Distributed Verification and Hardness of Distributed Approximation 2

Danupon Nanongkai
KTH

Based on

Distributed Verification and Hardness of Distributed Approximation, STOC 2011 & SICOMP 2012,

with Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Gopal Pandurangan, David Peleg, Roger Wattenhofer

Recap from last time

Direct Equality Verification lower bound $\Omega(b)$

Part 3.3

Distributed Equality Verification lower bound $\Omega(n^{1/2})$

Well-known result in communication complexity

By the Simulation theorem

Part 3.2

$\Omega(b)$

ST verification lower bound $\Omega(n^{1/2})$

Part 3.1

Approx MST lower bound $\Omega(n^{1/2})$

Notes

-The lower bounds hold on graphs of diameter **D=O(log n)**

- For simplicity, we will consider only **D=O(n$^{1/4}$)**

Simulation Theorem

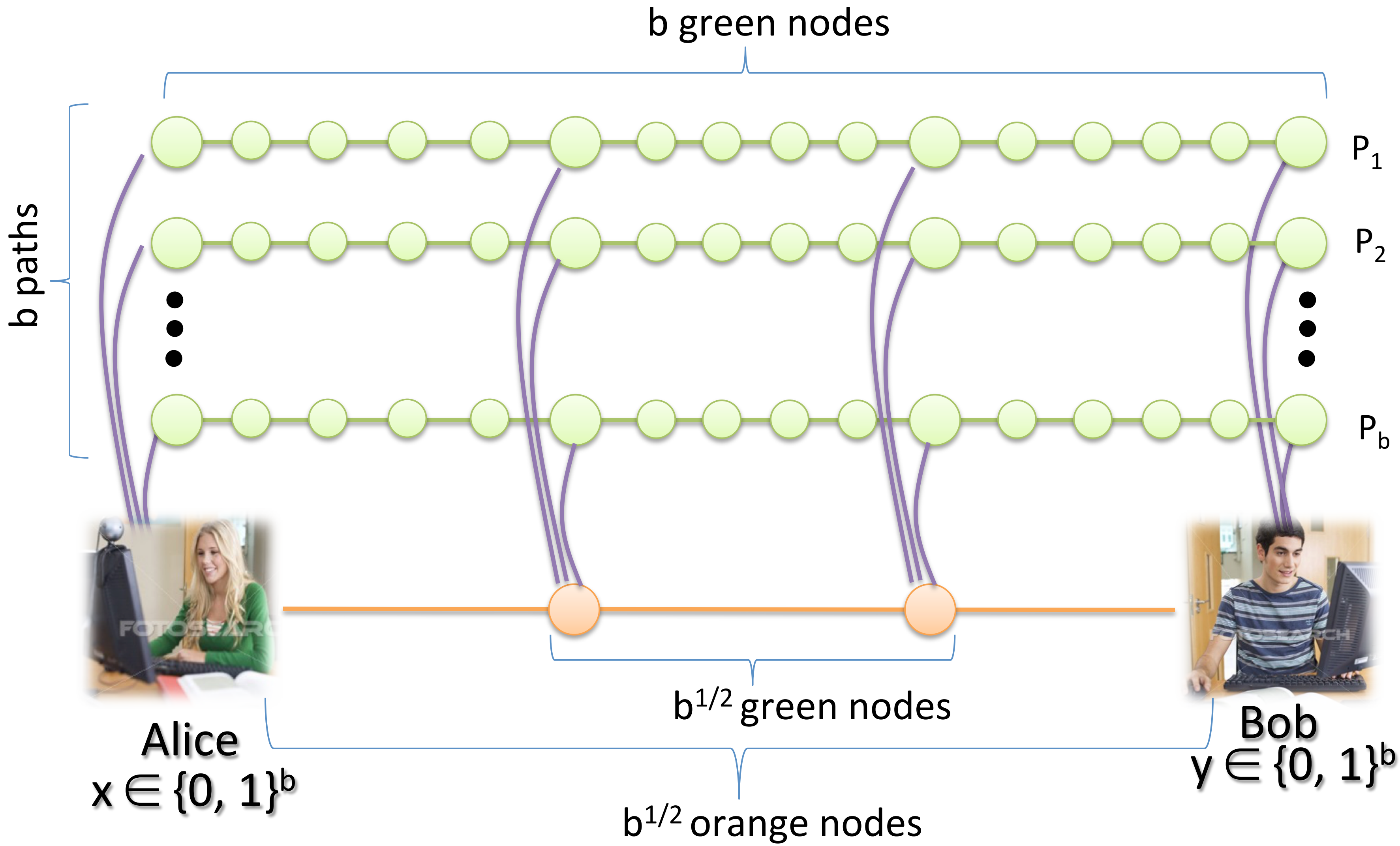
If the **distributed** equality verification can be solved in **T** days, for any **T ≤ b/2**, then the **direct** version can be solved in **≤T** days

time: **T < b/2**

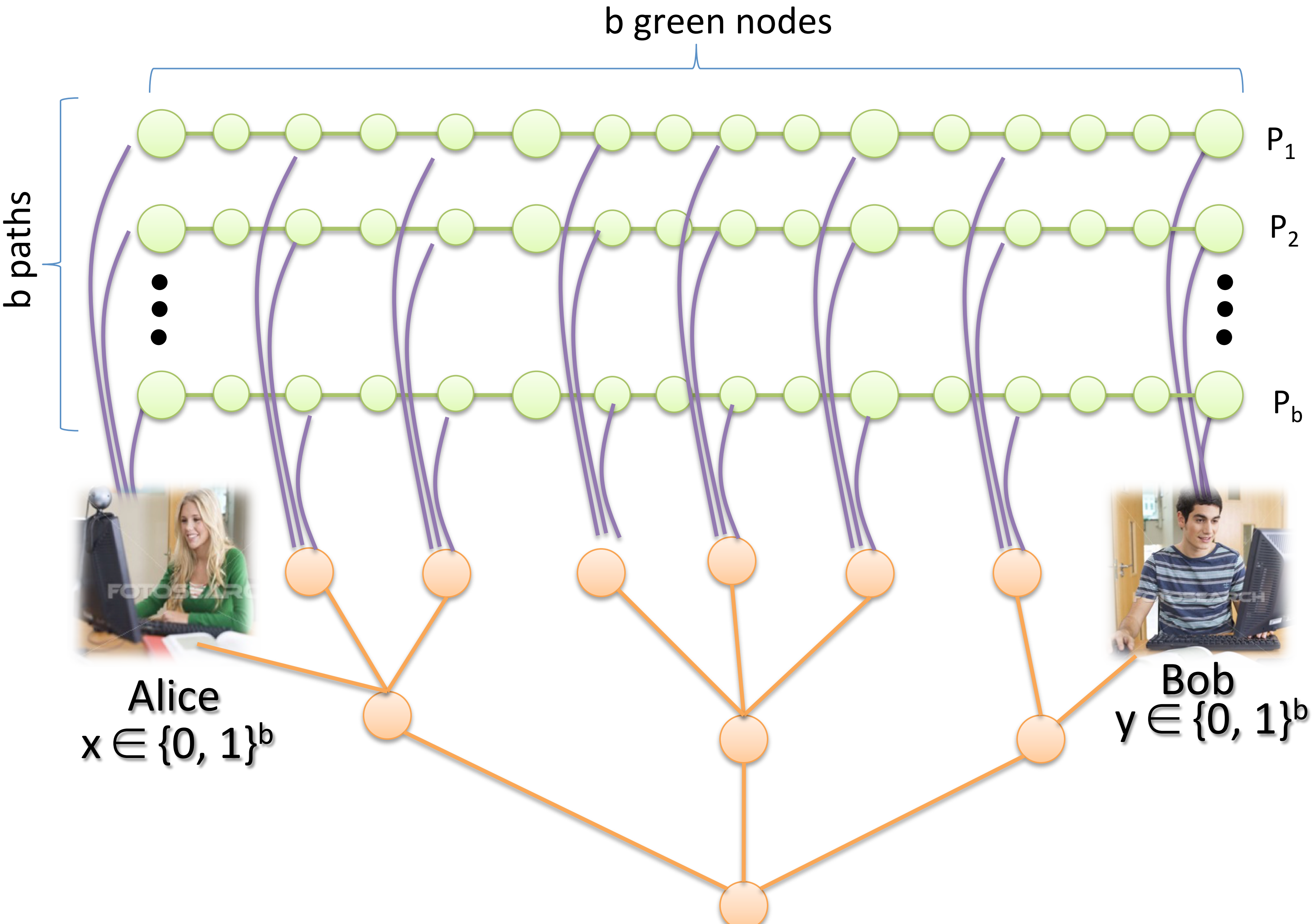
time = **T<b/2**
known: need **≥b**
Contradiction!

Graph G(b) has diameter $\mathbf{n^{1/4}}$

We can use a similar analysis on some graphs of diameter **O(log n)**



b green nodes
b paths
P_1
P_2
P_b
Alice
$x \in \{0, 1\}^b$
Bob
$y \in \{0, 1\}^b$
$b^{1/2}$ green nodes
$b^{1/2}$ orange nodes


b green nodes
b paths
P1
P2
Pb
Alice
x ∈ {0, 1}^b
Bob
y ∈ {0, 1}^b

We are done

with deterministic algorithms

How about randomized algorithms?

Today: Extensions

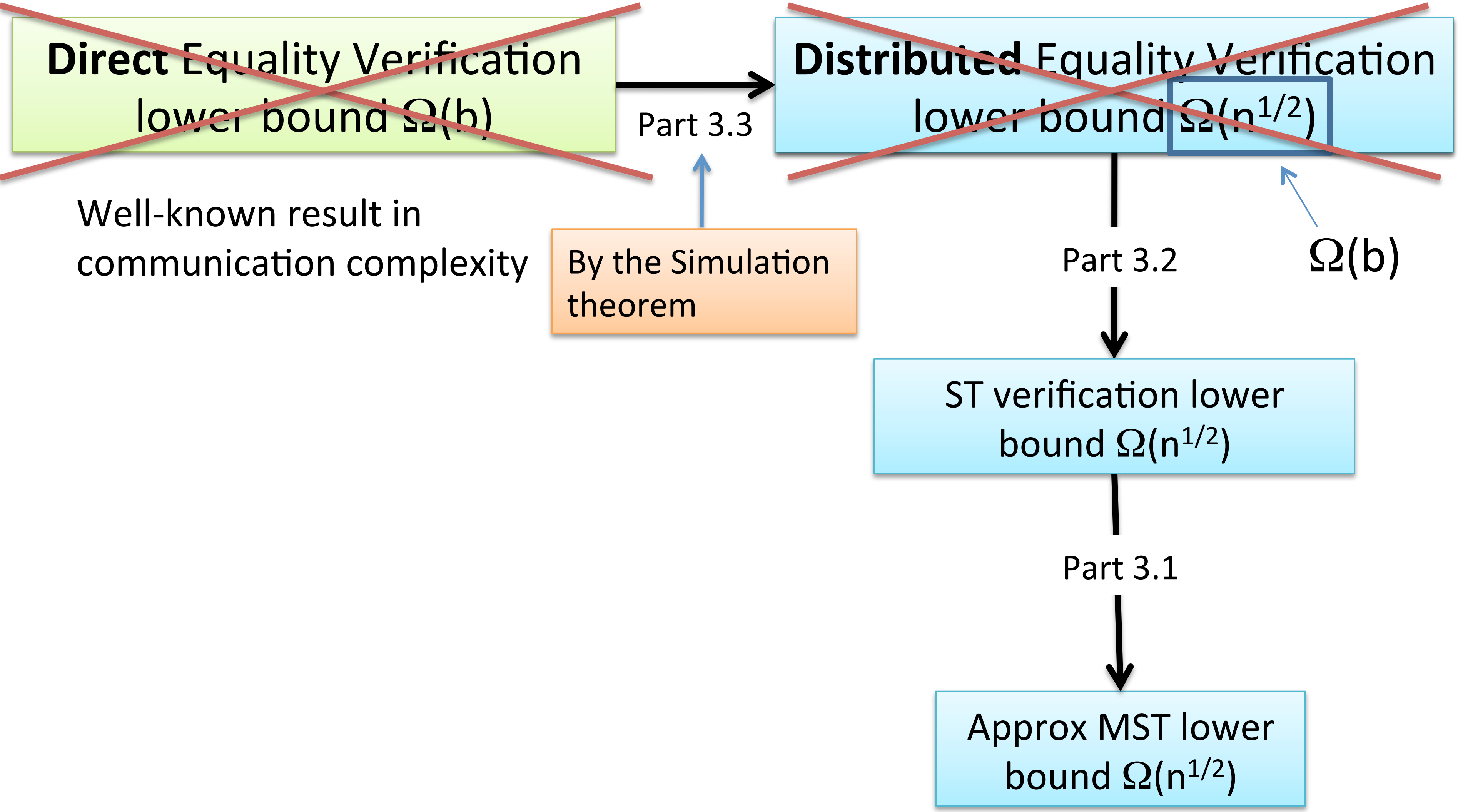
- Extension to lower bounds for randomized algorithms
- Follow-up works since 2011 + open research questions
- Extension to round-efficient Simulation Theorem
- Extension to lower bounds for quantum algorithms

Part 1

Extension to lower bounds for **randomized** algorithms

Bad news

Direct and distributed equality can be verified in **O(log b)** time by a randomized algorithm


Direct Equality Verification lower bound Ω(b)
Distributed Equality Verification lower bound Ω(n^1/2)
Part 3.3
Well-known result in communication complexity
By the Simulation theorem
Part 3.2
Ω(b)
ST verification lower bound Ω(n^1/2)
Part 3.1
Approx MST lower bound Ω(n^1/2)

Good news

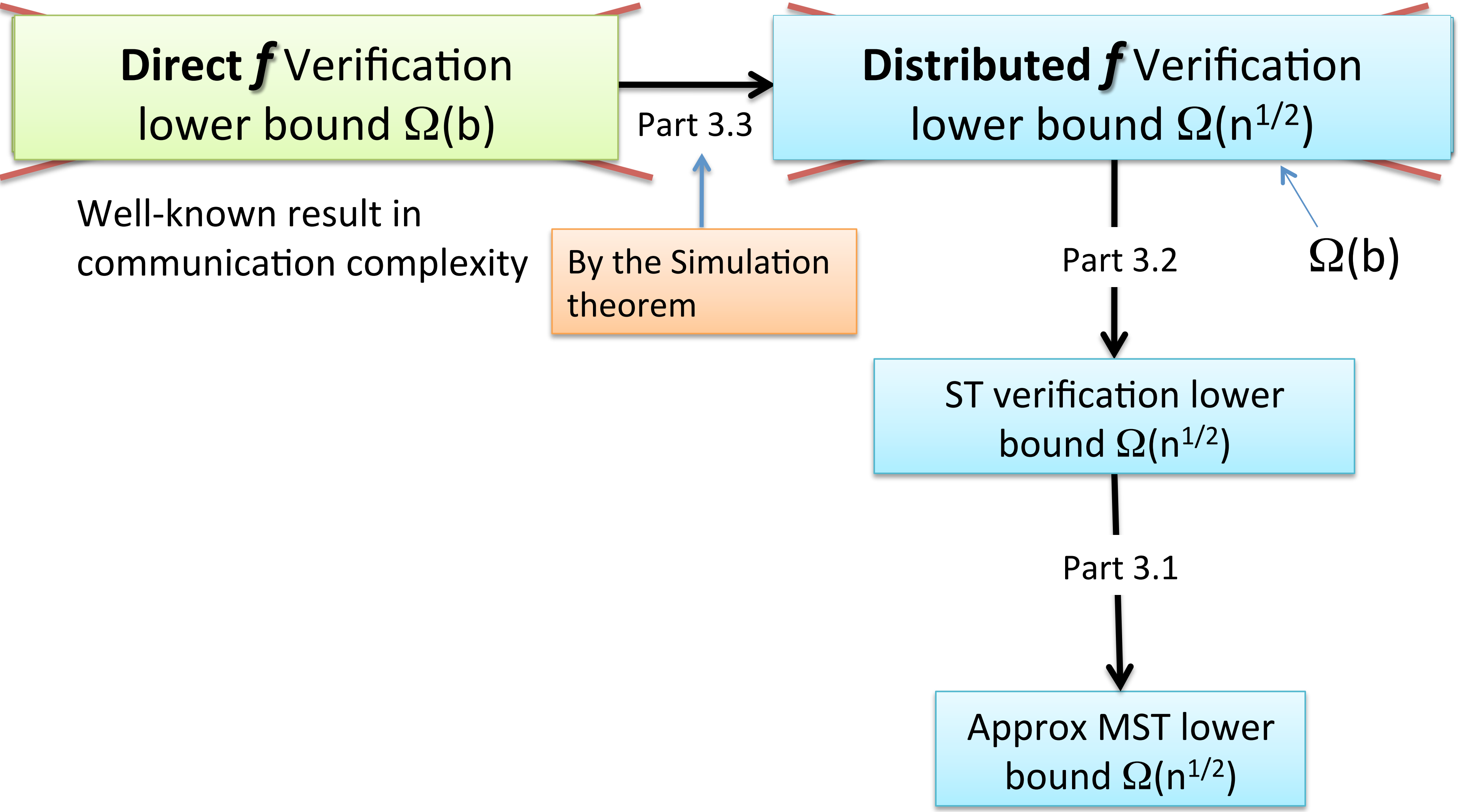
The simulation theorem is true for *any* function ***f***

(and for randomized algorithms)

Simulation Theorem

If ***f*** can be computed distributively in **T** days, for any **T ≤ (path length)/2**, then the communication complexity of ***f*** is **≤T**

Proof Alice and Bob can simulate any distributed algorithm for **b/2** days with one bit exchanged per day.


Direct f Verification lower bound Ω(b)
Part 3.3
Distributed f Verification lower bound Ω(n^{1/2})
Well-known result in communication complexity
By the Simulation theorem
Part 3.2
Ω(b)
ST verification lower bound Ω(n^{1/2})
Part 3.1
Approx MST lower bound Ω(n^{1/2})

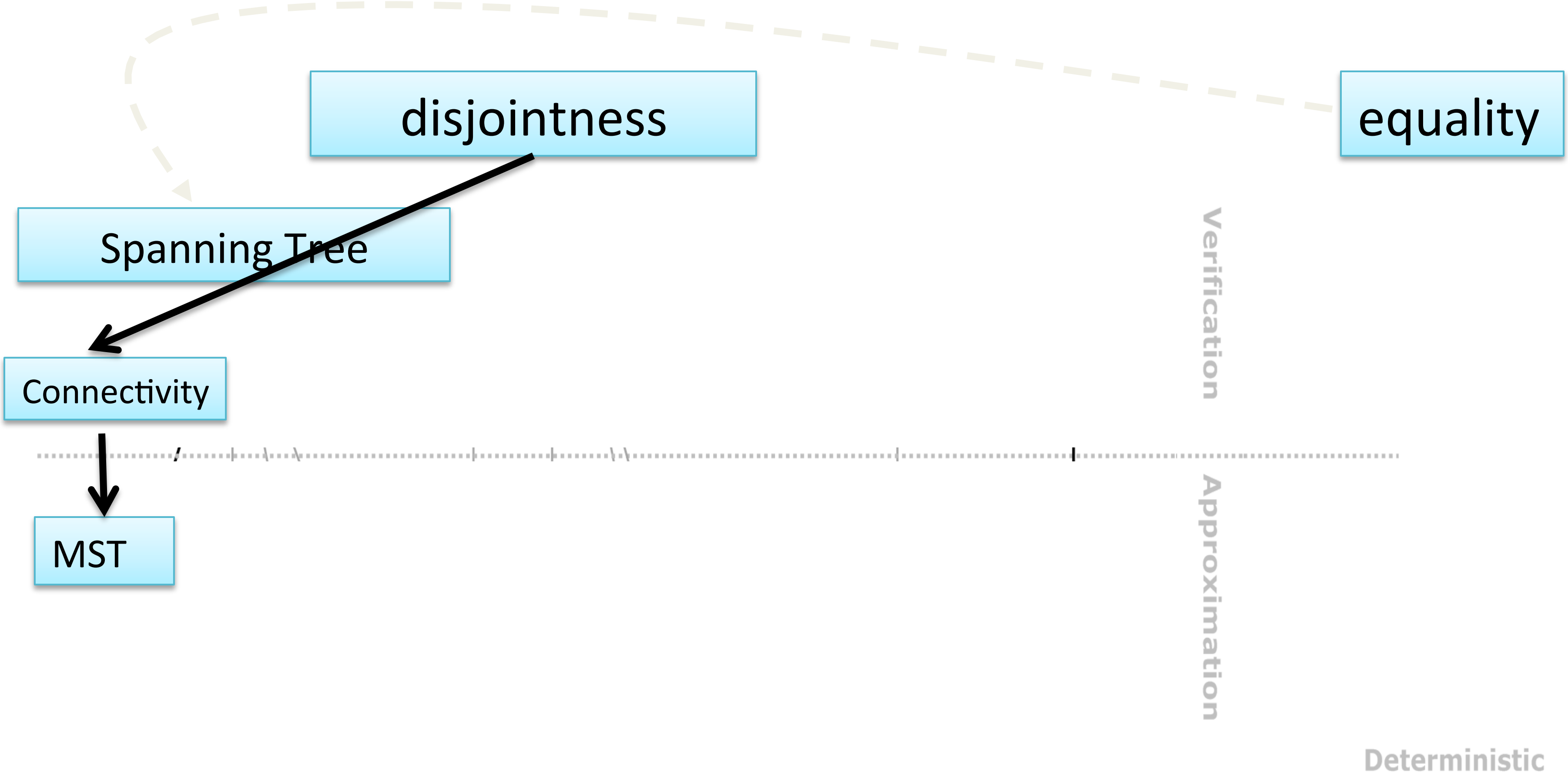
We will use

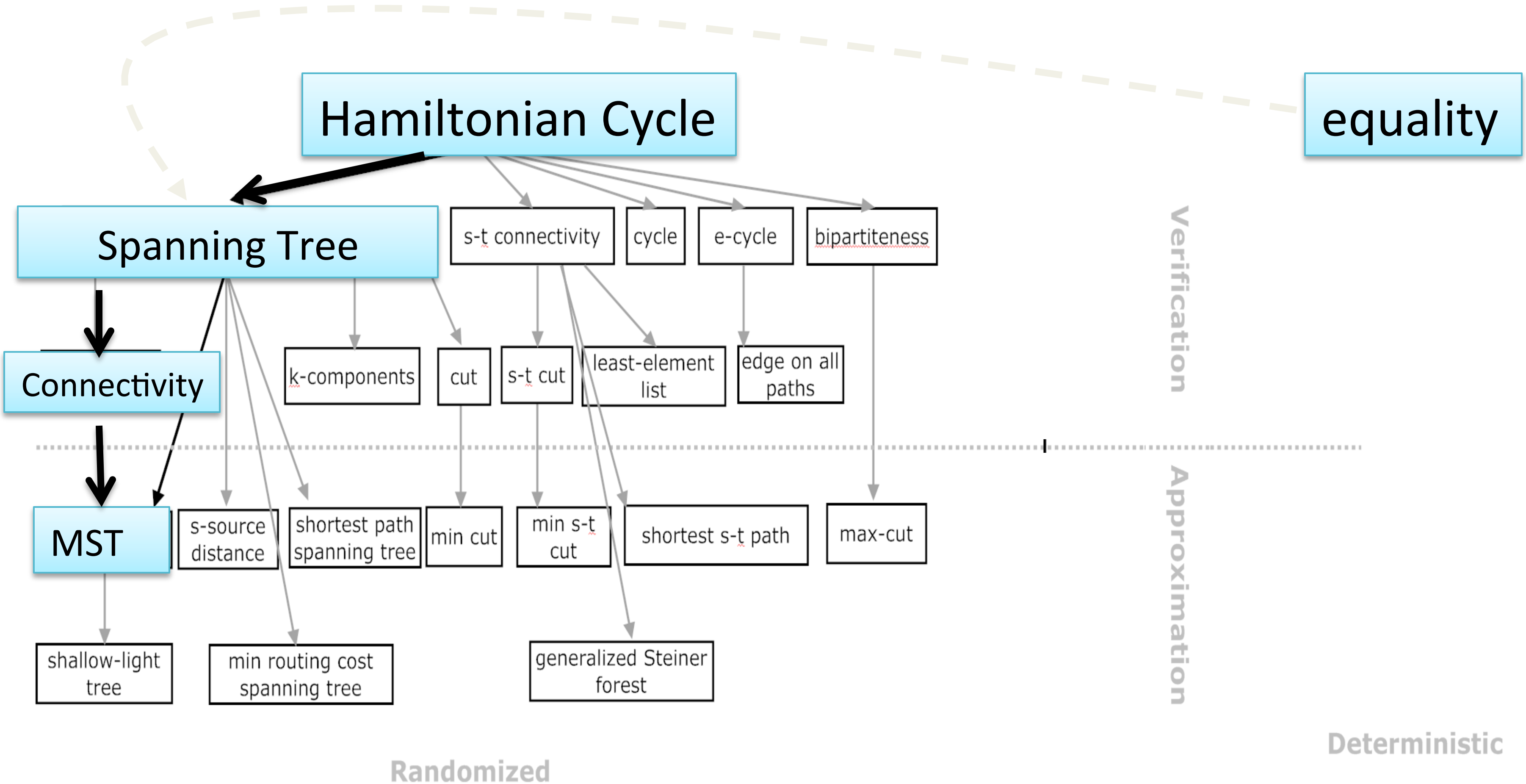
f_1 = "disjointness" function

f_2 = "Hamiltonian cycle" function

(Randomized lower bound = Ω(b))

(f$_2$ gives slightly better results)


disjointness
equality
Spanning Tree
Connectivity
MST
Verification
Approximation
Deterministic



To prove lower bound of Hamiltonian Cycle, we need the *IPmod3* problem

Part 1.1

Disjointness

Two parties are sharing a stadium

A(rgentina)

B(razil)

They want to have a disjoint schedule

June 2014							
Wk	Mo	Tu	We	Th	Fr	Sa	Su
22							1
23	2	3	4	5	6	7	8
24	9	10	11	12	13	14	15
25	16	17	18	19	20	21	22
26	23	24	25	26	27	28	29
27	30						

June 2014							
Wk	Mo	Tu	We	Th	Fr	Sa	Su
22							1
23	2	3	4	5	6	7	8
24	9	10	11	12	13	14	15
25	16	17	18	19	20	21	22
26	23	24	25	26	27	28	29
27	30						

There are two players, Alice and Bob

Alice

Bob

Each player received some numbers (e.g. dates of their matches)

Alice

1 4 7 ...

Bob

2 6 7 ...

Did they receive the same number?

Alice

1 4 7 …

Bob

2 6 7 …

Did they receive the same number?

Yes

Yes

Alice

1 4 7 …

Bob

2 6 7 …

Disjointness (more formally)

- Alice gets $x=\{0, 1\}^b$, Bob gets $y=\{0, 1\}^b$
- Wants to know $\langle x, y \rangle = 0$ or not, where $\langle x, y \rangle$ is the inner product
- Lower bound: $\Omega(b)$

Direct disjointness lower bound $\Omega(b)$ (randomized)

→

Distributed disjointness lower bound $\Omega(b) = \Omega(n^{1/2})$

By the Simulation theorem

Exercise

↓

Connectivity verification lower bound $\Omega(n^{1/2})$

↓

Approx MST lower bound $\Omega(n^{1/2})$

Connectivity verification problem

- Verify if the subgraph H is a connected graph that spans all nodes in the network

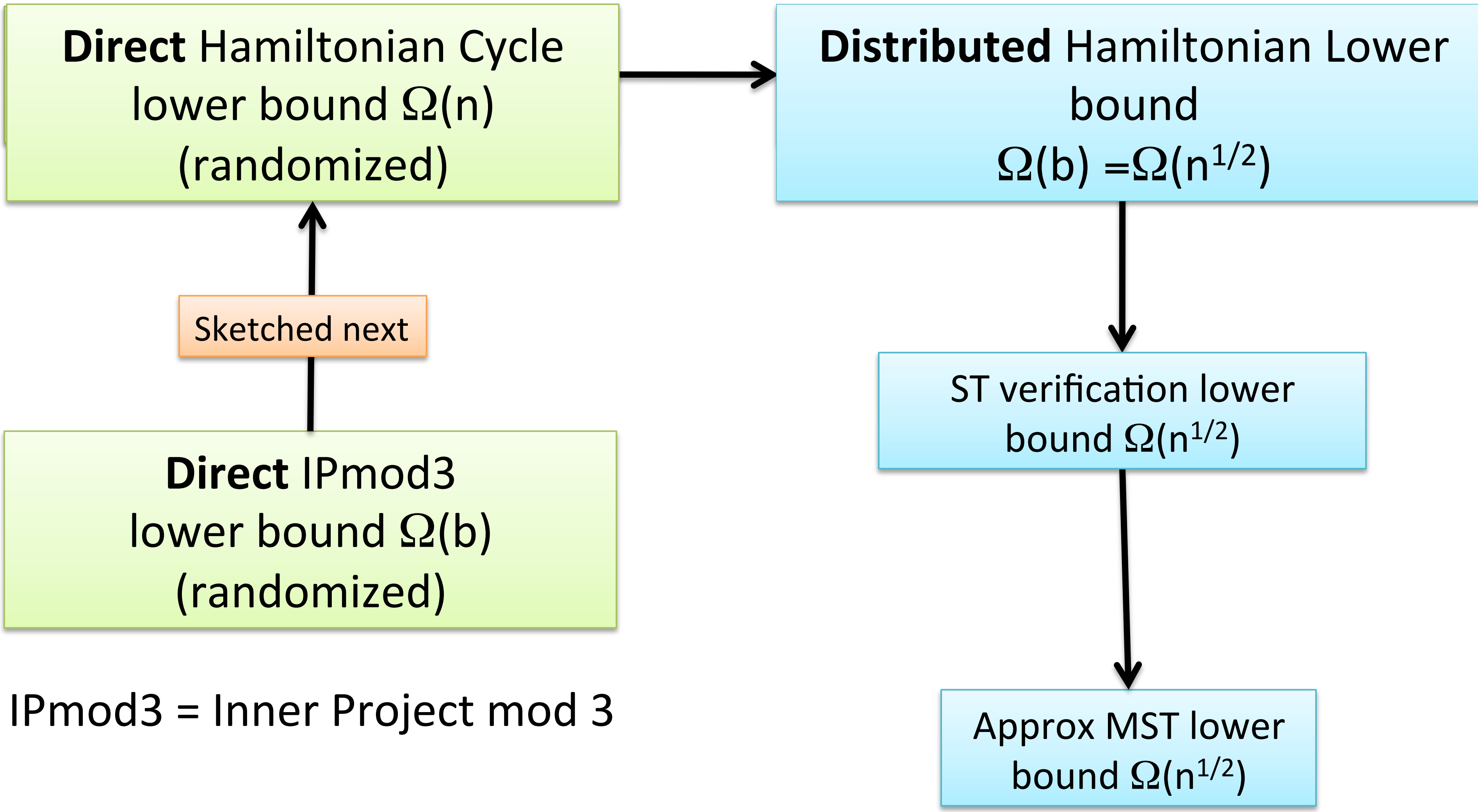
(We actually call this "spanning connected subgraph problem")

Part 1.2

Hamiltonian Cycle

Hamiltonian cycle problem

- Alice gets (V, E_1), Bob gets (V, E_2).
- Wants to know $G=(V, E_2 \cup E_1)$ is a Hamiltonian cycle or not, i.e. whether it is a cycle that includes all nodes
- Lower bound: $\Omega(|V|)$



IPmod3 = Inner Project mod 3

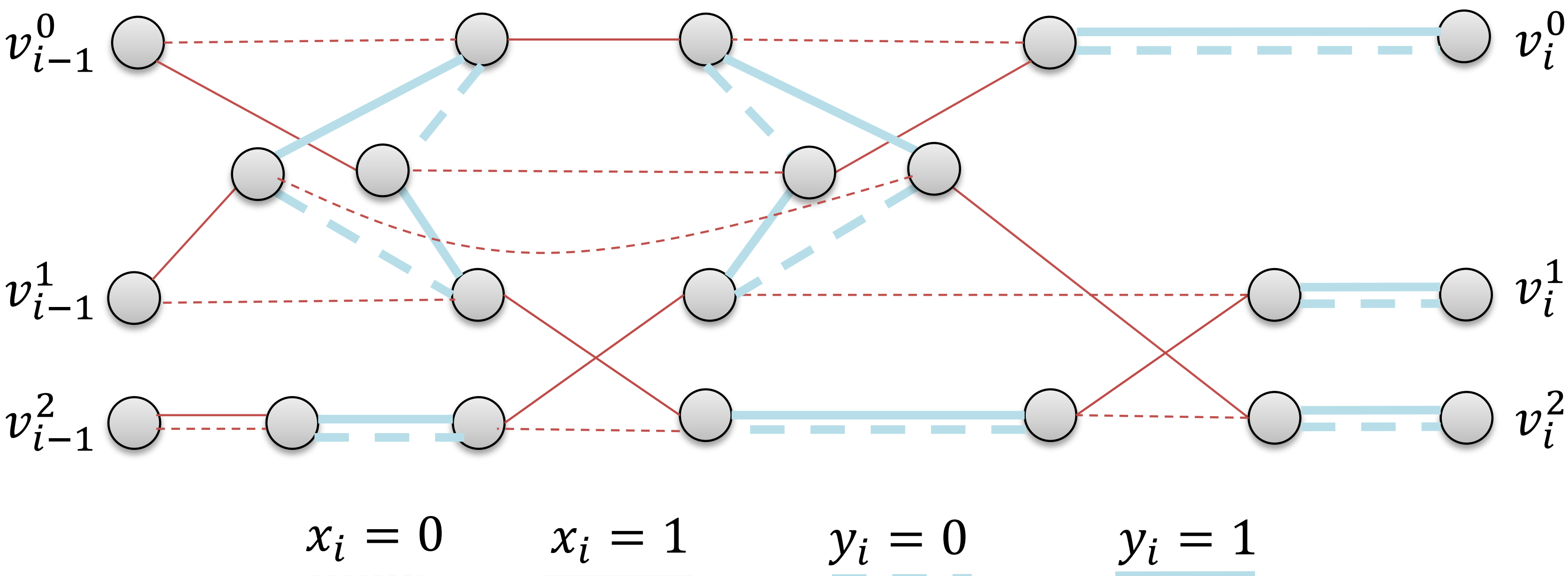
Direct Hamiltonian Cycle lower bound via **Direct** IPmod3 lower bound

Definition: IPmod3

- Alice gets $x=\{0, 1\}^b$, Bob gets $y=\{0, 1\}^b$
- Wants to know **<x, y> mod 3 = 0** or not, where <x, y> is the inner product
- Observe: disjointness = IPmod(n+1)
- Lower bound: Ω(b)
  - Holds even in the *quantum* setting

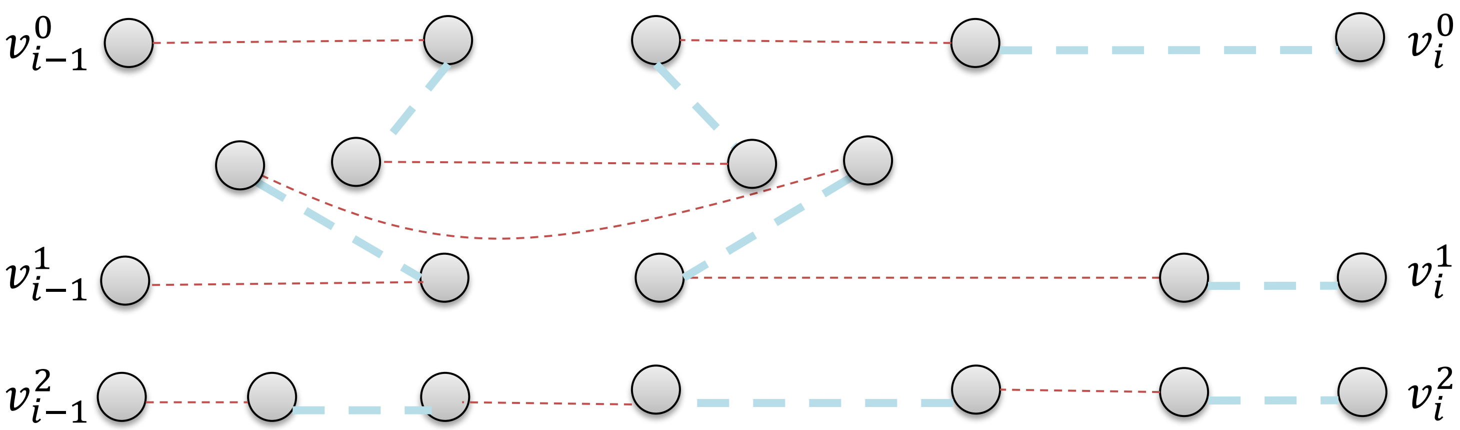
Reduction (sketched)

Gadget G_i for each bit i

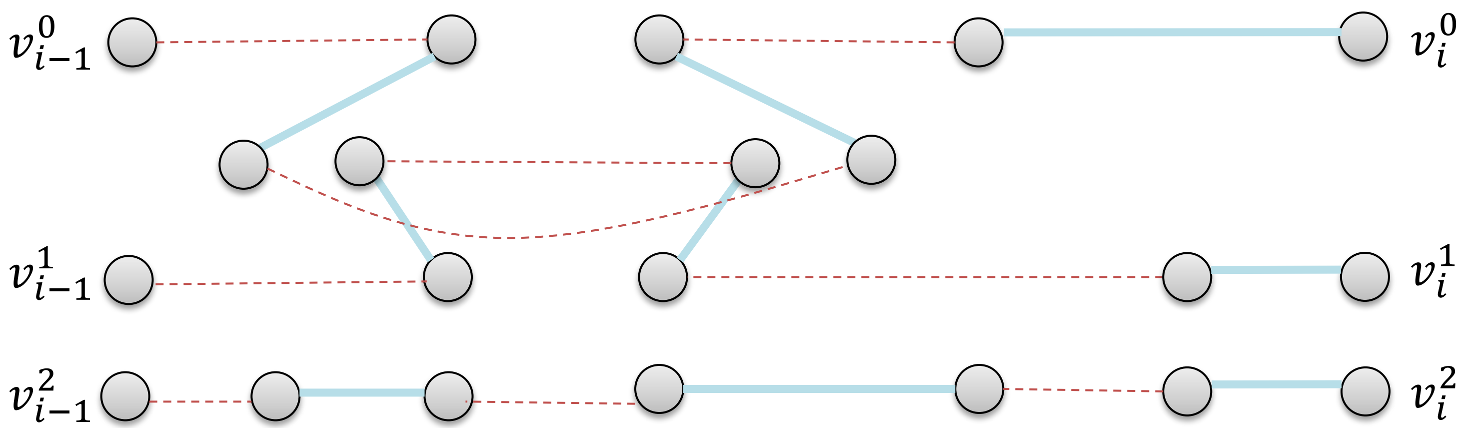


Red: Alice's edges, Blue: Bob's edges

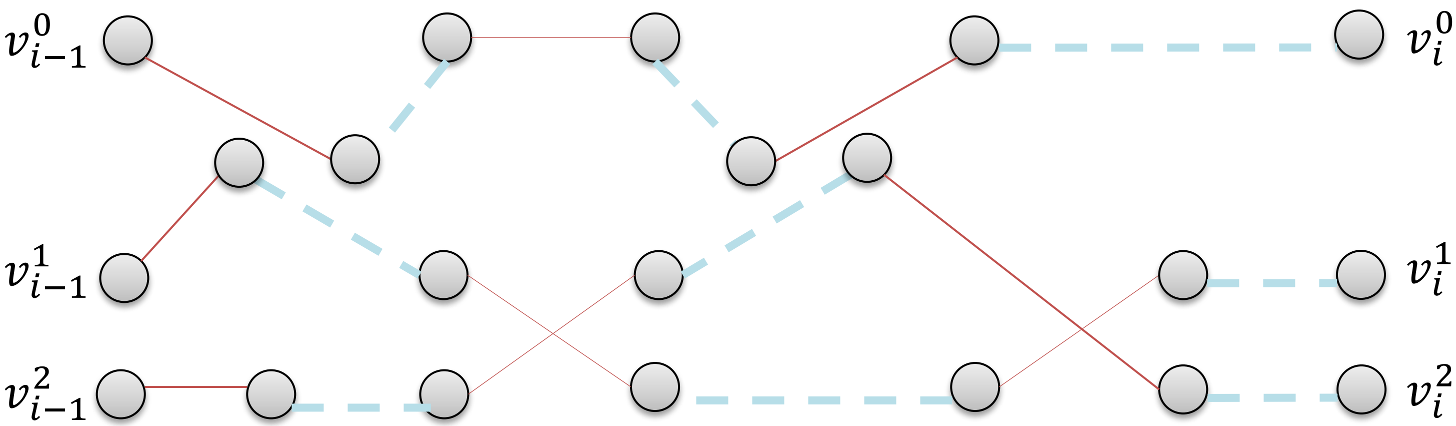
If $(x_i, y_i) = (0, 0)$



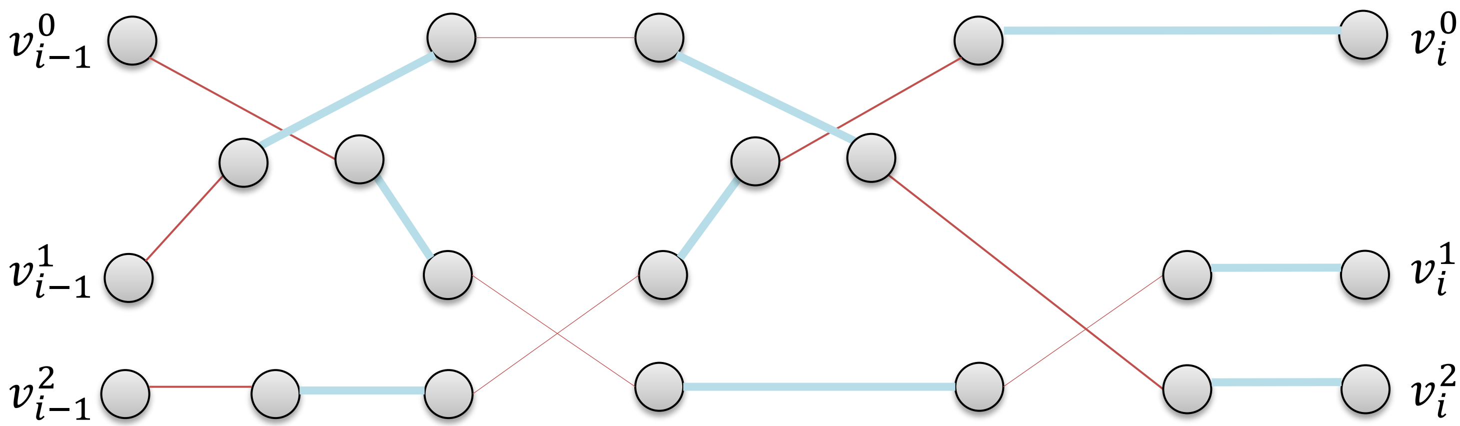
If $(x_i, y_i) = (0, 1)$



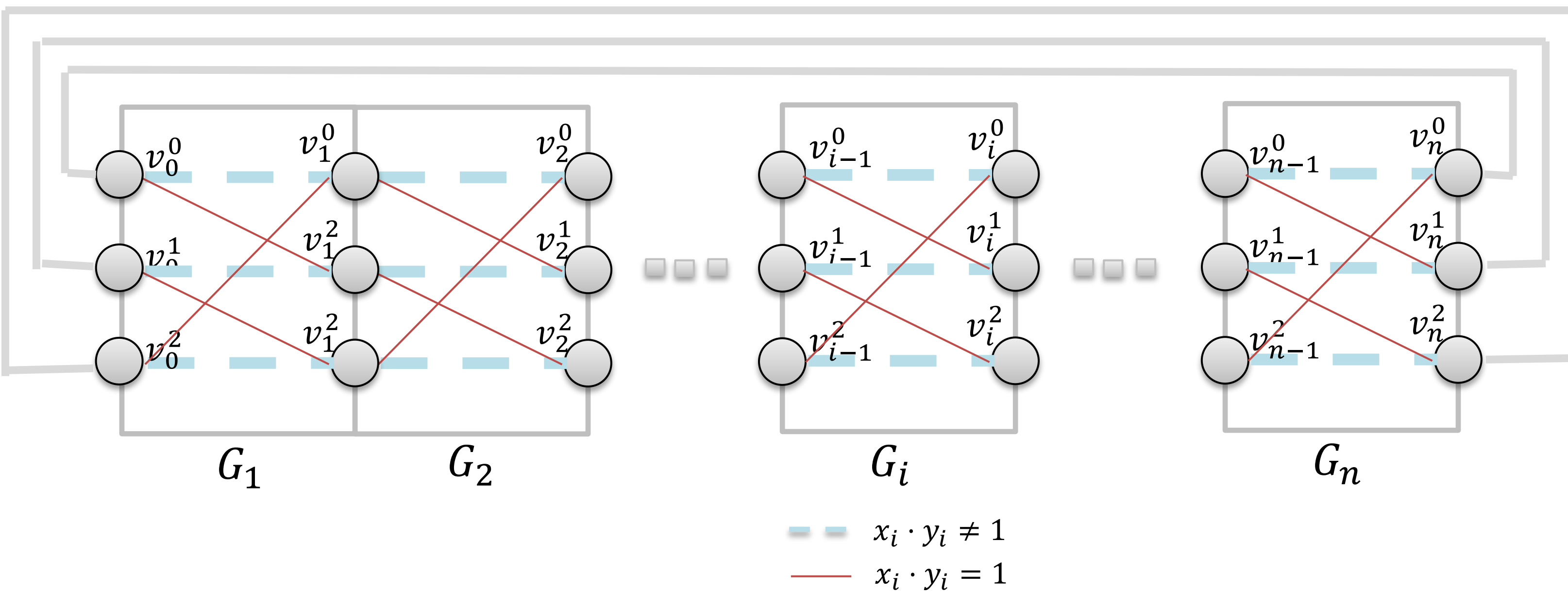
If $(x_i, y_i) = (1, 0)$



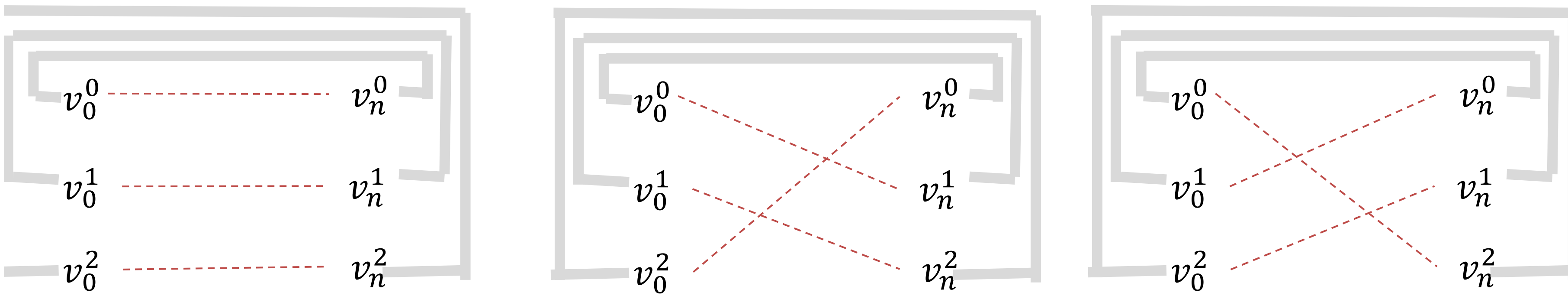
If $(x_i, y_i) = (1, 1)$



When connect everything together



Three possible end results



$$\sum_i x_i \cdot y_i \ mod\ 3 = 0$$

$$\sum_i x_i \cdot y_i \ mod\ 3 = 1$$

$$\sum_i x_i \cdot y_i \ mod\ 3 = 2$$

Exercise

- Reduce from direct Hamiltonian cycle to distributed spanning tree verification
- (Harder) Reduce from direct Hamiltonian cycle to distributed Hamiltonian cycle verification

Part 2

Some follow-up works

Part 2.1

Minimum Spanning Tree

Gallager, Humblet, Spira, **TOPLAS'83**

Chin, Teng, **FOCS'85**

Gafni, **PODC'85**

Awerbuch, **STOC'87**

Garay, Kutten, Peleg, **FOCS'93**

Kutten, Peleg, **PODC'95**

$O(D + n^{1/2} \log^* n)$-time

Lotker, Patt-Shamir, Peleg **PODC'01**

Lotker, Patt-Shamir, Peleg

Elkin **SODA'04**

Khan, Pandurangan **DISC'06**

Elkin + N + others **PODC'14**

Ookawa, Izumi **SOFSEM'15**

$\Omega(D+n^{1/2})$ –time lower bound

Peleg, Rubinovich **FOCS'99**

Elkin **STOC'04**

Das Sarma + N + 6 others **STOC'11**

"Any" approximation algorithm requires

$\Omega(D+(n/\log n)^{1/2})$ –time when $D=O(\log n)$

Approximation algorithm?

any = any poly(n)-approximation algorithm

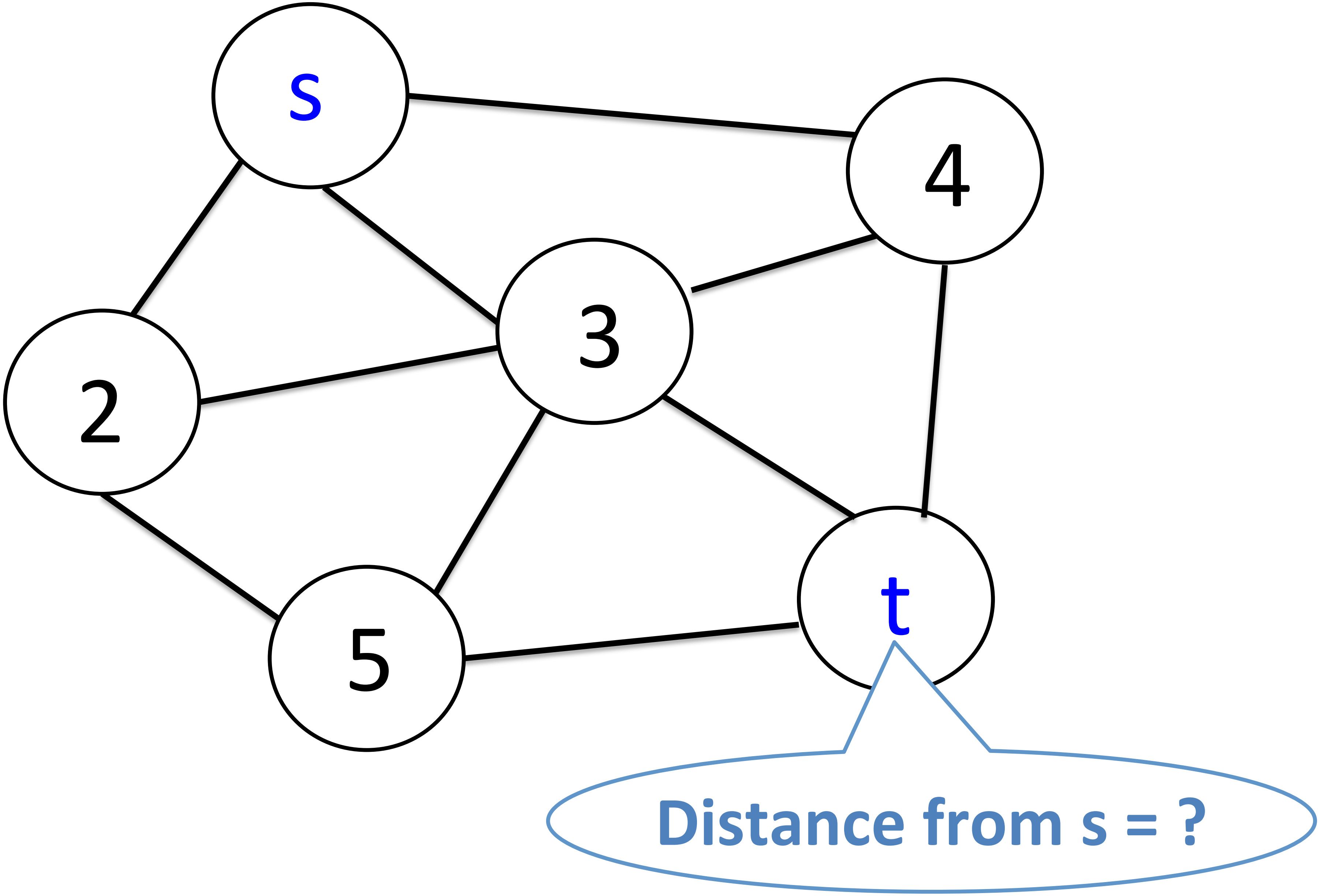
Distributed MST is essentially resolved

Still open: O(log* n) gap between upper and lower bounds

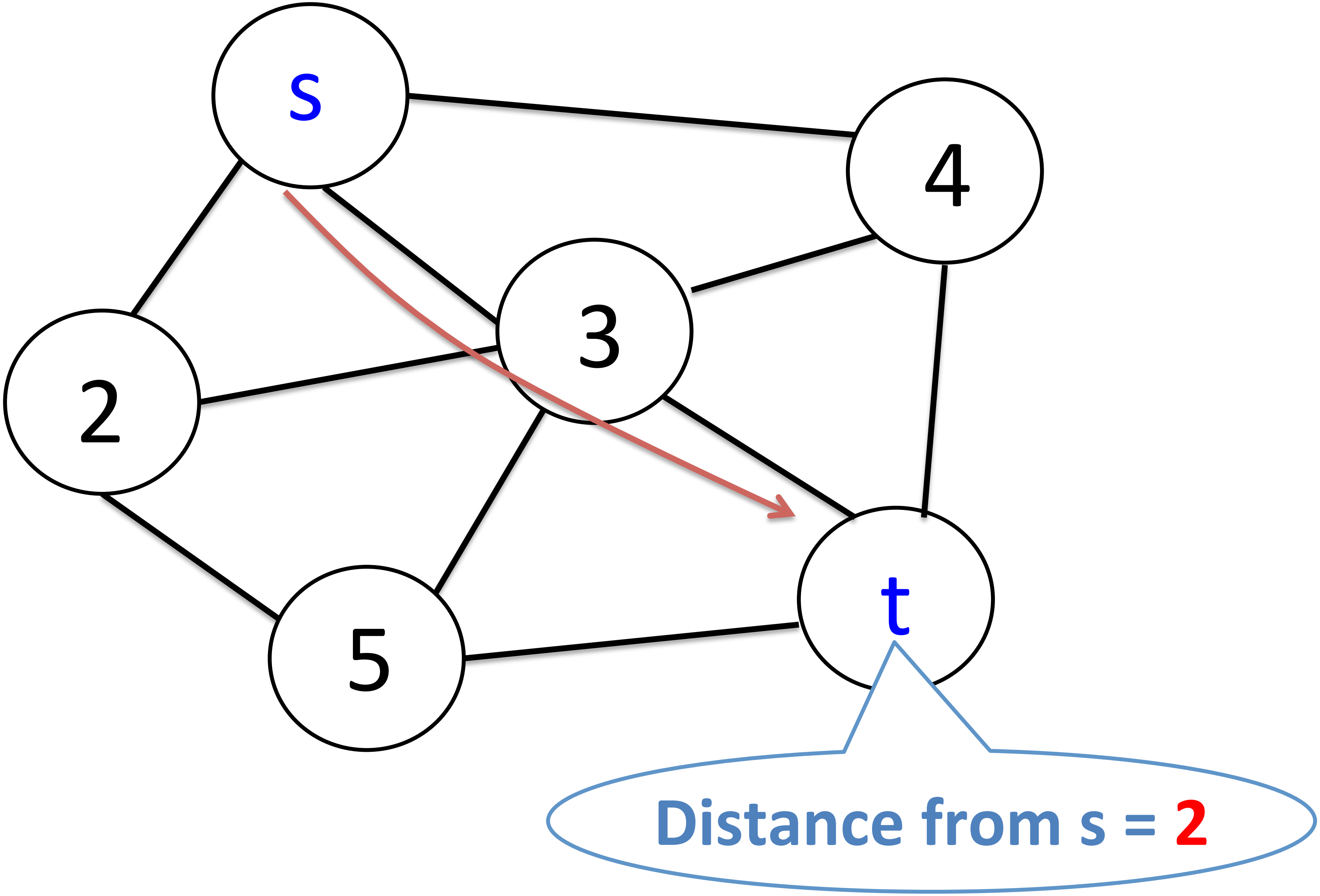
<u>**Part 2.2**</u>

s-t distance,
single-source distances

Definition: unweighted s-t distance



Goal: t knows distance from s

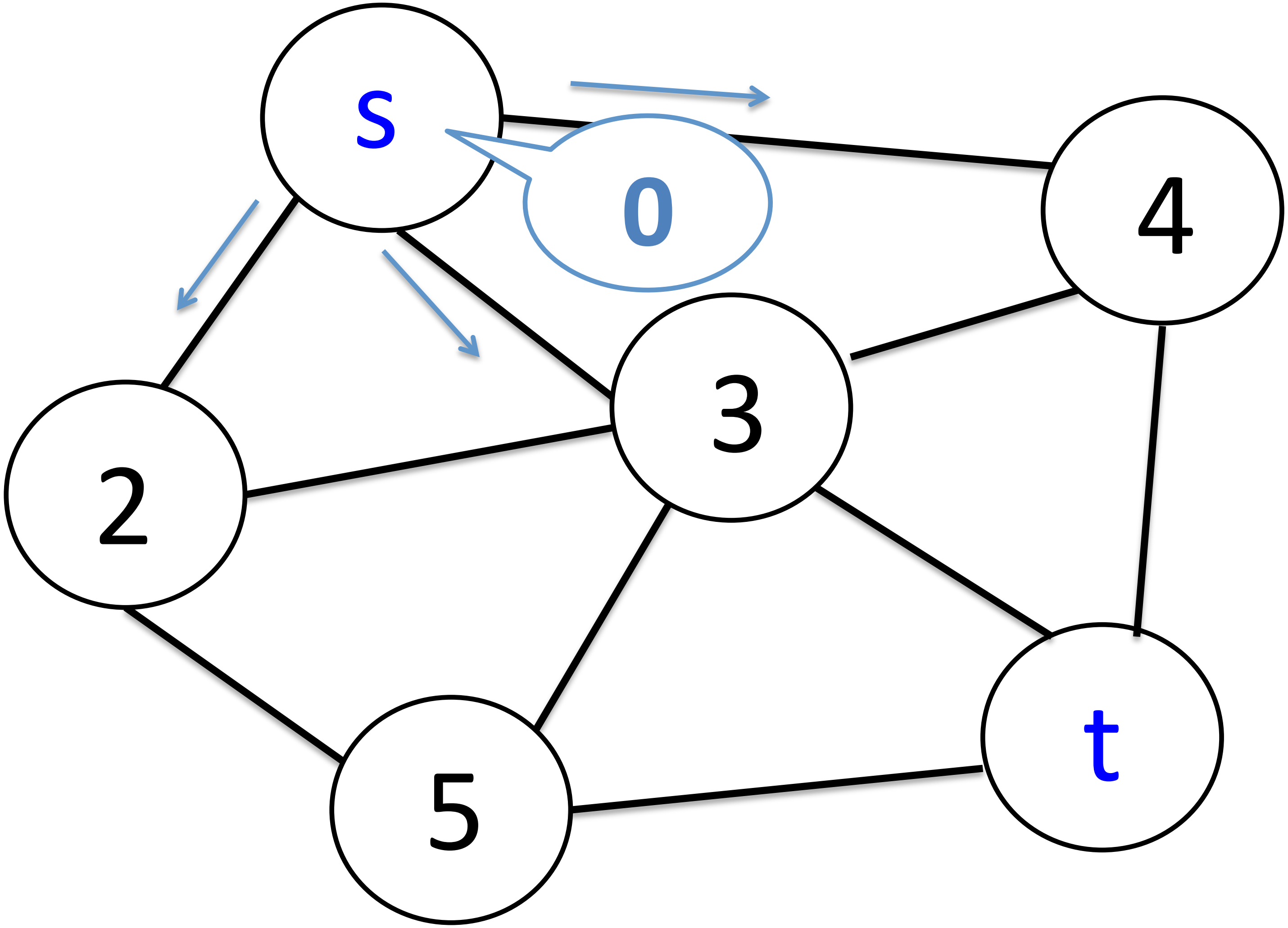


Goal: t knows distance from s

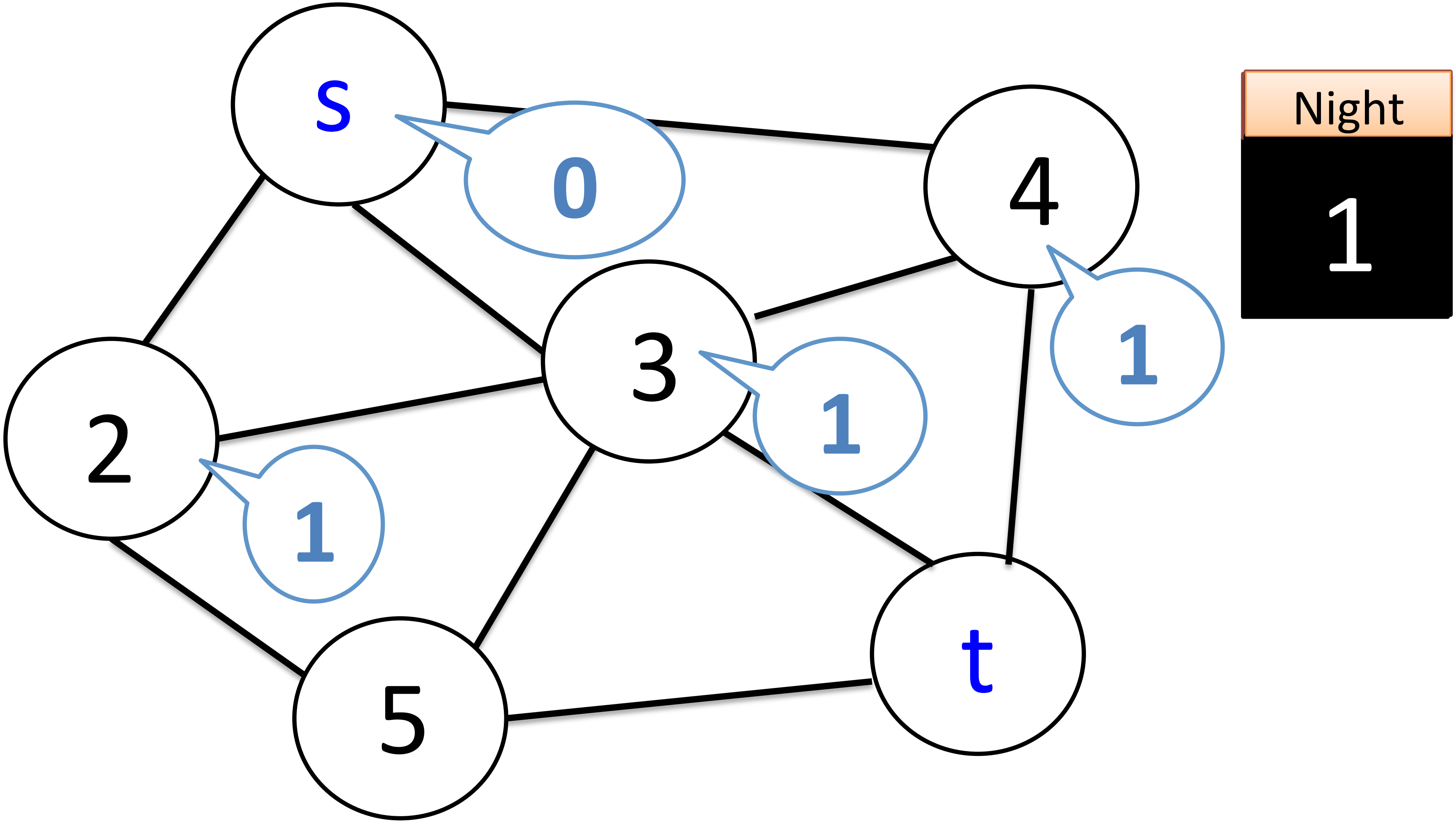
Claim

Computing s-t distance can be done in **O(D)** time by using the **Breadth-First Search (BFS)** algorithm.

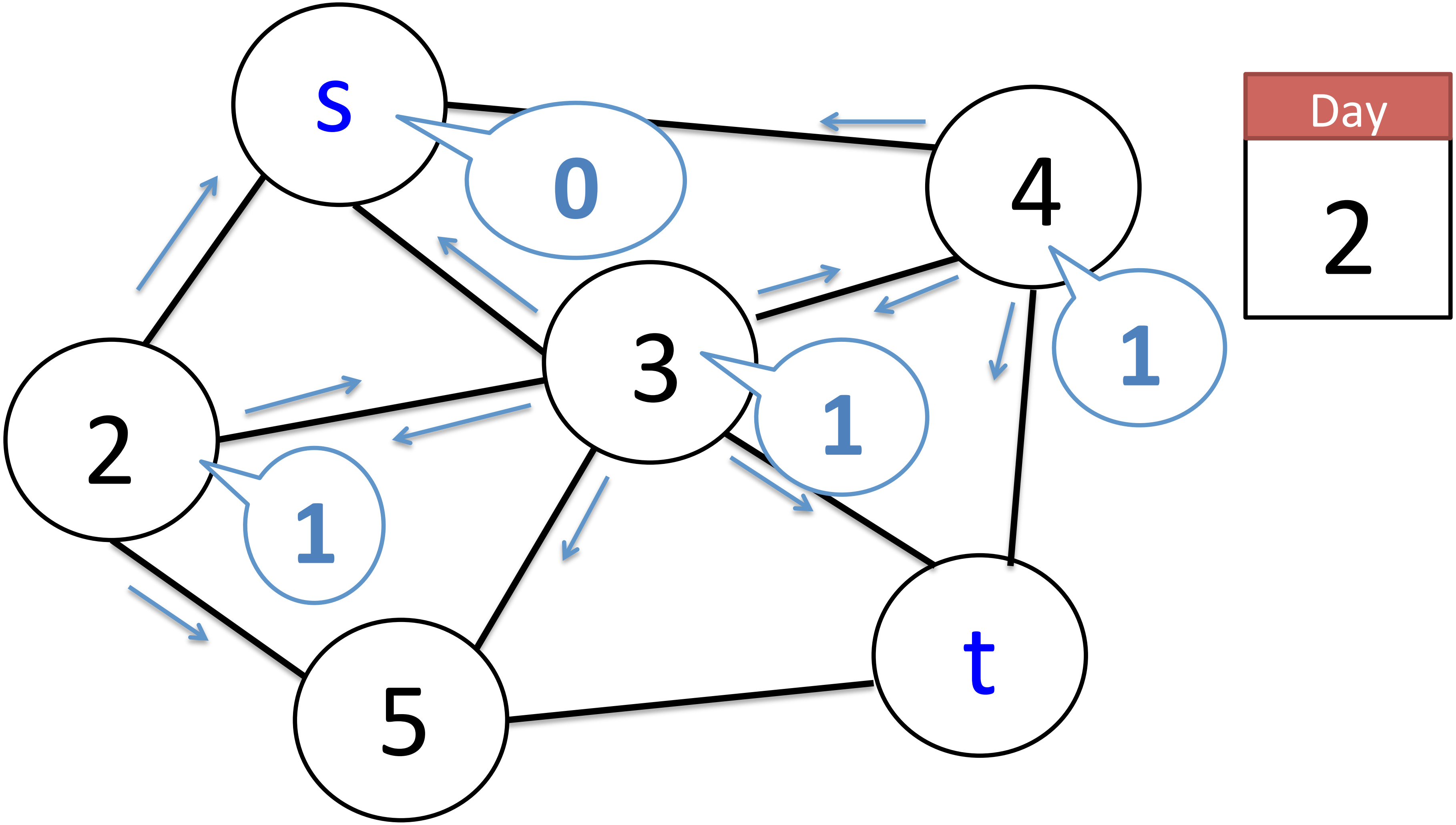
Reminder: D = diameter



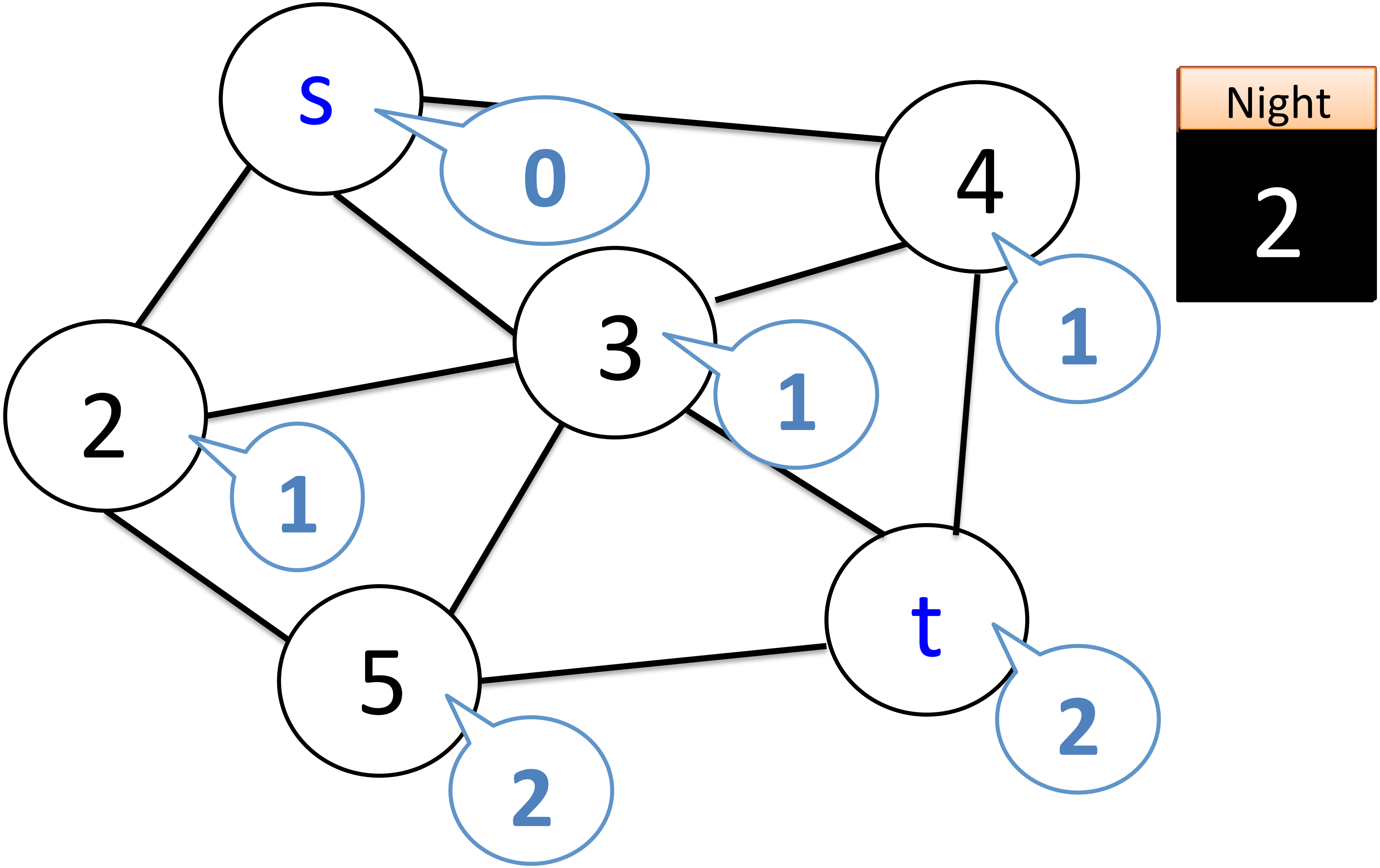
Source node sends its distance to neighbors



Each node updates its distance



Nodes tell new knowledge to neighbors



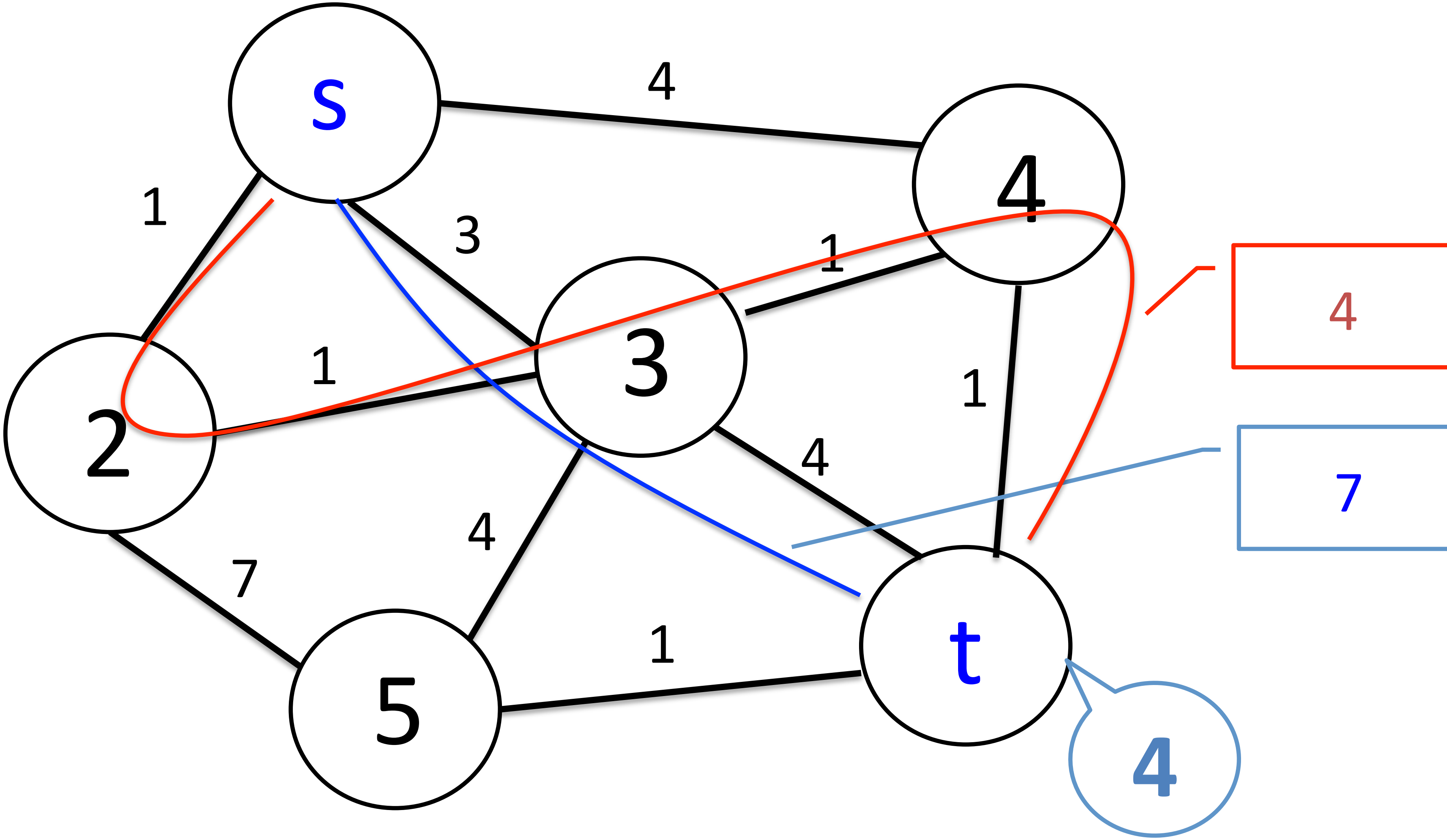
Each node updates its distance

Claim

s-t distance can be computed in **O(D)** time.

There is an **Ω(D)** lower bound.
So, the algorithm is tight.

How about **weighted** graphs?



s-t distance

Reference	Time	Approximation
Folklore	$\Omega(D)$	any

- Polylog n factors are hidden

Reference	Time	Approximation
Folklore	Ω(D)	any
⇨ Bellman&Ford [1950s]	O(n)	exact

- Polylog n factors are hidden

Reference	Time	Approximation
Folklore	$\Omega(D)$	any
Bellman&Ford [1950s]	$O(n)$	exact
⇨ Elkin [STOC 2006]	$\Omega((n/\alpha)^{1/2} + D)$	any α

- Polylog n factors are hidden

Reference	**Time**	**Approximation**
Folklore	$\Omega(D)$	any
Bellman&Ford [1950s]	$O(n)$	exact
Elkin [STOC 2006]	$\Omega((n/\alpha)^{1/2} + D)$	any α
Das Sarma et al [STOC 2011] Elkin et al. [PODC 2014]	$\Omega(n^{1/2} + D)$	any α also quantum

- Polylog n factors are hidden

Reference	Time	Approximation
Folklore	$\Omega(D)$	any
Bellman&Ford [1950s]	$O(n)$	exact
Elkin [STOC 2006]	$\Omega((n/\alpha)^{1/2} + D)$	any α
Das Sarma et al [STOC 2011] Elkin et al. [PODC 2014]	$\Omega(n^{1/2} + D)$	any α also quantum
⇨ Lenzen,Patt-Shamir [STOC 2013]	$O(n^{1/2+\varepsilon} + D)$	$O(1/\varepsilon)$

- Polylog n factors are hidden
- Lenzen&Patt-Shamir actually achieve more than computing distances

Reference	Time	Approximation
Folklore	$\Omega(D)$	any
Bellman&Ford [1950s]	$O(n)$	exact
Elkin [STOC 2006]	$\Omega((n/\alpha)^{1/2} + D)$	any α
Das Sarma et al [STOC 2011] Elkin et al. [PODC 2014]	$\Omega(n^{1/2} + D)$	any α also quantum
Lenzen,Patt-Shamir [STOC 2013]	$O(n^{1/2+\varepsilon} + D)$	$O(1/\varepsilon)$
⇨ N [STOC 2014]	$O(n^{1/2}D^{1/4} + D)$	$1+\varepsilon$

- Polylog n factors are hidden
- Lenzen&Patt-Shamir actually achieve more than computing distances

Reference	Time	Approximation
Folklore	$\Omega(D)$	any
Bellman&Ford [1950s]	$O(n)$	exact
Elkin [STOC 2006]	$\Omega((n/\alpha)^{1/2} + D)$	any α
Das Sarma et al [STOC 2011] Elkin et al. [PODC 2014]	$\Omega(n^{1/2} + D)$	any α also quantum
Lenzen,Patt-Shamir [STOC 2013]	$O(n^{1/2+\varepsilon} + D)$	$O(1/\varepsilon)$
N [STOC 2014]	$O(n^{1/2}D^{1/4} + D)$	$1+\varepsilon$
⇨ Henzinger,Krinninger,N [2015]	$O(n^{1/2+o(1)} + D^{1+o(1)})$	$1+\varepsilon$

- Polylog n factors are hidden
- Lenzen&Patt-Shamir actually achieve more than computing distances

Distributed s-t distance **approximation** is essentially resolved

Exercise (easy)

- Argue that approximating st-distance require $\Omega(n^{1/2})$ time on some network of diameter $n^{1/4}$

Open Problem

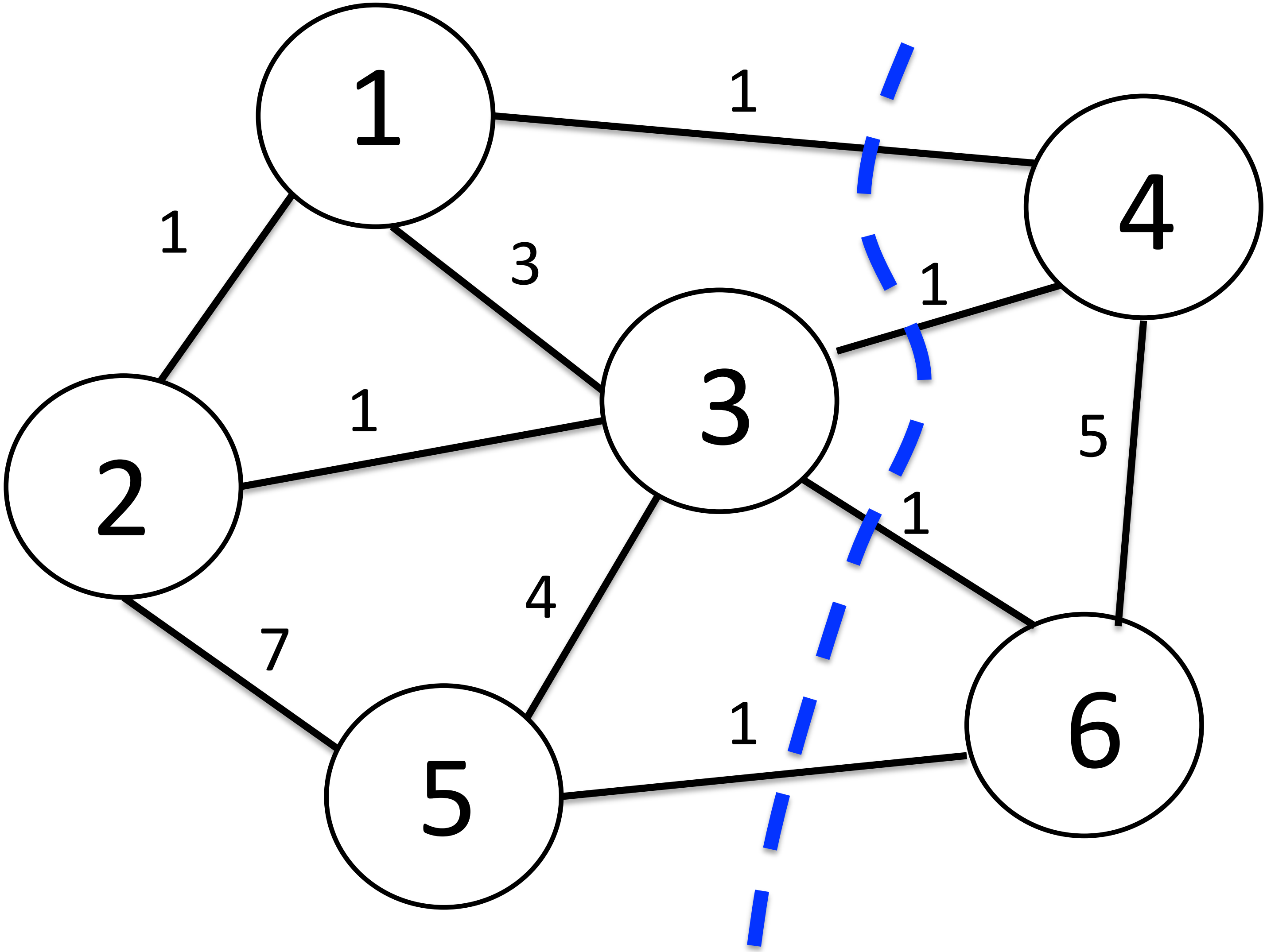
Computing s-t distance

exactly in **sublinear-time**

i.e. in $O(n^{1-\varepsilon}+D)$ time

Part 2.3

Some other distributed approximation algorithms



Minimum cut
(weight = 4)

Global min cut (a.k.a. edge-connectivity)

λ = optimal solution

Reference	Time	Approximation
Pritchard, Thurimella [TALG'11]	$O(D)$ for $\lambda \leq 2$	exact
	$O(n^{1/2} + D)$ for $\lambda \leq 3$	exact
N.-Su [DISC'14]	$O((n^{1/2} + D)\ \lambda^4)$ thus $O((n^{1/2} + D))$ for constant λ	exact

Das Sarma et al [STOC'11] Elkin et al. [PODC 2014]	$\Omega(n^{1/2} + D)$ for large enough λ	any also quantum
Ghaffari, Kuhn [DISC'13]	$O(n^{1/2} + D)$	2
N + Su [DISC'14]	$O(n^{1/2} + D)$	$1+\varepsilon$

*polylog n terms are omitted

Global min cut (a.k.a. edge-connectivity)

λ = optimal solution

Distributively **approximating** mincut is essentially resolved

<u>Open</u>:

- Sublinear-time **exact** algorithm.
- Lower bound when λ is small.

Probabilistic Tree Embedding (in particular, FRT embedding)

Reference	Time	Approximation
Das Sarma et al. [STOC'11]	$\Omega(n^{1/2} + D)$	any also quantum
Ghaffari, Lenzen [DISC'14]	$O(n^{1/2+\varepsilon} + D)$	$O(\log n/\varepsilon)$

Minimum-Weight Connected Dominating Set

Das Sarma et al. [STOC'11]	$\Omega(n^{1/2} + D)$	any also quantum
Ghaffari [ICALP'14]	$O(n^{1/2} + D)$	$O(\log n)$

Steiner Forest

Lenzen, Patt-Shamir [PODC'14]	$\Omega(n^{1/2} + D + k)$	any
Lenzen, Patt-Shamir [PODC'14]	$O(n^{1/2} + D + k)$	$O(\log n)$

Open problems

- Exact algorithms
  - st-distance O(n) vs. $\Omega(n^{1/2}+D)$
  - mincut O(m) vs. $\Omega(n^{1/2}+D)$
- k-edge connectivity when k is constant $O(n^{1/2}+D)$ vs. nothing

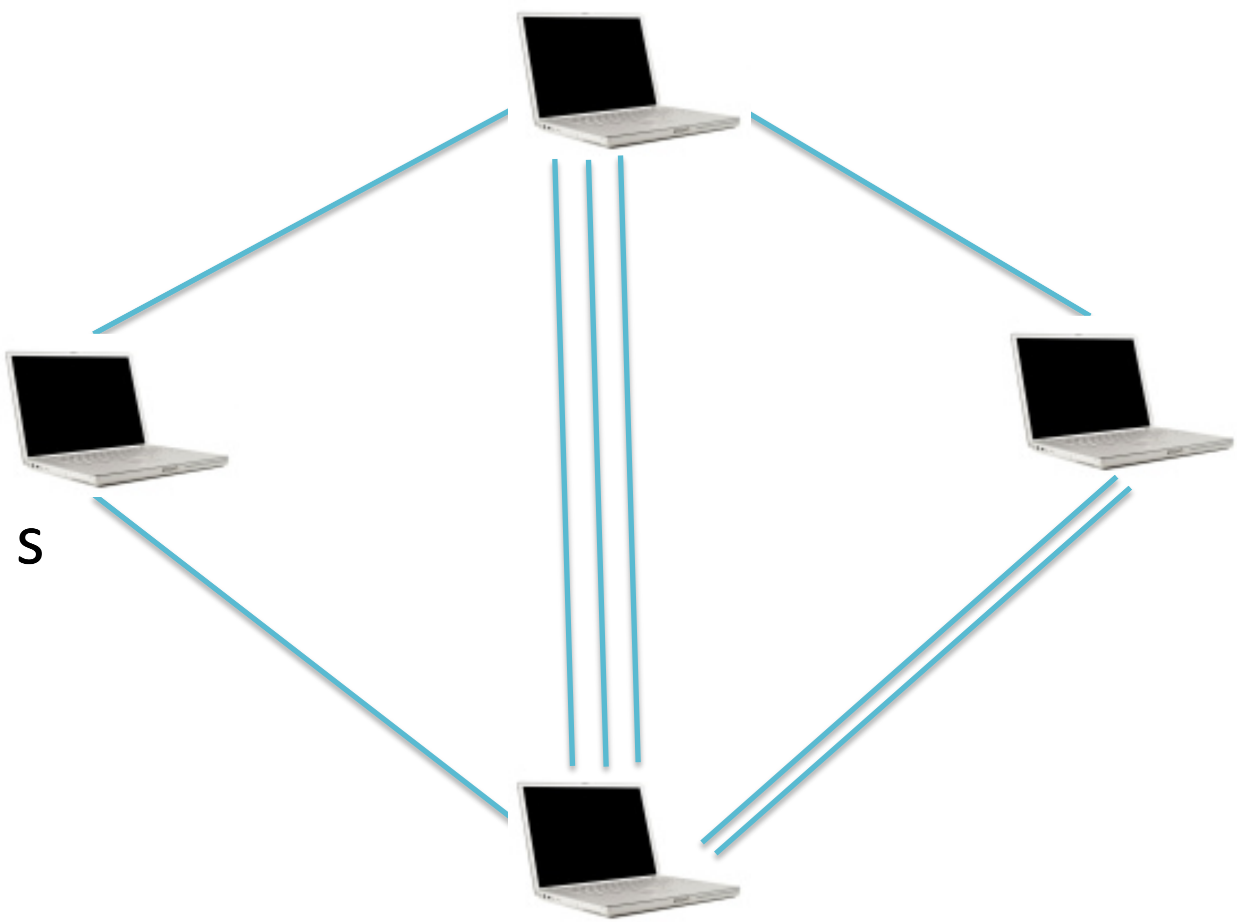
Part 3

Extension to round-efficient Simulation Theorem

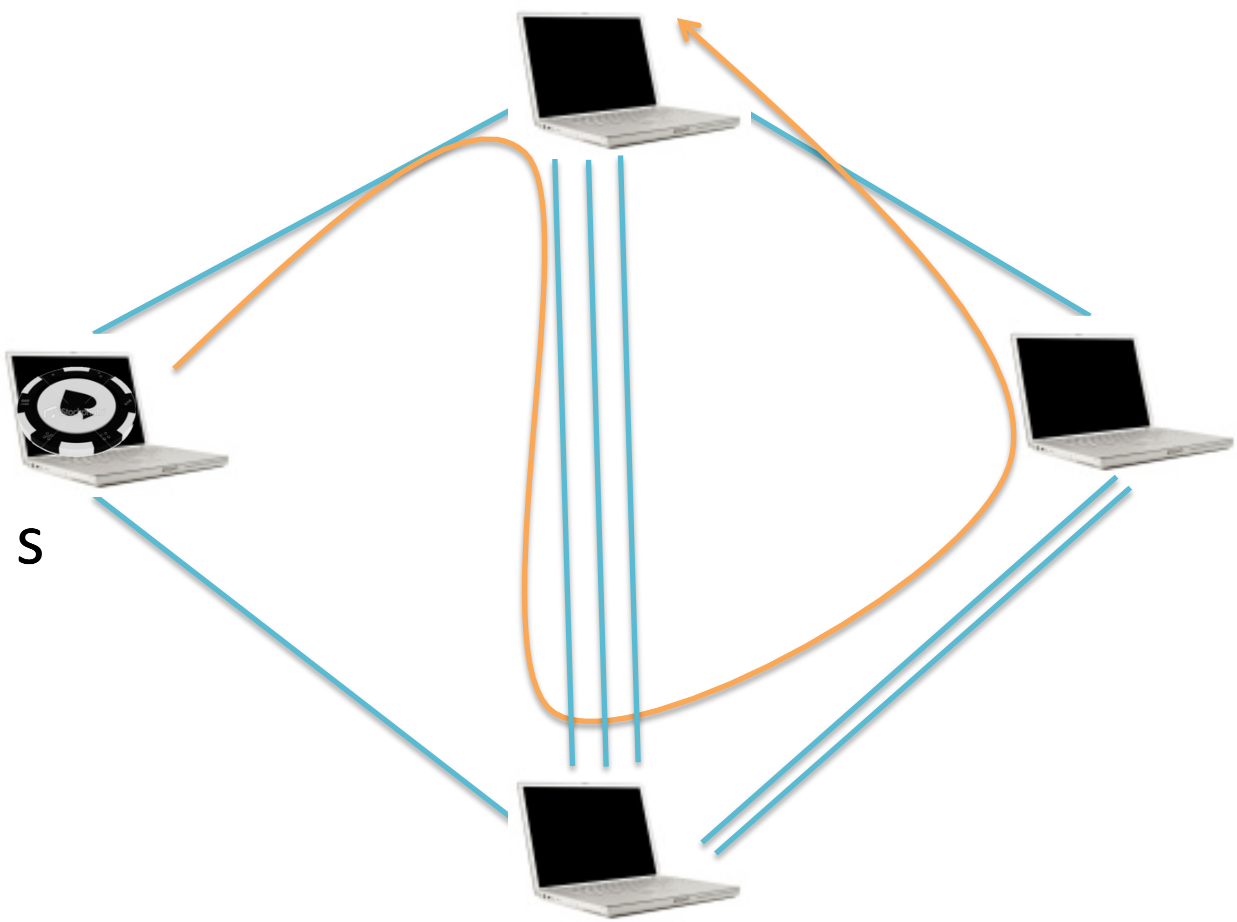
Motivation

Distributed Random Walks

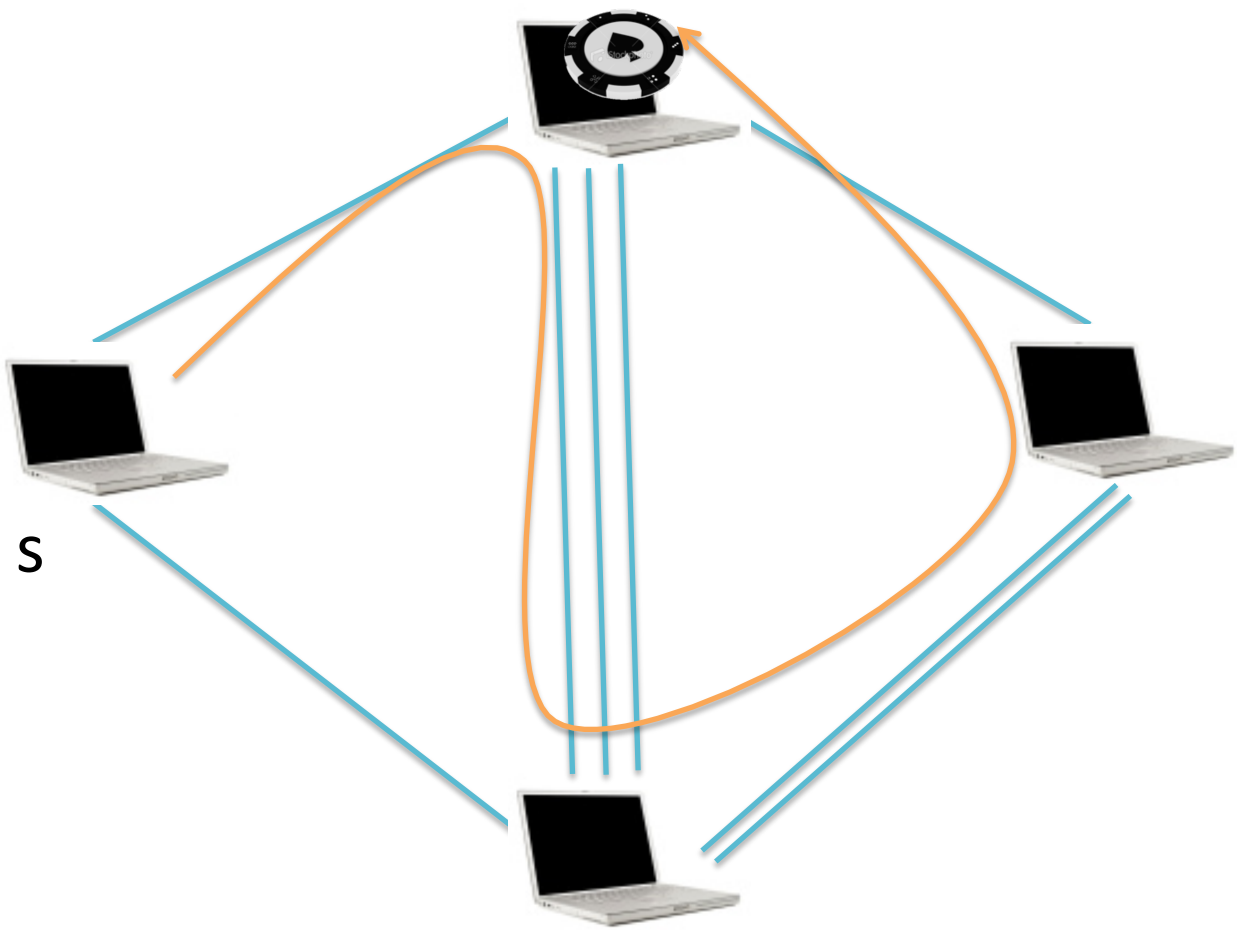
Want a random walk of length ℓ from s



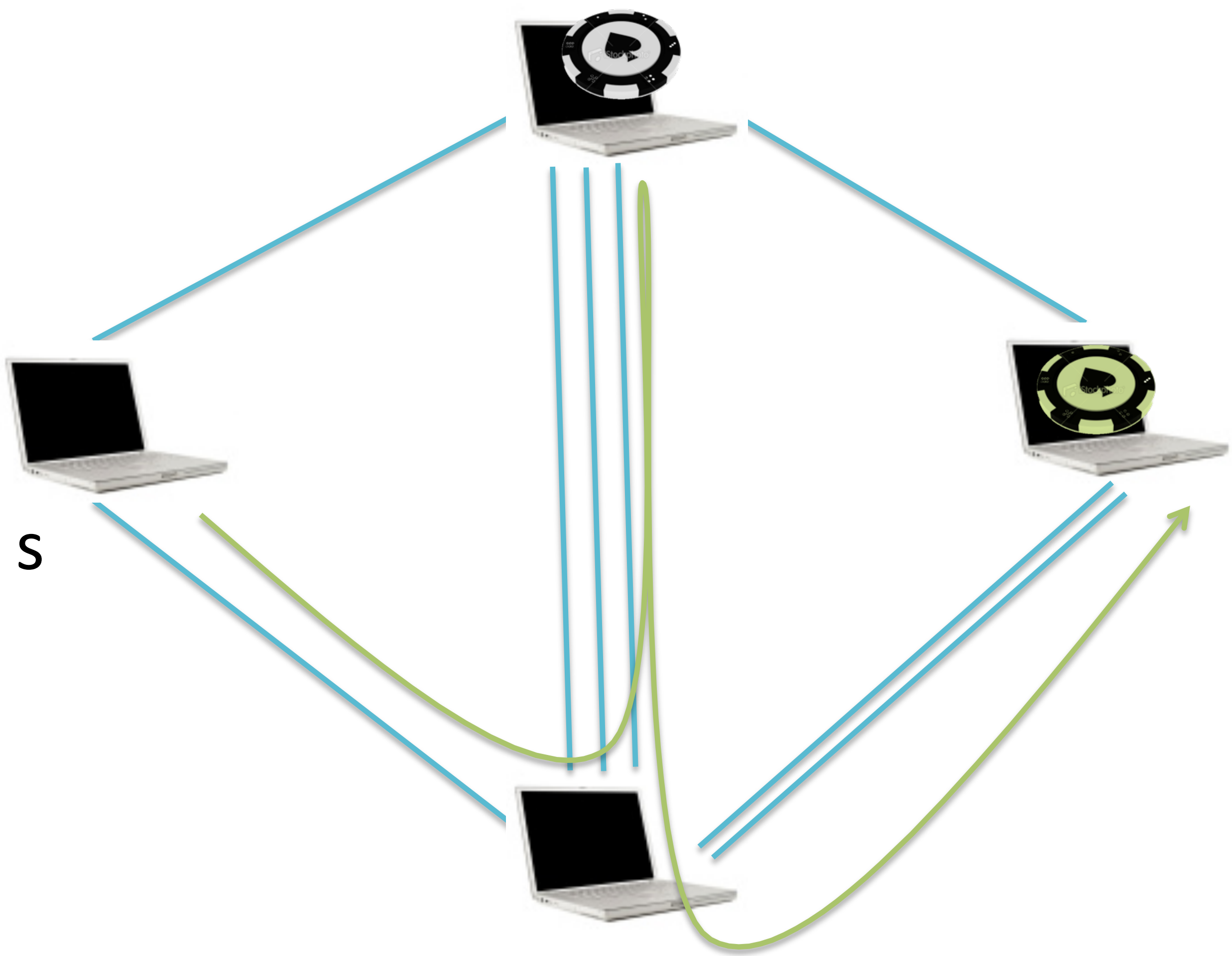
Trivial algorithm: Forward a token randomly for ℓ rounds



The token ends somewhere



If we repeat, the token might end in a different node



This process takes ℓ rounds to send a token in a random walk manner.

Distributed random walk problem

Can we forward the token in a random walk manner faster than ℓ rounds?

(Formally, we want to sample a destination node according to the distribution induced by the l –step random walk.)

Random walks

[Das Sarma, N., Pandurangan, Tetali, **PODC'09+10**]:

- A random walk of length ℓ can be found in $\mathbf{O((\ell D)^{1/2})}$ time
- *Conditional* lower bound of $\Omega(\ell^{1/2})$ time for small D on multigraphs

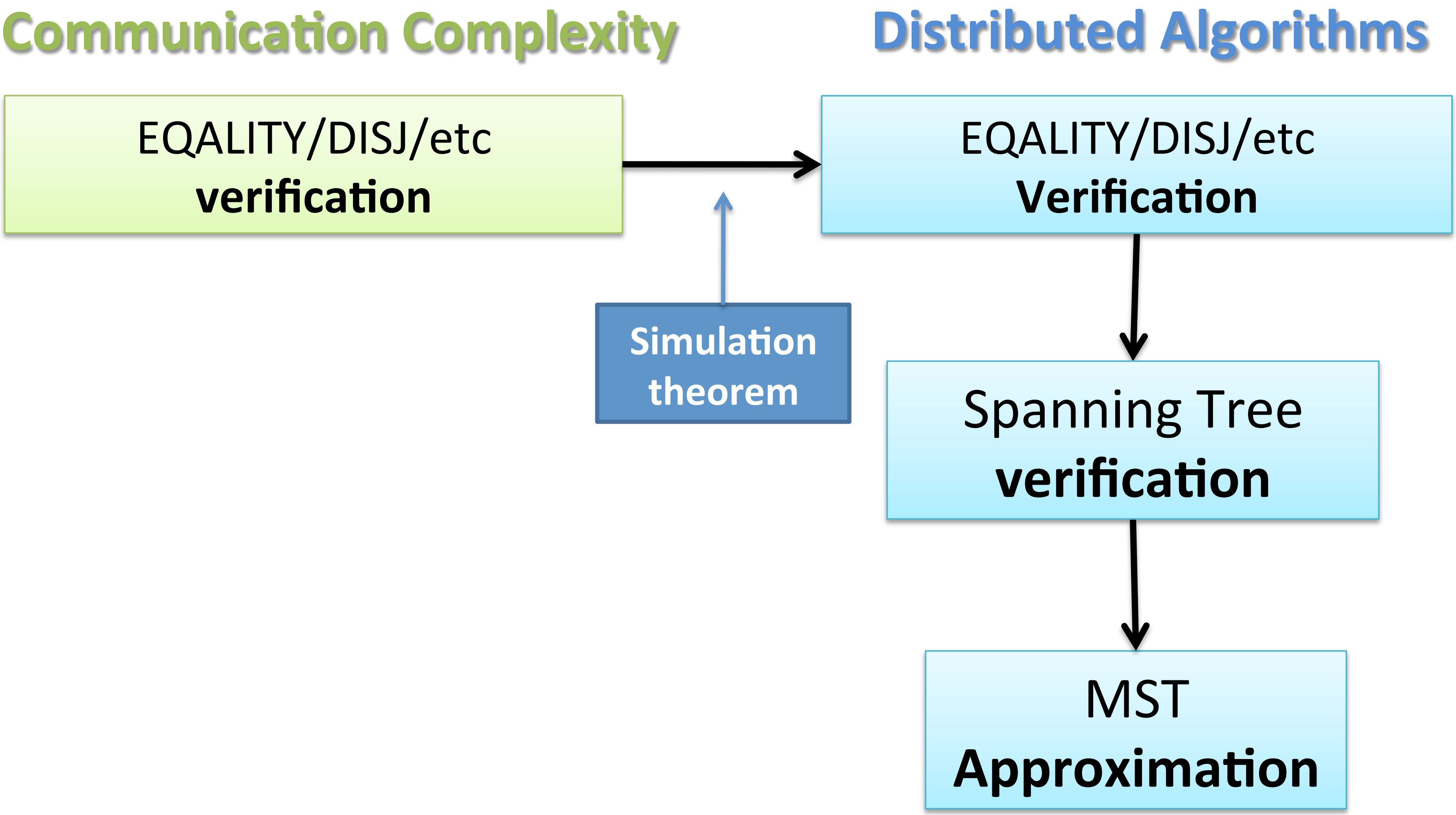
[N. Das Sarma, Pandurangan]:

- Lower bound of $\Omega((\ell D)^{1/2})$-time for any **n**, **D**, and $D \leq \ell \leq (n/D^3 \log n)^{1/4}$ on multigraphs
- First lower bound that D plays a role of multiplicative factor

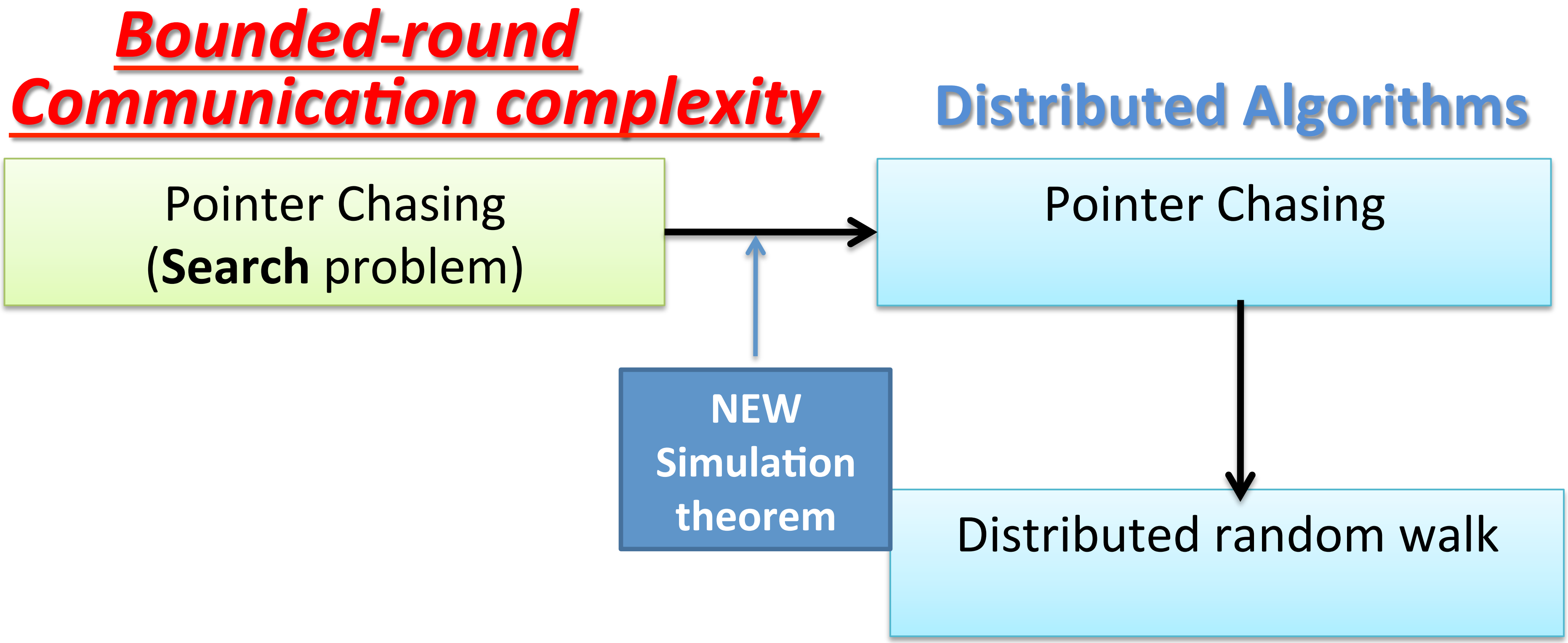
The Simulation Theorem is not Enough

Impossible to get D in the lower bound since D is not part of the Simulation Theorem

Previous Reductions



New Reductions



Previous Simulation Theorem

If *f* can be computed distributively in **T** days, for any **T ≤ (path length)/2**, then the communication complexity of *f* is **≤T**

Proof Alice and Bob can simulate any distributed algorithm for **b/2** days with one bit exchanged per day.

NEW Simulation Theorem

If *f* can be computed distributively in **T** days, for any **T ≤ (path length)/2**, then the communication complexity of *f* is **≤T** ***in ≤T/D rounds***

Proof Alice and Bob can simulate any distributed algorithm for **b/2** days with one bit exchanged per day.

NEW Simulation Theorem

<u>If</u> *f* can be computed distributively in **T** days, for any **T ≤ (path length)/2**, <u>then</u> the communication complexity of *f* is **≤T** ***<u>in ≤T/D rounds</u>***

<u>Proof</u> Alice and Bob can simulate any distributed algorithm for **b/2** days with one bit exchanged per day. **They wait for D rounds before sending messages**

Exercise

- Fill in the details for the proof of the New Simulation Theorem

Some changes are needed

Bad news

With quantum communication, disjointness is too easy

Good news

Many other problems are still hard

e.g. IPmod2, IPmod3, …

So, you can prove lower bounds for quantum algorithms using, e.g., IPmod3.

Part 4.1

Warning: You can't use arbitrary problem in the quantum communication complexity model

Bad news

We can't make the simulation theorem work for the quantum setting

Bad news

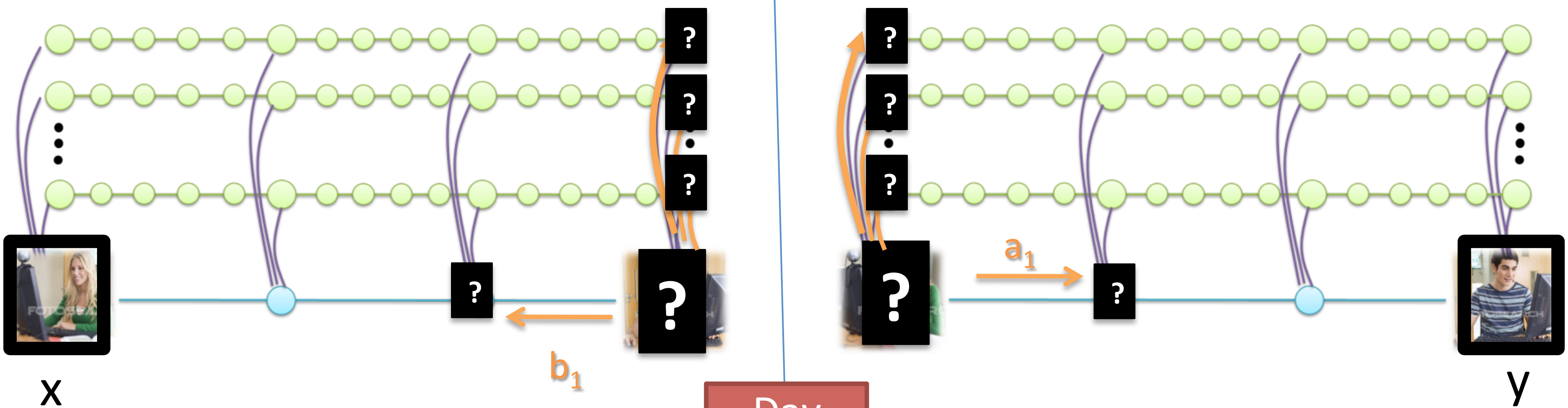
We can't make the simulation theorem work for the quantum setting

Reason

No-Cloning Theorem

(We can't make a copy of qubit)

Main problem:
Alice and Bob simulates the same machines



Alice
$x \in \{0, 1\}^b$

Bob
$y \in \{0, 1\}^b$

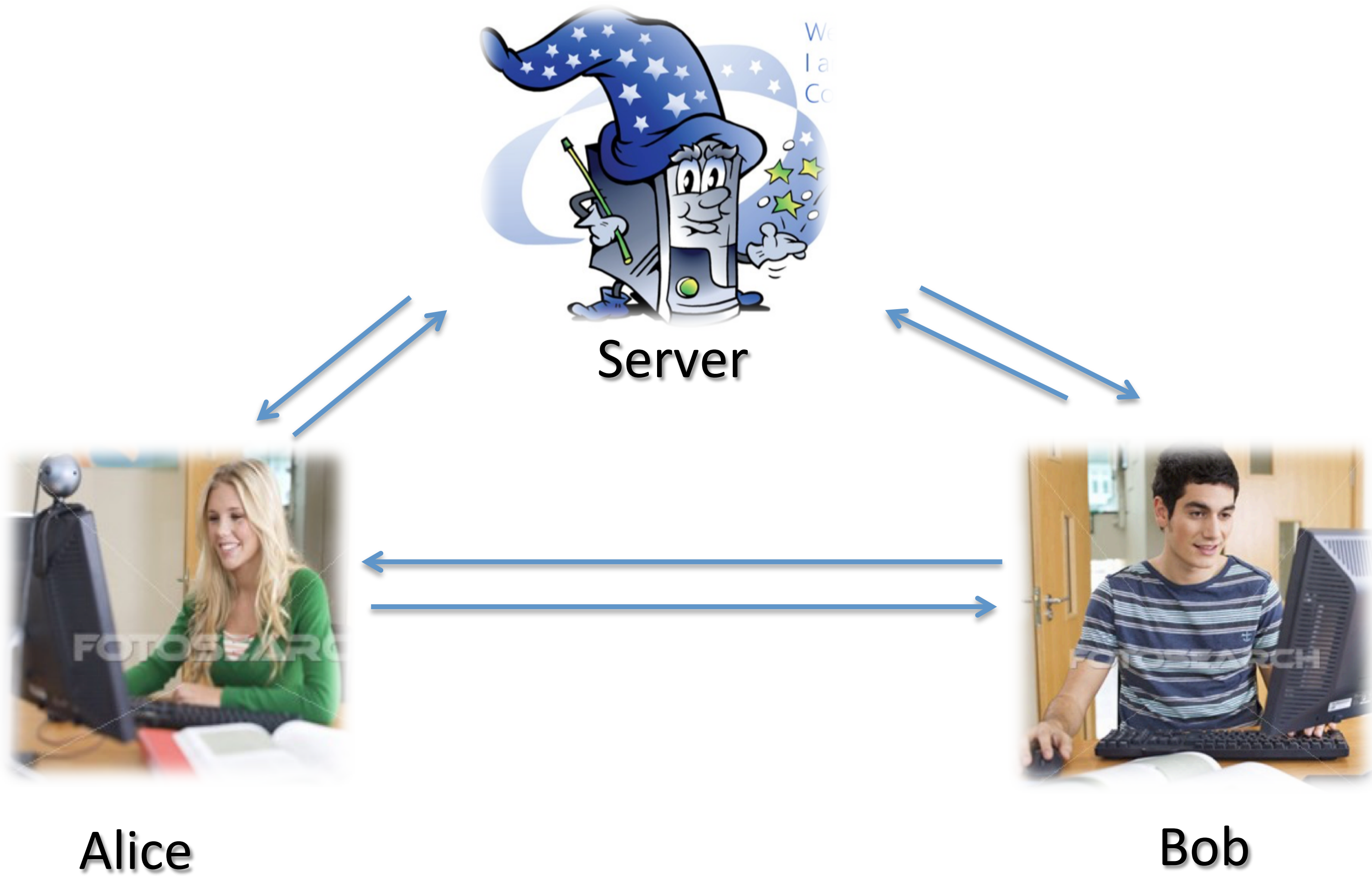
Main problem:Alice and Bob simulates the same machines

Good news

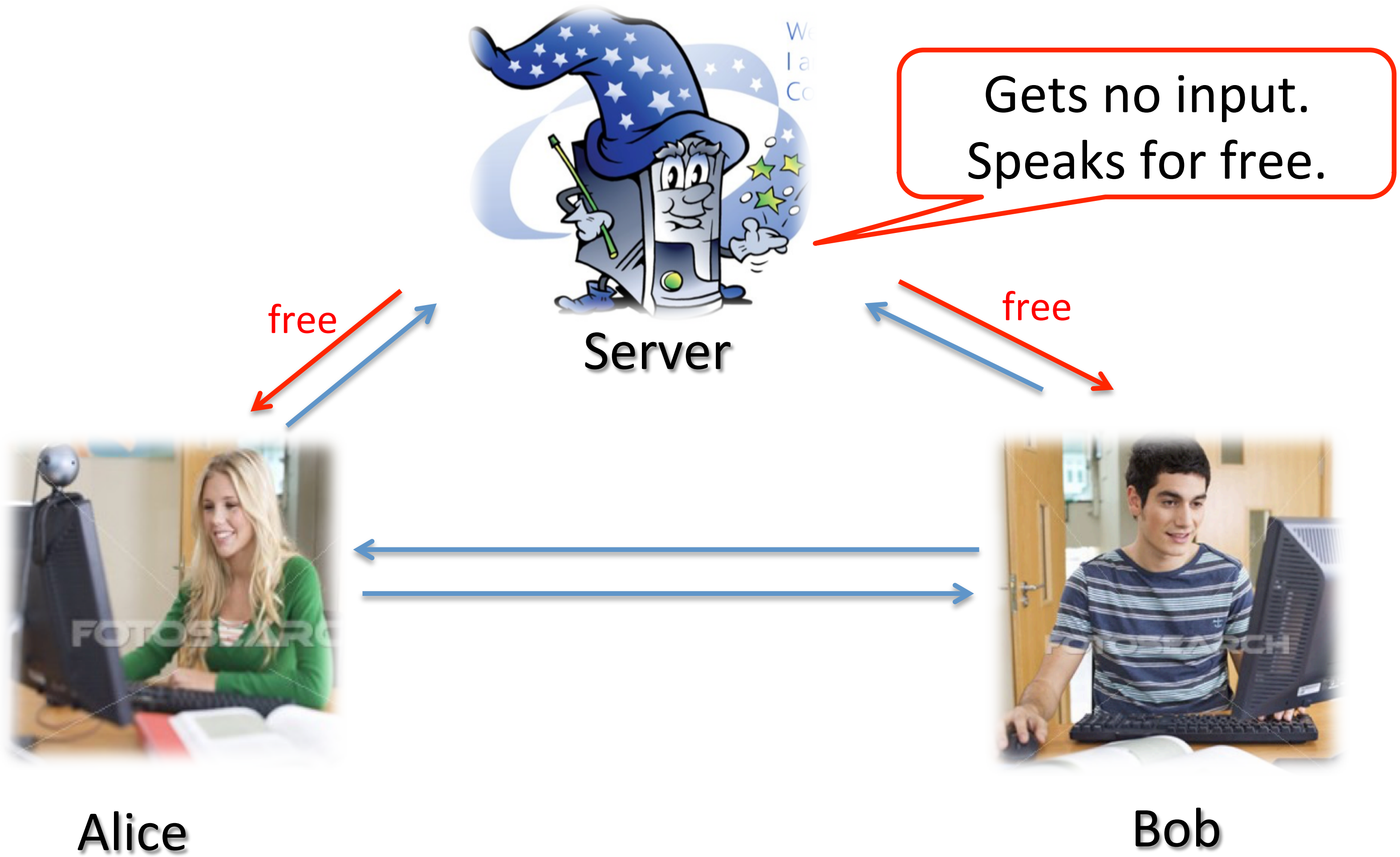
The simulation theorem works for a new model called

Server model

Server Model



Server Model



Good news

We show that problems such as IPmod2, IPmod3, … are still hard in the Server model.

Exercise

- Prove a new version of the Simulation Theorem where you start from the server model instead. Make sure that every machine in the network is simulated by exactly one party (among 3).