Multi-Party Quantum Communication Complexity with Routed Messages

Seiichiro Tani[†], Masaki Nakanishi[‡], Shigeru Yamashita[‡]

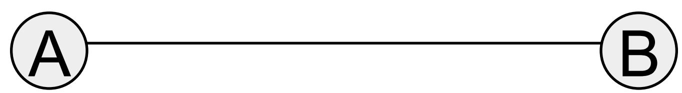
† **NTT Communication Science Labs**

‡ **Nara Institute of Science and Technology**

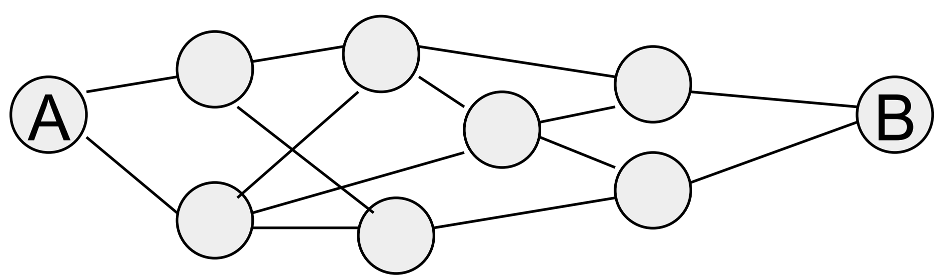
COCOON 2008

Motivation

- The amount of quantum communication needed to compute functions for distributed inputs has been intensively studied in the context of communication complexity.
- Most works assumes the standard two party model.



- On an actual communication network, however, two parties are usually connected by multiple paths on which there can be multiple parties.
- Only a few results are known in this case.



Summary of our results

- A general lower bound technique for the quantum communication complexity of a function that depends on the inputs given to two parties on an k-party network of any topology.

- Application of the technique to lower-bound the communication complexity of computing the distinctness problem on an k-party ring.

  - Almost matching upper bounds are also given.

List of Contents

- Statements of our results
  - General lower bound
  - Application
- Proof of general lower bound
  - Two lemmas
- Application
  - Problem definition
  - Lower bound
  - Upper bound
- Summary

Our results (1/3): A general lower bound technique

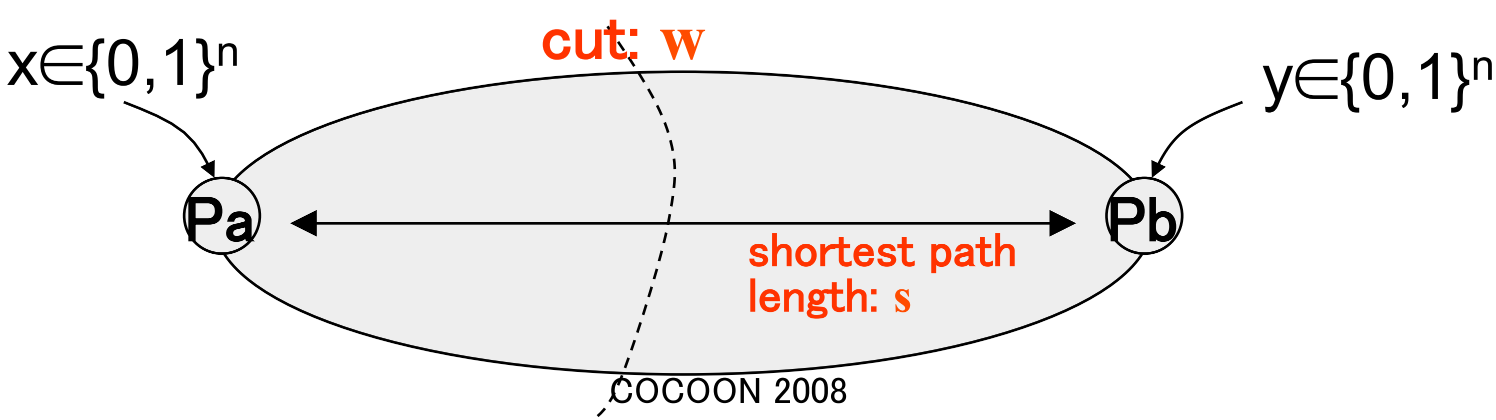
Theorem:

Suppose that $x,y \in \{0,1\}^n$ are given to two parties Pa and Pb, respectively, on network N of any topology.

The total quantum communication complexity over all links of computing a Boolean function $f(x,y)$ with bounded error is:

$$\Omega(s(Q_{1/3}(f(x,y)) - \log(\min\{s,n\}))/\log w),$$

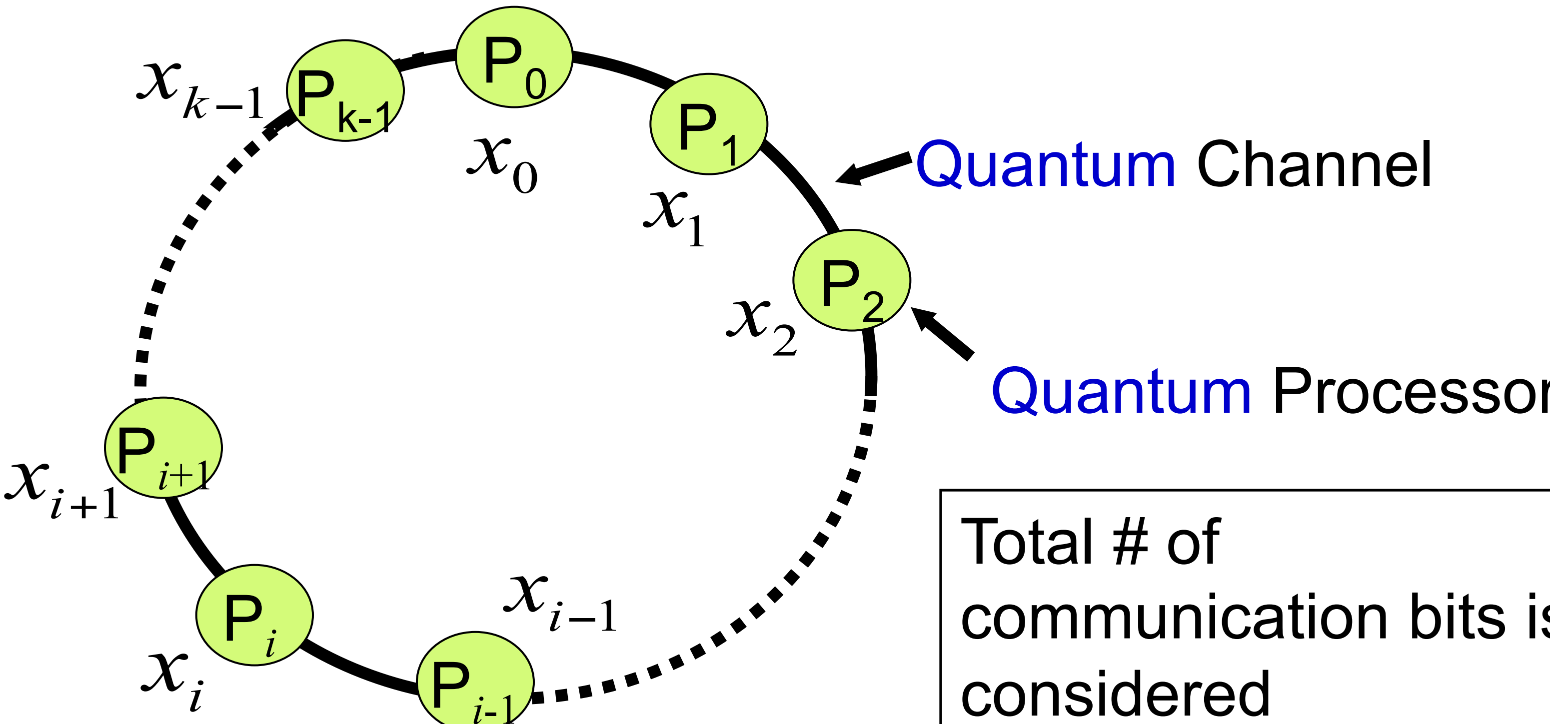
where $Q_{1/3}(f(x,y))$ is the quantum communication complexity of $f(x,y)$ in the ordinary two-party case.



Our results (2/3): Application

Our Problem: Distinctness on a ring

- Each of *k* parties has input $x_i \in \{0,\ldots,L-1\}$
- Determine whether two or more parties have the same value or not ($i \neq j \rightarrow x_i \neq x_j$)



Total # of communication bits is considered

Our results (3/3): Application

Complexity of computing Distinctness on an k-party ring.

L	Upper Bound	Lower Bound
$L \leq k (\log k)^2$	$O(k L^{1/2})$	$\Omega(k L^{1/2} / \log k)$ (or $\Omega(k^{3/2})$)
$n (\log n)^2 < L$	$O(k(k^{1/2} \log k + \log\log L))$	$\Omega(k(k^{1/2} + \log \log L))$

Our bounds are tight up to a log multiplicative factor
In particular, they are optimal $\Theta(k^{3/2})$ for $L = \Theta(k)$.

List of Contents

- Statements of our results
  - General lower bound
  - Application
- Proof of general lower bound
  - Two lemmas
- Application
  - Problem definition
  - Lower bound
  - Upper bound
- Summary

General Lower Bound Theorem.

Theorem:

Suppose that n-bit strings x and y are given to two parties Pa and Pb, respectively, on network G of any topology.

The total quantum communication complexity $Q^G_{1/3}(f)$ over all links of computing a Boolean function f(x,y) with bounded error is:

$$\Omega(s(Q_{1/3}(f(x,y)) - \log(\min\{s,n\}))/\log w),$$

where $Q_{1/3}(f(x,y))$ is the quantum communication complexity of f(x,y) in the ordinary two-party case.

By proving:

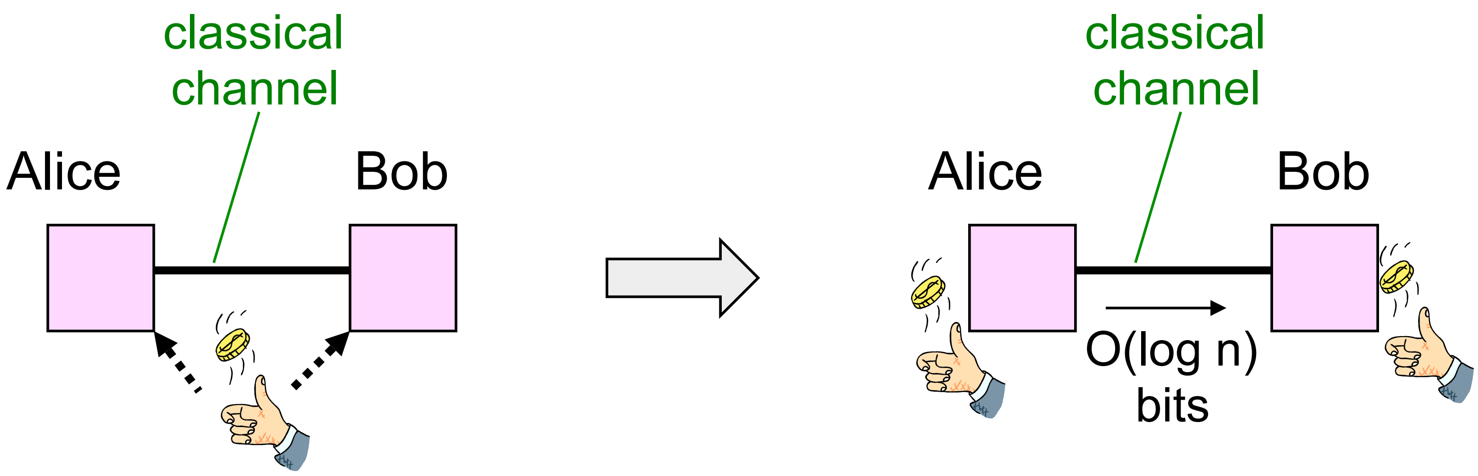
Lemma 1 $Q^G_{1/3}(f(x,y)) = \Omega(s(Q_{1/3}(f(x,y)) - \log n)/\log w)$

Lemma 2 $Q^G_{1/3}(f(x,y)) = \Omega(s(Q_{1/3}(f(x,y)) - \log s)/\log w)$

Public coin v.s. Private coin (1/2)

Theorem [Newman91]

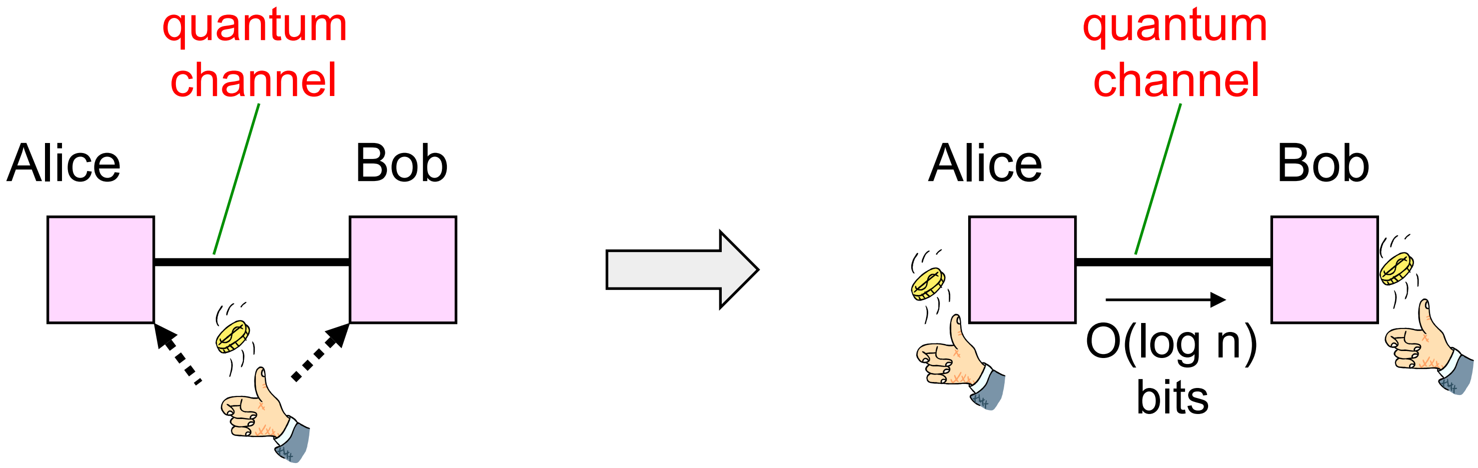
Any classical protocol using public coins with error probability at most 1/3
can be converted into
a protocol using only private coins with error probability at most 1/3
at the cost of O(log n) bits of additional communication,
where n is the number of input bits.



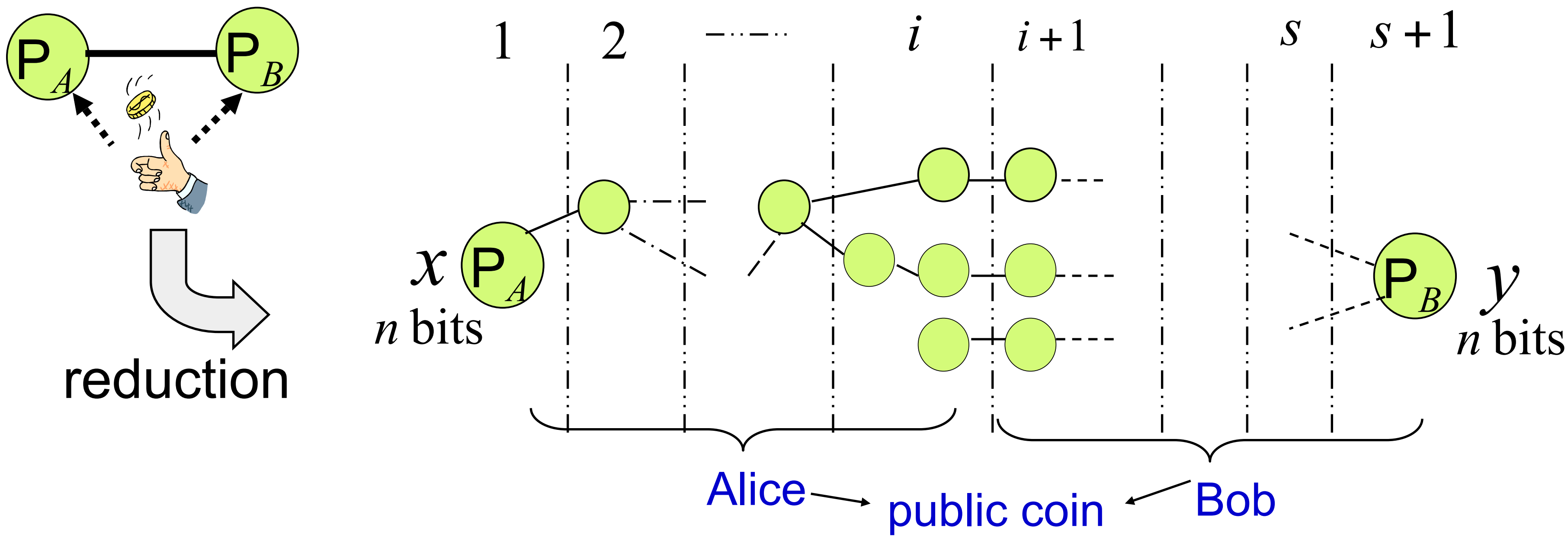
Public coin v.s. Private coin (2/2)

Theorem (Quantum version)

Any quantum protocol using public coins with error probability at most 1/3
can be converted into
a quantum protocol using only private coins with error probability at most 1/3
at the cost of O(log n) bits of additional classical communication,
where n is the number of input bits.



Proof of Lemma 1 (1/3)



Extension of the classical technique [Tiw87] to the quantum case:

Reduction from the two-party public coin model to the multi-party model on network G.

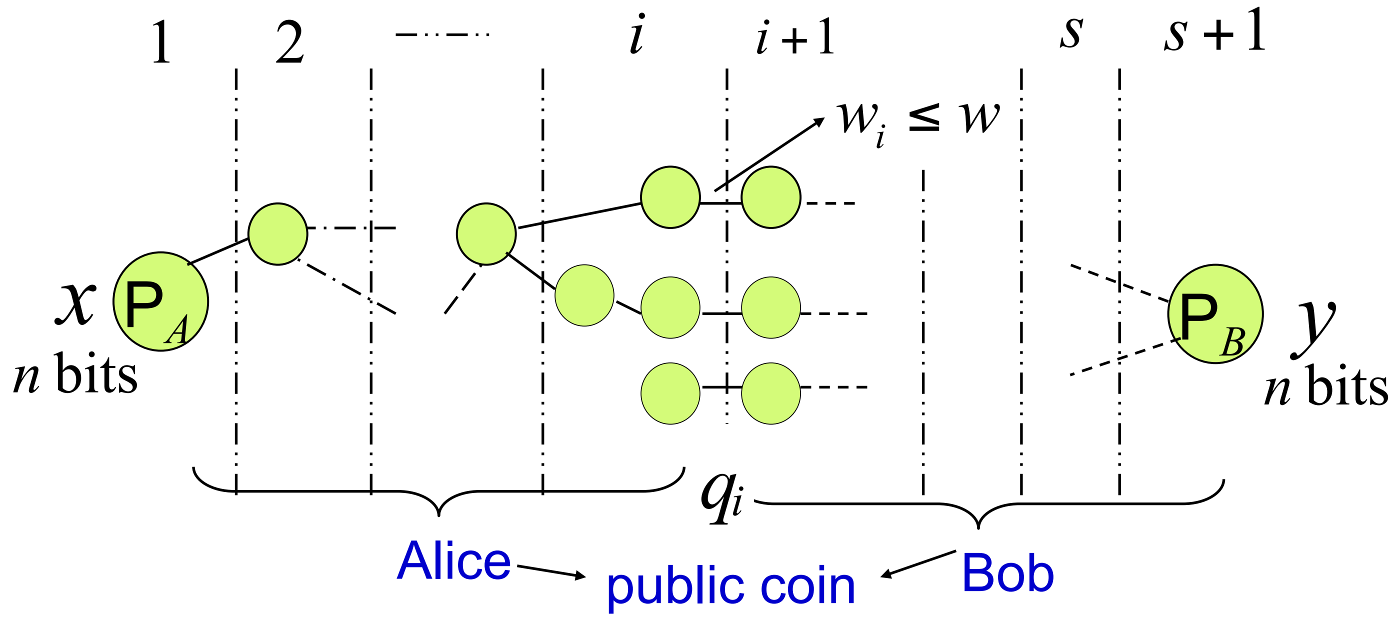
Let Φ be any protocol in the multi-party model.

(1) Pa and Pb sample value i∈{1,…,s} using public coins.

(2) Pa and Pb divide network G at the boundary of the i and (i+1)-st layers.

(3) Pa and Pb simulate the behavior of Φ at the left and right parts, resp.

Proof of Lemma 1 (2/3)



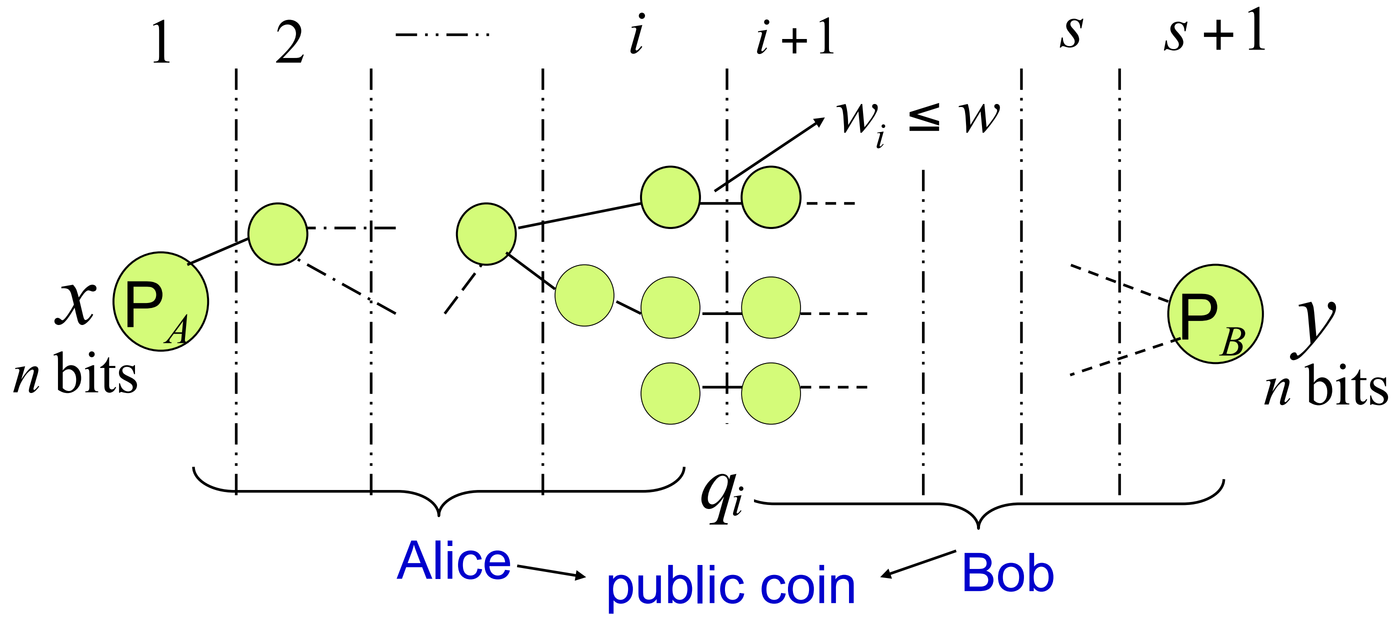
Let q_i be the number of qubits communicated by Φ on the edges across the boundary between the i-th and (i+1)-st layers.

$$\mathrm{E}\left[Q_{1/3}^{\mathrm{Pub}}\left(f(x,y)\right)\right] \le \sum_i \frac{1}{s}\left(\log w_i\right) q_i \le \frac{1}{s}\log w \sum_i q_i$$

By the standard technique,

$$Q_{1/3}^{\mathrm{Pub}}\left(f(x,y)\right) \le O\left(\frac{\log w}{s}\sum_i q_i\right) = O\left(\frac{\log w}{s} Q_{1/3}^{G}(f)\right)$$

Proof of Lemma 1 (3/3)



Applying the public-to-private conversion technique:

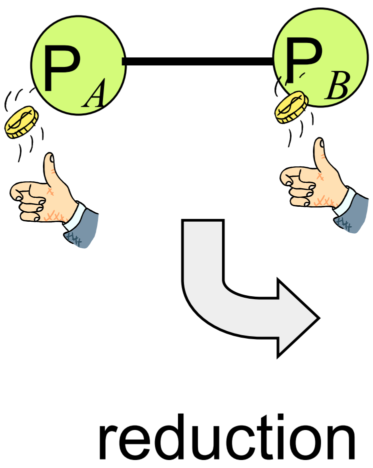
$$Q_{1/3}\big(f(x,y)\big) \le Q_{1/3}^{\text{pub}}(f(x,y)) + \mathrm{O}(\log n)$$

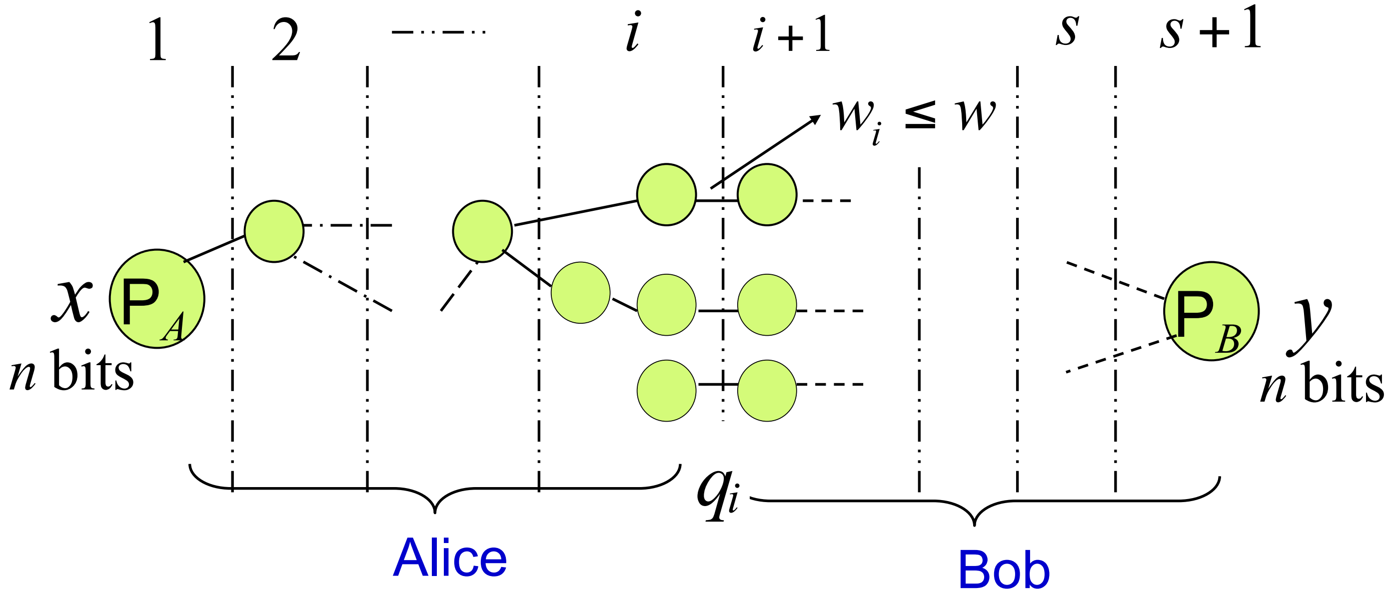
We have

$$Q_{1/3}^{G}\big(f(x,y)\big) = \Omega\Big(s\big(Q_{1/3}\big(f(x,y)\big) - \log n\big)/\log w\Big)$$

Proof of Lemma 2

Almost similar to the proof of Lemma 1 except it does not use public coins.





Let Φ be any protocol in the latter model.

(1) Pa samples value $i \in \{1,\ldots,s\}$ and send i to Pb with log s bits.

(2) Pa and Pb divide network G at the boundary of the i and (i+1)-st layers.

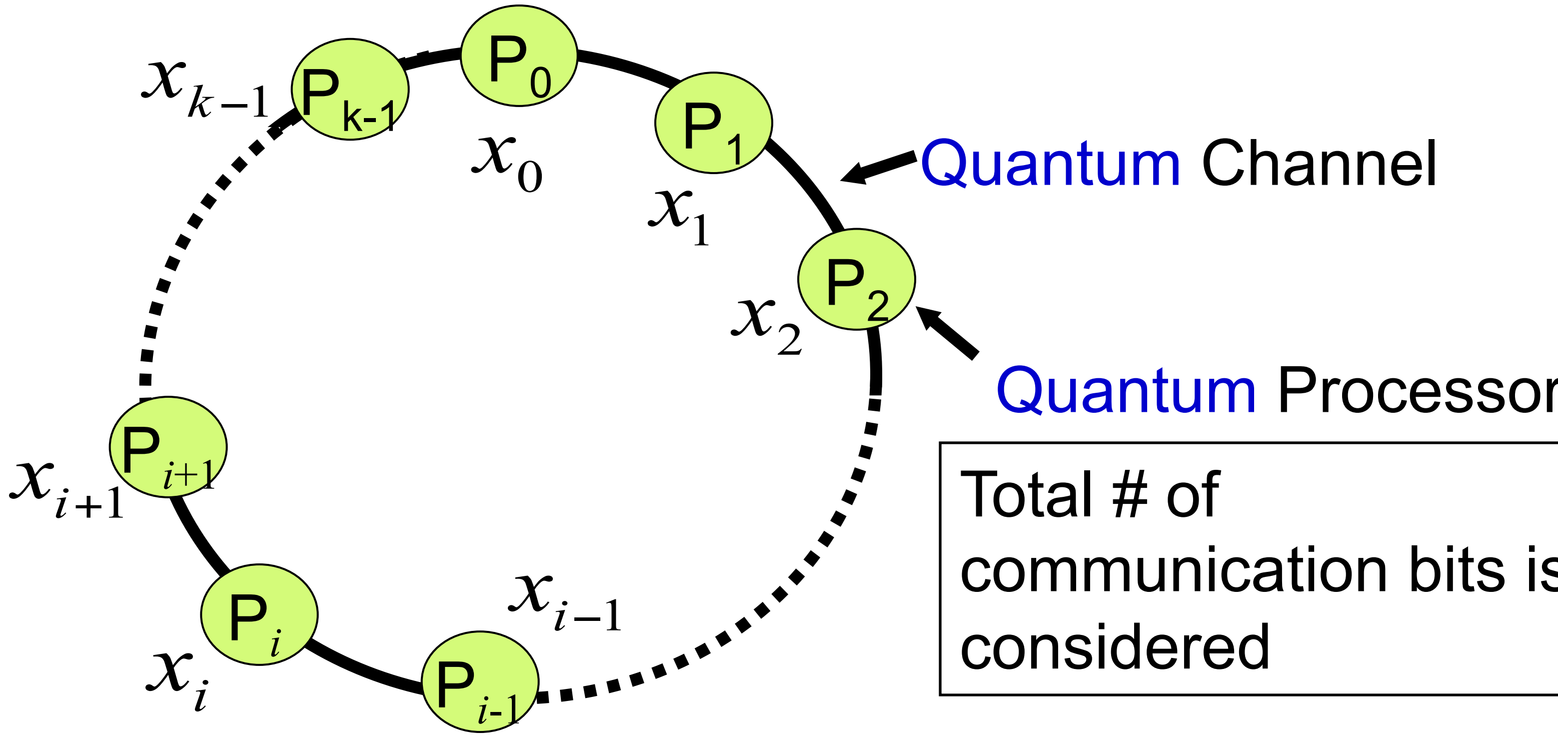
(3) Pa and Pb simulate the behavior of Φ at the left and right parts, resp.

$$Q^{G}_{1/3}\left(f(x,y)\right) = \Omega\left(s\left(Q_{1/3}\left(f(x,y)\right) - \log s\right) / \log w\right)$$

Application to Distinctness on a Ring

Distinctness on a ring

- For i=0,1,…,k-1, party P_i gets as input $x_i \in \{0,\ldots,L-1\}$
- Every party must output:
  - 0 if two or more parties have the same value
  - 1 otherwise ($i \neq j \rightarrow x_i \neq x_j$)



Total # of communication bits is considered

The lower bound of Distinctness on a ring

Theorem

DISTINCT^{ring} (k,L): Distinctness problem on a ring consisting of k parties, each of which is given a (log L)-bit value.

For $L=k+\Omega(k)$, the quantum communication complexity of DISTINCT ^{ring} (k,L) is

$$\Omega(k(k^{1/2} + \log \log L)).$$

Proof is by the following two lemmas.

Lemma 3: The quantum communication complexity of

DISTINCT^{ring} (k,L) is $\Omega(k^{3/2})$.

Lemma 4: The quantum communication complexity of

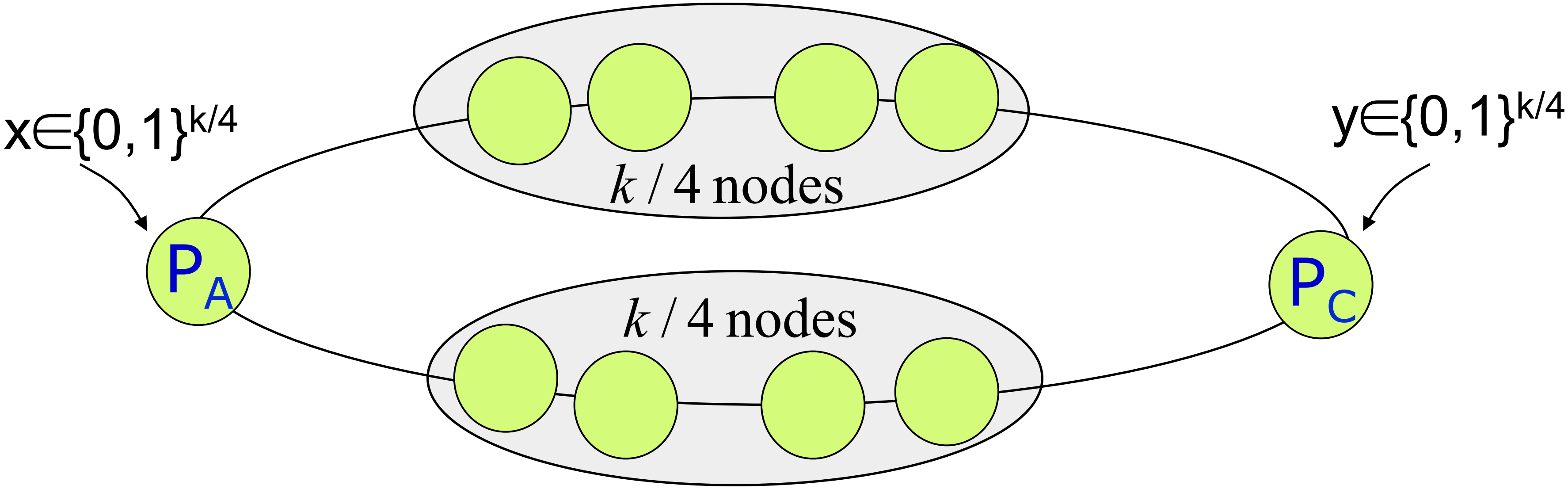
DISTINCT^{ring} (k,L) is $\Omega(k \log\log L)$ for $L=2^{\omega(poly(k))}$.

Proof of Lemma 3 (1/2)

$\text{DISJ}^{\text{ring}}$ (k/4)

P_A: x (k/4 bits) is given.
P_C: y (k/4 bits) is given.

⇨ Compute $\text{DISJ}(k/4) = \wedge_{i=1}^{k/4} \overline{x_i y_i}$ on a following network.

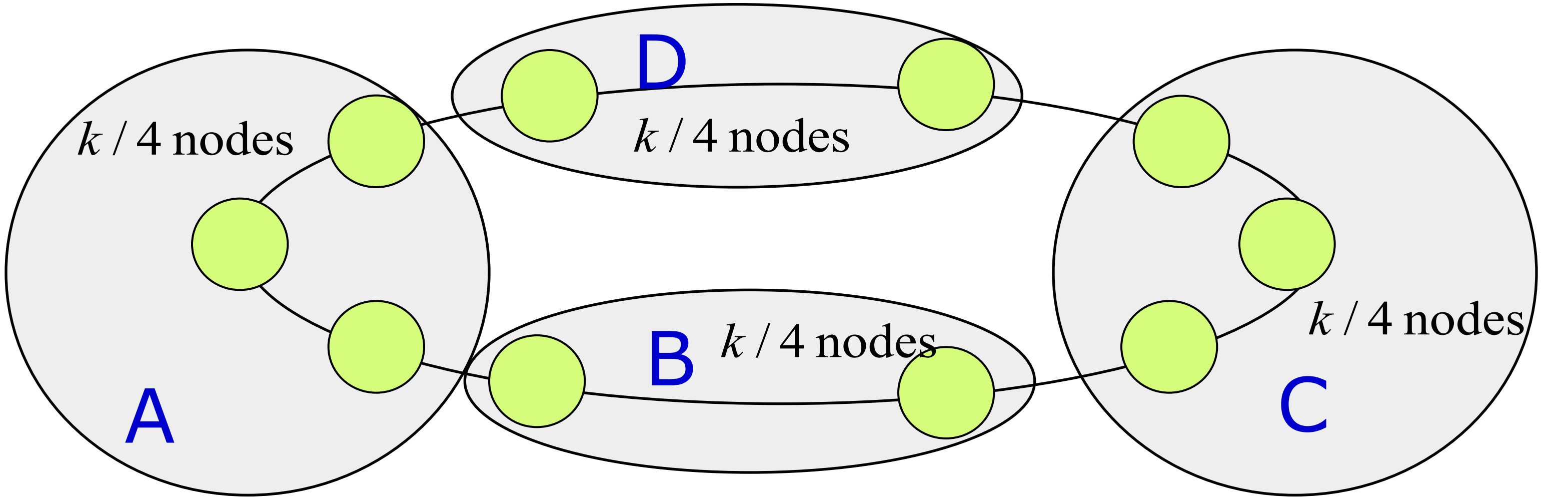


Since $Q_{1/3}(\text{DISJ}(k/4))=\Omega(k^{1/2})$ [Razborov03], our general lower bound implies that the quantum communication complexity of $\text{DISJ}^{\text{ring}}$ (k/4) is

$$\Omega\left(s\left(Q_{1/3}\left(\text{DISJ}(k/4)\right)-\log k\right)/\log w\right)=\Omega\left(k\sqrt{k}\right)$$

Proof Lemma 3 (2/2)

Reduction from DISJ^{ring}(k/4) to DISTINCT^{ring}(k,L)
with no extra communication cost.



P_A: x (k/4 bits) P_C: y (k/4 bits)

- Pa simulates the k/4 nodes in A: if x_k=1, the kth node gets as input $(k\text{-}1)\in I_1=\{0,1,\ldots,k/4\text{-}1\}$; otherwise it gets a distinct value in {k/4,…,k/2-1}.
- Pb simulates the k/4 nodes in C: if x_k=1, the kth node gets as input $(k\text{-}1)\in I_1$; otherwise it gets a distinct value In {k/2,…,3k/4-1}.
- The nodes in C and D gets as input distinct values in {3k/4,…,L-1}

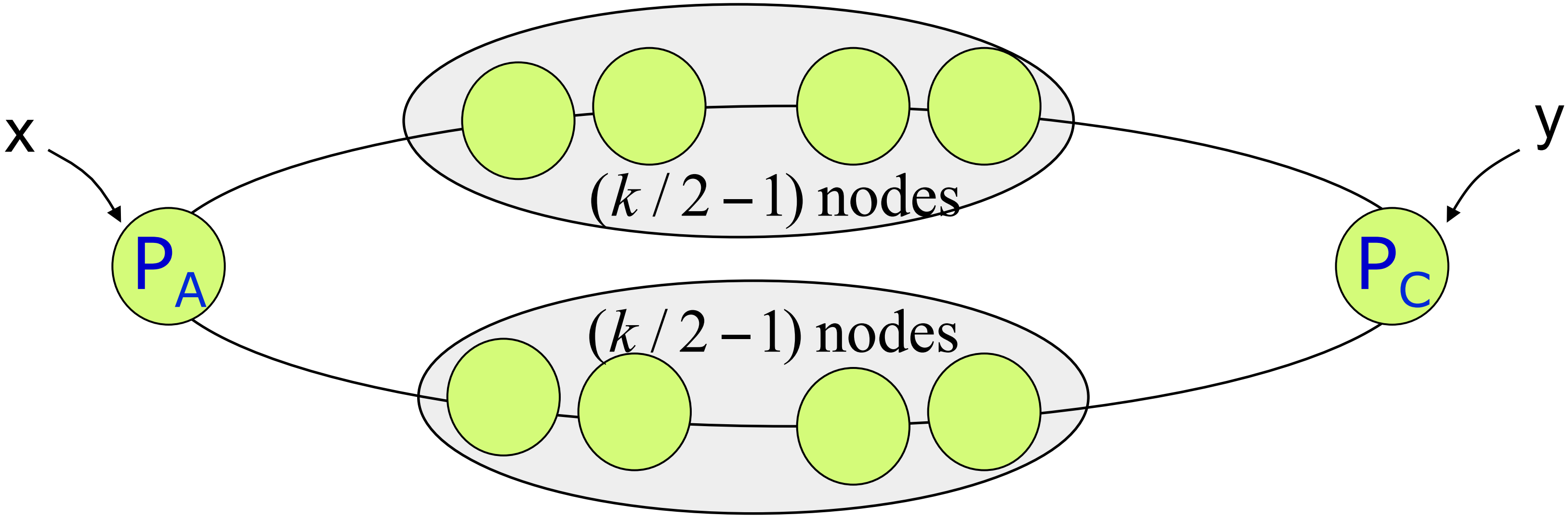
Proof of Lemma 4 (1/2)

$\mathrm{EQ}^{\mathrm{ring}}$ (k, log L -1)

P_A: x (log L-1 bits) is given.

P_C: y (log L-1 bits) is given.

Compute $\mathrm{EQ}(\log \mathrm{L} - 1) = \wedge_{i=1}^{\mathrm{Log}\, L-1}(x_i = y_i)$ on a following network.

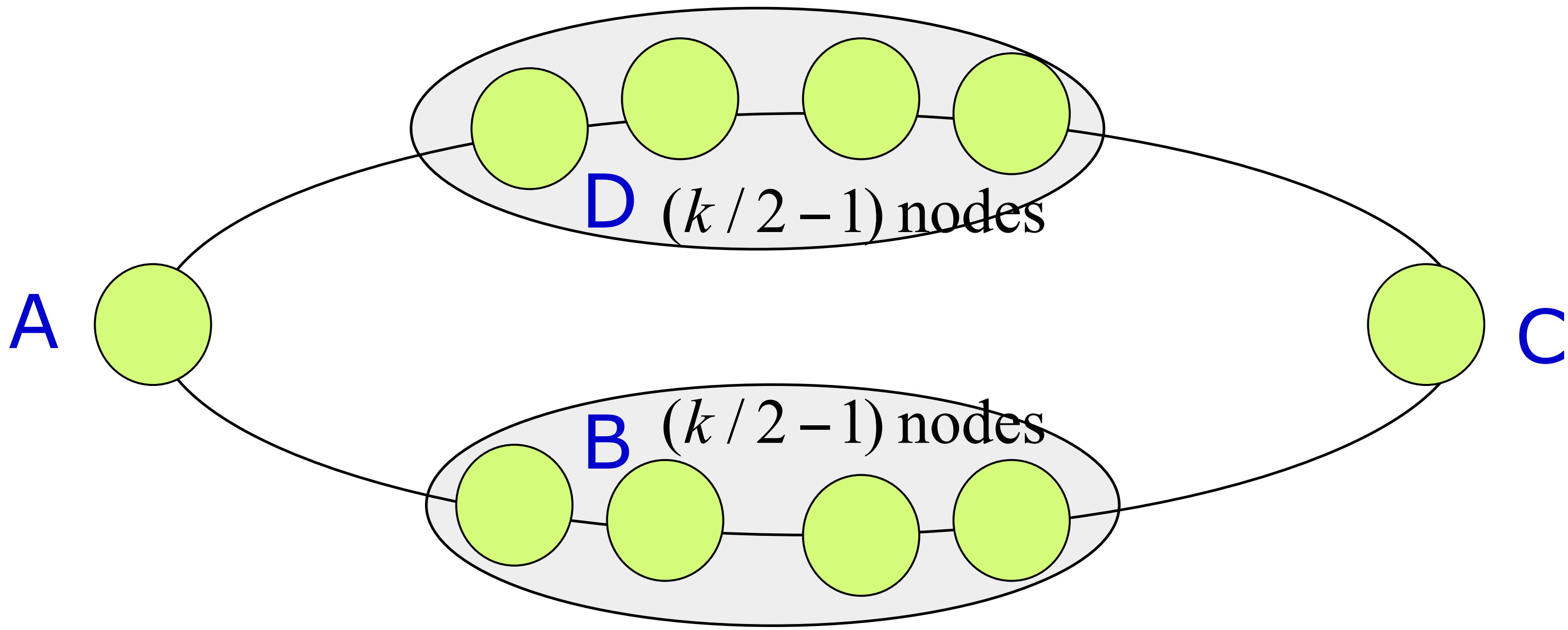


Since $Q_{1/3}(\mathrm{EQ}(\log \mathrm{L} -1))=\Omega(\log \log \mathrm{L})$, our general lower bound theorem implies that the quantum communication complexity of $\mathrm{EQ}^{\mathrm{ring}}$ (k, log L -1) is for $\mathrm{L}=2^{\omega(\mathrm{poly}(k))}$.

$$\Omega\left(s\left(Q_{1/3}\left(\mathrm{EQ}_{\log L}\right)-\log\min\{k,\log L\}\right)/\log w\right)=\Omega\left(k\log\log L\right)$$

Proof of Lemma 4 (2/2)

Reduction from $EQ_{\log L - 1}$ on a k-party ring to DISTINCT(k,L) without extra communication cost



- Pa simulates Node A: get as input 1x.
- Pb simulates Node C: get as input 1y.
- The nodes in C and D get as input distinct values 0z, where z is in {0,…,L/2-1}.

(Almost) Matching Upper Bound for Distinctness on a Ring

Almost Matching Upper Bound for Distinctness on a Ring

Lemma

The quantum communication complexity of $\mathrm{DISTINCT}^{\mathrm{ring}}(k,L)$ is $O(k(k^{1/2}\log k + \log\log L))$.

Idea is to solve the following search problem.

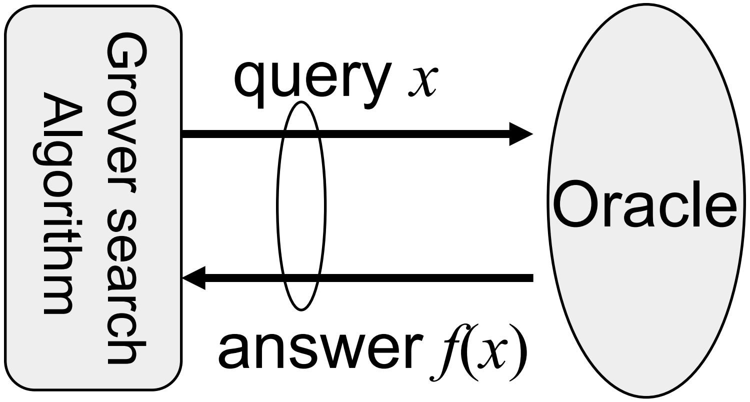
Search for $m \in \{0,\ldots,k\text{-}1\}$ that has the next property:
there is at least one party j $(\neq m)$
that gets the same value as x_m.

To do this, use Grover's quantum search algorithm [Gro96]
in a distributed fashion.

Grover's quantum search [Grover96]

- Boolean function $f:\{0,1\}^n \rightarrow \{0,1\}$ is given as an oracle
- Grover's algorithm can find $x \in \{0,1\}^n$ such that $f(x)=1$ with probability at least 2/3 by making $O(\sqrt{2^n})$ queries.

(In the classical setting, $O(\sqrt{2^n})$ queries are needed.)



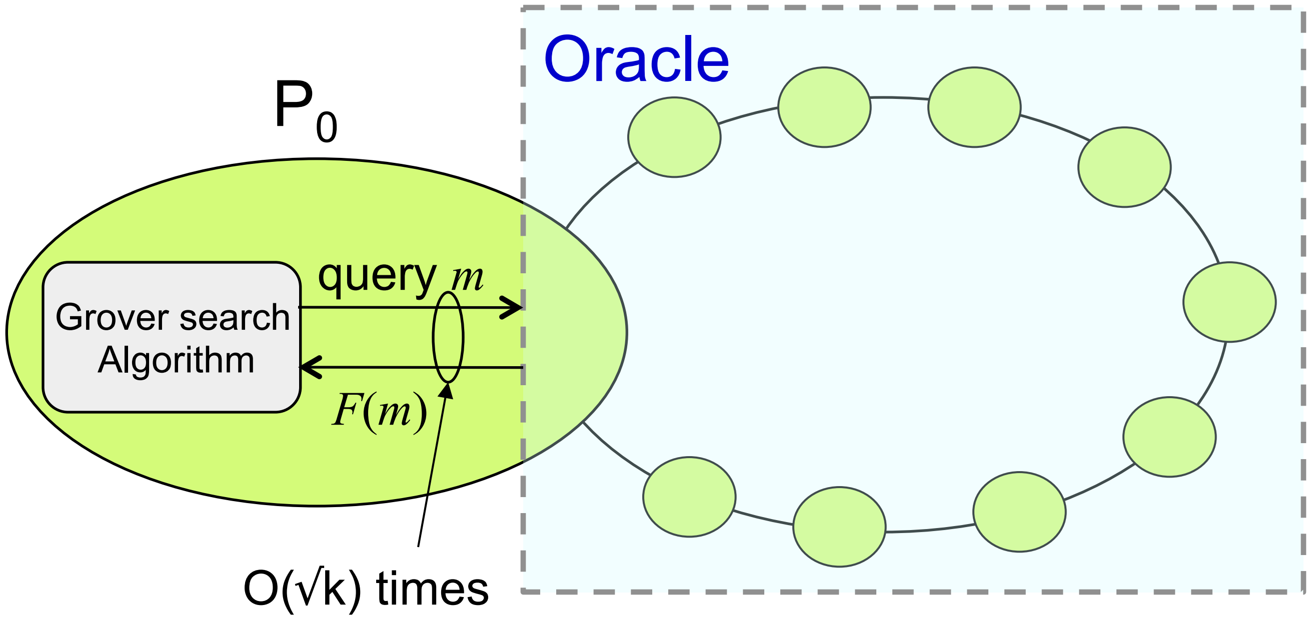
Application of Grover's algorithm to Distinctness

Def. F:{0,1,…,k-1}→{0,1} such that

$F(m)$ = 1 iff there is at least one party $j(\neq m)$ that gets the same value as x_m.

Idea:

- Party P1 runs Grover's algorithm
- All parties collaborate to simulate an oracle for F.



Distributed implementation of oracle

To compute F(m), it is sufficient to count the number of parties which have the same value as x_m.

- First phase gets information of x_m by conveying a message of the form (m,*value*) around the ring.
  - Initiator is P0
  - The message coming back to P0 should be (m, x_m).
  - Message consists of O(log k + log L) qubits
- Second phase counts the number of parties which have the same value as x_m by conveying message (x_m, *counter*).
  - Initiator is P0, transmitting (x_m, 0)
  - Message consists of O(log L + log k) qubits.
- Third phase inverts the first and second phases to disentangle work qubits.

Complexity

- Each oracle query needs $O(k(\log k + \log L))$-qubit communication.
  - Each message consists of $O(\log k + \log L)$ qubits.
- Since $O(\sqrt{k})$ queries need to be made, the complexity is:

$$O(k \sqrt{k} (\log k + \log L)).$$

This bound is almost optimal for L= poly (k), but for large L, it is much larger than

$$\Omega(k(\sqrt{k} + \log \log L)).$$

Improvement

Idea:

(1) To decreasing input size,

map original input of (log L) bits into a 3 log k-bit value

by using universal hashing.

(2) Use public coins so that every party can choose the same hash function.

(3) Convert the public-coin protocol into a private coin protocol.

Total Complexity: $O(k \sqrt{k} \log k) + O(k \log \log L)$

$= O(k(k^{1/2} \log k + \log \log L))$.

Hashing inputs

Idea: To decreasing input size,
map original input of (log L) bits into a 3 log k-bit value
by using universal hashing.

Algorithm: Assume all parties share public coins.
(This assumption will be removed later.)

1. Every party randomly chooses a hash function by using public coins.
2. Every party maps his original input into a 3log k bit value by using the hash function.
3. Run the $O(k \sqrt{k} (\log k + \log L))$ algorithm.

Complexity: $O(k \sqrt{k} \log k)$.

Analysis of error probability

- Hashing step
  - If party Pi and Pj has the same value xi=xj, the values are mapped into the same value h(xi)=h(xj); the output of Distinctness is unchanged.
  - If every party gets a distinct value, some distinct values are mapped into the same value with probability at most:

$$k(k-1)/2 \times 1/k^3 \approx 1/k.$$

- Grover's search step
  - Oracle contains no error.
  - Grover's search algorithm succeeds with at most constant error probability.

Over all error probability is at most constant.

Public coin -Private coin conversion for k parties

Theorem (Quantum k-party version)

Any quantum protocol using public coins for k parties
with error probability at most 1/3
can be converted into
a quantum protocol using only private coins for k parties
with error probability at most 1/3
at the cost of O(log kn) bits of additional classical communication,
where n is the number of input bits.

Since the total number of input bits is k log L,
the conversion needs
$O(k \log (k \log L)) = O(k \log\log L)$ for $L = \omega(\text{poly}(k))$.

Total Complexity: $O(k \sqrt{k} \log k) + O(k \log \log L)$
$= O(k(k^{1/2} \log k + \log \log L))$.

Another Upper Bound for Distinctness on a Ring

Lemma

The quantum communication complexity of $\mathrm{DISTINCT}^{\mathrm{ring}}(k,L)$ is $O(k\sqrt{L})$.

Idea is to solve the following search problem.

Search for $m \in \{0,\ldots,L\text{-}1\}$ that has the next property:

there is at least two parties that gets value m.

If we use Grover's search algorithm, the complexity is $O(k \sqrt{L} \log L)$.

It is possible to improve this bound to $O(k\sqrt{L})$ by using "recursive Grover search algorithm in [Aaronson&Ambainis03] instead.

Remark

- Q. Is it possible to remove log k factor of $O(k(k^{1/2}\log k + \log\log L))$?

- Q. Is it possible to improve $O(k\sqrt{L})$ by using universal hashing?

Summary

- A general lower bound of quantum communication complexity is given over multi-party network.
- As an application, the distinctness problem was considered on a ring. Almost tight bounds were given.

Open Problems

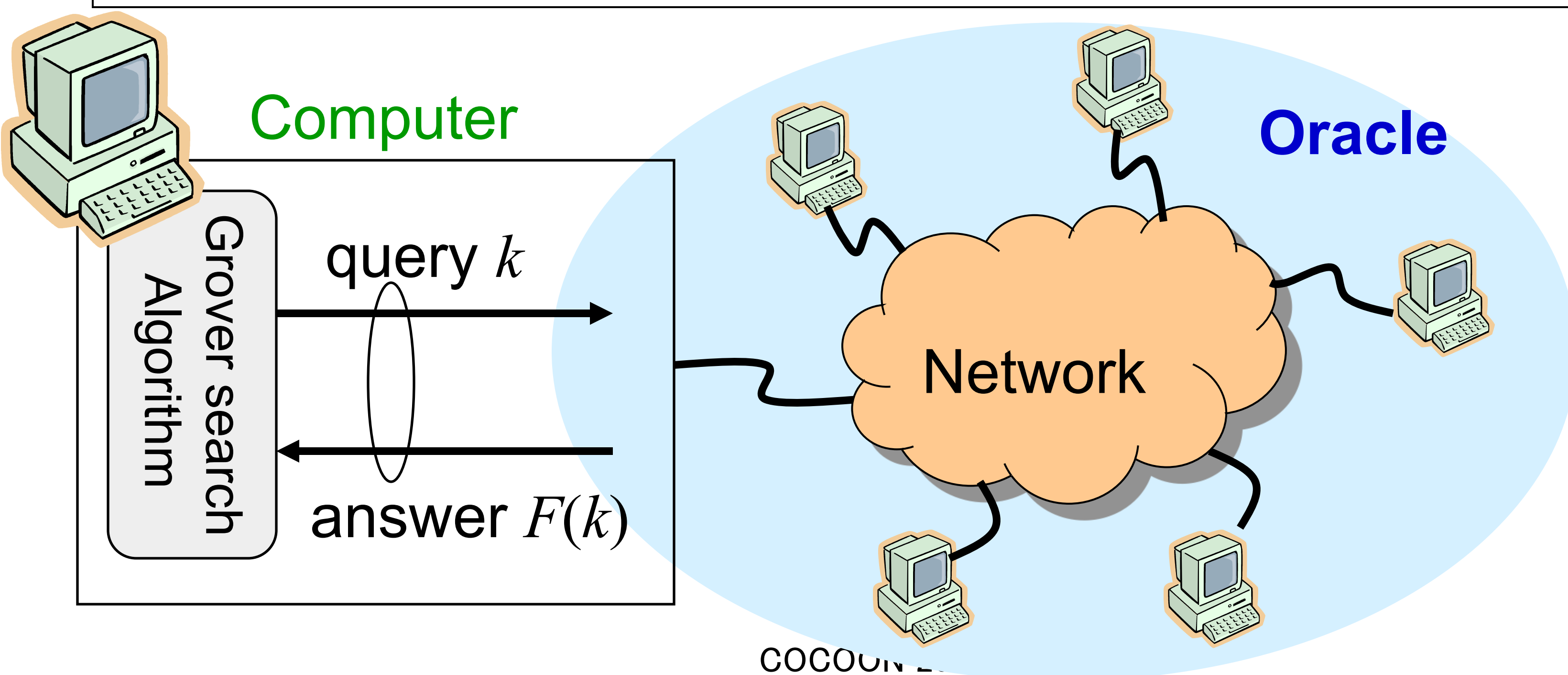
- Is it possible to get better lower bound, possibly by using other parameters?
- Is quantum communication complexity on a dense graph lower than that on a sparse graph?

Thank you!

Idea of algorithm computing Distinctness

Perform Grover search to find $k \in \{0,\ldots,L\text{-}1\}$
that has the next property:
there are two or more parties who get k as input.

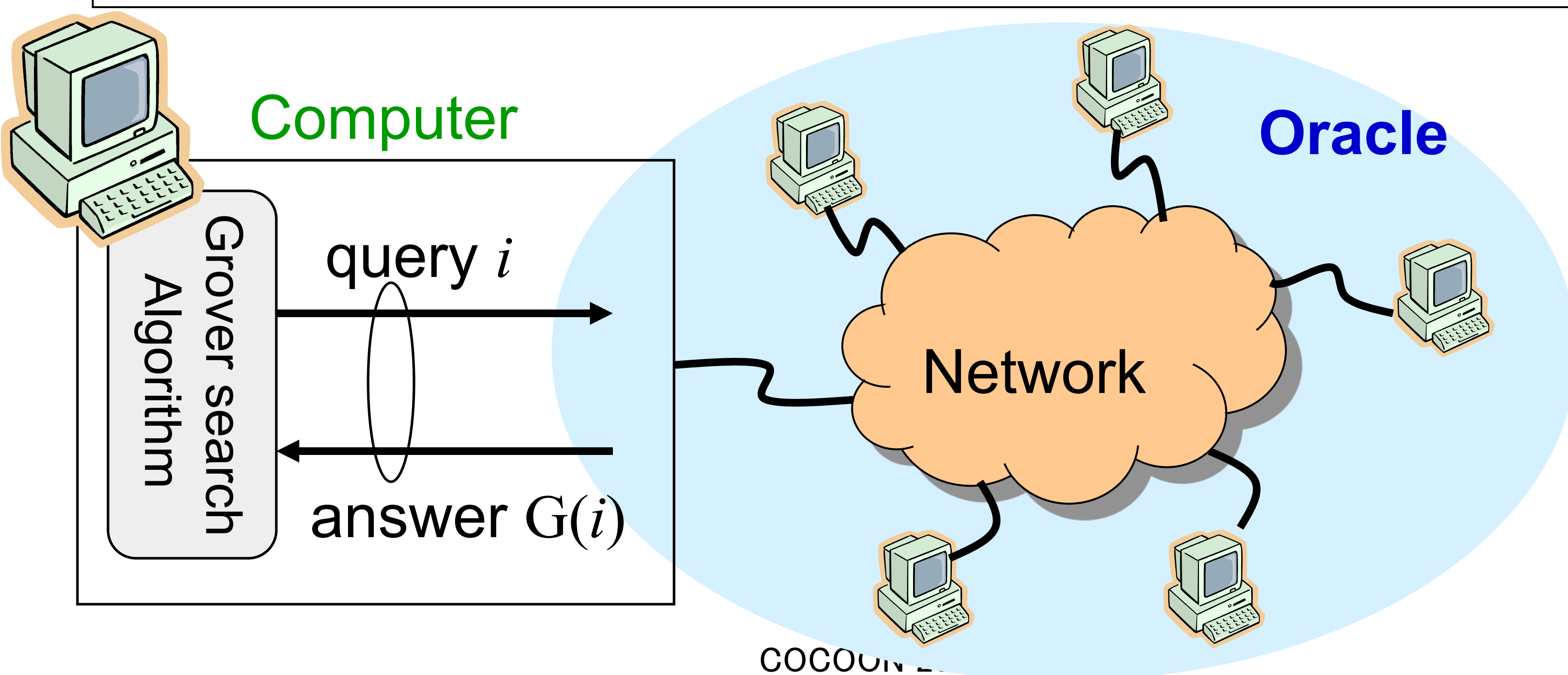
Def. $F(k) = 1$ if k has the property.
0 otherwise.



Another Idea of algorithm computing Distinctness

Perform Grover search to find $i \in \{1,\ldots,n\}$
that has the next property:
there is at least one $j \in \{1,\ldots,n\}$ such that $X_j = X_i$ for $i \neq j$

Def. $G(i)$ = 1 if i has the property.
0 otherwise.



Complexity: DISTINCTNESS on a ring

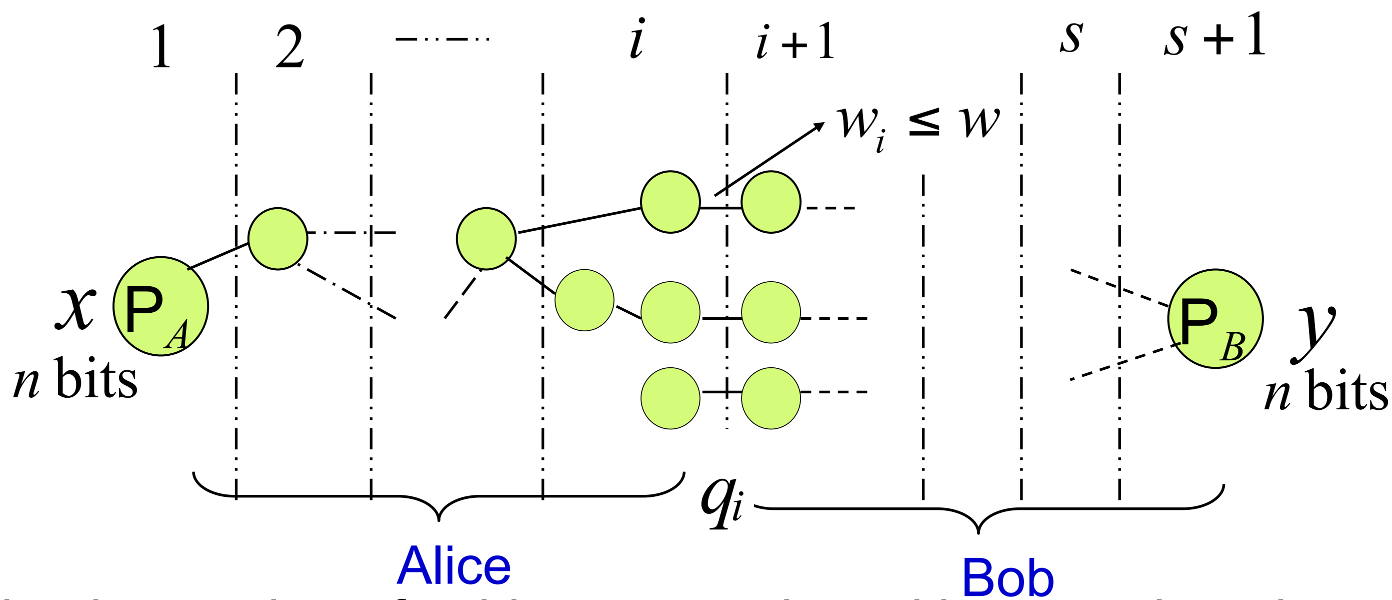
Idea 1 gives:

DISTINCTNESS for n computer on a ring network has the communication complexity $O(nL^{1/2})$.

Idea 2 gives:

DISTINCTNESS for n computer on a ring network has the communication complexity $O(n^{3/2}\log L)$.

Proof of Lemma 2 (2/3)



Let q_i be the number of qubits communicated by Φ on the edges across the boundary between the i-th and (i+1)-st layers.

$$E\left[Q_{1/3}\left(f(x,y)\right)\right] \le \log s + \sum_i \frac{1}{s}\left(\log w_i\right) q_i \le \log s + \frac{\log w}{s} \sum_i q_i$$

By the standard technique,

$$Q_{1/3}\left(f(x,y)\right) \le O\left(\log s + \frac{\log w}{s} \sum_i q_i\right) = O\left(\log s + \frac{\log w}{s} Q_{1/3}^{G}(f)\right)$$

The lower bound of Distinctness on a ring(1/5)

Lemma 1: The quantum communication complexity of Distinctness on a ring is $\Omega(k^{3/2})$.

Outline of Proof.

Step1: Apply the lower bound theorem to DISJ on a ring.

Step2: Reduce DISJ on a ring to Distinctness on a ring.

Lemma 2: The quantum communication complexity of Distinctness on a ring is $\Omega(n \log\log L)$ for $L=2^{\omega(poly(n))}$.

Outline of Proof.

Step1': Apply the lower bound theorem to EQ of log L bits on a ring.

Step2': Reduce EQ on a ring to Distinctness on a ring.

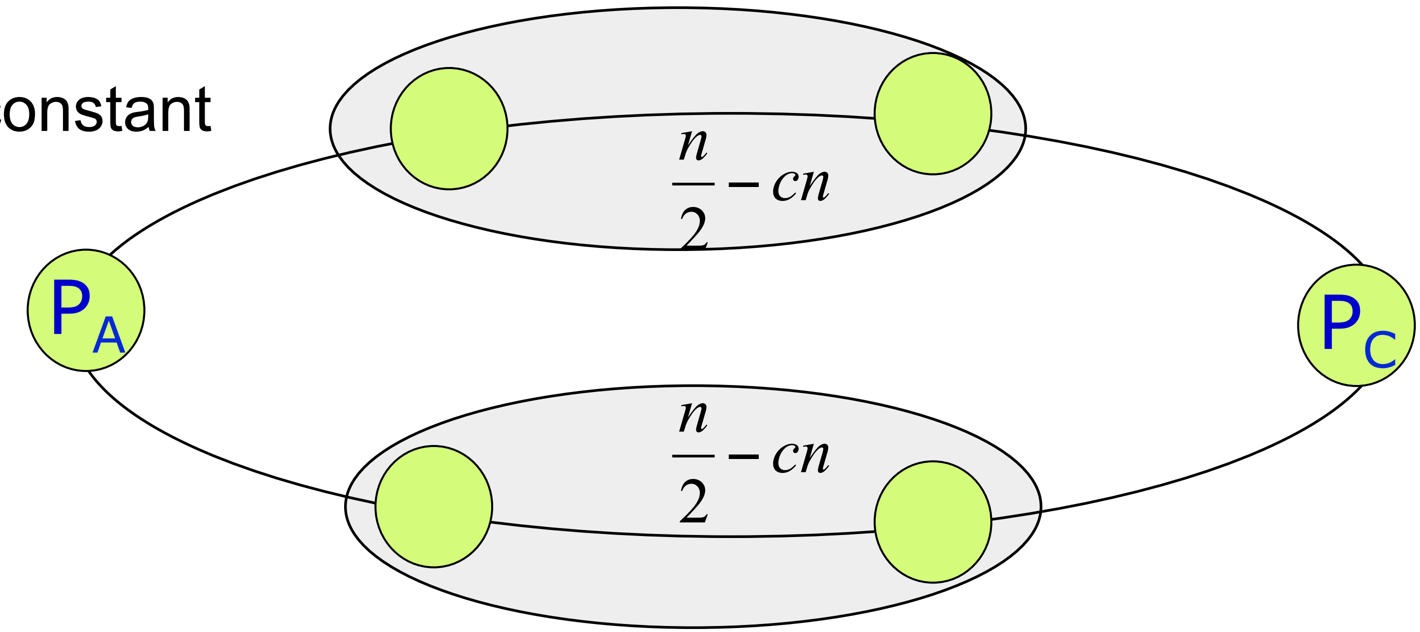
Step 1: Apply the lower bound theorem to DISJ on a Ring (2/5)

DISJ on a ring

P_A: x (cn bits) is given. P_C: y (cn bits) is given.

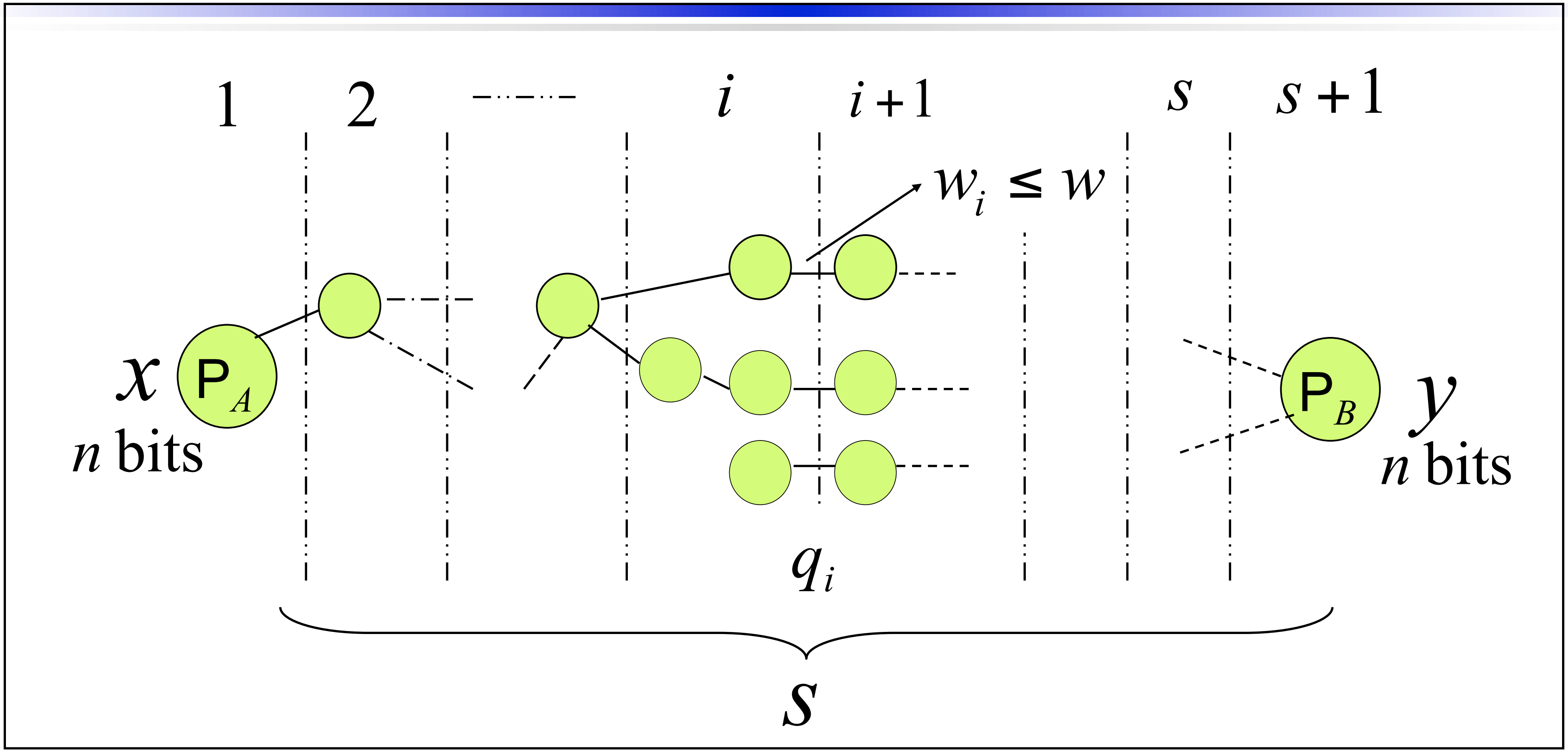
Compute $\wedge_{i=1}^{cn} \overline{x_i y_i}$ on a following network.

C: constant



$$Q_{1/3}^{RING}(DISJ) = \Omega\left(s\left(Q_{1/3}(DISJ) - O(\log n)\right)/\log w\right)$$
$$= \Omega\left(n\sqrt{n}\right)$$

Lower Bound on an arbitrary network (1/4)



Lemma 1

$$Q^{N}_{1/3}\left(f(x,y)\right) = \Omega\left(s\left(Q_{1/3}\left(f(x,y)\right) - \log n\right) / \log w\right)$$