Quantum Distributed Computing

LG and Magniez. **Sublinear-Time Quantum Computation of the Diameter in Distributed Networks**. PODC 2018.

LG, Nishimura and Rosmanis. **Quantum Advantage for the LOCAL model in Distributed Computing.** STACS 2019.

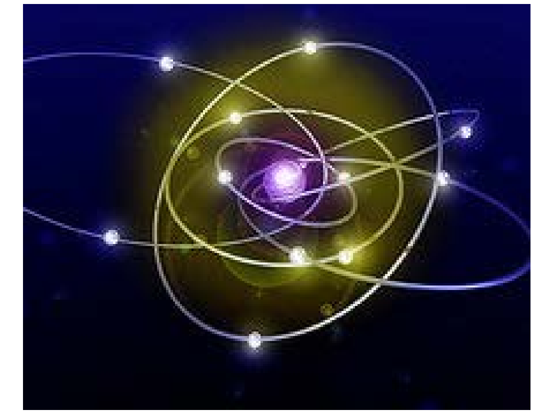
François Le Gall

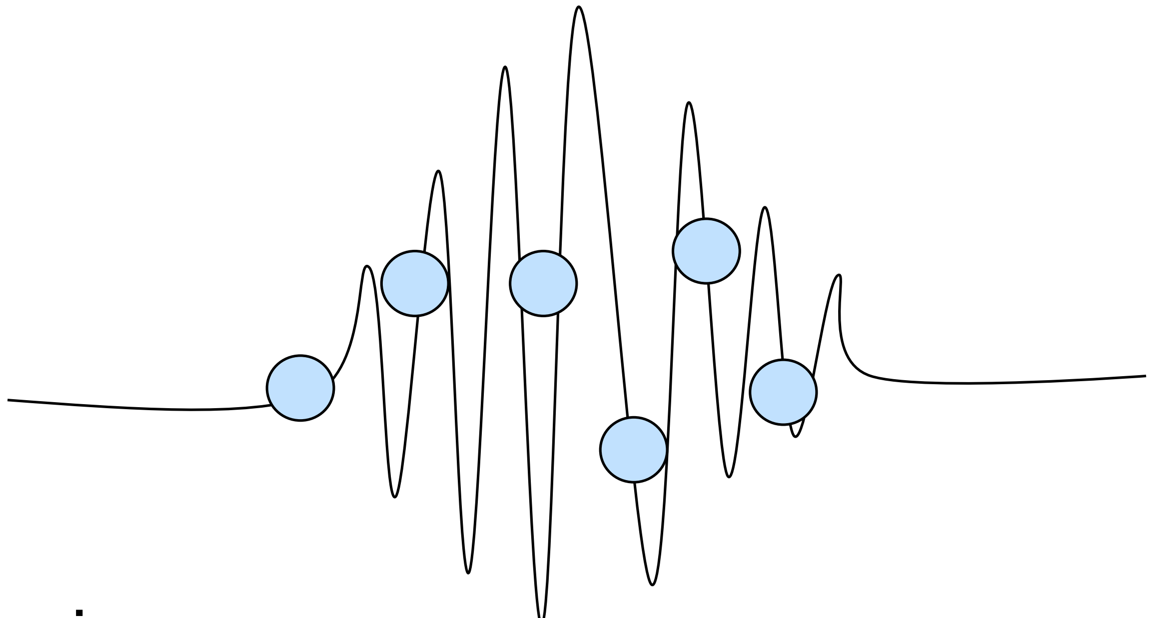
Kyoto University

Paris, 20 February 2018

Quantum Computing

- ✓ Computation paradigm based on the laws of quantum mechanics



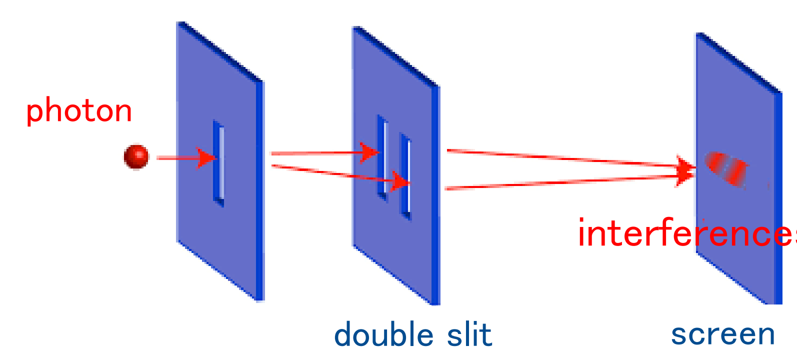


1 photon

quantum mechanics:

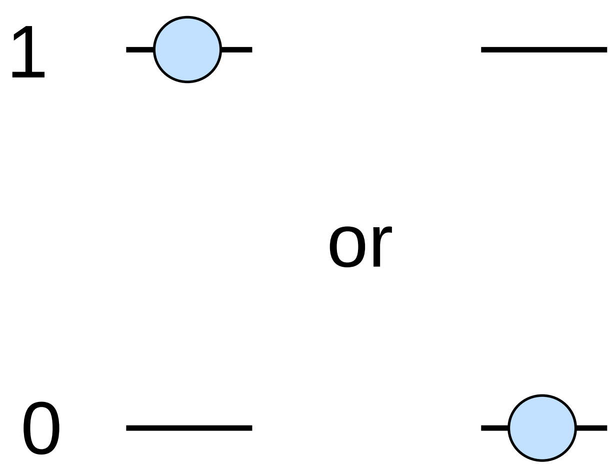
The position of a photon is described by ~~a probability distribution~~ a wave function

Double-slit experiment:



Quantum Mechanics: Discrete Case

1 bit of information



1 quantum bit (qubit) of information

<u>wave function</u> over 0 and 1
(<u>quantum superposition</u> over 0 and 1)

one 2-dimensional complex vector of norm 1

$\binom{\alpha}{\beta}$ with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$

$|\alpha|^2$ is the probability to observe the particle at state 0
$|\beta|^2$ is the probability to observe the particle at state 1

example: $\begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$ observing the qubit gives 0 with probability ½ and 1 with probability ½

Quantum Mechanics: Discrete Case

n bits of information

n quantum bits of information

one binary string of length n

quantum superposition over all the binary strings of length n

one 2^n-dimensional complex vector of norm 1

$$\begin{pmatrix} \alpha_1 \\ \\ \alpha_{2^n} \end{pmatrix} \quad \text{with } \alpha_i \in \mathbb{C} \text{ and } \sum_i |\alpha_i|^2 = 1$$

$|\alpha_i|^2$ is the probability to observe the i-th binary string

- ✓ Quantum information is attractive since it can store and manipulate an exponentially large amount of information (as a quantum superposition)
- ✓ Observing the quantum particles, however, does not give more than a random string (with probabilities depending of the coefficients in the superposition)
- ✓ But since the coefficients can be negative we can exploit interferences to amplify the probabilities of observing a good outcome and reducing the probability to observing a bad outcome

the art of quantum programming

Quantum Algorithms

What can we do with a quantum computer?

quantum algorithm for integer factoring [Shor 1994]

➡ breaks RSA cryptosystem

quantum algorithm for search [Grover 1996]

➡ fast for generic search problems

Quantum Algorithms

What can we do with a quantum c

quantum algorithm for integer factorin

➡ breaks RSA cryptosystem

quantum algorithm for search [Grover 1

➡ fast for generic search prol

- quantum algorithms with amplitude amplification [Bras
- quantum algorithms for adiabatic evolution [Fahri et al.
- quantum algorithms for element disjointness [Ambainis
- quantum algorithms for Gauss sums [van Dam et al. 2
- quantum algorithms for solving Pell's equation [Hallgre
- quantum algorithms for quantum simulations [Childs 2
- quantum algorithms for hidden subgroups [Kuperberg
- quantum algorithms for finding an unit group [Hallgren
- quantum algorithms for triangle finding [Magniez et al.
- quantum algorithms for computing knot invariants [Aha
- quantum algorithms for data streams [LG 2006]
- quantum algorithms for hidden nonlinear structures [Childs et al. 2007]
- quantum algorithms for evaluating NAND formulas [Fahri et al. 2007]
- quantum algorithms using span programs [Belovs 2011]
- quantum algorithms for matrix multiplication [LG 2012]
- quantum algorithms for matrix inversion [Ta-Shma 2013]
- quantum algorithms for the edit distance [Boroujeni et al. 2017]
- quantum algorithms for dynamic programming [Ambainis+ 2018]

Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@nist.gov. Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an *n*-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/3+o(1)})$ [252]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in croptography is given in [271]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, *cf.* [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Algorithm: Discrete-log
Speedup: Superpolynomial
Description: We are given three *n*-bit numbers *a*, *b*, and *N*, with the promise that $b = a^s \mod N$ for some *s*. The task is to find *s*. As shown by Shor [82], this can be achieved on a quantum computer in poly(*n*) time. The fastest known classical algorithm requires time superpolynomial in *n*. By similar techniques to those in [82], quantum computers can solve the discrete logarithm problem on elliptic curves, thereby breaking elliptic curve cryptography [109]. The superpolynomial quantum speedup has also been extended to the discrete logarithm problem on semigroups [203, 204]. See also Abelian Hidden Subgroup.

Algorithm: Pell's Equation
Speedup: Superpolynomial
Description: Given a positive nonsquare integer *d*, Pell's equation is $x^2 - dy^2 = 1$. For any such *d* there are infinitely many pairs of integers (*x,y*) solving this equation. Let (x_1, y_1) be the pair that minimizes $x + y\sqrt{d}$. If *d* is an *n*-bit integer (*i.e.* $0 \leq d < 2^n$), (x_1, y_1) may in general require

278 entries (2019/2/11)

Quantum Distributed Computing

- ✓ Mostly been studied in the framework of 2-party communication complexity
- ✓ Relatively few results focusing on more than two parties:
  - ➢ exact quantum protocols for leader election on anonymous networks
    [Tani, Kobayashi, Matsumoto PODC'09]
  - ➢ study of quantum distributed algorithms on non-anonymous networks

[Gavoille, Kosowski, Markiewicz DISC'09] ← LOCAL model

no significant advantage reported

[Elkin, Klauck, Nanongkai, Pandurangan PODC'14] ← CONGEST model

negative results: shows impossibility of quantum distributed computing faster than classical distributed computing for many important problems (shortest paths, MST,…)

Question: can quantum distributed computing be useful?

- ✓ Yes, in the CONGEST model [LG and Magniez PODC 2018]
  sublinear-time quantum distributed algorithm for computing the diameter
- ✓ Maybe also in the LOCAL model [LG, Rosmanis and Nishimura STACS 2019]
  evidences that quantum can be superior to classical

Quantum CONGEST model

Quantum CONGEST model

CONGEST model where quantum bits can be sent instead of usual bits

one quantum bit (qubit) = one quantum particle (e.g., one photon)

- ✓ can be created using a laser and sent using optical fibers
- ✓ generalizes the concept of bit (hence quantum distributed computing can trivially simulate classical distributed computing)

More formally:

- ✓ network G=(V,E) of n nodes (all nodes have distinct identifiers)
- ✓ each node knows the identifiers of all its neighbors
- ✓ synchronous communication between adjacent nodes: one message of O(log n) qubits per round
- ✓ each node is a quantum processor (i.e., a quantum computer)

Complexity: the number of rounds needed for the computation

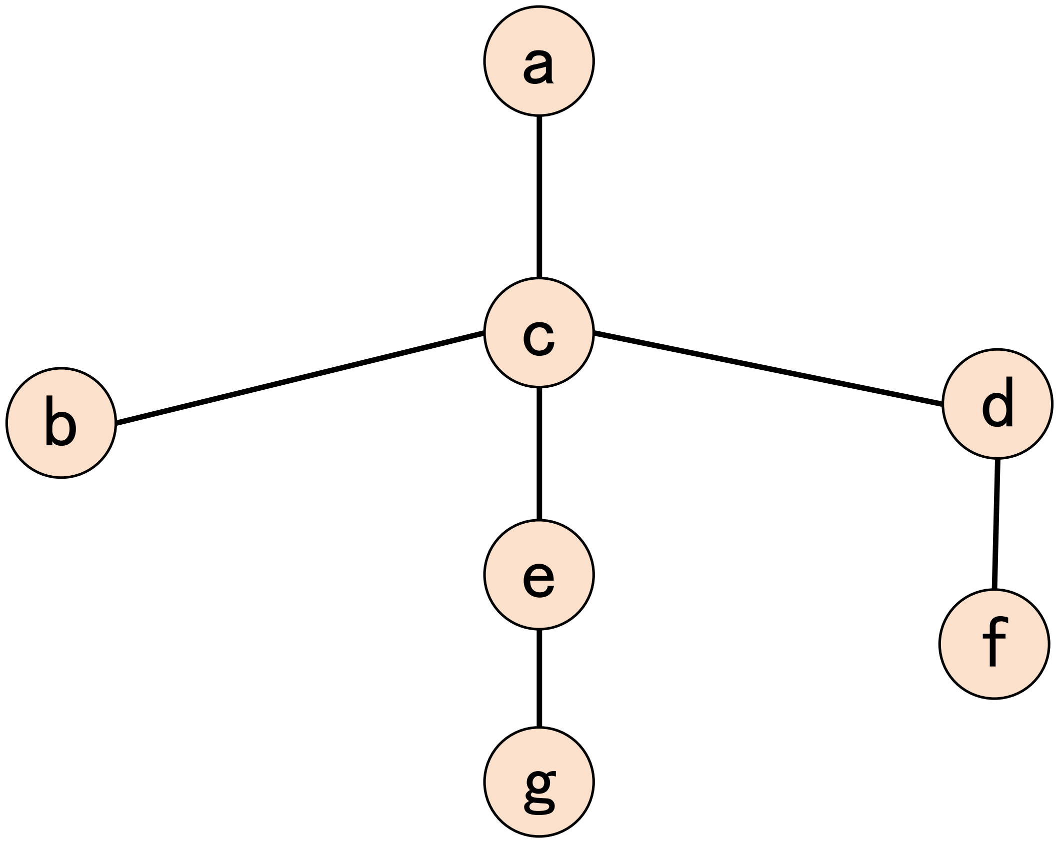
Diameter and Eccentricity

Consider an undirected and unweighted network G = (V,E) with n nodes

The <u>diameter</u> of the graph is the maximum distance between two nodes

$$D = \max_{u,v \in V} \{d(u,v)\}$$

d(u,v) = distance between u and v



Diameter and Eccentricity

Consider an undirected and unweighted network G = (V,E) with n nodes

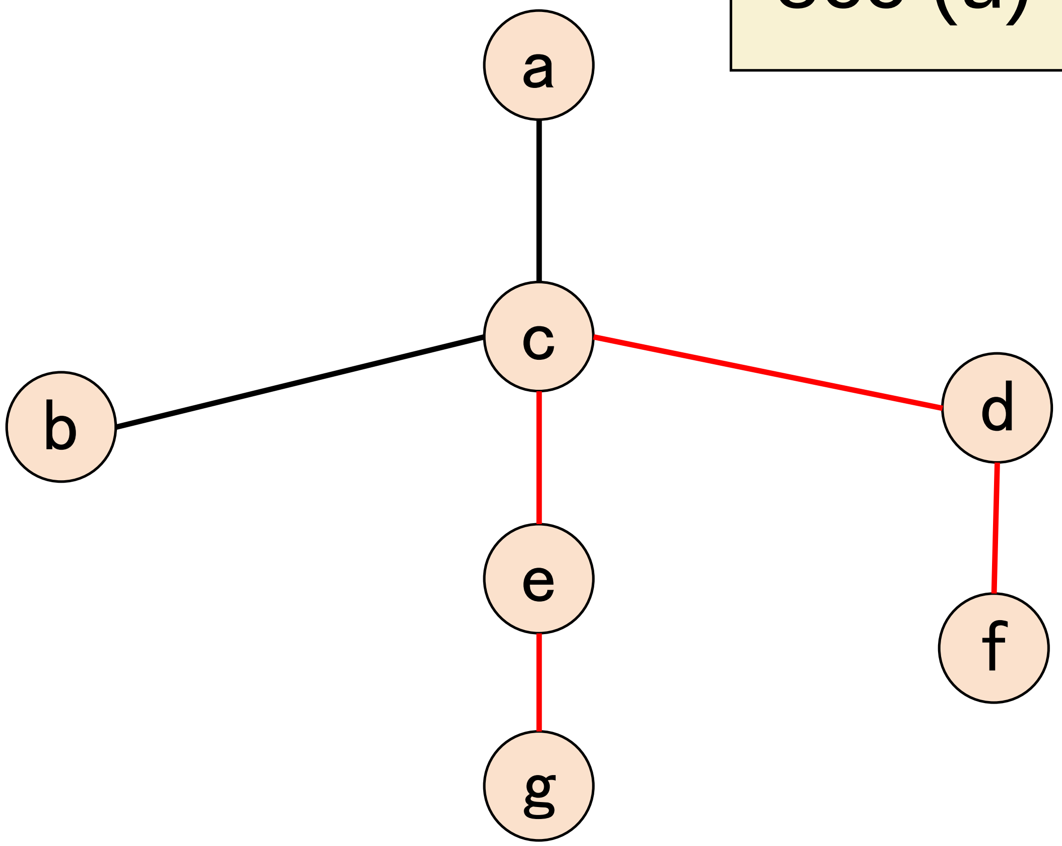
The diameter of the graph is the maximum distance between two nodes

$$D = \max_{u,v \in V} \{d(u,v)\} = \max_{u \in V} \{ecc(u)\}$$

d(u,v) = distance between u and v

The eccentricity of a node u is defined as

$$ecc(u) = \max_{v \in V} \{d(u,v)\}$$



D = 4

ecc (a) = 3
ecc (b) = 3
ecc (c) = 2
ecc (d) = 3
ecc (e) = 3
ecc (f) = 4
ecc (g) = 4

d(a,a) = 0
d(a,b) = 2
d(a,c) = 1
d(a,d) = 2
d(a,e) = 2
d(a, f) = 3
d(a,g) = 3

Diameter and Eccentricity

Consider an undirected and unweighted network G = (V,E) with n nodes

The diameter of the graph is the maximum distance between two nodes

$$D = \max_{u,v \in V} \{d(u,v)\} = \max_{u \in V} \{ecc(u)\}$$

d(u,v) = distance between u and v

The eccentricity of a node u is defined as

$$ecc(u) = \max_{v \in V} \{d(u,v)\}$$

In the classical (i.e., non-quantum) CONGEST model:

- ✓ ecc(u) can be computed in O(D) rounds by constructing a Breadth-First Search tree rooted at u
- ✓ computing the diameter (i.e., the maximum eccentricity) requires Θ(n) rounds even for constant D
  [Frischknecht+12, Holzer+12, Peleg+12, Abboud+16]

Computation of the Diameter in the CONGEST model

main result: sublinear-round quantum computation of the diameter whenever D=o(n)
(our algorithm uses $O((\log n)^2)$ qubits of quantum memory per node)

first gap between classical and quantum in the CONGEST model for a major problem of interest to the distributed computing community

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$
Exact computation (lower bounds)	$\tilde{\Omega}(n)$ [Frischknecht+12]	$\tilde{\Omega}(\sqrt{nD})$ [conditional]

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

condition: holds for quantum distributed algorithms using only polylog(n) qubits of memory per node

	Classical	Quantum (our results)
3/2-approximation (upper bounds)	$O(\sqrt{n}+D)$ [Lenzen+13, Holzer+14]	$O(\sqrt[3]{nD}+D)$
(3/2-ε)-approximation (lower bounds)	$\tilde{\Omega}(n)$ [Holzer+12, Abboud+16]	$\tilde{\Omega}(\sqrt{n}+D)$ [unconditional]

Our Upper Bound

main result: sublinear-round quantum computation of the diameter whenever D=o(n)

(our algorithm uses $O((\log n)^2)$ qubits of quantum memory per node)

first gap between classical and quantum in the CONGEST model for a major problem of interest to the distributed computing community

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

Quantum Distributed Computation of the Diameter

Computation of the diameter (decision version)

Given an integer d, decide if diameter ≥ d

there is a vertex u such that ecc (u) ≥ d

This is a search problem

Idea: use the technique called “quantum search”

Centralized Quantum Search: Grover's algorithm

Let f: X → {0,1} be a Boolean function given as a black box

Goal: find an element x ∈ X such that f(x) = 1

Classically this can be done using $O(|X|)$ calls to the black box
("brute force search: try all the elements x")

There is a quantum centralized algorithm solving this problem with $O(\sqrt{|X|})$ calls to the black box

Quantum search
[Grover 96]

Example of application: quantum algorithm for Boolean satisfiability (SAT)

SAT: given a Boolean formula f of poly size on M variables, find a satisfying assignment (if such an assignment exists)

Centralized Quantum Search: Grover's algorithm

Let f: X → {0,1} be a Boolean function given as a black box

Goal: find an element x ∈ X such that f(x) = 1

Classically this can be done using $O(|X|)$ calls to the black box
("brute force search: try all the elements x")

> There is a quantum centralized algorithm solving this problem with $O(\sqrt{|X|})$ calls to the black box

Quantum search
[Grover 96]

Example of application: quantum algorithm for Boolean satisfiability (SAT)

> SAT: given a Boolean formula f of poly size on M variables, find a satisfying assignment (if such an assignment exists)

X = set of all possible assignments ← $|X| = 2^M$

Black box: computes f(x) from x ← poly(M) time

Centralized Quantum Search: Grover’s algorithm

Let f: X → {0,1} be a Boolean function given as a black box

Goal: find an element x ∈ X such that f(x) = 1

Classically this can be done using $O(|X|)$ calls to the black box
(“brute force search: try all the elements x”)

> There is a quantum centralized algorithm solving this problem with $O(\sqrt{|X|})$ calls to the black box

Quantum search [Grover 96]

<u>Example of application</u>: quantum algorithm for Boolean satisfiability (SAT)

> SAT: given a Boolean formula f of poly size on M variables, find a satisfying assignment (if such an assignment exists)

X = set of all possible assignments ← $|X| = 2^M$

Black box: computes f(x) from x ← poly(M) time

⇨ Quantum search solves SAT in $O(2^{M/2} \times poly(M))$ time

Quantum Distributed Computation of the Diameter

Define the function f: V → {0,1} such that $f(u) = \begin{cases} 1 & \text{if } ecc(u) \geq d \\ 0 & \text{otherwise} \end{cases}$

Goal: find u such that f(u) = 1 (or report that no such vertex exist)

There is a quantum <u>centralized</u> algorithm for this search problem using $O(\sqrt{n})$ calls to a black box evaluating f

Quantum search [Grover 96]

Computation of the diameter (decision version)

Given an integer d, decide if diameter ≥ d

there is a vertex u such that ecc (u) ≥ d

This is a search problem

Idea: use the technique called “quantum search”

Quantum Distributed Computation of the Diameter

Define the function f: V → {0,1} such that f(u) = $\begin{cases} 1 \text{ if ecc (u)} \geq d \\ 0 \text{ otherwise} \end{cases}$

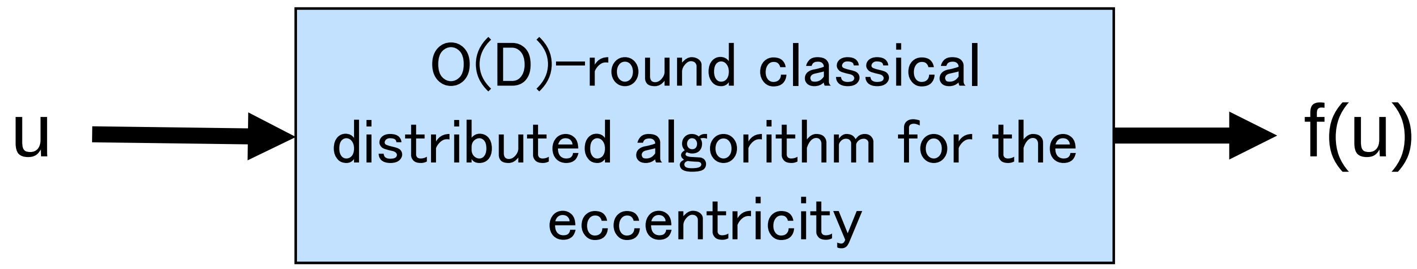
Goal: find u such that f(u) = 1 (or report that no such vertex exist)

There is a quantum centralized algorithm for this search problem using $O(\sqrt{n})$ calls to a black box evaluating f

Quantum search [Grover 96]

Quantum distributed algorithm computing the diameter

- ✓ The network elects a leader
- ✓ The leader locally runs this centralized quantum algorithm for search, in which each call to the black box is implemented by executing the standard O(D)-round classical algorithm computing the eccentricity



Quantum Distributed Computation of the Diameter

Classically in O(D) rounds it is possible to simultaneously compute the eccentricities of D vertices [Peleg+12]

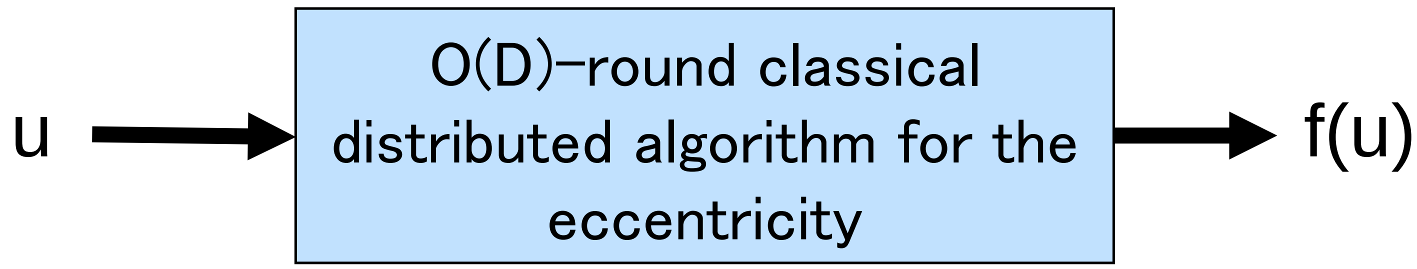
Thus we can instead do a Grover search over groups of D vertices (there are n/D groups) in

$$O(\sqrt{n/D} \times \mathrm{D}) = O(\sqrt{nD}) \text{ rounds}$$

Quantum distributed algorithm computing the diameter

- ✓ The network elects a leader
- ✓ The leader locally runs this centralized quantum algorithm for search, in which each call to the black box is implemented by executing the standard O(D)-round classical algorithm computing the eccentricity

Complexity: $O(\sqrt{n} \times \mathrm{D})$ rounds



With further work, the complexity can be reduced to $O(\sqrt{nD})$ rounds

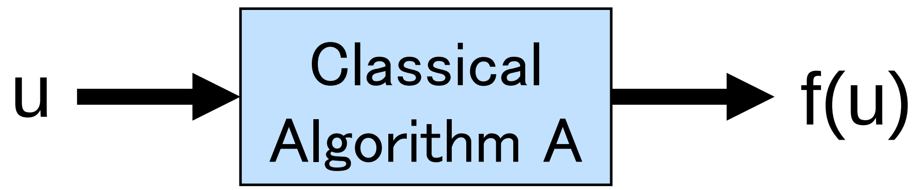
Subtlety: Quantum Access to the Black Box

Define the function f: V → {0,1} such that $f(u) = \begin{cases} 1 \text{ if ecc}(u) \geq d \\ 0 \text{ otherwise} \end{cases}$

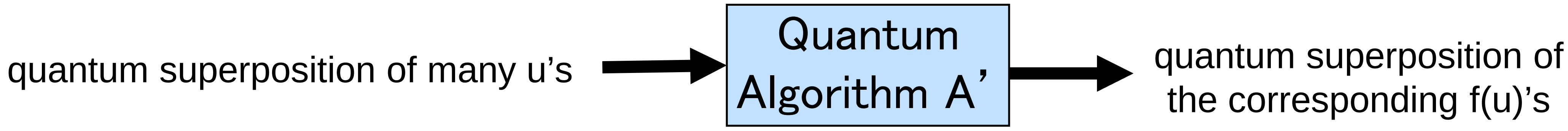
Goal: find u such that f(u) = 1 (or report that no such vertex exist)

There is a quantum centralized algorithm for this search problem using $O(\sqrt{n})$ calls to a black box evaluating f

Quantum search [Grover 96]



Subtlety: quantum search requires accessing the black box “in superposition”



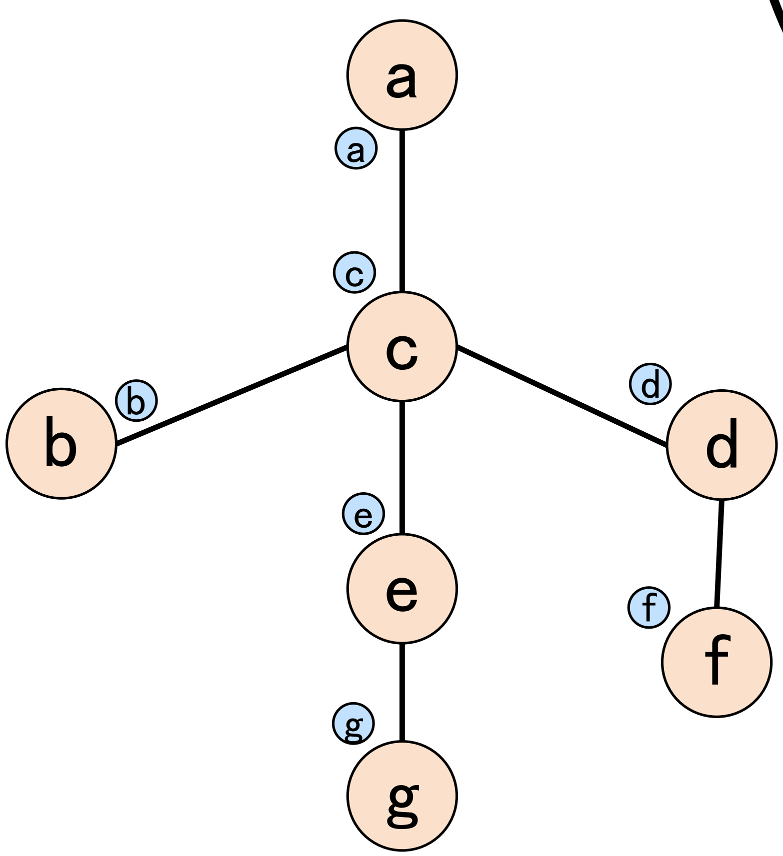
Why does this not introduce congestions?

Implementati

$$\sum_{u \in V} \alpha_u |u\rangle |0\rangle$$

Node a introduces 1 register	$\sum_{u \in V} \alpha_u \|u\rangle_a \|0\rangle$
Node a applies CNOTS	$\sum_{u \in V} \alpha_u \|u\rangle_a \|u\rangle$
Node a sends the second register to c	$\sum_{u \in V} \alpha_u \|u\rangle_a \|u\rangle_c$
Node c introduces 3 registers	$\sum_{u \in V} \alpha_u \|u\rangle_a \|u\rangle_c \|0\rangle \|0\rangle \|0\rangle$
Node c applies CNOTS	$\sum_{u \in V} \alpha_u \|u\rangle_a \|u\rangle_c \|u\rangle \|u\rangle \|u\rangle$
Node c sends the registers to b,e,d	$\sum_{u \in V} \alpha_u \|u\rangle_a \|u\rangle_c \|u\rangle_b \|u\rangle_e \|u\rangle_d$
……	

<u>Example</u>:
V={a,b,c,d,e,f,g}
here leader = node a



Initially node a owns $\sum_{u \in V} \alpha_u |u\rangle_a$

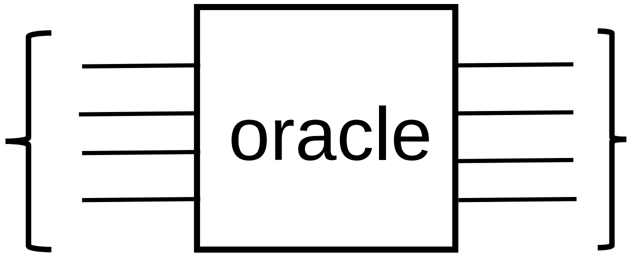
1. "<u>Broadcast</u>" this state, which gives [ecc(a) ≤ D rounds]

$$\sum_{u \in V} \alpha_u |u\rangle_a |u\rangle_b |u\rangle_c |u\rangle_d |u\rangle_e |u\rangle_f |u\rangle_g$$

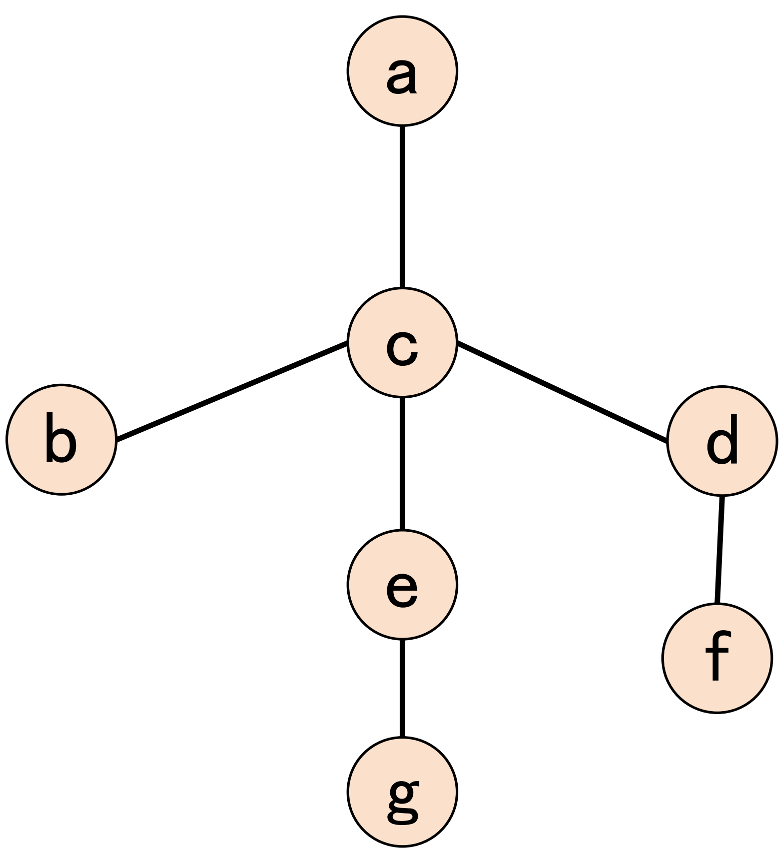
2. The nodes implement the classical protocol for computing the eccentricity of u, which gives [O(D) rounds]

$$\sum_{u \in V} \alpha_u |u\rangle_a |u\rangle_b |u\rangle_c |u\rangle_d |u\rangle_e |u\rangle_f |u\rangle_g |ecc(u)\rangle_a$$

Implementation of the Oracle in O(D) rounds

$\sum_{u \in V} \alpha_u |u\rangle_{\mathrm{a}} |0\rangle_{\mathrm{a}}$  $\sum_{u \in V} \alpha_u |u\rangle_{\mathrm{a}} |ecc(u)\rangle_{\mathrm{a}}$

V={a,b,c,d,e,f,g}



Initially node a owns $\sum_{u \in V} \alpha_u |u\rangle_{\mathrm{a}}$

1. "Broadcast" this state, which gives [ecc(a) ≤ D rounds]

$$\sum_{u \in V} \alpha_u |u\rangle_{\mathrm{a}} |u\rangle_{\mathrm{b}} |u\rangle_{\mathrm{c}} |u\rangle_{\mathrm{d}} |u\rangle_{\mathrm{e}} |u\rangle_{\mathrm{f}} |u\rangle_{\mathrm{g}}$$

2. The nodes implement the classical protocol for computing the eccentricity of u, which gives [O(D) rounds]

$$\sum_{u \in V} \alpha_u |u\rangle_{\mathrm{a}} |u\rangle_{\mathrm{b}} |u\rangle_{\mathrm{c}} |u\rangle_{\mathrm{d}} |u\rangle_{\mathrm{e}} |u\rangle_{\mathrm{f}} |u\rangle_{\mathrm{g}} |ecc(u)\rangle_{\mathrm{a}}$$

3. The nodes revert Step 1 [ecc(a) ≤ D rounds]

The Upper Bound

- ✓ We have just described a O($\sqrt{n}$ x D)-round quantum distributed algorithm for computing (with high probability) the diameter
- ✓ With further work, the complexity can be reduced to O($\sqrt{nD}$) rounds

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$

The Lower Bounds

	Classical	Quantum (our results)
Exact computation (lower bounds)	$\widetilde{\Omega}(n)$ [Frischknecht+12]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional] $\widetilde{\Omega}(\sqrt{nD})$ [conditional]

via two-party communication complexity of the disjointness function (DISJ)

classical lower bound

- ✓ reduce DISJ to the distributed computation of diameter [Frischknecht+12]
- ✓ the (two-party) communication complexity of DISJ_n is Ω(n) bits [Kalyanasundaram+92]

unconditional quantum lower bound

- ✓ same reduction from DISJ to the distributed computation of diameter
- ✓ the (two-party) communication complexity of DISJ_n is $\Omega(\sqrt{n})$ qubits [Razborov03]

conditional quantum lower bound

- ✓ Claim: if the quantum distributed algorithm for diameter uses few quantum memory per node, then the reduction can be adjusted to give a two-party protocol for DISJ using few messages (idea: send communication in batches)
- ✓ the (two-party) r-message quantum communication complexity of DISJ_n is $\Omega(n/r + r)$ qubits [Braverman+15]

Summary of the first part

main result: sublinear-round quantum computation of the diameter in the CONGEST model (when D is small enough)

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$
Exact computation (lower bounds)	$\widetilde{\Omega}(n)$ [Frischknecht+12]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional] $\widetilde{\Omega}(\sqrt{nD})$ [conditional]

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

3/2-approximation (upper bounds)	$O(\sqrt{n}+D)$ [Lenzen+13, Holzer+14]	$O(\sqrt[3]{nD}+D)$
(3/2-ε)-approximation (lower bounds)	$\widetilde{\Omega}(n)$ [Holzer+12, Abboud+16]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional]

Summary of the first part

Useful for problems in distributed computing where the bottleneck is a search problem

"Recipe" to build a quantum distributed algorithm
(even without knowing anything about quantum computation):

If you need to find a good element among N candidates and have a r-round procedure to check if an element is good, there is a $O(r\sqrt{N})$-round quantum algorithm for this search problem.

- ✓ Our upper bounds are obtained by showing how to implement quantum search in a distributed setting
- ✓ Interesting research direction: apply this technique to other problems in distributed computing

Quantum Distributed Computing

- ✓ Mostly been studied in the framework of 2-party communication complexity
- ✓ Relatively few results focusing on more than two parties:
  - ➢ exact quantum protocols for leader election on anonymous networks
    [Tani, Kobayashi, Matsumoto PODC'09]
  - ➢ study of quantum distributed algorithms on non-anonymous networks

[Gavoille, Kosowski, Markiewicz DISC'09] ⟵ LOCAL model

no significant advantage reported

[Elkin, Klauck, Nanongkai, Pandurangan PODC'14] ⟵ CONGEST model

negative results: shows impossibility of quantum distributed computing faster than classical distributed computing for many important problems (shortest paths, MST,…)

Question: can quantum distributed computing be useful?

- ✓ Yes, in the CONGEST model [LG and Magniez PODC 2018]
  sublinear-time quantum distributed algorithm for computing the diameter
- ✓ Maybe also in the LOCAL model [LG, Rosmanis and Nishimura STACS 2019]
  evidences that quantum can be superior to classical

Quantum Distributed Computing

- ✓ Mostly been studied in the framework of 2-party communication complexity
- ✓ Relatively few results focusing on more than two parties:
  - ➢ exact quantum protocols for leader election on anonymous networks
    [Tani, Kobayashi, Matsumoto PODC'09]
  - ➢ study of quantum distributed algorithms on non-anonymous networks

[Gavoille, Kosowski, Markiewicz DISC'09] ⟵ LOCAL model

no significant advantage reported

[Elkin, Klauck, Nanongkai, Pandurangan PODC'14] ⟵ CONGEST model

negative results: shows impossibility of quantum distributed computing faster than classical distributed computing for many important problems (shortest paths, MST,…)

Question: can quantum distributed computing be useful?

- ✓ Yes, in the CONGEST model [LG and Magniez PODC 2018]
  sublinear-time quantum distributed algorithm for computing the diameter
- ✓ Maybe also in the LOCAL model [LG, Rosmanis and Nishimura STACS 2019]
  evidences that quantum can be superior to classical

Quantum LOCAL model

Messages can now have arbitrary length

Quantum CONGEST model

- ✓ network G=(V,E) of n nodes (all nodes have distinct identifiers)
- ✓ each node knows the identifiers of all its neighbors
- ✓ synchronous communication between adjacent nodes: one message of O(log n) qubits per round
- ✓ each node is a quantum processor (i.e., a quantum computer)

Complexity: the number of rounds needed for the computation

Quantum LOCAL model

- ✓ network G=(V,E) of n nodes (all nodes have distinct identifiers)
- ✓ each node knows the identifiers of all its neighbors
- ✓ synchronous communication between adjacent nodes: one message of arbitrary length per round
- ✓ each node is a quantum processor (i.e., a quantum computer)

Complexity: the number of rounds needed for the computation

Superiority of the Quantum LOCAL model

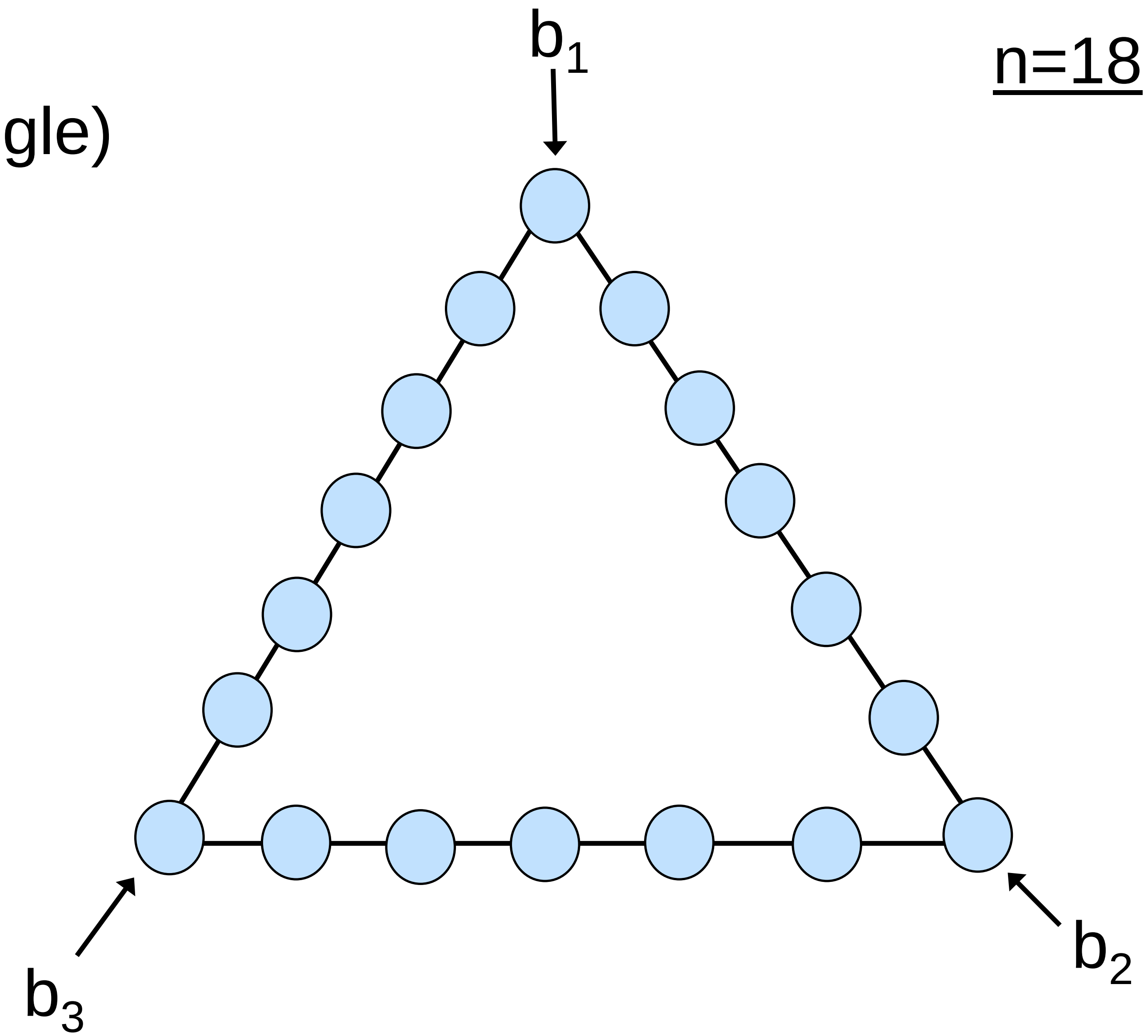
[LG, Rosmanis and Nishimura STACS 2019]

multiple of 3

Consider a ring of size n (seen as a triangle)

Each "corner" gets a bit as input

Each node will output one bit



Superiority of the Quantum LOCAL model

[LG, Rosmanis and Nishimura STACS 2019]

Consider a ring of size n (seen as a triangle) ← multiple of 3

Each "corner" gets a bit as input

Each node will output one bit

Define the following four bits:

$m_R = z_2 \oplus z_4 \oplus z_6$

(parity of the outputs of the nodes of even index on the right)

$m_B = z_8 \oplus z_{10} \oplus z_{12}$

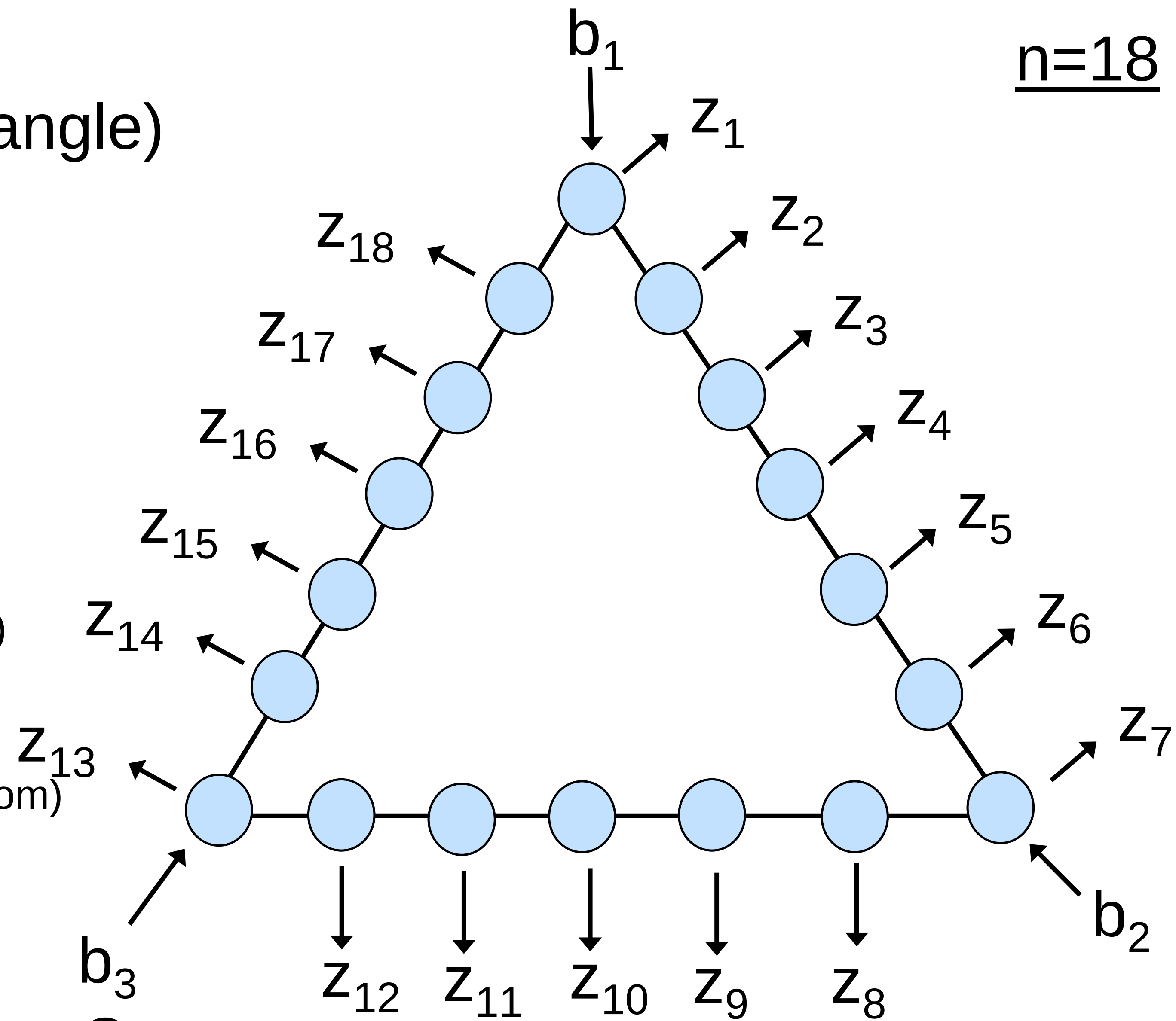
(parity of the outputs of the nodes of even index on the bottom)

$m_L = z_{14} \oplus z_{16} \oplus z_{18}$

(parity of the outputs of the nodes of even index on the left)

$m_{odd} = z_1 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_9 \oplus z_{11} \oplus z_{13} \oplus z_{15} \oplus z_{17}$

(parity of the outputs of all the nodes of odd index)



1. Each node creates 1 qubit
2. Each node makes its qubit interact with its two neighbors (2 rounds)
3. Each non-corner node makes a “measurement in the X basis” to its qubit, and outputs the bit corresponding to the measurement outcome
4. Each corner node makes a “measurement in the X basis” to its qubit if its input bit is 0, or makes a “measurement in the Y basis” to its qubit if its input bit is 1, and outputs the bit corresponding to the measurement outcome

2019]

18

Define the following four bits:

$$m_R = z_2 \oplus z_4 \oplus z_6$$

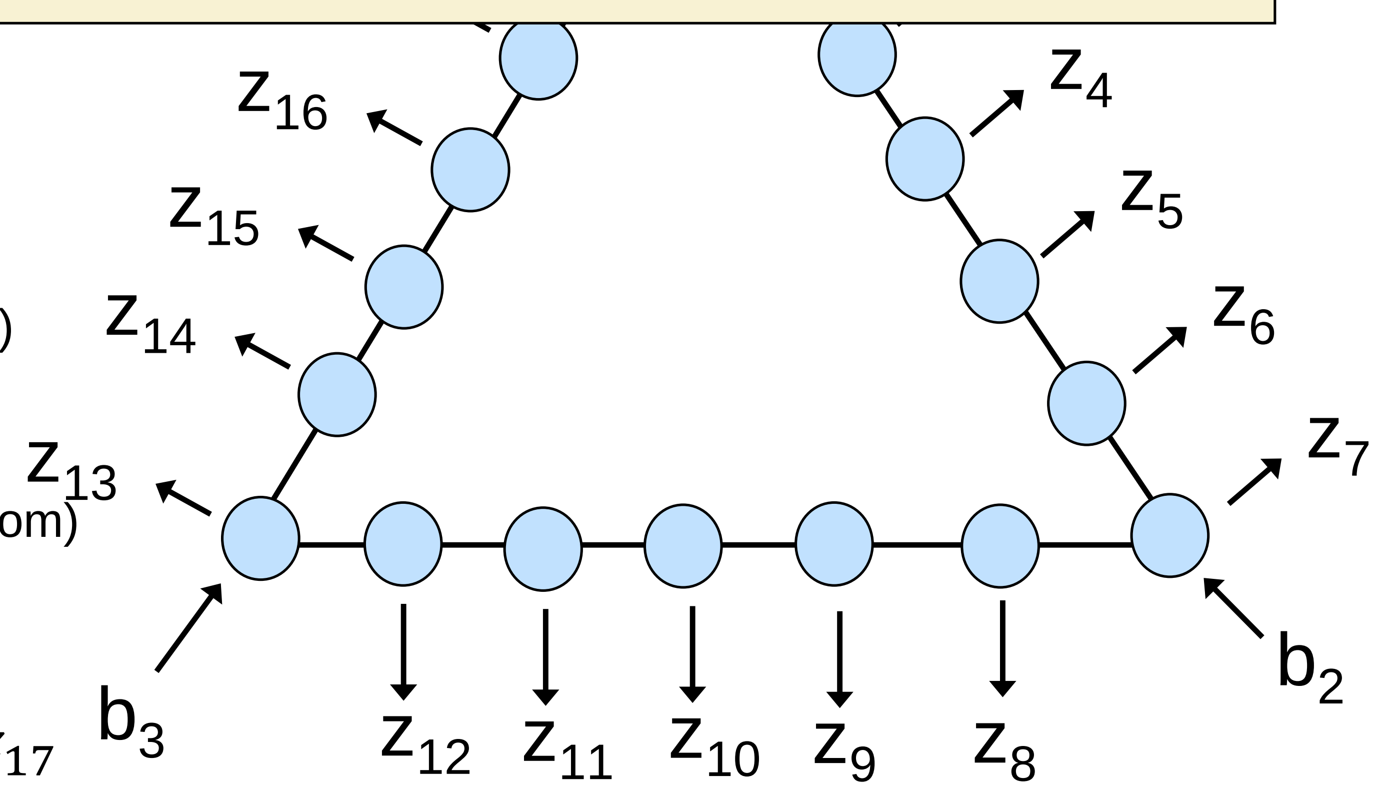
(parity of the outputs of the nodes of even index on the right)

$$m_B = z_8 \oplus z_{10} \oplus z_{12}$$

(parity of the outputs of the nodes of even index on the bottom)

$$m_L = z_{14} \oplus z_{16} \oplus z_{18}$$

$$m_{odd} = z_1 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_9 \oplus z_{11} \oplus z_{13} \oplus z_{15} \oplus z_{17}$$



Claim 1: There is a 2-round quantum algorithm that outputs the uniform distribution over all binary strings $(z_1, z_2, \ldots, z_n) \in \{0,1\}^n$ satisfying the following condition:

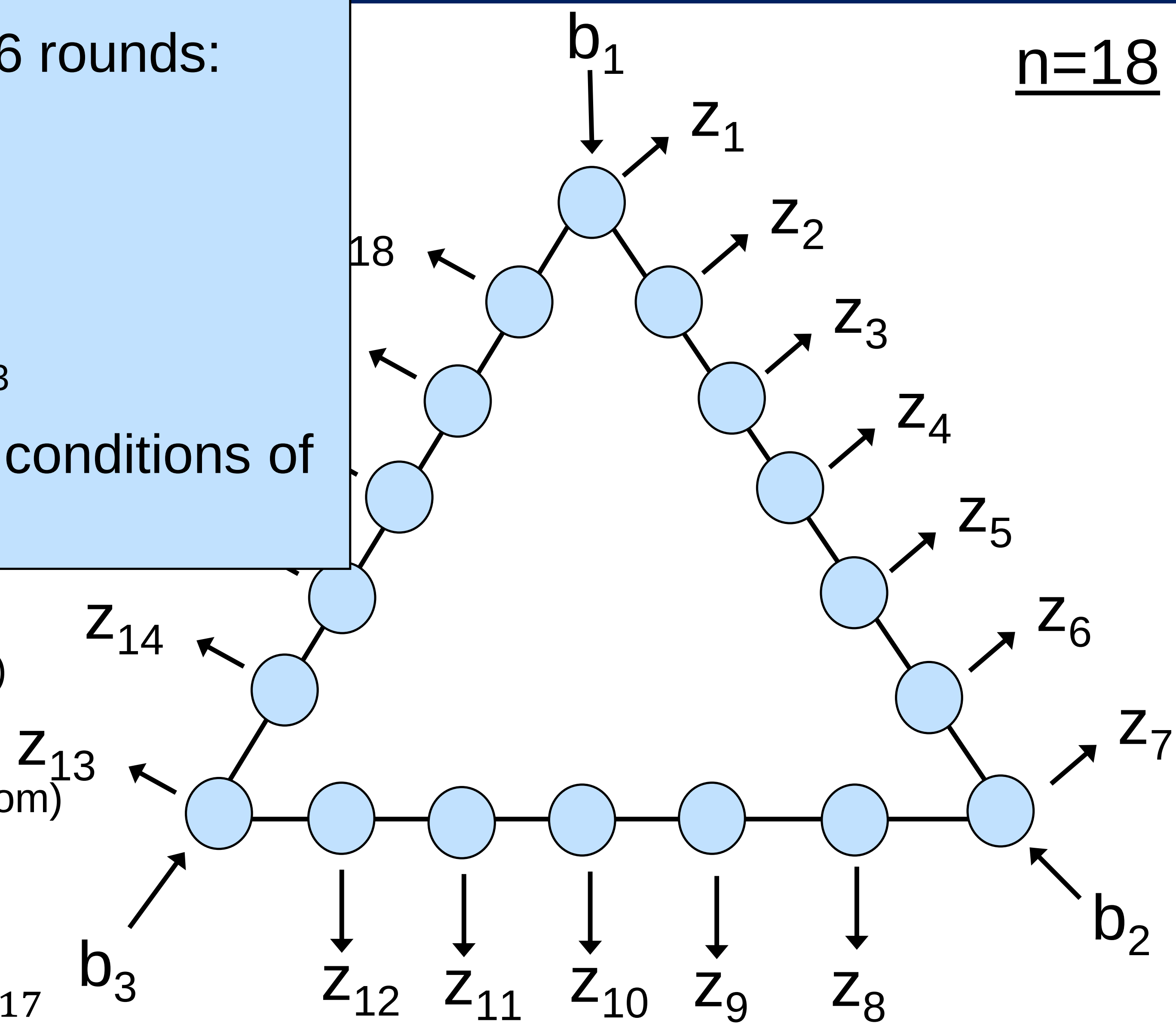
$$\begin{cases} m_{odd} = 0 & \text{if } (b_1, b_2, b_3) = (0,0,0) \\ m_{odd} \oplus m_R = 1 & \text{if } (b_1, b_2, b_3) = (1,1,0) \\ m_{odd} \oplus m_B = 1 & \text{if } (b_1, b_2, b_3) = (0,1,1) \\ m_{odd} \oplus m_L = 1 & \text{if } (b_1, b_2, b_3) = (1,0,1) \end{cases}.$$

Claim 2: In the LOCAL model, any classical algorithm that outputs the same distribution must use at least n/6 rounds.

- ✓ In any classical protocol using less than n/6 rounds:

  m_R is an affine function of b_1 and b_2
  m_B is an affine function of b_2 and b_3
  m_L is an affine function of b_1 and b_3
  m_{odd} is an affine function of b_1, b_2 and b_3

- ✓ Such functions cannot satisfy all the linear conditions of Claim 1



$m_R = z_2 \oplus z_4 \oplus z_6$
(parity of the outputs of the nodes of even index on the right)

$m_B = z_8 \oplus z_{10} \oplus z_{12}$
(parity of the outputs of the nodes of even index on the bottom)

$m_L = z_{14} \oplus z_{16} \oplus z_{18}$

$m_{odd} = z_1 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_9 \oplus z_{11} \oplus z_{13} \oplus z_{15} \oplus z_{17}$

Claim 1: There is a 2-round quantum algorithm that outputs the uniform distribution over all binary strings $(z_1, z_2, \dots, z_n) \in \{0,1\}^n$ satisfying the following condition:

$$\begin{cases} m_{odd} = 0 & \text{if } (b_1, b_2, b_3) = (0,0,0) \\ m_{odd} \oplus m_R = 1 & \text{if } (b_1, b_2, b_3) = (1,1,0) \\ m_{odd} \oplus m_B = 1 & \text{if } (b_1, b_2, b_3) = (0,1,1) \\ m_{odd} \oplus m_L = 1 & \text{if } (b_1, b_2, b_3) = (1,0,1) \end{cases}.$$

Conclusions

- ✓ We have shown that in the CONGEST model the diameter of the network can be computed faster using quantum distributed algorithms (for constant diameter: $\Theta(\sqrt{n})$ rounds quantumly vs. $\Theta(n)$ rounds classically)

- ✓ We have shown that in the LOCAL model quantum distributed algorithms can also be faster, at least for some computational task (for our ring problem: 2 rounds quantumly vs. $\Theta(n)$ rounds classically)

Interesting research directions:

- ✓ Consider other applications of quantum distributed algorithms in the CONGEST model

- ✓ Find one interesting application of quantum distributed algorithms in the LOCAL model

- ✓ Consider other models (e.g., asynchronous computation) in the quantum setting