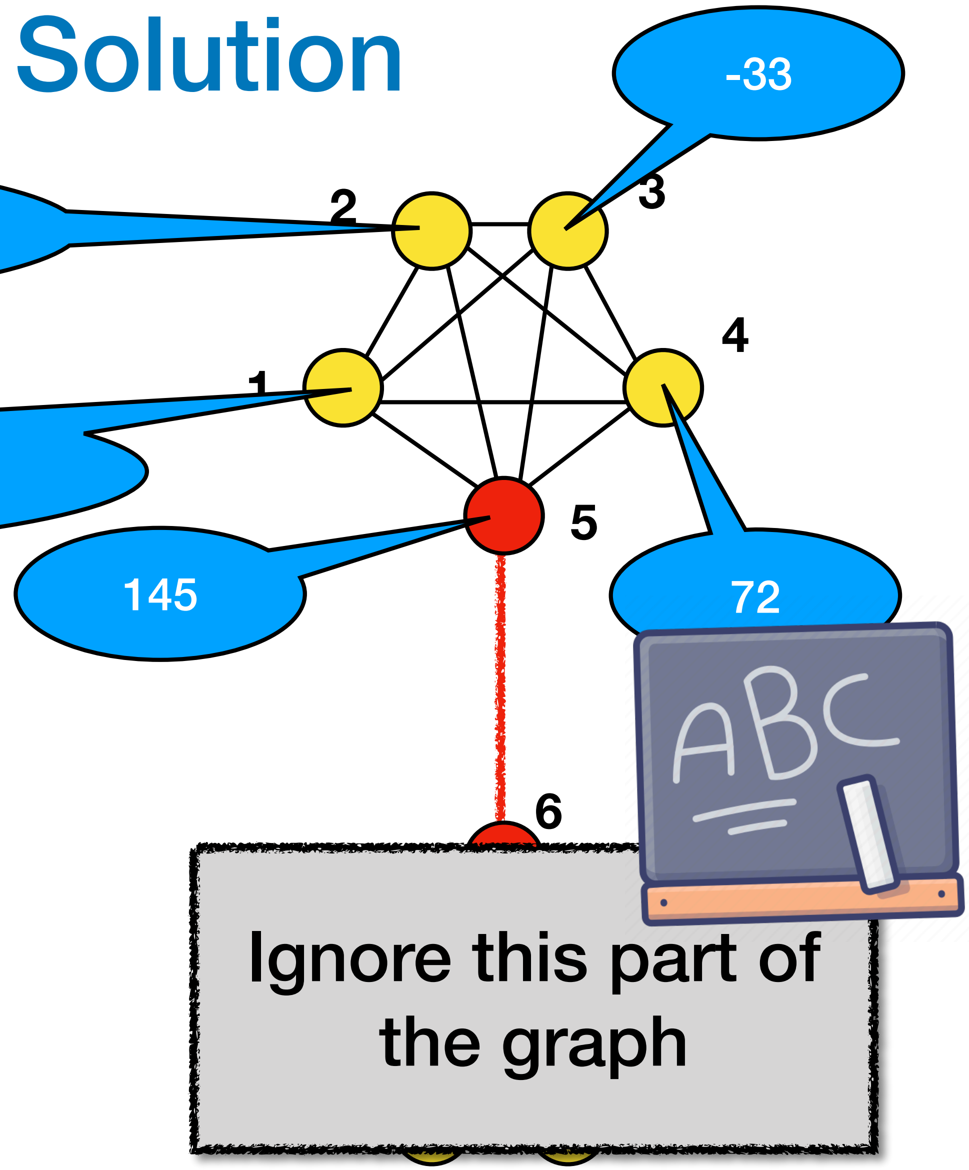
- Assign a unique number to each edge

- Di… number or its negation on each edge based on direction

- Each node sends sum of the numbers on its edges



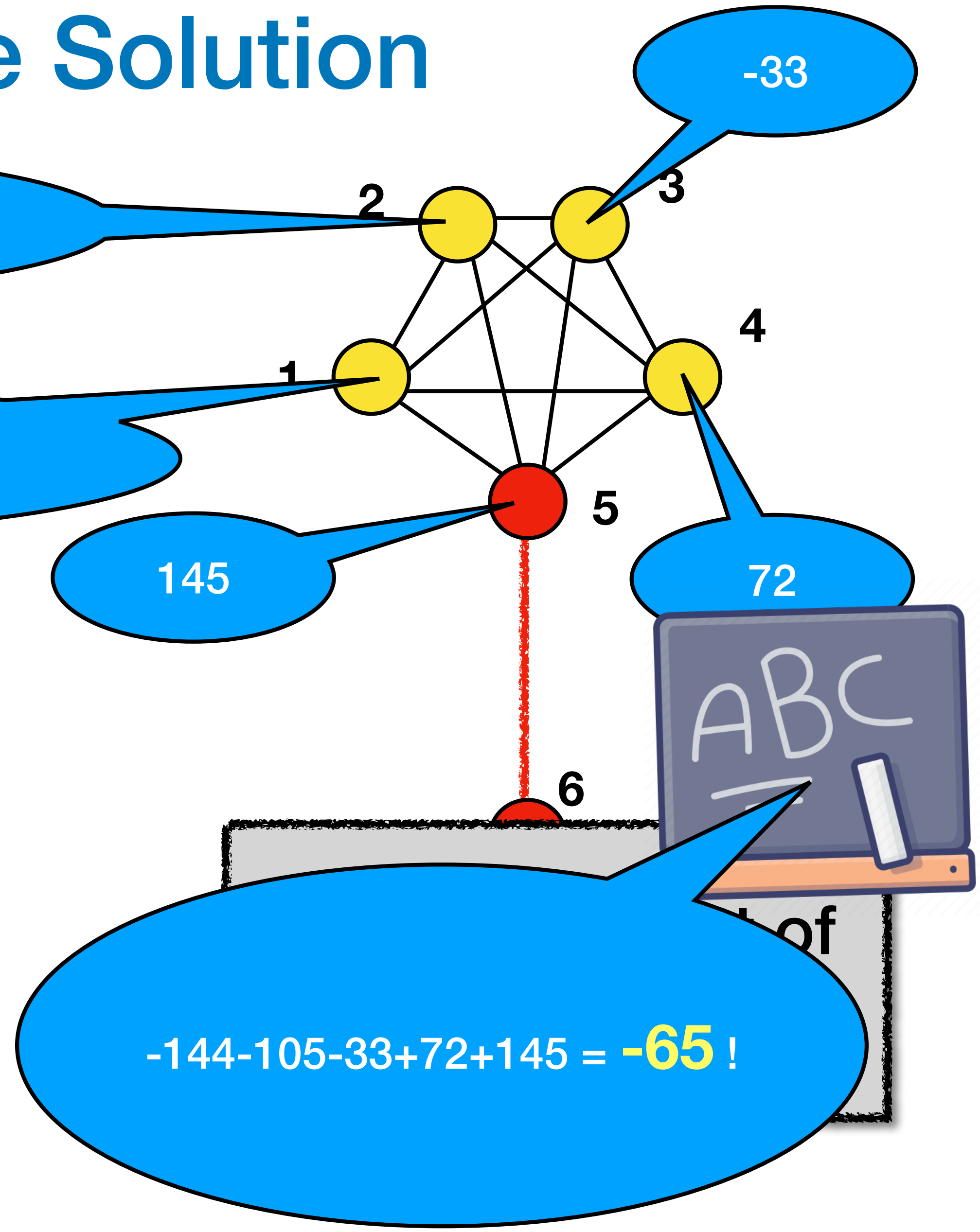
A Concrete Solution



- Assign a unique number to each edge
- Di… number or its negation on each edge based on direction
- Each node sends sum of the numbers on its edges
- Add up the numbers on the board in one of the partitions to find the crucial edge

A Concrete Solution

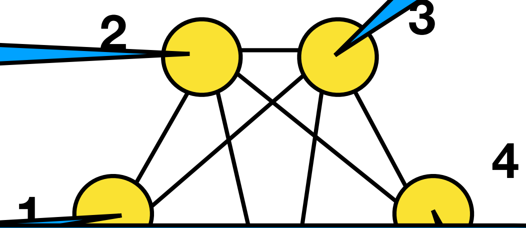
- Assign a unique number to each edge
- Di number or its negation on each edge based on direction
- Each node sends sum of the numbers on its edges
- Add up the numbers on the board in one of the partitions to find the crucial edge



A Concrete Solution

-52-42-32+21 = -105

-33



- Assign a unique number to each edge

Takeaway: vertices can indirectly inform the referee about edges of other vertices

- Add up the numbers on the board in one of the partitions to find the crucial edge

-144-105-33+72+145 = -65 !

Distributed Sketching Algorithms

- AGM sketch of Ahn, Guha, and McGregor [AGM12a]: $\mathcal{O}(\log^3(n))$ size messages for finding a spanning forest of every graph with high probability.

n: number of vertices

Distributed Sketching Algorithms

- AGM sketch of Ahn, Guha, and McGregor [AGM12a]: $\mathcal{O}(\log^3(n))$ size messages for finding a spanning forest of every graph with high probability.
- Extended to various other problems:
  - MST and edge connectivity [AGM12a]
  - Vertex connectivity [GMT15][**A**S22]
  - Subgraph Counting [AGM12b]
  - Sparsifiers and approximate min/max cuts [AGM13, KLMMS14]
  - Spanners and approximate shortest paths [FKN21]
  - Densest Subgraph, degeneracy, and arboricity [BHNT15, MTSV15, CT16]
  - Graph coloring [**A**CK19][BCG20][**A**KM22][HKNT22]

n: number of vertices

Majority of these results only need a single round!

What About Lower Bounds?

Lower Bounds

- A key tool for proving lower bounds: communication complexity

Lower Bounds

- A key tool for proving lower bounds: communication complexity
- Proving lower bounds in this model can be challenging

Lower Bounds

- A key tool for proving lower bounds: communication complexity
- Proving lower bounds in this model can be challenging
- Some reasons:
  - There are surprisingly strong algorithms that defy intuition
  - Edge sharing makes the model different from ``typical" distributed communication complexity lower bounds
    - Number-in-hand vs Number-on-forehead

Lower Bounds

- A key tool for proving lower bounds: communication complexity
- Proving lower bounds in this model can be challenging

  It is more challenging but not impossible!
- Some reasons:
  - There are surprisingly strong algorithms that defy intuition
  - Edge sharing makes the model different from ``typical" distributed communication complexity lower bounds
    - Number-in-hand vs Number-on-forehead

Distributed Sketching Lower Bounds

- Three categories of lower bounds:

Distributed Sketching Lower Bounds

- Three categories of lower bounds:

- Problems that are "hard" even for close to n rounds

  $\Omega(n)$ total communication is needed regardless of # rounds

  - E.g., testing triangle-freeness [BMNRST11][BMRT12]

Distributed Sketching Lower Bounds

- Three categories of lower bounds:

- Problems that are "hard" even for close to n rounds

  $\Omega(n)$ total communication is needed regardless of # rounds

  - E.g., testing triangle-freeness [BMNRST11][BMRT12]

- Problems that are "easy" even for one round

  $\Omega(\log^3 n)$ communication is needed for one-round

  - E.g., spanning forest and connectivity [NY19,Y21]

Distributed Sketching Lower Bounds

- Three categories of lower bounds:

- Problems that are "hard" even for close to n rounds

  $\Omega(n)$ total communication is needed regardless of # rounds

  - E.g., testing triangle-freeness [BMNRST11][BMRT12]

- Problems that are "easy" even for one round

  $\Omega(\log^3 n)$ communication is needed for one-round

  - E.g., spanning forest and connectivity [NY19,Y21]

- Problems that are "hard" initially but become "easy" in a few number of rounds (are round-sensitive)

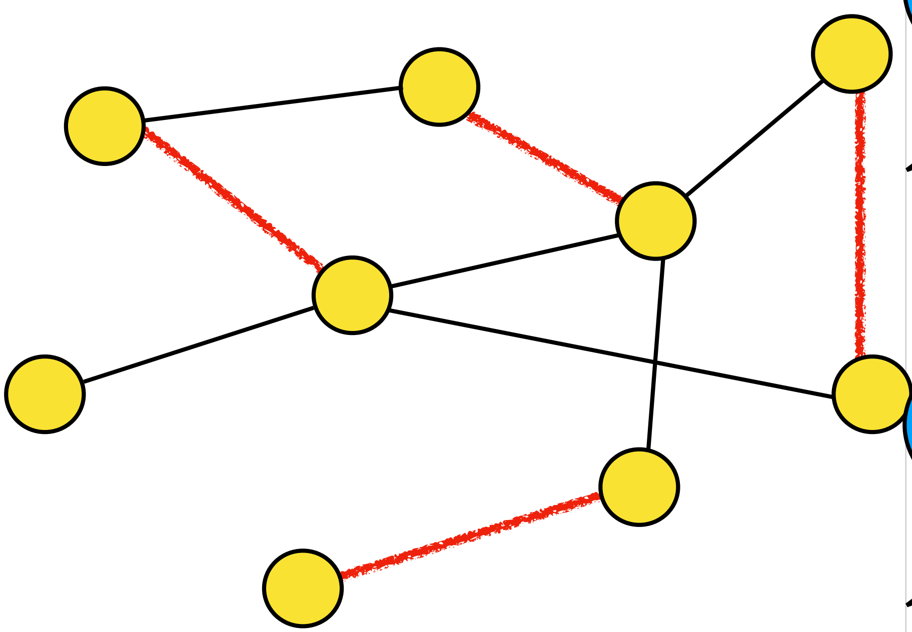
  $\log^{O(1)}(n)$ communication needs at least $\Omega(\log \log n)$ rounds

  - E.g., MIS and maximal matching [**A**KO20,**A**KZ22]

Distributed Sketching Lower Bounds for MIS and Maximal Matching

Maximal Matching and Maximal Independent Set

- Maximal Matching:



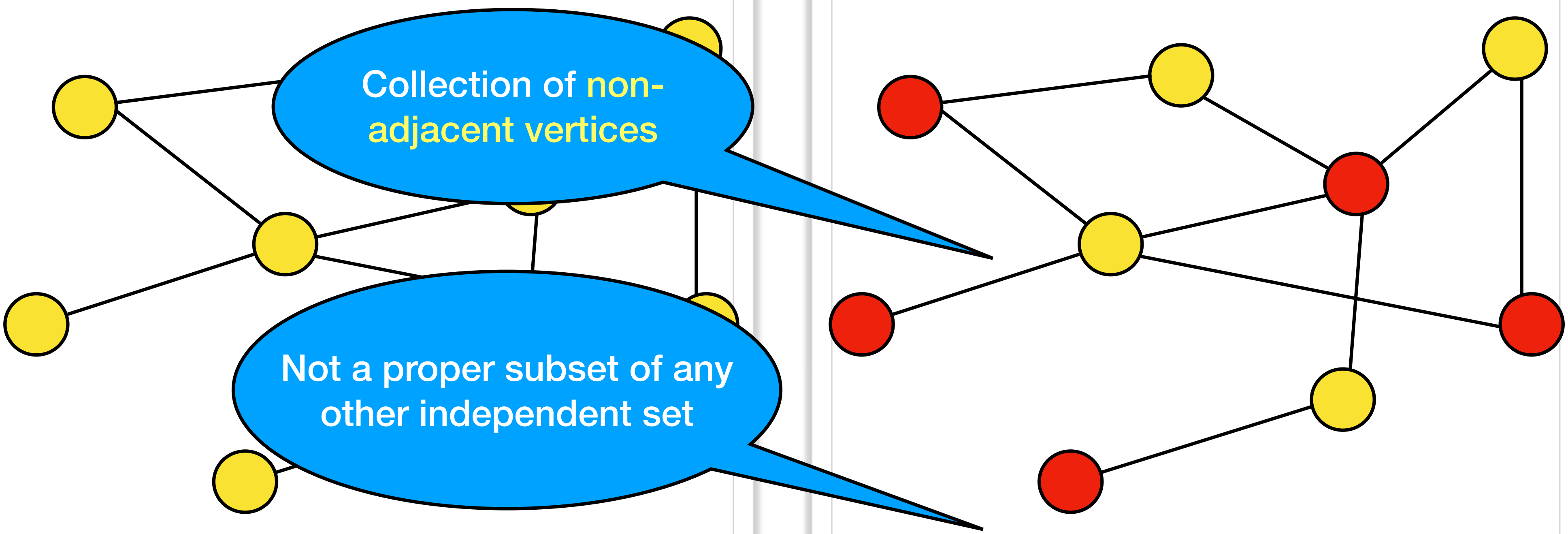
Collection of vertex-disjoint edges

Not a proper subset of any other matching

Maximal Matching and Maximal Independent Set

- Maximal Matching:

- Maximal Independent Set (MIS):



Algorithms for MIS and Matching?

- Luby's algorithm [L86] gives protocols for both problems with $O(\log n)$ round and $O(\log n)$ communication per-player
- [GGKMR18] gives an $O(\log \log n)$ round algorithm for MIS with $\log^{O(1)}(n)$ communication per-player on average

Lower Bounds

Lower Bounds

[**A**, Kol, Oshman; 2020]

Any single-round protocol for MIS or maximal matching requires $n^{1/2-o(1)}$ communication per-player even on average

Lower Bounds

[**A**, Kol, Oshman; 2020]

Any single-round protocol for MIS or maximal matching requires $n^{1/2-o(1)}$ communication per-player even on average

[**A**, Kol, Zhang; 2022]

Any r-round protocol for MIS or maximal matching requires $n^{1/20^{r+1}}$ communication per-player in the worst case

Lower Bounds

[**A**, Kol, Oshman; 2020]

Any single-round protocol for MIS or maximal matching requires $n^{1/2-o(1)}$ communication per-player even on average

[**A**, Kol, Zhang; 2022]

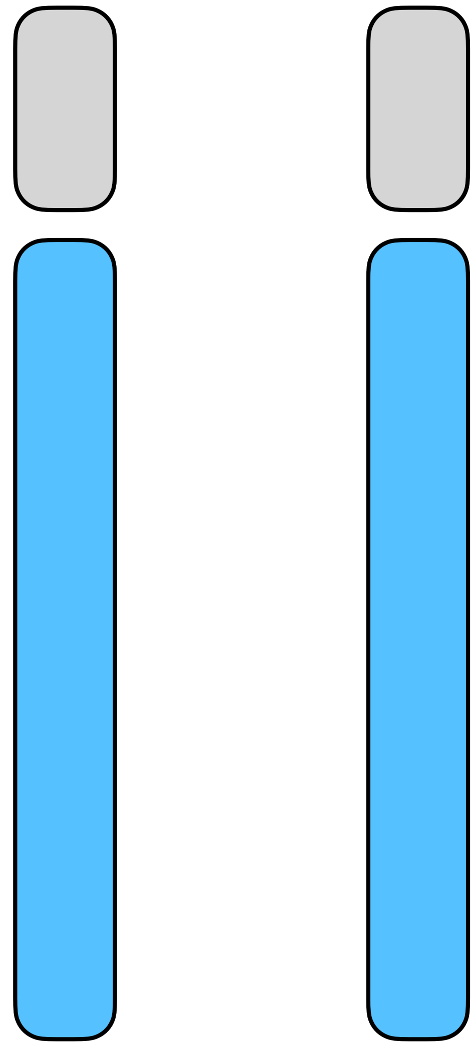
Any r-round protocol for MIS or maximal matching requires $n^{1/20^{r+1}}$ communication per-player in the worst case

Corollary: $\Omega(\log\log n)$ rounds are necessary for $\log^{O(1)}(n)$ communication

Lower Bounds for Maximal Matchings

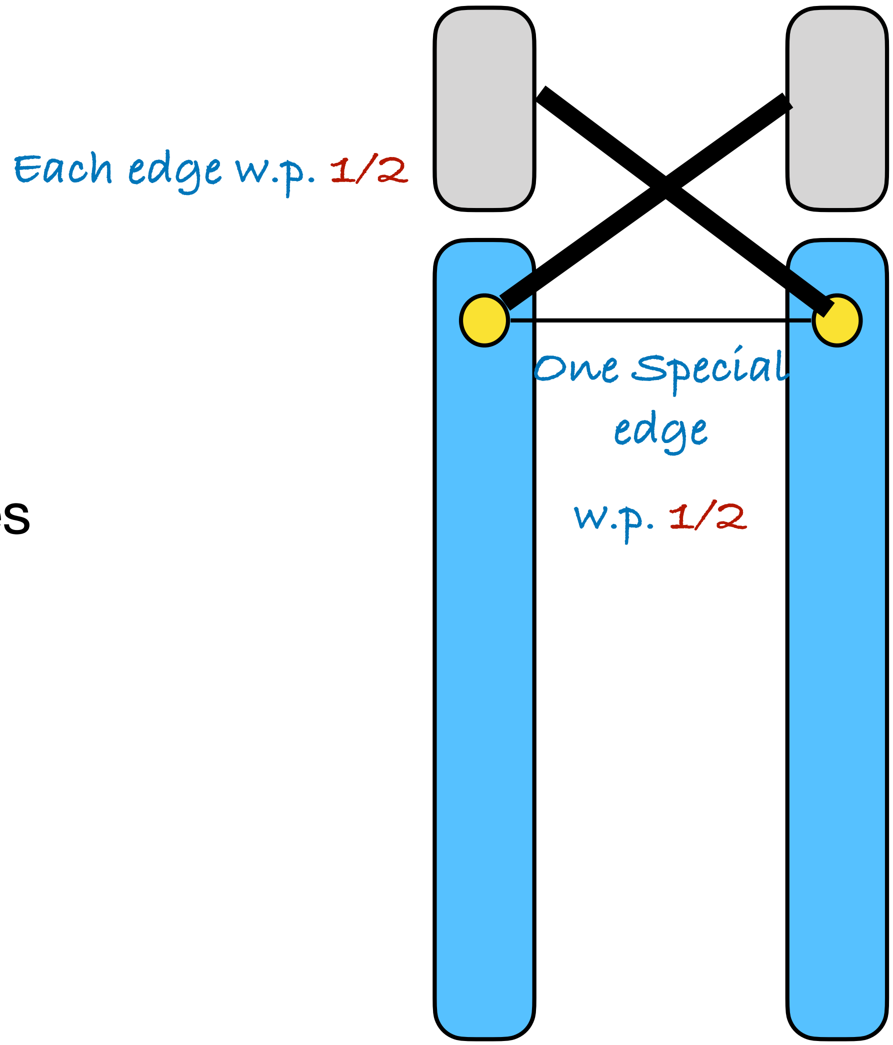
Warm-Up: One-Round Lower Bound

- On each side:
  - n vertices
  - $\sqrt{n}$ fooling vertices
  - $n - \sqrt{n}$ principal vertices



Warm-Up: One-Round Lower Bound

- On each side:
  - n vertices
  - $\sqrt{n}$ fooling vertices
  - $n - \sqrt{n}$ principal vertices

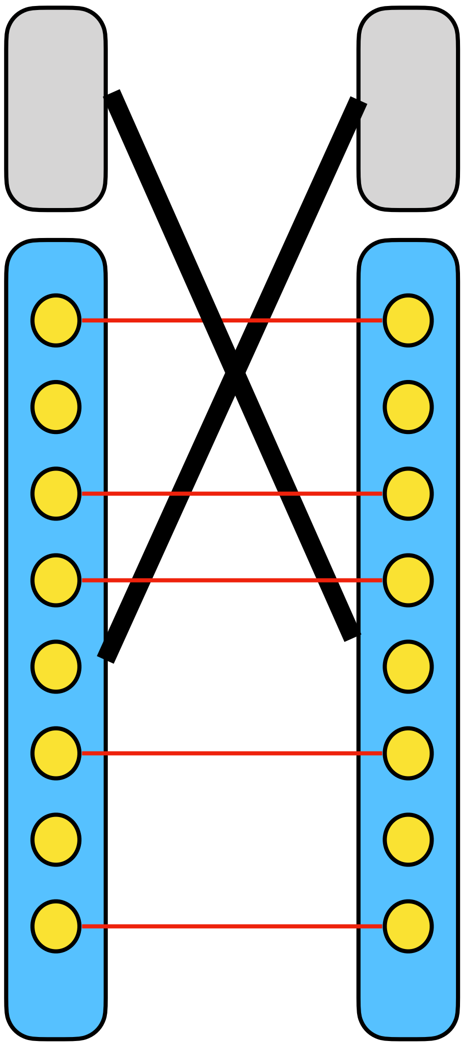


Warm-Up: One-Round Lower Bound

- On each side:
  - n vertices
  - $\sqrt{n}$ fooling vertices
  - $n - \sqrt{n}$ principal vertices

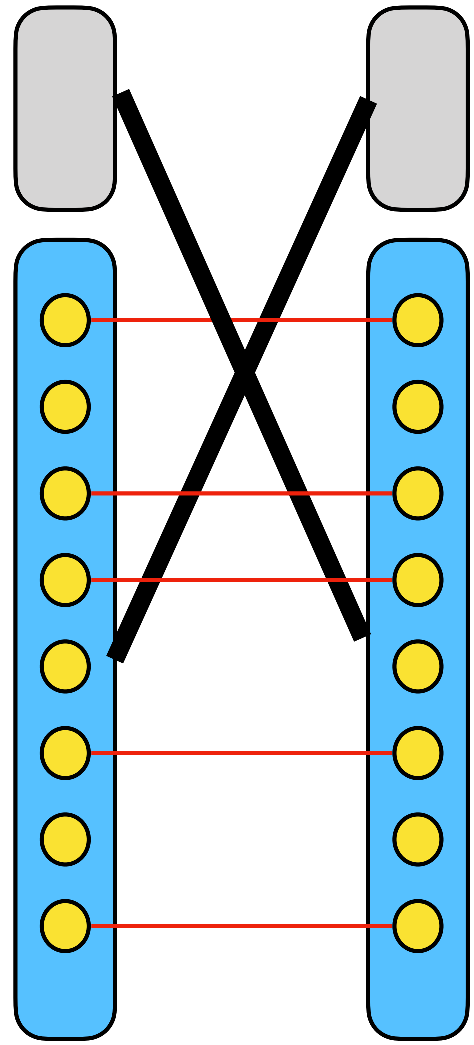
Any maximal matching needs many special edges

A similar-in-spirit construction to lower bounds for approximate matching in dynamic streams [K15,AKLY16]



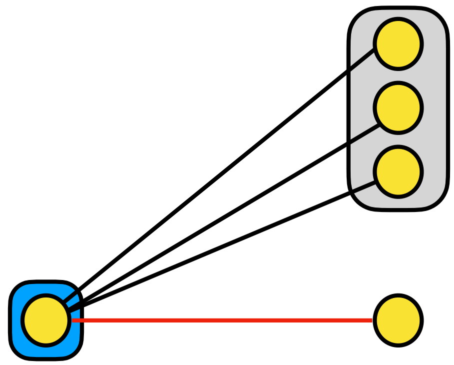
Warm-Up: One-Round Lower Bound

- Analysis?



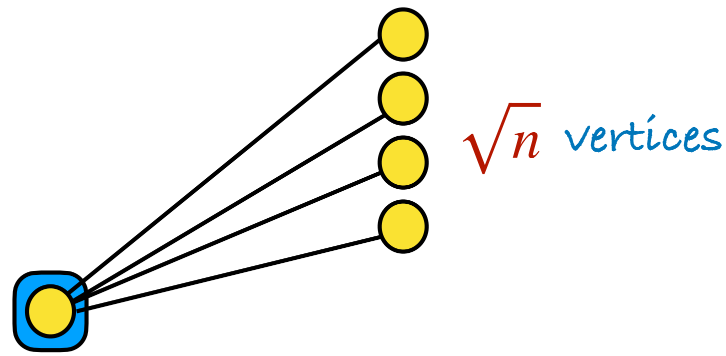
Warm-Up: One-Round Lower Bound

- Analysis?
- From the perspective of principal vertices:



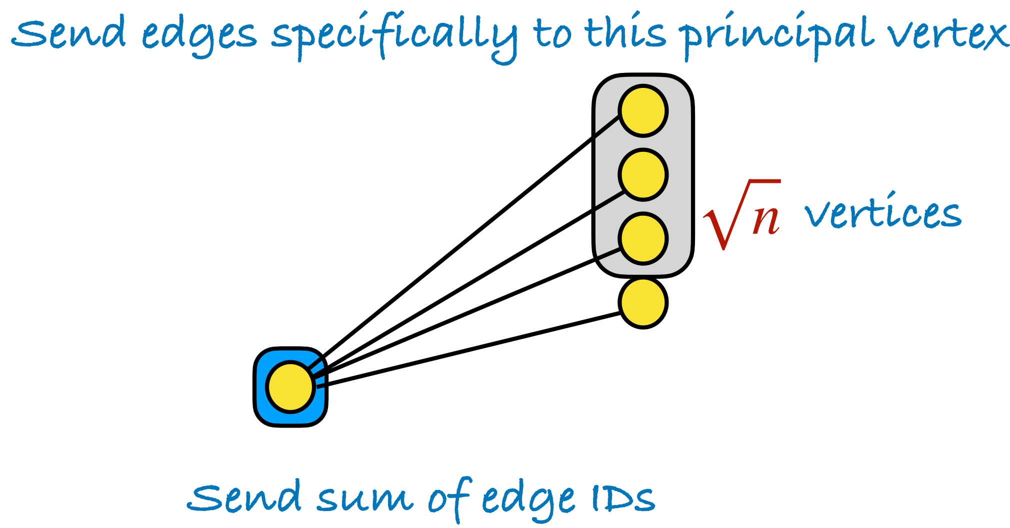
Warm-Up: One-Round Lower Bound

- Analysis?
- From the perspective of principal vertices:
- "On their own", sending $o(\sqrt{n})$ bits only reveals $o(1)$ bits about the special edges
- But fooling vertices's messages can change this



Warm-Up: One-Round Lower Bound

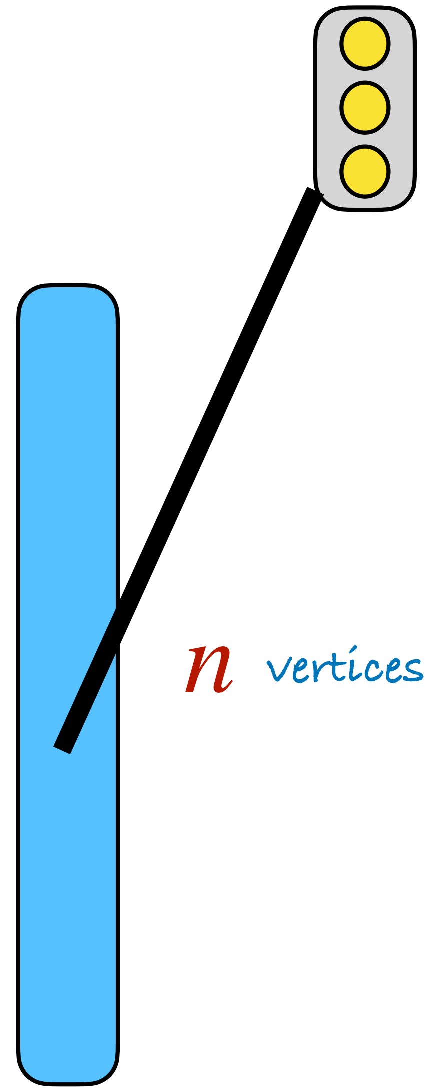
- Analysis?
- From the perspective of principal vertices:
- “On their own”, sending $o(\sqrt{n})$ bits only reveals $o(1)$ bits about the special edges
- But fooling vertices’s messages can change this



Warm-Up: One-Round Lower Bound

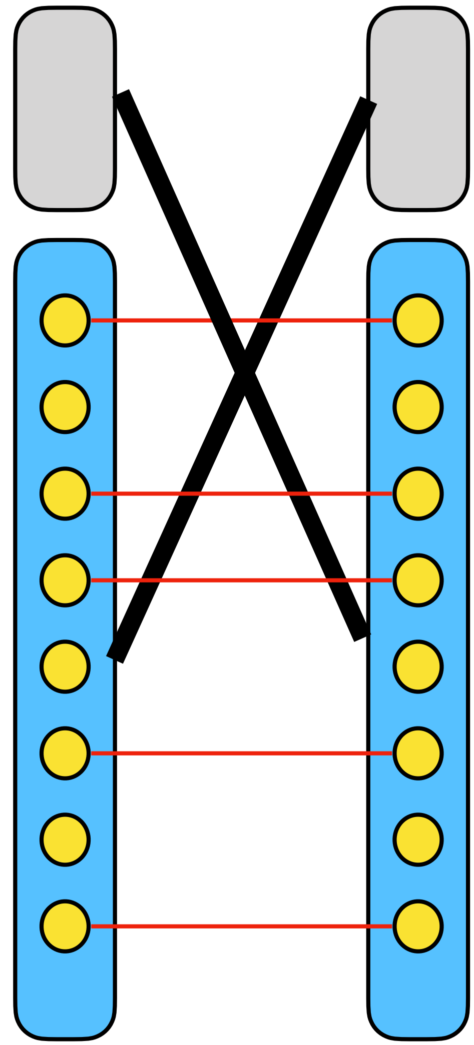
- Analysis?
- From the perspective of fooling vertices:
- $\sqrt{n}$ fooling vertices can reveal $o(n)$ information in total no matter what they know
- So we can "decompose" the revealed information between fooling and principal vertices

A similar idea was used first by [NY19,Y21] for spanning forest



Warm-Up: One-Round Lower Bound

- Analysis?
- The total information revealed about the special edges:

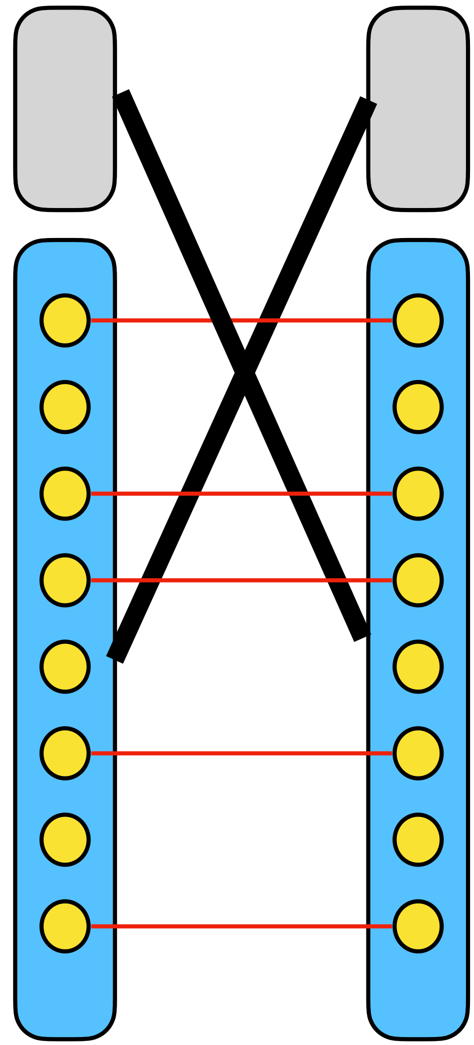


Warm-Up: One-Round Lower Bound

- Analysis?
- The total information revealed about the special edges:

$$\mathbb{I}(\textbf{special edges}; \textbf{msg}) \leqslant$$

$$n \cdot \frac{o(\sqrt{n})}{\sqrt{n}} + \sqrt{n} \cdot o(\sqrt{n}) = o(n)$$



Warm-Up: One-Round Lower Bound

- Analysis?

- The total information revealed about the special edges:

Shannon mutual information

$$\mathbb{I}(\textbf{special edges}; \textbf{msg}) \leqslant$$

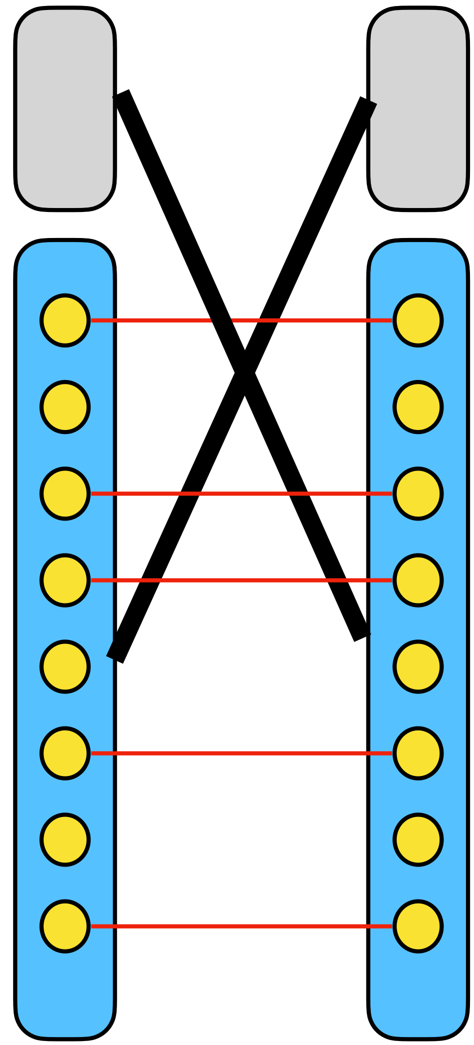
$$n \cdot \frac{o(\sqrt{n})}{\sqrt{n}} + \sqrt{n} \cdot o(\sqrt{n}) = o(n)$$

# of principal vertices

Their limited knowledge

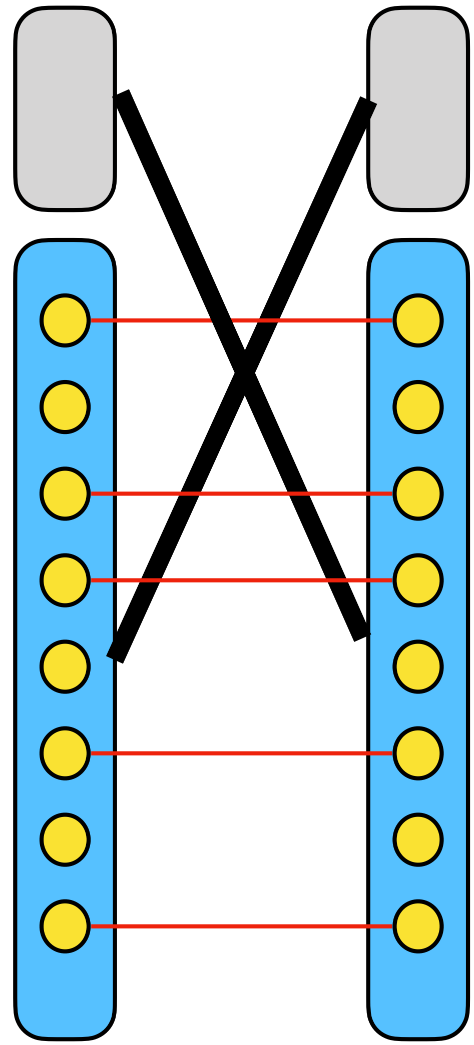
# of fooling vertices

Their limited bandwidth



Warm-Up: One-Round Lower Bound

- Analysis?
- The total information revealed about the special edges: $o(n)$ bits
- Not enough to reveal enough edges for the maximal matching



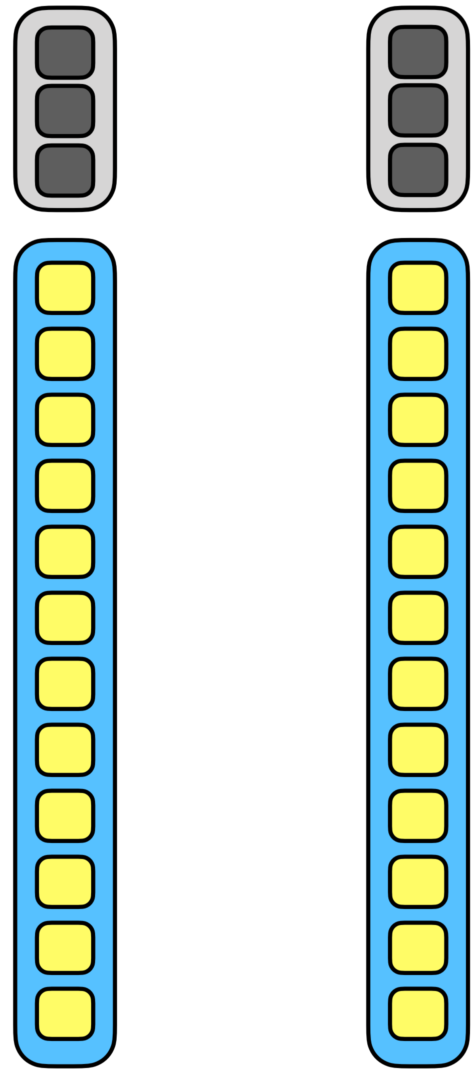
Warm-Up: One-Round Lower Bound

- Conclusion:

Any single-round protocol for maximal matching requires $\Omega(\sqrt{n})$ communication per-player in worst case

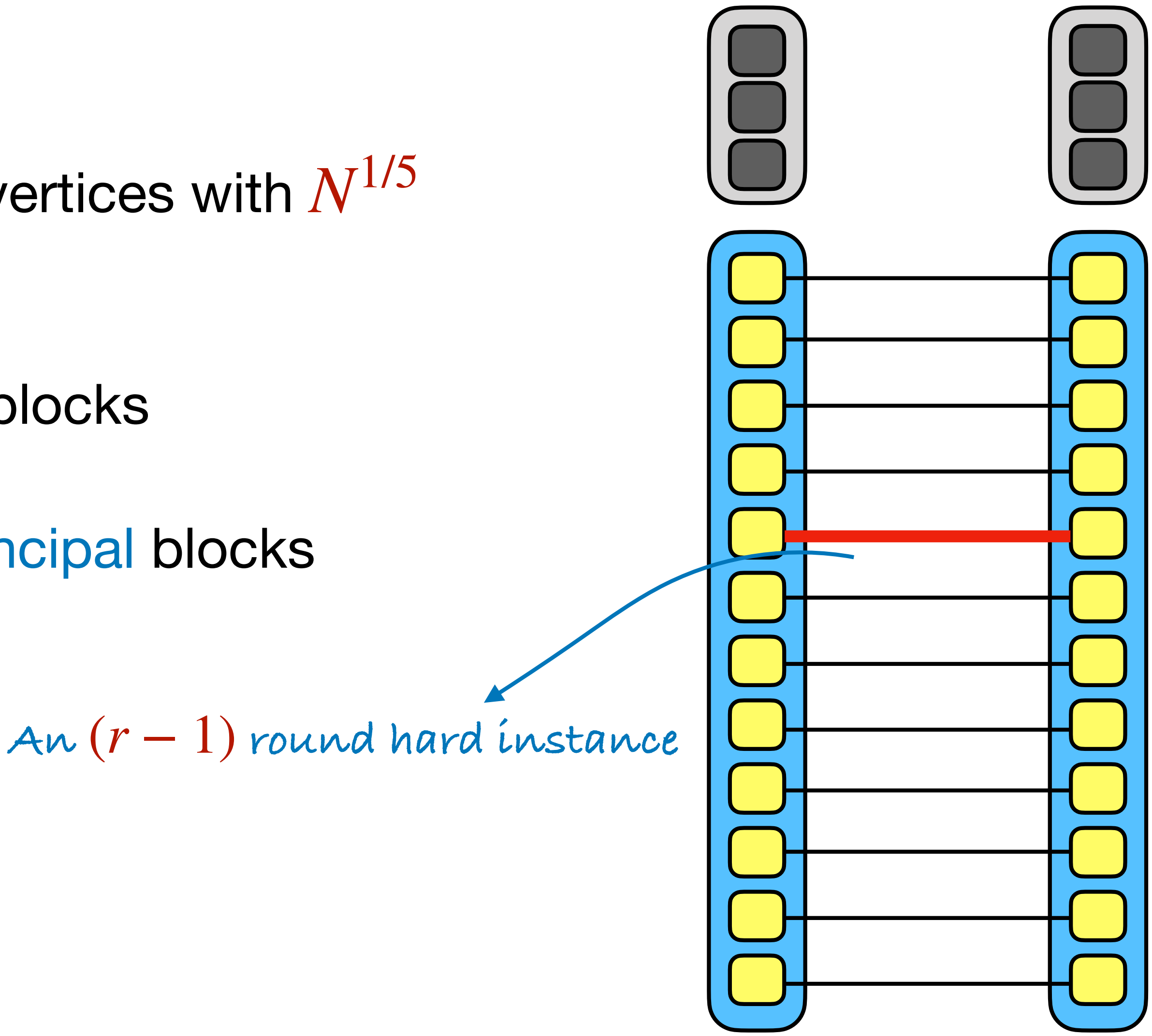
An r-Round Lower Bound

- On each side:
  - N blocks of vertices with $N^{1/5}$ vertices
  - $N^{2/5}$ fooling blocks
  - $N - N^{2/5}$ principal blocks



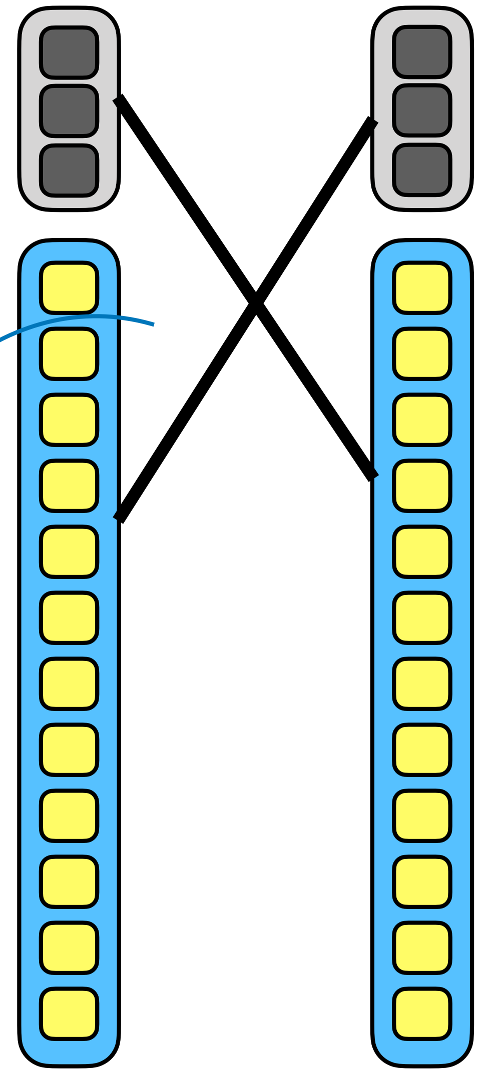
An r-Round Lower Bound

- On each side:
  - N blocks of vertices with $N^{1/5}$ vertices
  - $N^{2/5}$ fooling blocks
  - $N - N^{2/5}$ principal blocks



An r-Round Lower Bound

- On each side:
  - N blocks of vertices with $N^{1/5}$ vertices
  - $N^{2/5}$ fooling blocks
  - $N - N^{2/5}$ principal blocks

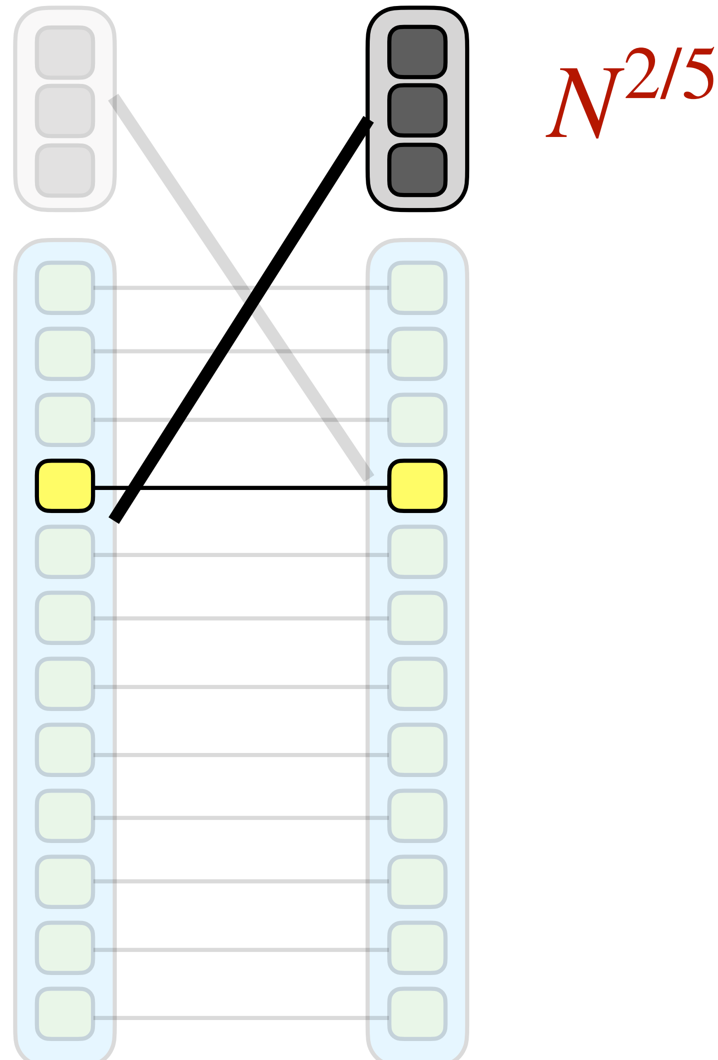


A fooling instance:
Marginally to each vertex looks
like an actual $(r-1)$ instance

Based on a brilliant idea introduced by [ANRW15]
for one-sided matching problem

An r-Round Lower Bound

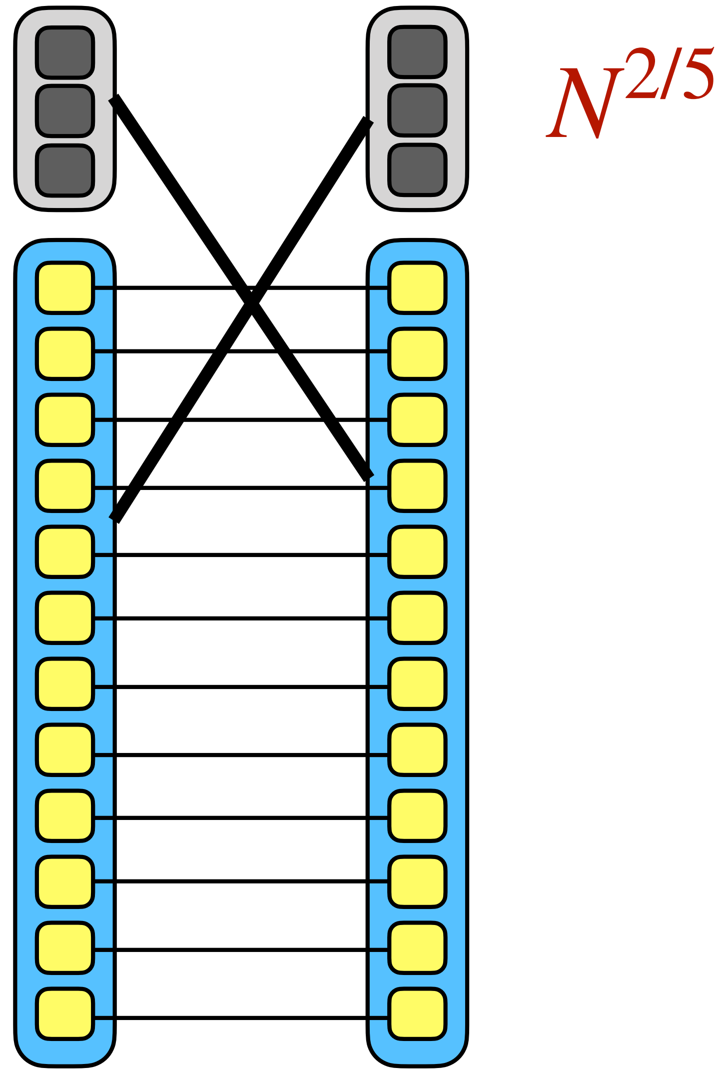
- From the perspective of principal blocks:
  - They play in $N^{2/5}$ $(r-1)$-round instances
  - They need to solve their special instance
- Obliviousness implies with $o(N^{1/5})$ communication per-player, only $o(1)$ bits is revealed about their special instance in the first round



An r-Round Lower Bound

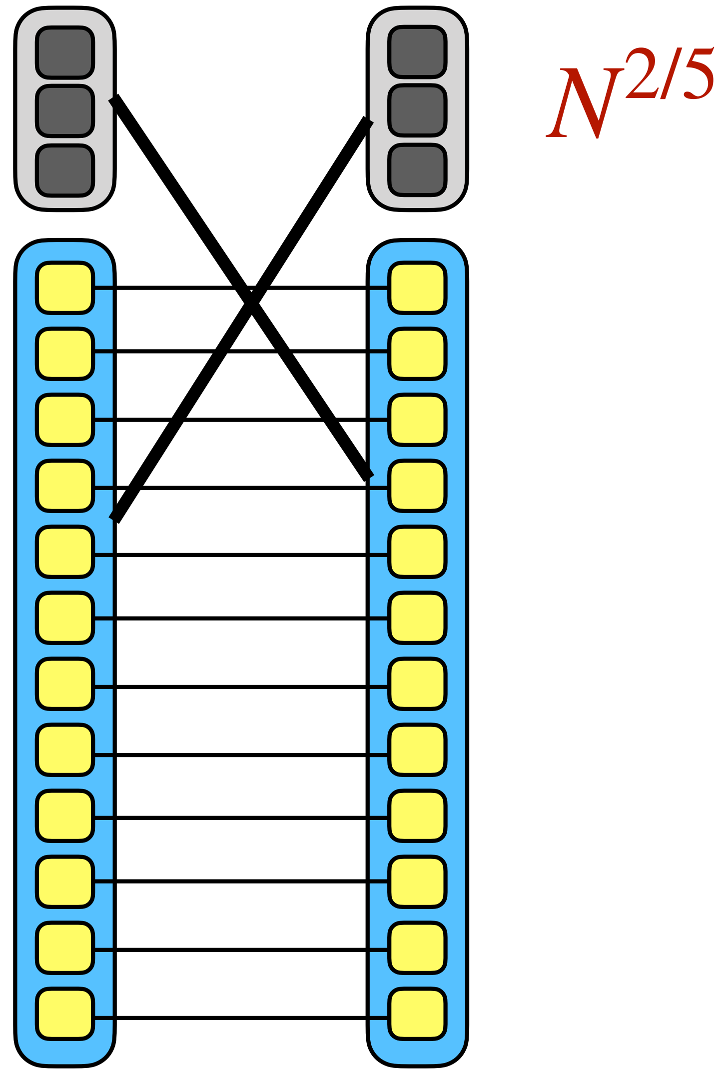
- From the perspective of fooling blocks:

- Limited bandwidth implies with $o(N^{1/5})$ communication per-player, they can only reveal $o(N^{4/5})$ bits in total in the first round



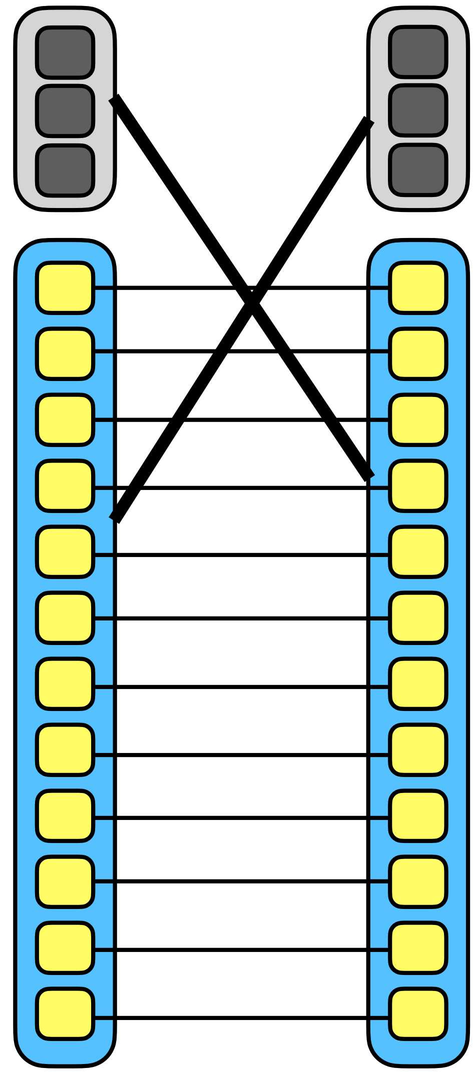
An r-Round Lower Bound

- So, after the first round:
- $o(1)$ bits is revealed about most principal $(r-1)$ round instances
- So, the distributions of these instances remain almost the same even after the first message
- But the players now only have $(r-1)$ rounds to solve these instances



An r-Round Lower Bound

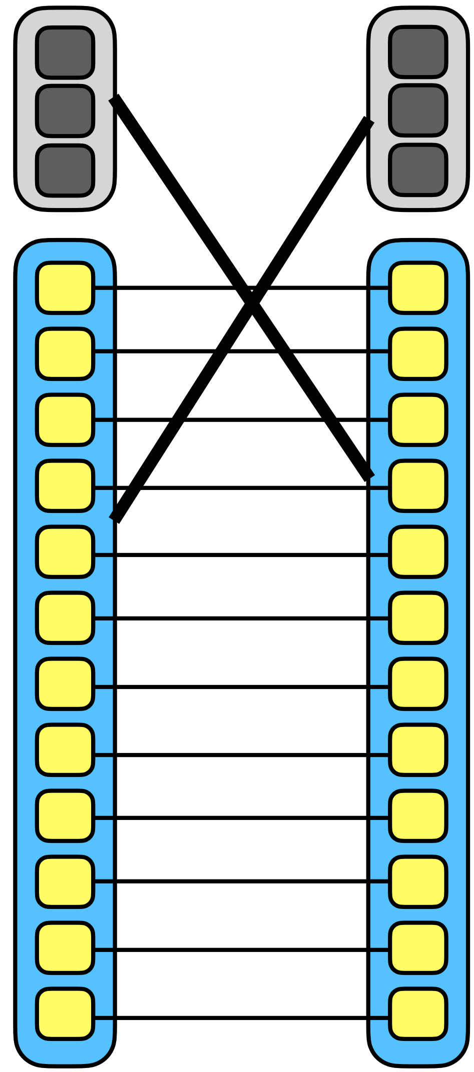
- The standard approach at this point: round elimination



$N^{2/5}$

An r-Round Lower Bound

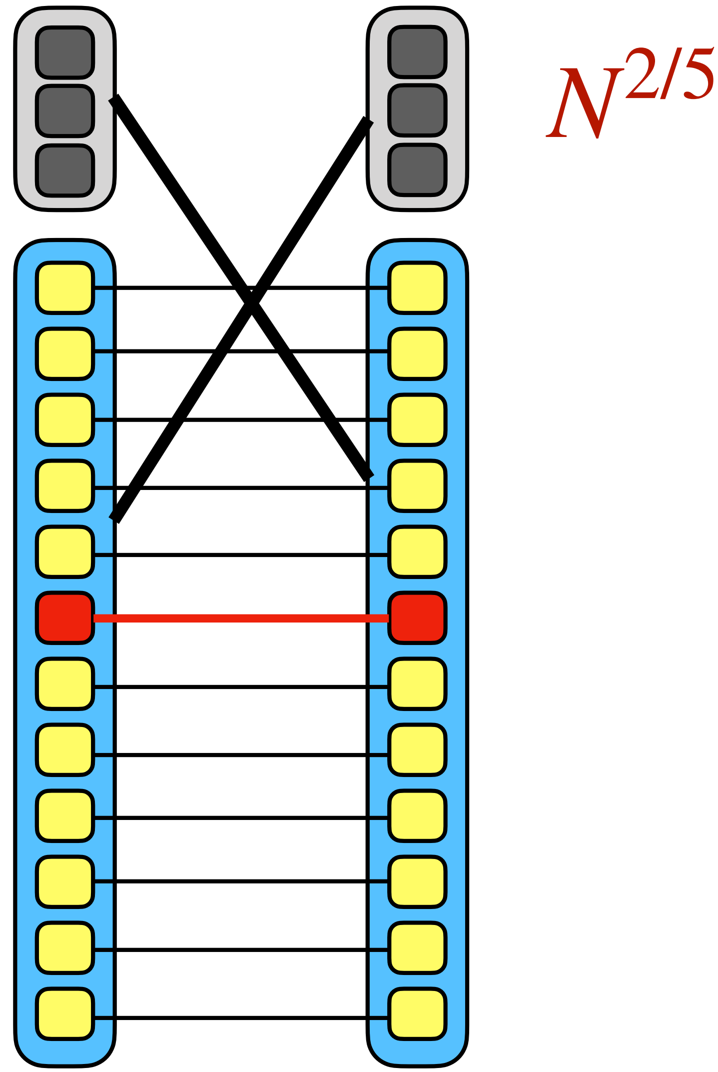
- The standard approach at this point: round elimination
- This is "communication complexity round elimination" as opposed to "LOCAL round elimination"
  - [Duris, Galil, and Schnitger; STOC 1984]
  - [Nisan, Wigderson; STOC 1991]



$N^{2/5}$

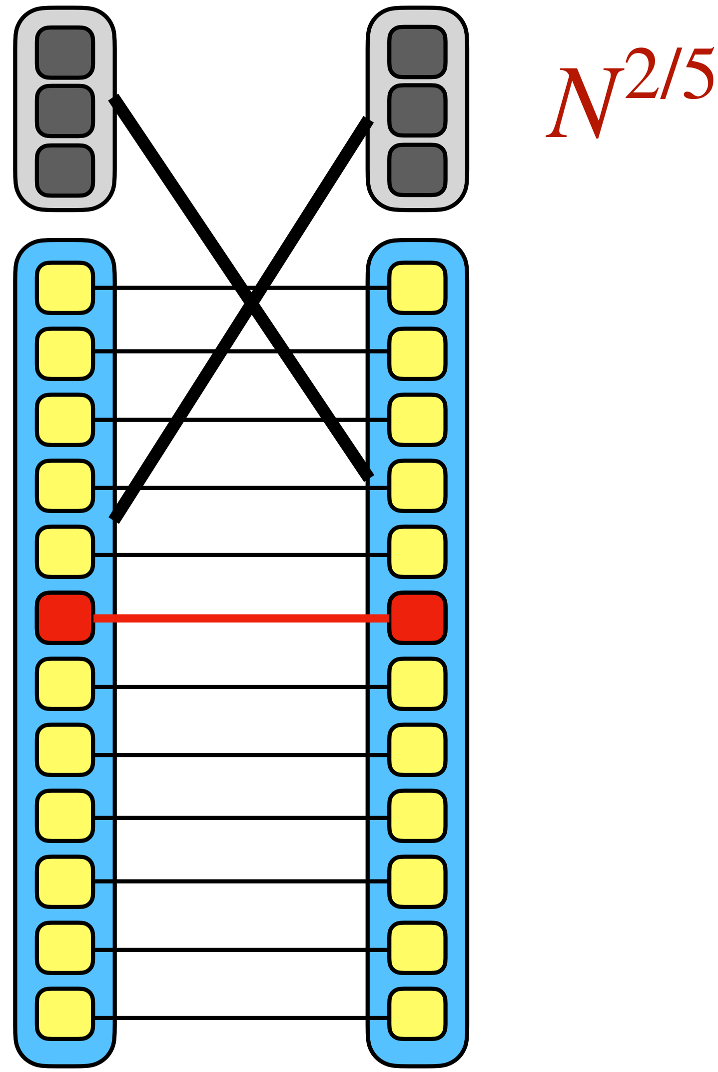
An r-Round Lower Bound

- Use any r-round protocol to get an $(r-1)$-protocol:
  - Embed the $(r-1)$-round instance in a random principal instance
  - Sample the first message of the protocol without communication
  - Run the protocol from its second-round onwards
  - Simulate the remaining players



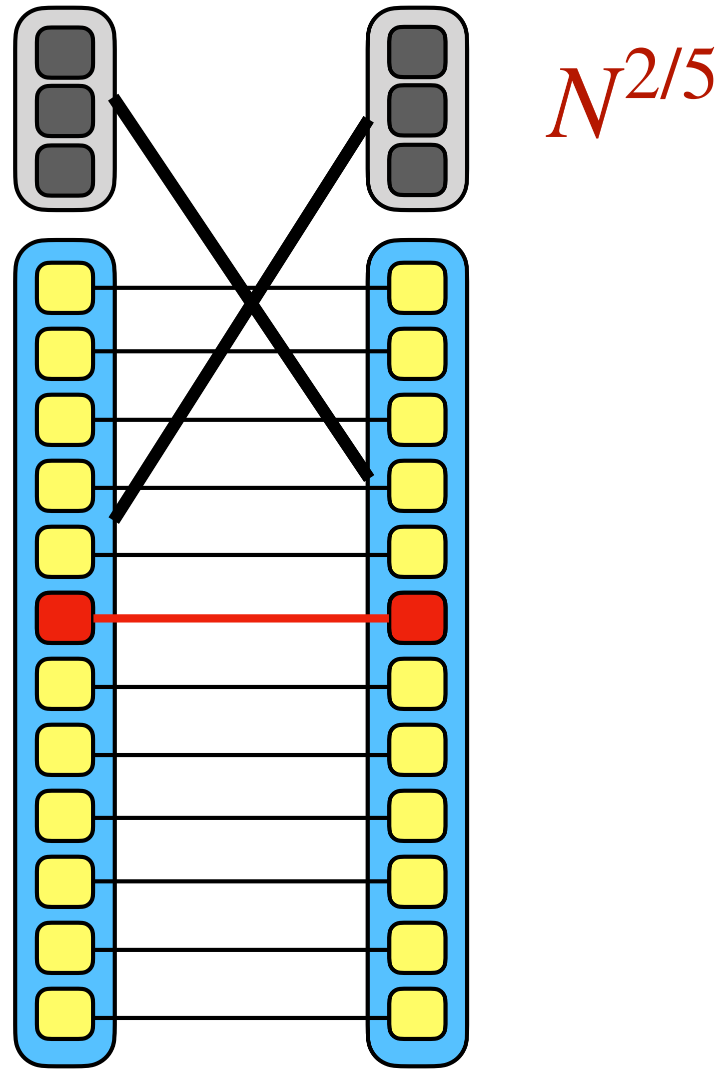
An r-Round Lower Bound

- This step is quite technical because of persistence of fooling blocks across subsequent rounds



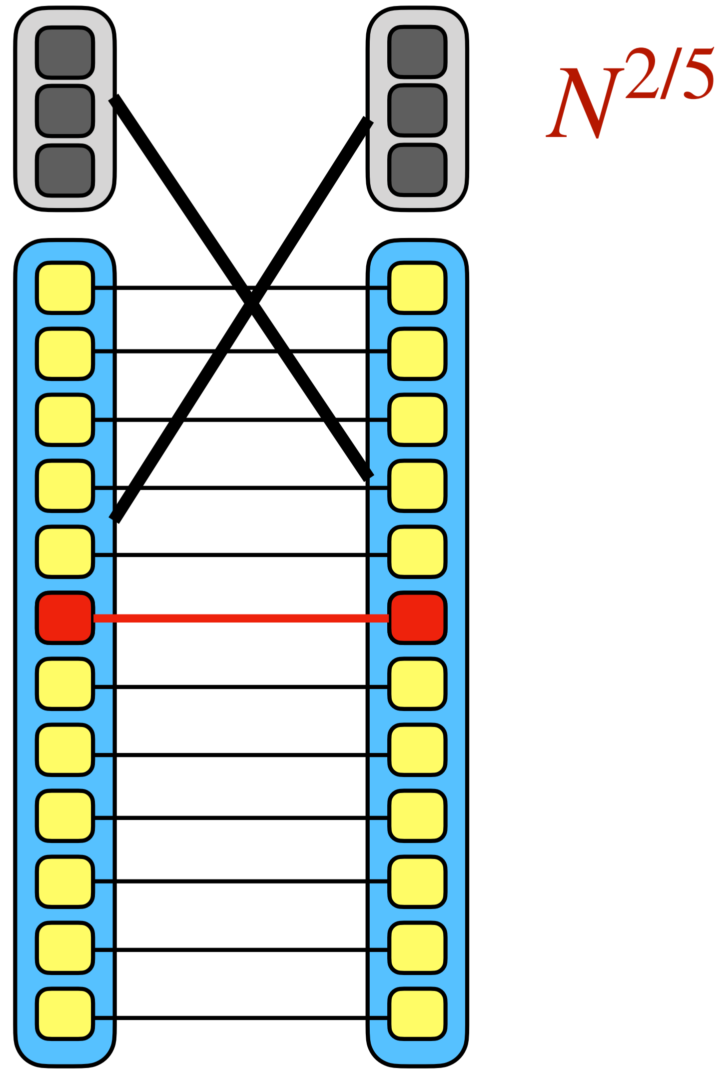
An r-Round Lower Bound

- This step is quite technical because of persistence of fooling blocks across subsequent rounds
  - They continue “injecting noise” throughout the protocol, not only in the first round



An r-Round Lower Bound

- This step is quite technical because of persistence of fooling blocks across subsequent rounds
  - They continue "injecting noise" throughout the protocol, not only in the first round
- **Main novelty**: a new round elimination approach based on a non-simultaneous simulation of other players via only their messages not their inputs



An r-Round Lower Bound

- Conclusion:

Any r-round protocol for maximal matching requires $\Omega(n^{1/10^{r+1}})$ communication per-player in worst case

Concluding Remarks

Concluding Remarks

- We can prove "some" multi-round lower bounds for round-sensitive problems

- Two key techniques:

  - A novel round elimination via simulating players through their messages and not their inputs

  - Core component: break the dependency of edge sharing by limiting the number of "fooling vertices"

    - There are "few" players whose removal makes proving the rest of the argument an "easy" number-in-hand communication argument

Concluding Remarks

- We can prove "some" multi-round lower bounds for round-sensitive problems

- We are quite far from "right" answers/techniques yet!

  - There are "few" players whose removal makes proving the rest of the argument an "easy" number-in-hand communication argument

Concluding Remarks

They don't seem tight even for one-round protocols

- We can prove "some" multi-round lower bounds for round-sensitive problems

This is a lossy argument in general (both here and elsewhere)

- Two key techniques:
  - A novel round elimination via simulating players through their messages and not their inputs
  - Core component: break the dependency of edge sharing by limiting the number of "fooling vertices"
    - There are "few" players whose removal makes proving the rest of the argument an "easy" number-in-hand communication argument

This is a very crude argument: we have no "handle" on the type of information revealed by these vertices only it size

Open Problems

- "Easier" immediate open questions:
  - Extend [**A**KZ22] lower bound to average communication (to match [GGKMR18] protocol)
  - Prove $\Omega(n)$ one-round lower bounds for MIS and matching (to match trivial bounds)

Open Problems

And/or extend their protocol to worst-case communication

- "Easier" immediate open questions:
  - Extend [**A**KZ22] lower bound to average communication (to match [GGKMR18] protocol)
  - Prove $\Omega(n)$ one-round lower bounds for MIS and matching (to match trivial bounds)

Open Problems

And/or extend their protocol to worst-case communication

- "Easier" immediate open questions:
  - Extend [AKZ22] lower bound to average communication (to match [GGKMR18] protocol)
  - Prove $\Omega(n)$ one-round lower bounds for MIS and matching (to match trivial bounds)
- "Harder" open questions:
  - Prove strong lower bounds for shortest path even in one round
  - Prove $\approx \Omega(\log n)$ round lower bounds for matching
  - Prove $n^{\Omega(1)}$ lower bound for deterministic connectivity in one round

Open Problems

And/or extend their protocol to worst-case communication

- "Easier" immediate open questions:
  - Extend [AKZ22] lower bound to average communication (to match [GGKMR18] protocol)
  - Prove $\Omega(n)$ one-round lower bounds for MIS and matching (to match trivial bounds)
- "Harder" open questions:
  - Prove strong lower bounds for shortest path even in one round
  - Prove $\approx \Omega(\log n)$ round lower bounds for matching
  - Prove $n^{\Omega(1)}$ lower bound for deterministic connectivity in one round

At this point, a longstanding open question

Open Problems

And/or extend their protocol to worst-case communication

- "Easier" immediate open questions:
  - Extend [AKZ22] lower bound to average communication (to match [GGKMR18] protocol)
  - Prove $\Omega(n)$ one-round lower bounds for MIS and matching (to match trivial bounds)
- "Harder" open questions:
  - Prove strong lower bounds for shortest path even in one round
  - Prove $\approx \Omega(\log n)$ round lower bounds for matching
  - Prove $n^{\Omega(1)}$ lower bound for deterministic connectivity in one round *At this point, a longstanding open question*
- "Conceptual" open questions:
  - Characterize protocols in this model: are they all linear sketches?
  - Connections between "LOCAL" vs "communication" round elimination?

Open Problems

- "Easier" immediate open questions:
  - Extend [AKZ22] lower bound to average communication (to match [GGKMR18] protocol)

    And/or extend their protocol to worst-case communication
  - Prove $\Omega(n)$ one-round lower bounds for MIS and matching (to match trivial bounds)
- "Harder" open questions:
  - Prove strong lower bounds for shortest path even in one round
  - Prove $\approx \Omega(\log n)$ round lower bounds for matching
  - Prove $n^{\Omega(1)}$ lower bound for deterministic connectivity in one round

    At this point, a longstanding open question
- "Conceptual" open questions:
  - Characterize protocols in this model: are they all linear sketches?
  - Connections between "LOCAL" vs "communication" round elimination?

Thank you!