Optimal Gossip Algorithms for Exact and Approximate Quantile Computations

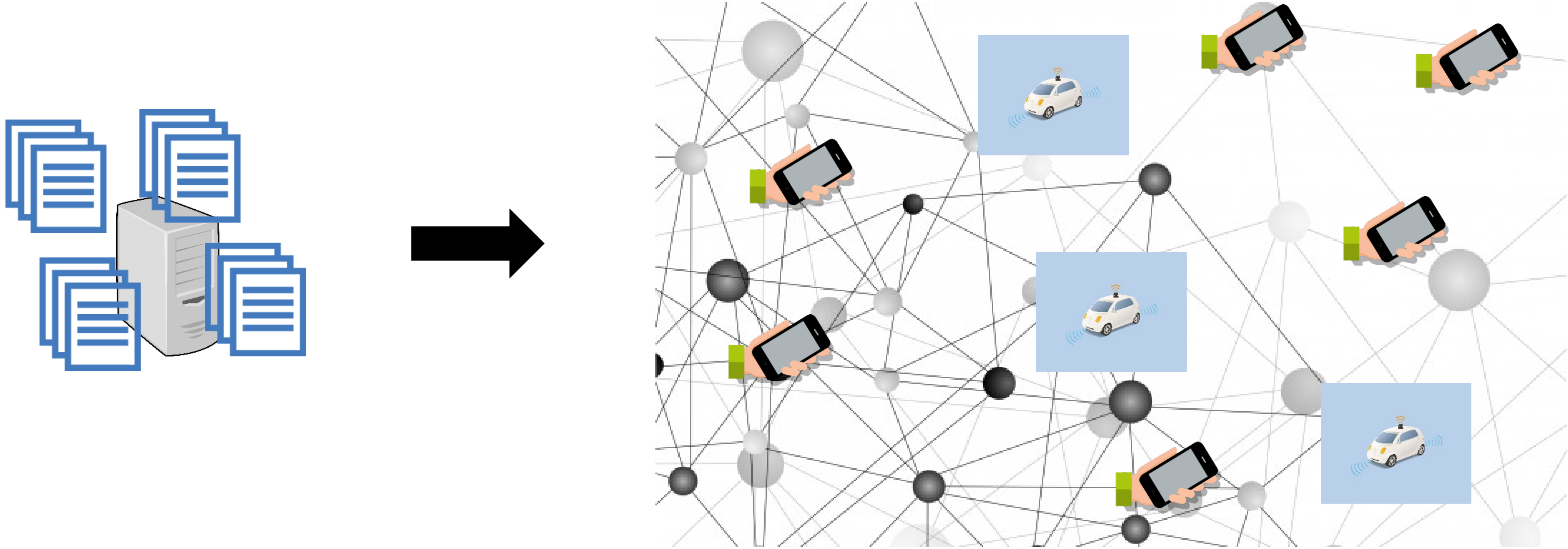
Hsin-Hao Su
(UNC Charlotte => Boston College)

joint work with Bernhard Haeupler (CMU) and Jeet Mohapatra (MIT)

WOLA ‘18

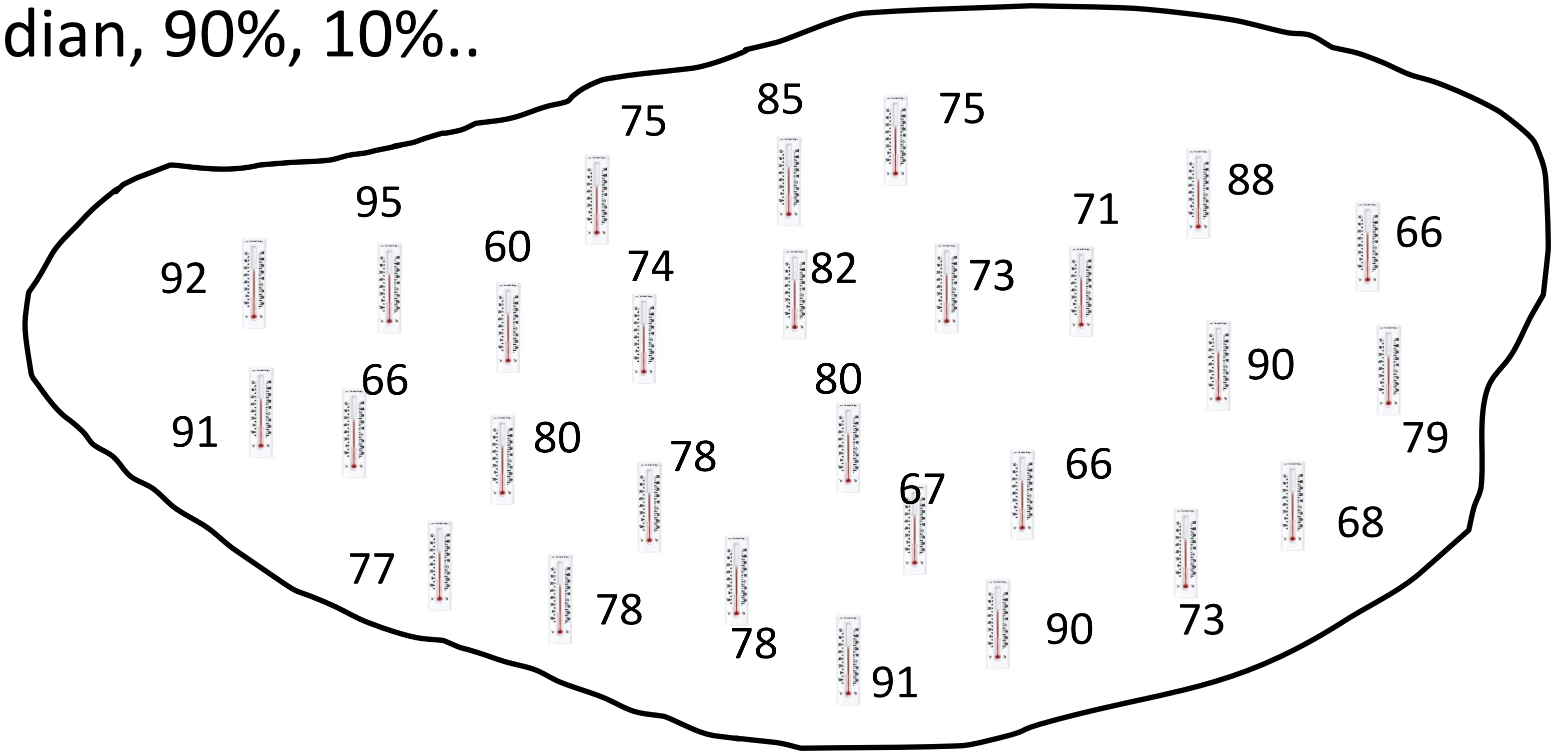
Shifting in Computing Paradigm

- Reduced costs on devices
- Increase in the amount of data
- Advances in connectivity between computers



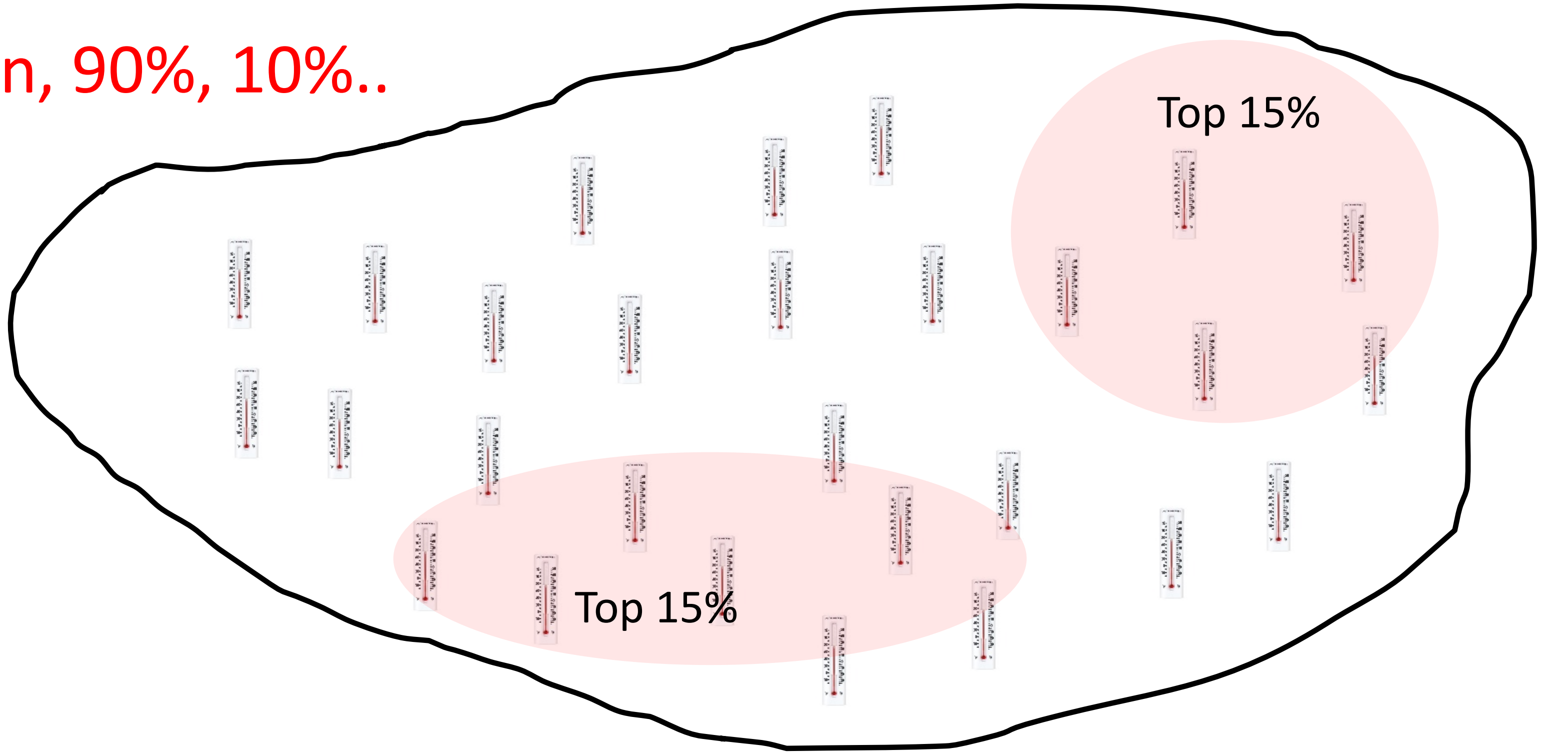
Basic Aggregation Problems

- Basic Problems
  - Sum
  - Average
  - Min, Max
  - Quantiles: Median, 90%, 10%..



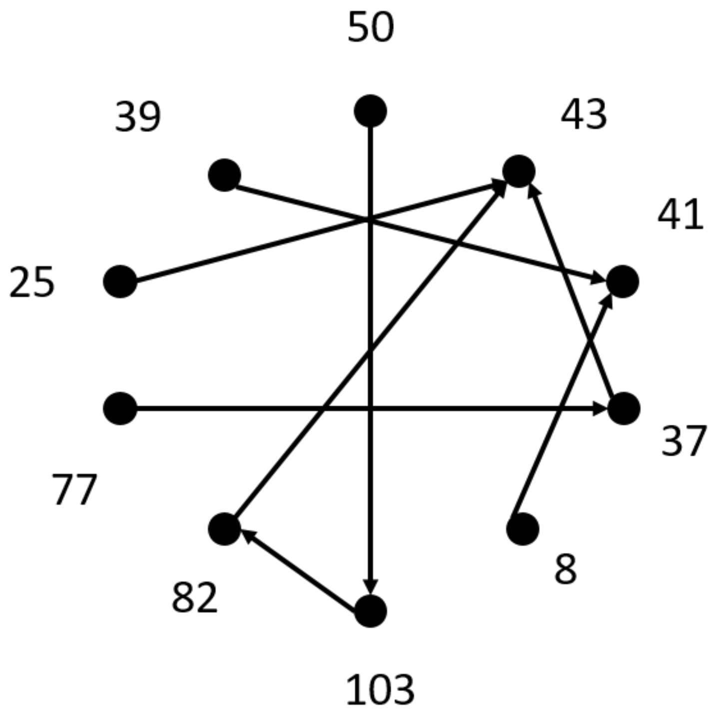
Basic Aggregation Problems

- Basic Problems
  - Sum
  - Average
  - Min, Max
  - Quantiles: Median, 90%, 10%..



Gossip Algorithms

- Gossip Algorithms / Epidemic Algorithms/ Population Protocols
  - Each node interacts with another node *t(v)*, chosen uniformly at random
  - **PUSH** or **PULL** $O(\log n)$ bits
  - Nice properties:
    - Scalable
    - Low overhead (i.e. $O(n)$ messages per round)
    - Fast convergence
    - Fault-tolerant
  - Captures interaction patterns in Nature:
    - Molecules interactions in chemical reactions
    - Rumor spreading



Previous Results

- Max, Min
  - Folklore: $O(\log n)$ rounds w.h.p.

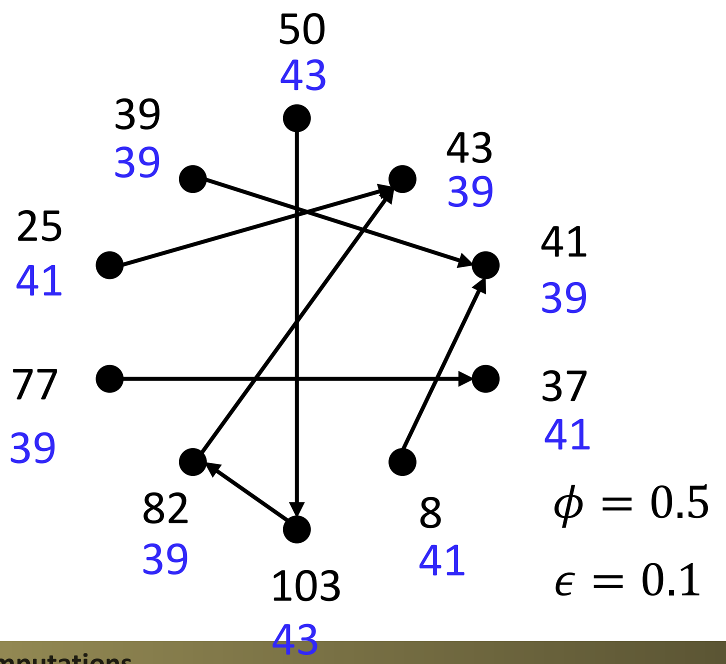
- Sum, Average
  - [Kempe, Dobra, Gehrke '03]
    PUSH-SUM: approximate within $(1 \pm \epsilon)$ in $O(\log n + \log(\frac{1}{\epsilon}))$ rounds w.h.p.

Previous Results

- Quantile Computation
  - **ϕ-quantile:** Given $0 < \phi < 1$, every node outputs a value whose rank is $\lfloor \phi n \rfloor$
    - [Kempe, Dobra, Gehrke '03]: $O(\log^2 n)$
  - **ϵ-approximate ϕ-quantile**: Given $0 < \phi, \epsilon < 1$, every node outputs a value whose rank is $(\phi \pm \epsilon)n$
    - [Doerr et al. '11]:

      $O(\log n)$ rounds alg. for $\epsilon = O(\sqrt{\frac{\log n}{n}})$

      and $\phi = 0.5$ (median)



New results

- **ϕ-quantile:** Given $0 < \phi < 1$, every node outputs a value whose rank is $\lfloor \phi n \rfloor$

$O(\log n)$ rounds	Optimal

- **ϵ-approximate ϕ-quantile**: Given $0 < \phi, \epsilon < 1$, every node outputs a value whose rank is $(\phi \pm \epsilon)n$

$O\left(\log \log n + \log(\frac{1}{\epsilon})\right)$ rounds	Optimal

New results

- **ϕ-quantile:** Given $0 < \phi < 1$, every node outputs a value whose rank is $\lfloor \phi n \rfloor$

$O(\log n)$ rounds	Optimal

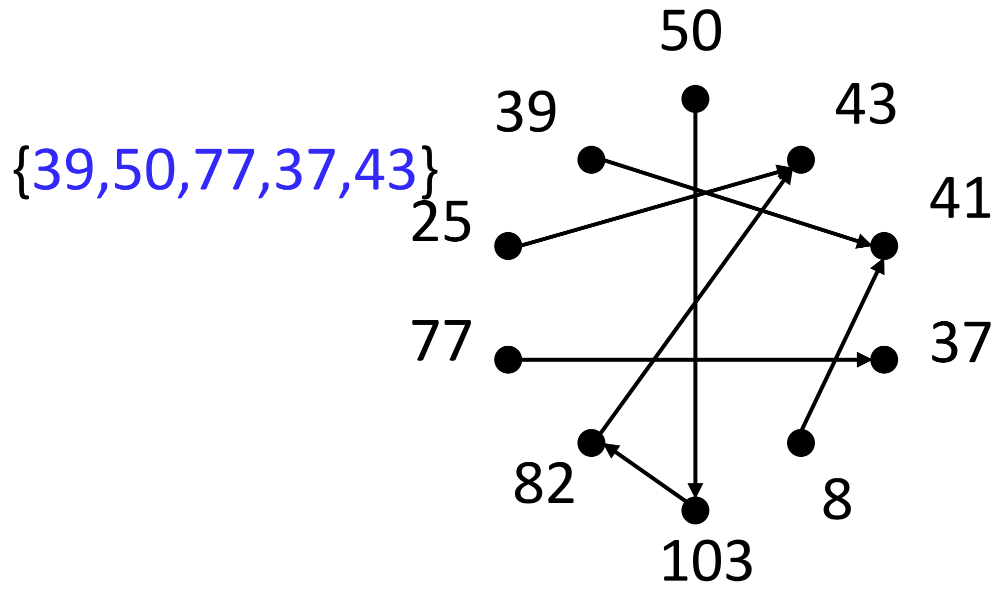
- **ϵ-approximate ϕ-quantile**: Given $0 < \phi, \epsilon < 1$, every node outputs a value whose rank is $(\phi \pm \epsilon)n$

$O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ rounds	Optimal

First Attempt (Sampling)

Suppose each node randomly samples $\Theta\left(\frac{\log n}{\epsilon^2}\right)$ values and outputs the ϕ-quantile of the sampled values

Then, w.h.p. the quantile of the value is $(\phi \pm \epsilon)$



$O\left(\frac{\log n}{\epsilon^2}\right)$ rounds

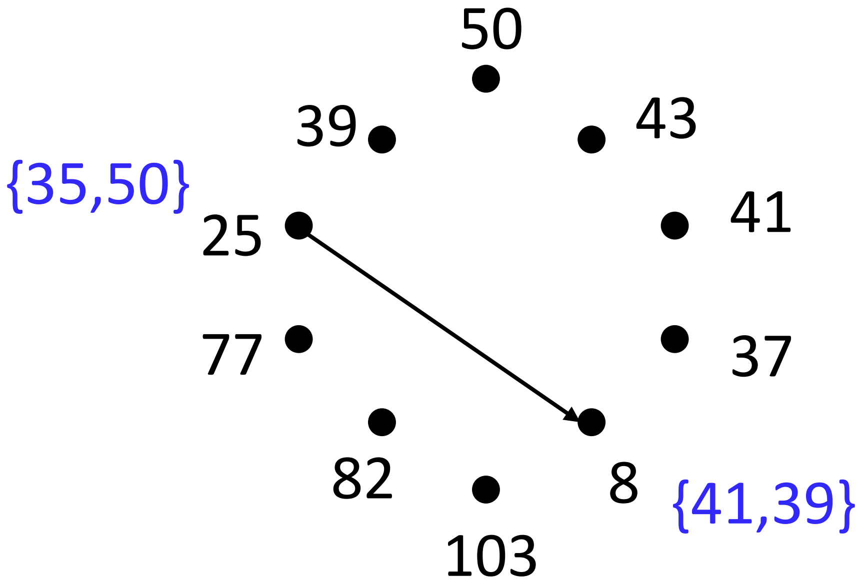
⇩

$O\left(\log\log n + \log\left(\frac{1}{\epsilon}\right)\right)$ rounds ?

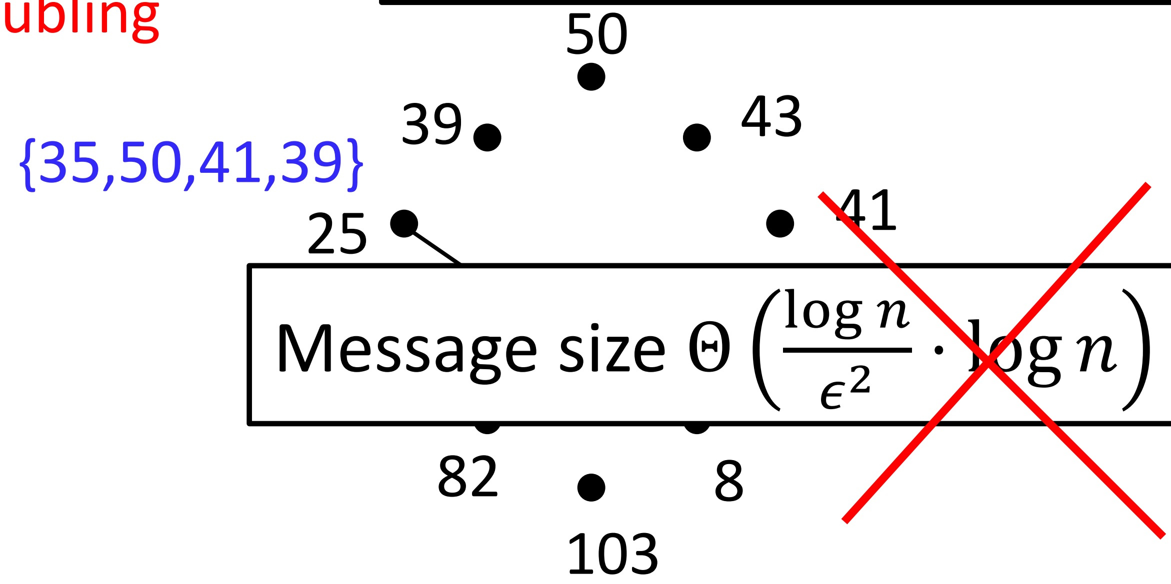
Second Attempt (Doubling)

The sampled values can be doubled in every round.

$O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ rounds



Doubling

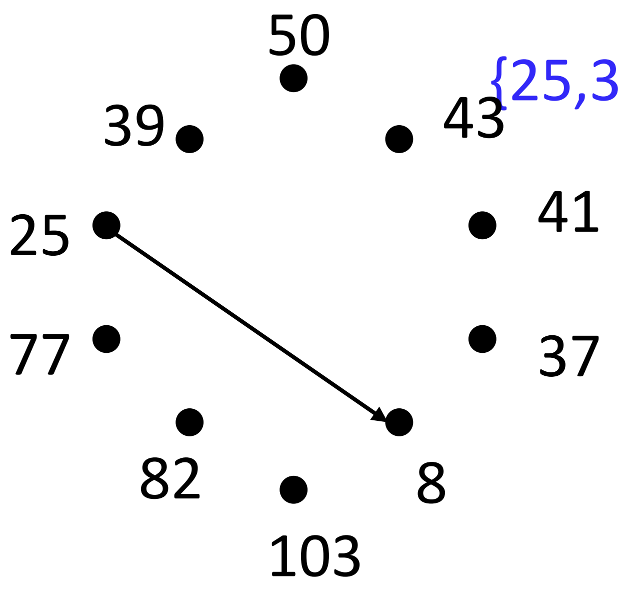


Message size $\Theta\left(\frac{\log n}{\epsilon^2} \cdot \log n\right)$ bits

Third Attempt (Quantile Sketch)

Instead of storing the whole set of sampled values, only keep a sketch of it.

Quantile Sketch: [Munro and Patterson ’80, Manku et al. ’99, Greenwald and Khanna ‘01]



{25,35,39,41,41,50, 77, 103} ➡ {~~25~~,35,~~39~~,41,~~41~~,50, ~~77~~, 103}

Message size:
$\Theta((\log\log n + (\log 1/\epsilon)) \cdot \log n)$ bits

$O(\log n)$ bits possible?

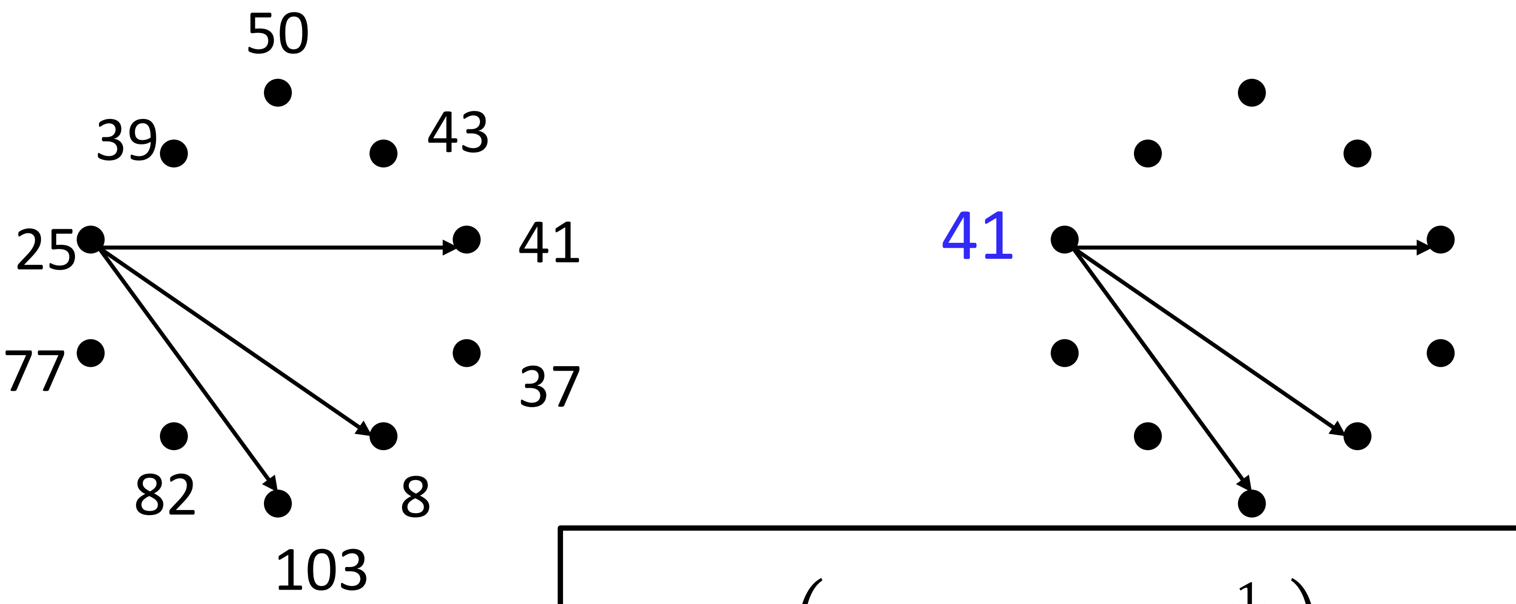
Our Approach

- Phase I: Shift the ϕ-quantile to the approximate median

- Phase II: Compute the approximate median

The TOURNAMENT Algorithm

3-Tournament
For each iteration:
Each node v randomly samples 3 values and sets itself to the middle one



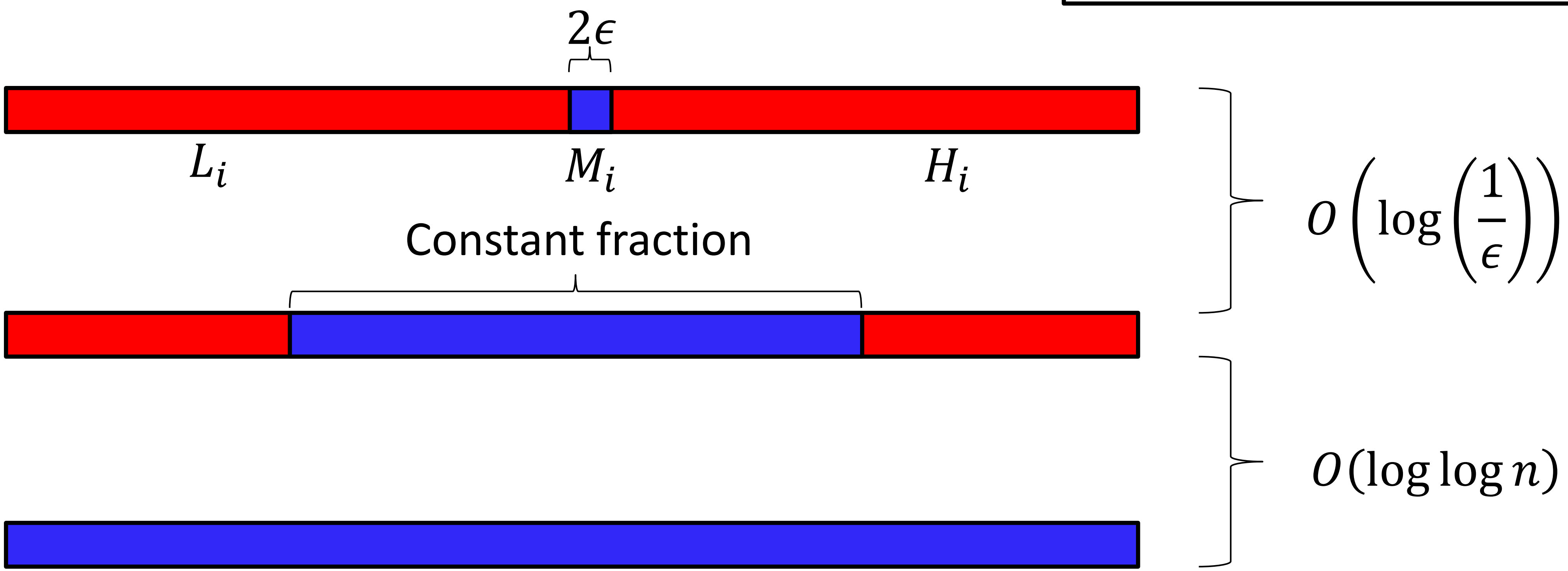
After $O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ iterations, the rank of every value is in $(0.5 \pm \epsilon)n$

Running Time

- Blue: Nodes whose values are in $(0.5 \pm \epsilon)$ quantile
- Red: Nodes whose values are not in $(0.5 \pm \epsilon)$ quantile

Expected Behavior

$$H_{i+1} = 3H_i^2 - 2H_i^3$$
$$L_{i+1} = 3L_i^2 - 2L_i^3$$



Approximate Median

- Compute the approximate median ($\phi = 0.5$)

$$O\left(\log\log n + \log(\frac{1}{\epsilon})\right) \text{ rounds}$$

- ϕ-quantiles for other ϕ?

ϕ-quantiles for other ϕ?

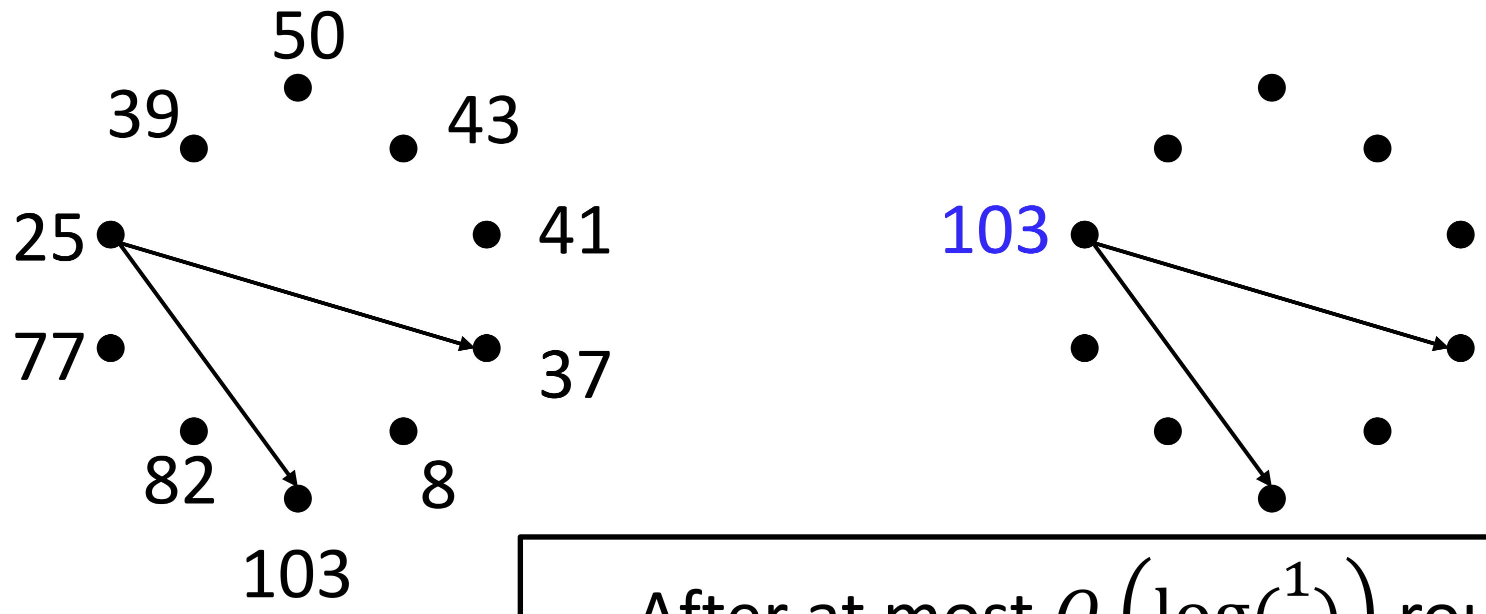
- Phase I: Shift the ϕ-quantile to the approximate median

- Phase II: Compute the approximate median

$O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ rounds

Phase I: Shifting

2-Tournament $(\phi > 1/2)$
For each iteration:
Each node v randomly samples 2 values and sets itself to the higher one



After at most $O\left(\log(\frac{1}{\epsilon})\right)$ rounds, the $(\phi \pm \epsilon)$-quantiles become the current approximate median

Running Time

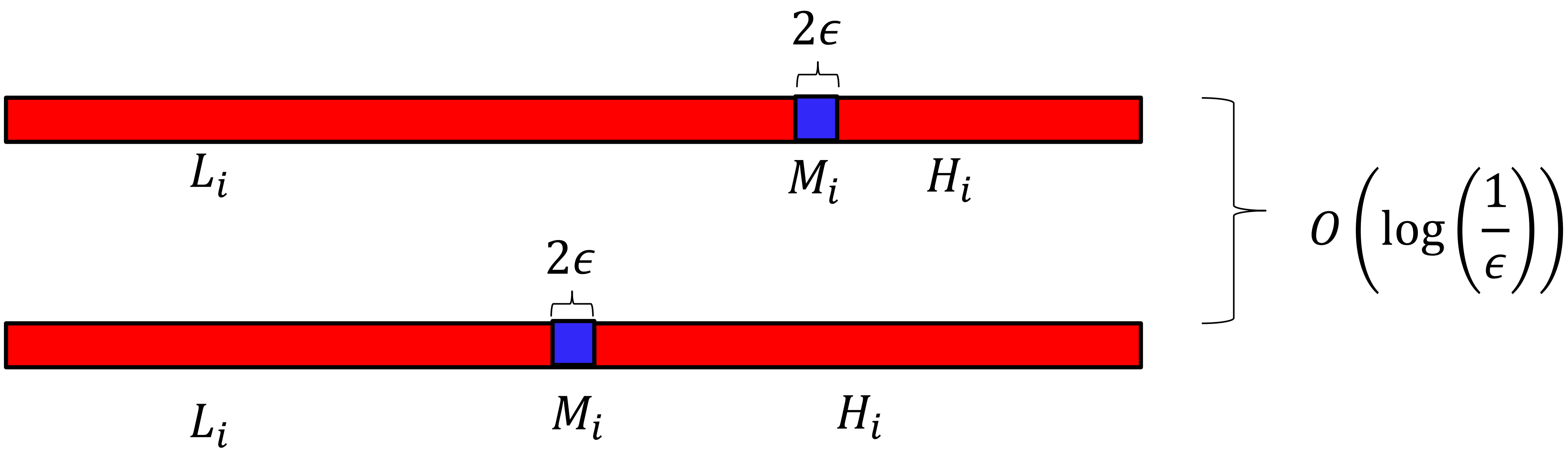
Nodes whose values are in $(\phi \pm \epsilon)$ quantile

Nodes whose values are not in $(\phi \pm \epsilon)$ quantile

Expected Behavior

$L_{i+1} = L_i^2$

$M_{i+1} \geq M_i$



Our Approach

- Phase I: Shift the ϕ-quantile to approximate median

$$O\left(\log(\frac{1}{\epsilon})\right) \text{ rounds}$$

- Phase II: Compute the approximate median

$$O\left(\log\log n + \log(\frac{1}{\epsilon})\right) \text{ rounds}$$

Caveat

- Given $0 \leq \phi \leq 1$ and $0 < \epsilon < 1$, every node outputs a value whose rank is $(\phi \pm \epsilon)n$

$O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ rounds w.h.p., but only for $\epsilon \geq 1/n^{0.01}$

- Example: $\phi = 0.5$ (median), $\epsilon = 1/(2n)$ (exact quantile computation)
  - After the first round of the 3-Tournament algorithm, with a constant probability, the answer is erased.

Quantile Computation for Small ϵ

- **ϕ-quantile:** Given $0 < \phi < 1$, every node outputs a value whose rank is $\lceil \phi n \rceil$

  $O(\log n)$ rounds

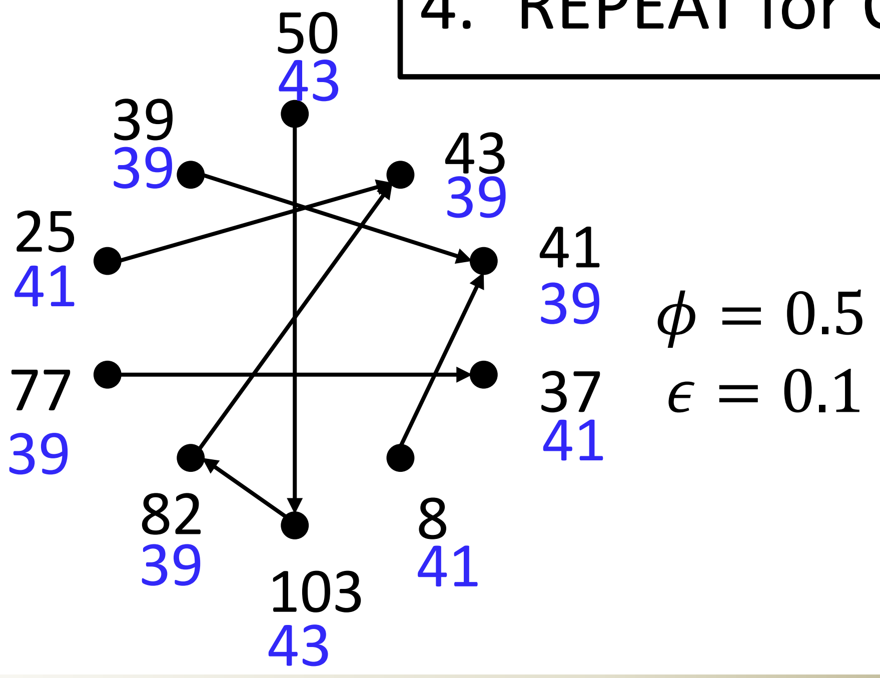
- **ϵ-approximate ϕ-quantile**: Given $0 < \phi < 1$ and $1/n^{0.01} < \epsilon < 1$,every node outputs a value whose rank is $(\phi \pm \epsilon)n$

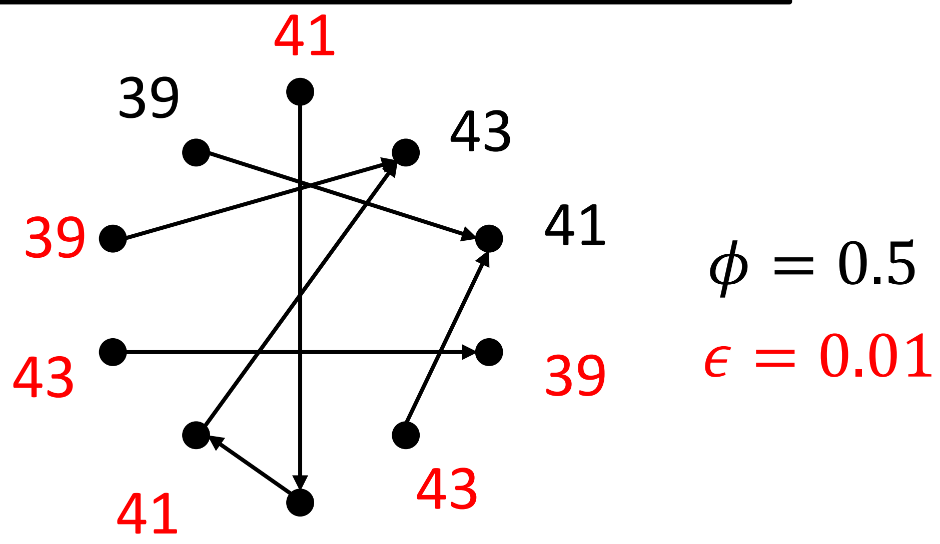
  $O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ rounds

Bootstrapping

- Given $0 \leq \phi \leq 1$, every node outputs a value whose rank is $\lceil \phi n \rceil$
- Bootstrap the approximation algorithm:

1. Run the algorithm for $\epsilon = 1/n^{0.01}$
2. Discard values lying outside $\phi \pm \epsilon$ quantiles
3. Duplicate the remaining values
4. REPEAT for O(1) rounds





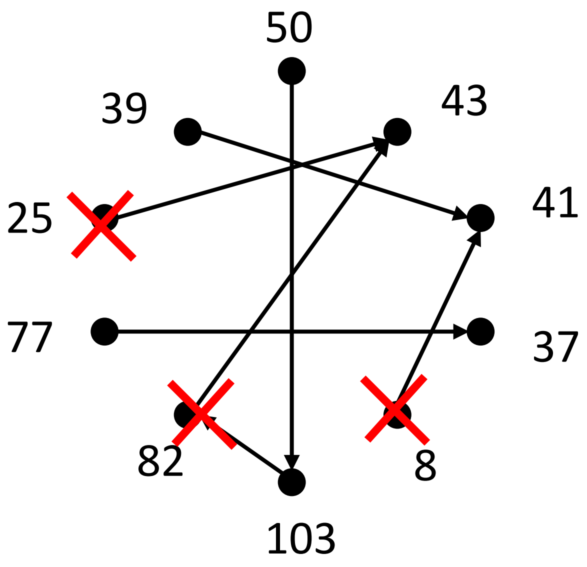
Quantile Computation

- Given $0 \leq \phi \leq 1$ and $0 < \epsilon < 1$, every node outputs a value whose rank is $(\phi \pm \epsilon)n$

$O\left(\log\log n + \log(\frac{1}{\epsilon})\right)$ rounds, but ~~only for $\epsilon \geq 1/n^{0.01}$~~

Robustness

- Our algorithm also tolerates constant probability node failure in each round with the same asymptotic running time.

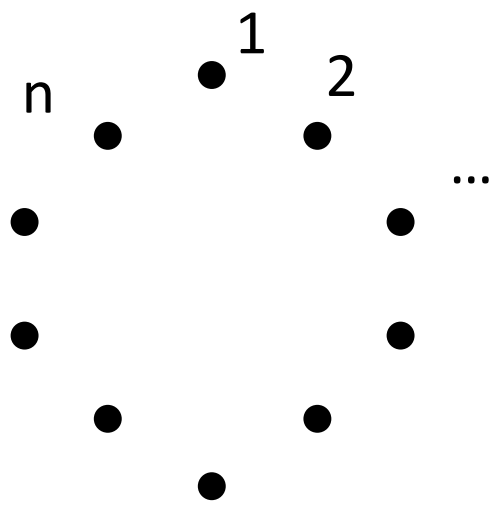


Lower Bound

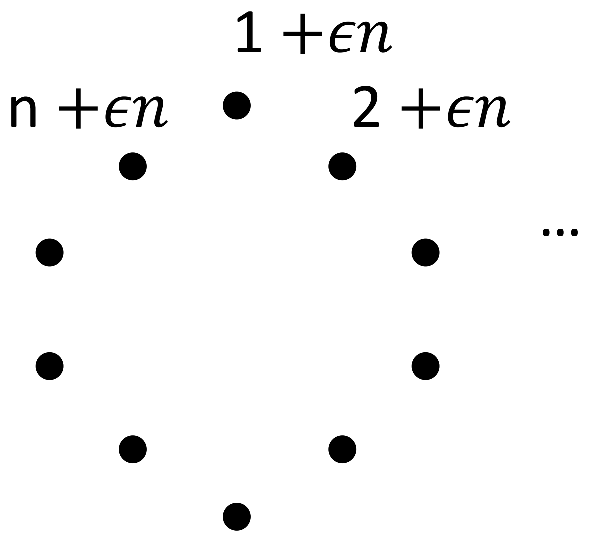
Theorem: Given $\frac{\log n}{n} < \epsilon < 1$, any gossip algorithms that uses $o\left(\log\log n + \log\left(\frac{1}{\epsilon}\right)\right)$ rounds fail to ϵ-approximate the median with prob. at least $\frac{1}{3}$.

Lower Bound

- Indistinguishable Arguments
  - Scenario 1: the values are $\{1, 2, \ldots, n\}$
  - Scenario 2: the values are $\{1 + \epsilon n, 2 + \epsilon n, \ldots, n + \epsilon n\}$



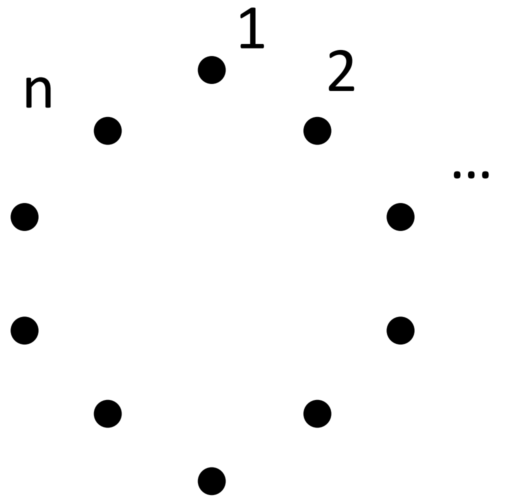
Scenario 1



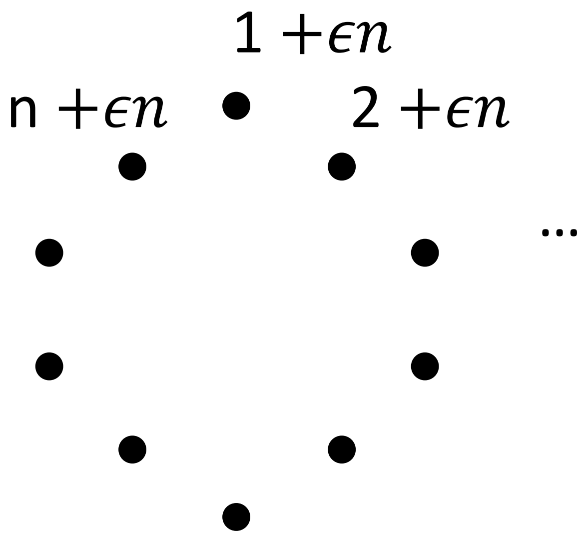
Scenario 2

Lower Bound

- Indistinguishable Arguments
  - The median in the first scenario and the second differ by ϵn
  - In Scenario 1, each node must receive a value in $S = \{1, 2, \ldots, \epsilon n\}$ to ensure it is not in Scenario 2



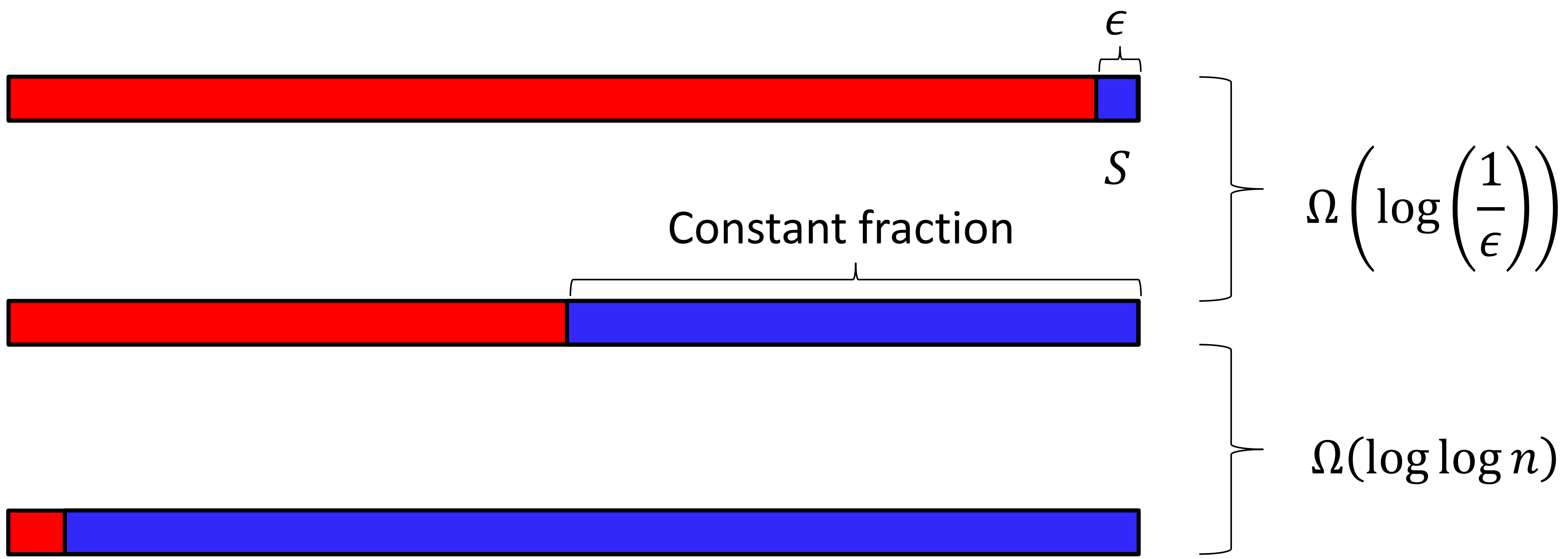
Scenario 1



Scenario 2

Lower Bound

- Lower Bound
  - $|S| = \epsilon n$
  - It takes $\Omega\left(\log\log n + \log\left(\frac{1}{\epsilon}\right)\right)$ to spread messages from S to every node using both PUSH and PULL.



Open Problems

- **For all** ϵ-quantile computation problem
  - Each node v outputs $Q(v)$ such that $|Q(v) - \phi(v)| \leq \epsilon$, where $\phi(v)$ is the quantile of *v*
  - **Approach 1:**
    - Use our algorithm to select the $\frac{\epsilon}{2}, \frac{2\epsilon}{2}, \frac{3\epsilon}{2}, \ldots$ quantiles
    - Each node outputs the nearest quantile
    - $O\left(\frac{1}{\epsilon}\left(\log\log n + \log\left(\frac{1}{\epsilon}\right)\right)\right)$ rounds
  - **Approach 2:**
    - Broadcast every value to every node in $O(n + \log n)$ rounds using Network Coding [Haeupler '11]
    - $O(n)$ rounds for exact answers

Open Problems

- Pipeline problems
  - [Haeupler '11] Broadcast k messages in $O(k + \log n)$ rounds by gossiping.
  - [Kempe, Dobra, Gehrke '03] Compute the sum in $O(\log n)$ rounds

  - Compute k sums in $O(k + \log n)$ rounds?

Thank you