# Randomization and Quantization for Average Consensus

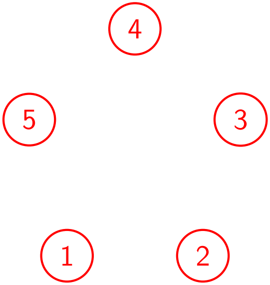
Bernadette Charron-Bost    Patrick Lambein-Monette

Laboratoire d'informatique, École polytechnique

13th International Conference on Reachability Problems
2019–09–12

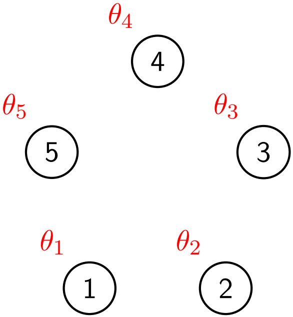
# Problem description

- *V*, a set of *n agents*.



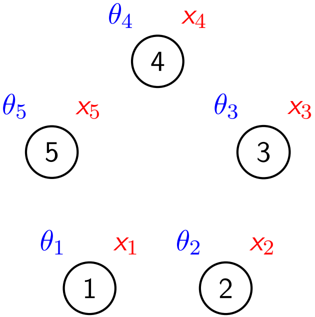
# Problem description

- $V$, a set of $n$ *agents*.
- Each $u \in V$ has input $\theta_u \in \mathbb{R}$.



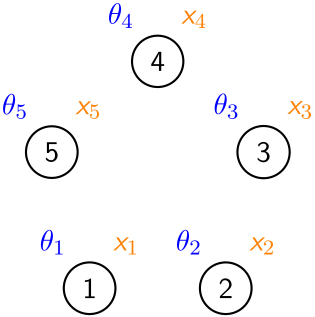
# Problem description

- $V$, a set of $n$ *agents*.
- Each $u \in V$ has input $\theta_u \in \mathbb{R}$.
- Each $u \in V$ has output $x_u(t)$.



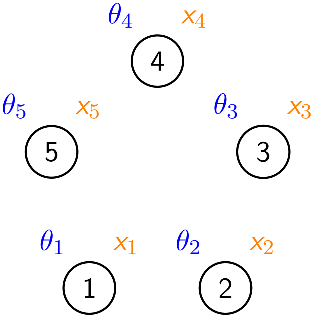
# Problem description

- $V$, a set of $n$ *agents*.
- Each $u \in V$ has input $\theta_u \in \mathbb{R}$.
- Each $u \in V$ has output $x_u(t)$.
- **Goal:** consensus around $\theta := s/n$, with $s := \sum_{u \in V} \theta_u$.



# Problem description

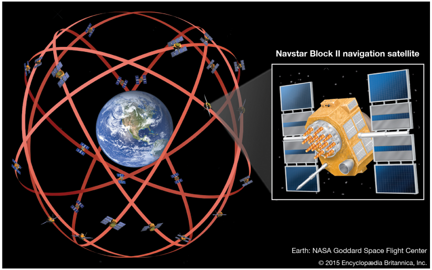
- $V$, a set of $n$ *agents*.
- Each $u \in V$ has input $\theta_u \in \mathbb{R}$.
- Each $u \in V$ has output $x_u(t)$.
- **Goal:** consensus around $\theta := s/n$, with $s := \sum_{u \in V} \theta_u$.




$$\rightsquigarrow x_u(t) \in [\theta - \varepsilon, \theta + \varepsilon]$$

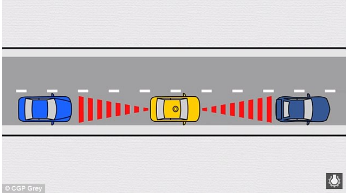
# Applications

- Sensor fusion



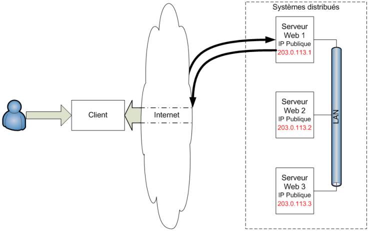
# Applications

- Sensor fusion
- Motion coordination



# Applications

- Sensor fusion
- Motion coordination
- Load balancing



# Applications

- Sensor fusion
- Motion coordination
- Load balancing
- (...)

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |
| Oliva et al. 17 | $\mathcal{O}(nD)$ | $\mathcal{O}(\log n)$ | fixed topology<br>$D \geq$ diameter known to all agents<br>unique identifiers |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |
| Oliva et al. 17 | $\mathcal{O}(nD)$ | $\mathcal{O}(\log n)$ | fixed topology<br>$D \geq$ diameter known to all agents<br>unique identifiers |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |
| Oliva et al. 17 | $\mathcal{O}(nD)$ | $\mathcal{O}(\log n)$ | fixed topology<br>$D \geq$ diameter known to all agents<br>unique identifiers |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |
| Oliva et al. 17 | $\mathcal{O}(nD)$ | $\mathcal{O}(\log n)$ | fixed topology<br>$D \geq$ diameter known to all agents<br>unique identifiers |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |
| Oliva et al. 17 | $\mathcal{O}(nD)$ | $\mathcal{O}(\log n)$ | fixed topology<br>$D \geq$ diameter known to all agents<br>unique identifiers |

# State of the art

| Algorithm | Time | Message size | Restrictions |
|---|---|---|---|
| Flooding | $\mathcal{O}(n)$ | $\mathcal{O}(n \log n)$ | unique identifiers |
| Nedic et al. 09 | $\mathcal{O}(n^2)$ | $\mathcal{O}(1)$ | bidirectional topology<br>3-round stability |
| Yuan et al. 13 | $\mathcal{O}(n)$ | $\infty$ | fixed and bidirectional topology<br>computation intensive |
| Olshevsky 14 | $\mathcal{O}(N)$ | $\infty$ | fixed and bidirectional topology<br>$N \geq n$ known to all agents |
| Oliva et al. 17 | $\mathcal{O}(nD)$ | $\mathcal{O}(\log n)$ | fixed topology<br>$D \geq$ diameter known to all agents<br>unique identifiers |
| Algorithm $\overline{\mathcal{R}}$ | $n-1$ | $\mathcal{O}(\log \log n)$ | Monte Carlo |

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence
- $\mathcal{O}(\log\log n)$ space use

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence
- $\mathcal{O}(\log \log n)$ space use
- no global knowledge

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence
- $\mathcal{O}(\log \log n)$ space use
- no global knowledge
- no bidirectional links

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}\left(n\right)$ convergence
- $\mathcal{O}\left(\log\log n\right)$ space use
- no global knowledge
- no bidirectional links
- no stability of the topology

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence
- $\mathcal{O}(\log \log n)$ space use
- no global knowledge
- no bidirectional links
- no stability of the topology
- no identifiers

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence
- $\mathcal{O}(\log\log n)$ space use
- no global knowledge
- no bidirectional links
- no stability of the topology
- no identifiers

# Properties of the algorithm $\overline{\mathcal{R}}$

- $\mathcal{O}(n)$ convergence
- $\mathcal{O}(\log\log n)$ space use
- no global knowledge
- no bidirectional links
- no stability of the topology
- no identifiers

Builds on the works of Mosk-Aoyama and Shah (2006), and Kuhn et alii (2010).

# Communication

- Closed rounds $t = 1, 2, \ldots$

# Communication

- Closed rounds $t = 1, 2, \ldots$
- During a round $t$, each agent $u$:

# Communication

- Closed rounds $t = 1, 2, \ldots$
- During a round $t$, each agent $u$:
  1. Sends a message $m_u(t)$.

# Communication

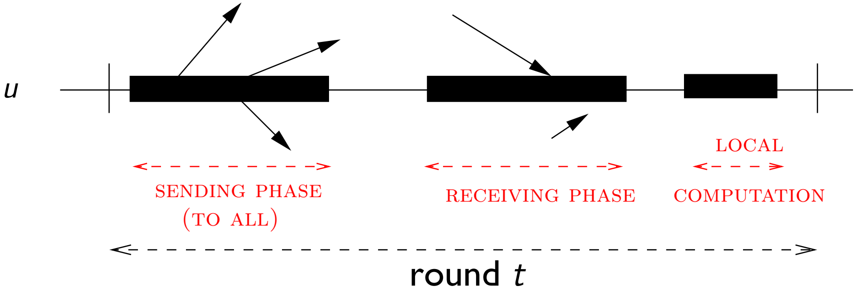
- Closed rounds $t = 1, 2, \ldots$
- During a round $t$, each agent $u$:
  1. Sends a message $m_u(t)$.
  2. Receives some messages $m_{v_1}(t), \ldots, m_{v_k}(t)$ sent by agents $v_1, \ldots, v_k$.

# Communication

- Closed rounds $t = 1, 2, \ldots$
- During a round $t$, each agent $u$:
  1. Sends a message $m_u(t)$.
  2. Receives some messages $m_{v_1}(t), \ldots, m_{v_k}(t)$ sent by agents $v_1, \ldots, v_k$.
  3. Transitions to a new state.

# Communication

- Closed rounds $t = 1, 2, \ldots$
- During a round $t$, each agent $u$:
  1. Sends a message $m_u(t)$.
  2. Receives some messages $m_{v_1}(t), \ldots, m_{v_k}(t)$ sent by agents $v_1, \ldots, v_k$.
  3. Transitions to a new state.



# Communication graph

For each round $t$,

# Communication graph

For each round $t$,

- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$

# Communication graph

For each round $t$,

- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$
- directed graph $\mathbb{G}(t) = (V, E(t))$

# Communication graph

For each round $t$,

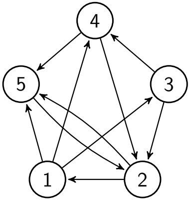
- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$
- directed graph $\mathbb{G}(t) = (V, E(t))$
- $(u, v) \in E(t) \iff u \sim_t v$

# Communication graph

For each round $t$,

- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$
- directed graph $\mathbb{G}(t) = (V, E(t))$
- $(u, v) \in E(t) \iff u \sim_t v$

$\mathbb{G}(1)$:

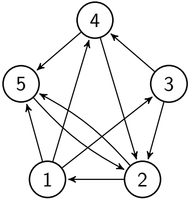


# Communication graph

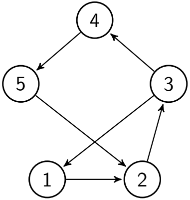
For each round $t$,

- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$
- directed graph $\mathbb{G}(t) = (V, E(t))$
- $(u, v) \in E(t) \iff u \sim_t v$

$\mathbb{G}(1)$:




$\mathbb{G}(2)$:

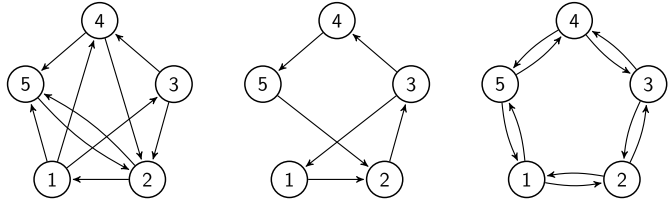


# Communication graph

For each round $t$,

- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$
- directed graph $\mathbb{G}(t) = (V, E(t))$
- $(u, v) \in E(t) \iff u \sim_t v$

$\mathbb{G}(1)$: $\mathbb{G}(2)$: $\mathbb{G}(3)$:

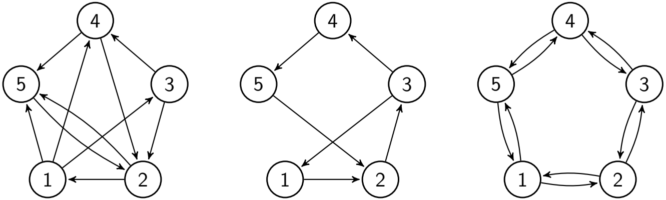


# Communication graph

For each round $t$,

- binary relationship: $u \sim_t v \iff v$ receives $m_u(t)$ at round $t$
- directed graph $\mathbb{G}(t) = (V, E(t))$
- $(u, v) \in E(t) \iff u \sim_t v$

$\mathbb{G}(1)$: $\mathbb{G}(2)$: $\mathbb{G}(3)$:




$(\cdots)$

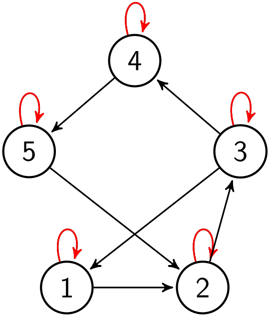
# Communication assumptions

In each round $t$, the graph $\mathbb{G}(t)$:

# Communication assumptions

In each round $t$, the graph $\mathbb{G}(t)$:

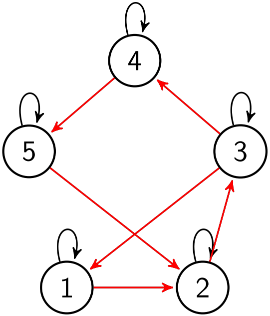
- has a self-loop $(u, u)$ for each vertex $u \in V$



# Communication assumptions

In each round $t$, the graph $\mathbb{G}(t)$:

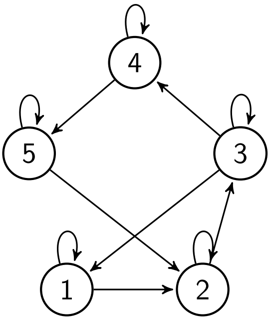
- has a self-loop $(u, u)$ for each vertex $u \in V$
- is strongly connected



# Communication assumptions

In each round $t$, the graph $\mathbb{G}(t)$:

- has a self-loop $(u, u)$ for each vertex $u \in V$
- is strongly connected




$$\rightsquigarrow \mathbb{G}(t) \circ \cdots \circ \mathbb{G}(t + n - 1) = K_V$$

# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$

$x_u \leftarrow \theta_u$

**for** $t = 1, 2, \ldots$ **do**

  Send $x_u$.

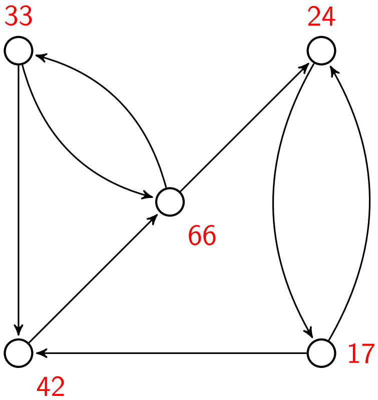
  Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.

  $x_u \leftarrow \min \{x_{v_1}, \ldots, x_{v_k}\}$

**end for**

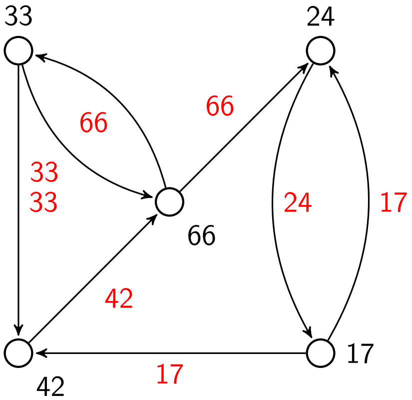
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
  Send $x_u$.
  Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
  $x_u \leftarrow \min\{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



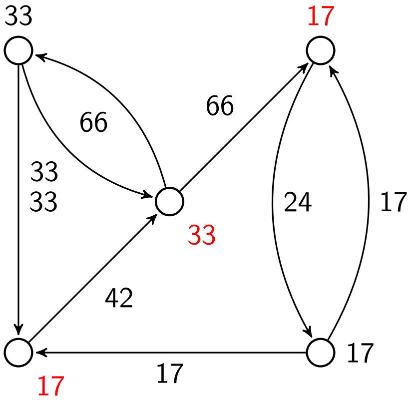
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
  Send $x_u$.
  Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
  $x_u \leftarrow \min\{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



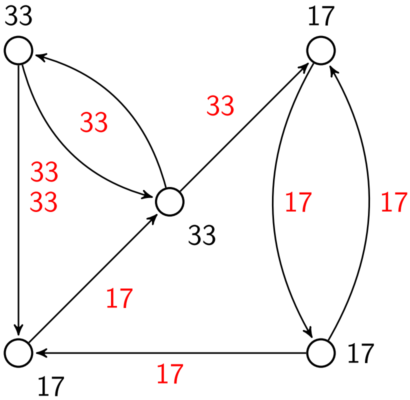
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
Send $x_u$.
Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
$x_u \leftarrow \min\{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



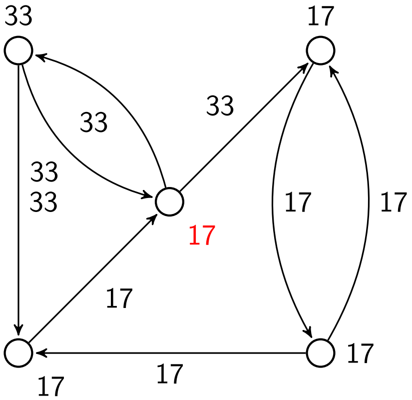
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
    Send $x_u$.
    Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
    $x_u \leftarrow \min\{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



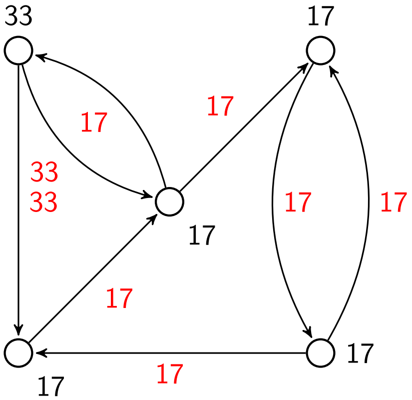
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
  Send $x_u$.
  Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
  $x_u \leftarrow \min\{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



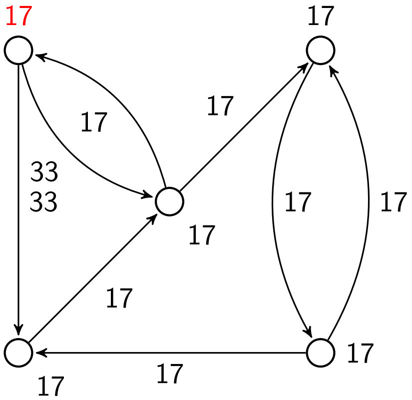
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
Send $x_u$.
Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
$x_u \leftarrow \min \{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



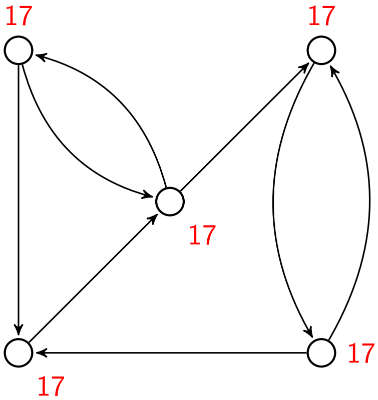
# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
    Send $x_u$.
    Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
    $x_u \leftarrow \min\{x_{v_1}, \ldots, x_{v_k}\}$
**end for**



# E.g., computing the minimum

**Input:** $\theta_u \in \mathbb{R}$
$x_u \leftarrow \theta_u$
**for** $t = 1, 2, \ldots$ **do**
  Send $x_u$.
  Receive $x_{v_1}, \ldots, x_{v_k}$ from neighbors.
  $x_u \leftarrow \min \{x_{v_1}, \ldots, x_{v_k}\}$
**end for**




$\rightsquigarrow$ Converges in at most $n - 1$ rounds.

# Impossibility result (Hendrickx and Tsitsiklis, 2015)

In this model, deterministically computable functions are **order**- and **multiplicity**-independent in the input values,
unless the agents have either:

# Impossibility result (Hendrickx and Tsitsiklis, 2015)

In this model, deterministically computable functions are **order**- and **multiplicity**-independent in the input values,
unless the agents have either:

- unique identifiers

# Impossibility result (Hendrickx and Tsitsiklis, 2015)

In this model, deterministically computable functions are **order**- and **multiplicity**-independent in the input values,
unless the agents have either:

- unique identifiers
- knowledge of their out-degree

# Impossibility result (Hendrickx and Tsitsiklis, 2015)

In this model, deterministically computable functions are **order**- and **multiplicity**-independent in the input values,
unless the agents have either:

- unique identifiers
- knowledge of their out-degree

even for a fixed topology.

# Impossibility result (Hendrickx and Tsitsiklis, 2015)

In this model, deterministically computable functions are **order**- and **multiplicity**-independent in the input values,
unless the agents have either:

- unique identifiers
- knowledge of their out-degree

even for a fixed topology.

$\rightsquigarrow$ the average is not computable.

# Impossibility result (Hendrickx and Tsitsiklis, 2015)

In this model, deterministically computable functions are **order**- and **multiplicity**-independent in the input values,
unless the agents have either:

- unique identifiers
- knowledge of their out-degree

even for a fixed topology.

$\rightsquigarrow$ the average is not deterministically computable.

# Exponential random variables

$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda_1), \\ \left(\vdots\right) \\ X_k \sim \mathsf{Exp}(\lambda_k) \end{array}\right\} \quad \text{independent}$$

# Exponential random variables

$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda_1), \\ \left(\vdots\right) \\ X_k \sim \mathsf{Exp}(\lambda_k) \end{array}\right\} \quad \text{independent}$$

$$\rightsquigarrow \min\{X_1, \ldots, X_k\}$$

# Exponential random variables

$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda_1), \\ \left(\vdots\right) \\ X_k \sim \mathsf{Exp}(\lambda_k) \end{array}\right\} \text{ independent}$$

$$\rightsquigarrow \min\{X_1, \ldots, X_k\} \sim \mathsf{Exp}(\lambda_1 + \cdots + \lambda_k)$$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

$\forall u \in V, X_u \sim \mathsf{Exp}(\theta_u)$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

$$\forall u \in V, X_u \sim \mathsf{Exp}(\theta_u) \qquad \Rightarrow X := \min\left\{X_u \mid u \in V\right\} \sim \mathsf{Exp}(s)$$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

$$\forall u \in V, X_u \sim \mathsf{Exp}(\theta_u) \qquad \Rightarrow X := \min\left\{X_u \mid u \in V\right\} \sim \mathsf{Exp}(s)$$

$$\forall u \in V, Y_u \sim \mathsf{Exp}(1)$$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

$$\forall u \in V, X_u \sim \mathsf{Exp}(\theta_u) \qquad \Rightarrow X := \min\left\{X_u \mid u \in V\right\} \sim \mathsf{Exp}(s)$$

$$\forall u \in V, Y_u \sim \mathsf{Exp}(1) \qquad \Rightarrow Y := \min\left\{Y_u \mid u \in V\right\} \sim \mathsf{Exp}(n)$$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

$$\forall u \in V, X_u \sim \mathsf{Exp}(\theta_u) \qquad \Rightarrow X := \min\left\{X_u \mid u \in V\right\} \sim \mathsf{Exp}(s)$$

$$\forall u \in V, Y_u \sim \mathsf{Exp}(1) \qquad \Rightarrow Y := \min\left\{Y_u \mid u \in V\right\} \sim \mathsf{Exp}(n)$$

$$\rightsquigarrow \mathbb{E}\left[X\right] = 1/s, \qquad \mathbb{E}\left[Y\right] = 1/n$$

# First idea

$$\mathbb{E}\left[X \sim \mathsf{Exp}(\lambda)\right] = 1/\lambda$$

$$\forall u \in V, X_u \sim \mathsf{Exp}(\theta_u) \qquad \Rightarrow X := \min\left\{X_u \mid u \in V\right\} \sim \mathsf{Exp}(s)$$

$$\forall u \in V, Y_u \sim \mathsf{Exp}(1) \qquad \Rightarrow Y := \min\left\{Y_u \mid u \in V\right\} \sim \mathsf{Exp}(n)$$

$$\rightsquigarrow \mathbb{E}\left[X\right] = 1/s, \qquad \mathbb{E}\left[Y\right] = 1/n$$

$$\ldots \mathbb{E}\left[Y/X\right] = +\infty$$

# Exponential random variable (continued)

$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda), \\ \left(\vdots\right) \\ X_\ell \sim \mathsf{Exp}(\lambda) \end{array}\right\} \quad \text{i.i.d.}$$

# Exponential random variable (continued)

$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda), \\ \left(\vdots\right) \\ X_\ell \sim \mathsf{Exp}(\lambda) \end{array}\right\} \quad \text{i.i.d.}$$

$$\mathsf{Pr}\left[\left|\frac{X_1 + \cdots + X_\ell}{\ell} - \frac{1}{\lambda}\right| \geq \frac{\alpha}{\lambda}\right] \leq 2\,\mathsf{exp}\left(-\frac{\ell\alpha^2}{3}\right)$$

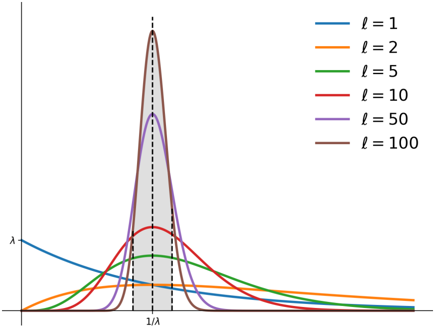
(Cramér-Chernoff bound)

# Exponential random variable (continued)

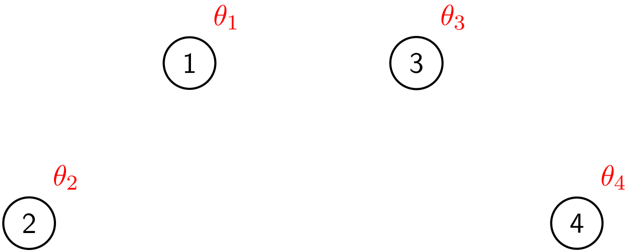
$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda), \\ \left(\vdots\right) \\ X_\ell \sim \mathsf{Exp}(\lambda) \end{array}\right\} \quad \text{i.i.d.}$$

$$\mathsf{Pr}\left[\left|\frac{X_1 + \cdots + X_\ell}{\ell} - \frac{1}{\lambda}\right| \geq \frac{\alpha}{\lambda}\right] \leq 2\,\mathsf{exp}\left(-\frac{\ell\alpha^2}{3}\right)$$

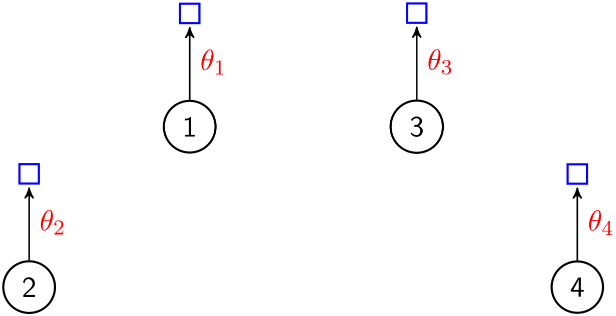
(Cramér-Chernoff bound)



# The algorithm $\mathcal{R}$

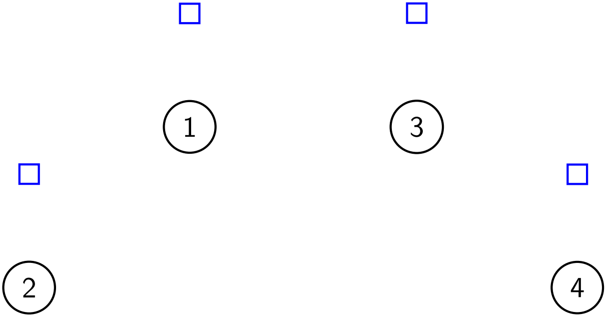


# The algorithm $\mathcal{R}$



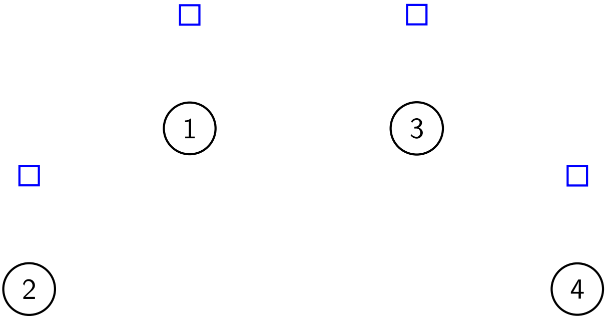
# The algorithm $\mathcal{R}$

$\sigma_1^{(1)} \sim \mathsf{Exp}(\theta_1)$



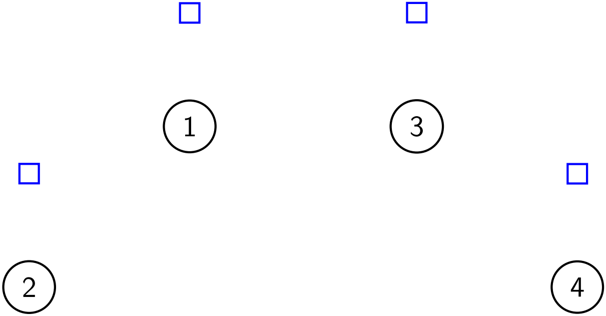
# The algorithm $\mathcal{R}$

$\sigma_1^{(1)}, \dots, \sigma_1^{(\ell)} \overset{\text{iid}}{\sim} \text{Exp}(\theta_1)$

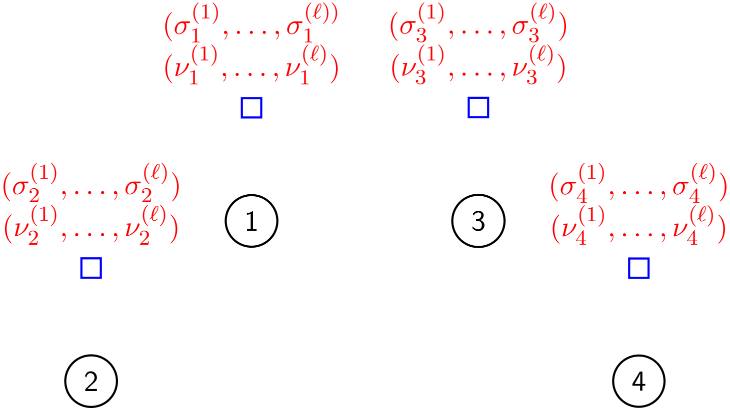


# The algorithm $\mathcal{R}$

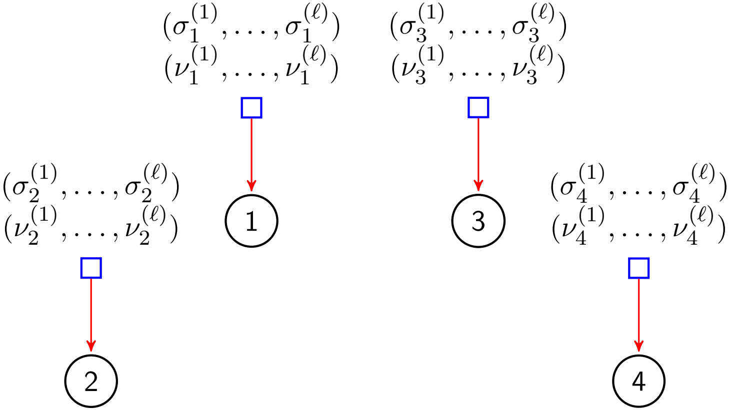
$$\sigma_1^{(1)}, \ldots, \sigma_1^{(\ell)} \overset{\text{iid}}{\sim} \mathsf{Exp}(\theta_1)$$
$$\nu_1^{(1)}, \ldots, \nu_1^{(\ell)} \overset{\text{iid}}{\sim} \mathsf{Exp}(1)$$



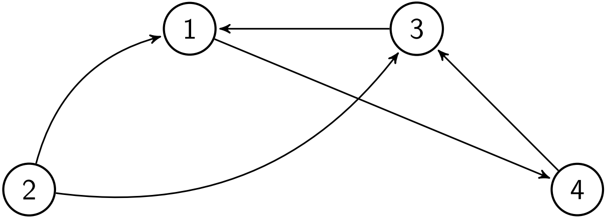
# The algorithm $\mathcal{R}$



# The algorithm $\mathcal{R}$

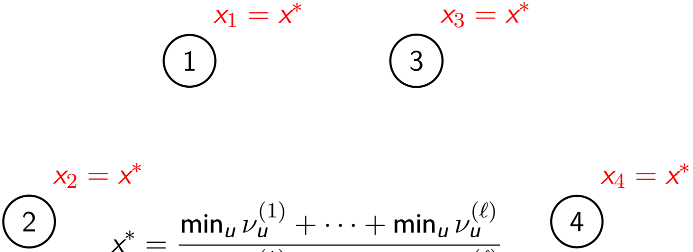


# The algorithm $\mathcal{R}$




(pointwise minimum)

# The algorithm $\mathcal{R}$




$$x^* = \frac{\min_u \nu_u^{(1)} + \cdots + \min_u \nu_u^{(\ell)}}{\min_u \sigma_u^{(1)} + \cdots + \min_u \sigma_u^{(\ell)}}$$

# Convergence of the algorithm $\mathcal{R}_{\varepsilon,\eta}$

$\ell$?

# Convergence of the algorithm $\mathcal{R}_{\varepsilon,\eta}$

$$\ell = \left\lceil 3(2+\varepsilon)^2(\ln 4 - \ln \eta)(b-a+1)^2/\varepsilon^2 \right\rceil$$

Theorem:

- $\forall t \geq n-1, x_u = x^*$
- $\mathbf{Pr}\left[x^* \notin [\theta - \varepsilon, \theta + \varepsilon]\right] \leq \eta$

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0,1])$, agents generate random identifiers, unique with probability $1$.

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0,1])$, agents generate random identifiers, unique with probability $1$.
- Via flooding, we compute the *exact* average.

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0,1])$, agents generate random identifiers, unique with probability $1$.
- Via flooding, we compute the *exact* average.

However...

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0,1])$, agents generate random identifiers, unique with probability $1$.
- Via flooding, we compute the *exact* average.

However...

- With finite memory, random IDs collide with non-null probability.

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0, 1])$, agents generate random identifiers, unique with probability $1$.
- Via flooding, we compute the *exact* average.

However...

- With finite memory, random IDs collide with non-null probability.
- Generating unique identifiers w.h.p. requires agents to know a bound on $n$ to circumvent the birthday paradox.

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0,1])$, agents generate random identifiers, unique with probability $1$.
- Via flooding, we compute the *exact* average.

However...

- With finite memory, random IDs collide with non-null probability.
- Generating unique identifiers w.h.p. requires agents to know a bound on $n$ to circumvent the birthday paradox.
- The algorithm $\mathcal{R}$ does not require global information.

# Relevance of the algorithm $\mathcal{R}$

- Sampling from $\mathcal{U}([0,1])$, agents generate random identifiers, unique with probability $1$.
- Via flooding, we compute the *exact* average.

However...

- With finite memory, random IDs collide with non-null probability.
- Generating unique identifiers w.h.p. requires agents to know a bound on *n* to circumvent the birthday paradox.
- The algorithm $\mathcal{R}$ does not require global information.
- Does it degrade gracefully when real numbers cannot be used?

# Rounding and quantization

We can adapt the algorithm $\mathcal{R}$ so that it requires finite memory and bandwidth.

# Rounding and quantization

We can adapt the algorithm $\mathcal{R}$ so that it requires finite memory and bandwidth.

By appropriately rounding each $\sigma_u^{(i)}$ and $\nu_u^{(i)}$, this new algorithm $\overline{\mathcal{R}}$ uses

$$\mathcal{O}\left((-\log \eta/\varepsilon^2)(\log(\log n - \log \eta) - \log \varepsilon)\right)$$

bits of memory/bandwidth,
at the cost of making $\ell' \leq K\ell$ samples.

# To sum up

- Our Monte Carlo algorithm computes a good approximation of the average $\theta$

# To sum up

- Our Monte Carlo algorithm computes a good approximation of the average $\theta$
- Fast: $n-1$ rounds

# To sum up

- Our Monte Carlo algorithm computes a good approximation of the average $\theta$
- Fast: $n - 1$ rounds
- Efficient: $\mathcal{O}(\log \log n)$ bits of memory/bandwidth

# To sum up

- Our Monte Carlo algorithm computes a good approximation of the average $\theta$
- Fast: $n - 1$ rounds
- Efficient: $\mathcal{O}(\log\log n)$ bits of memory/bandwidth
- Completely distributed: no identifiers, no global information

# To sum up

- Our Monte Carlo algorithm computes a good approximation of the average $\theta$
- Fast: $n - 1$ rounds
- Efficient: $\mathcal{O}(\log \log n)$ bits of memory/bandwidth
- Completely distributed: no identifiers, no global information
- Works with any strongly connected communication topology

# To sum up

- Our Monte Carlo algorithm computes a good approximation of the average $\theta$
- Fast: $n - 1$ rounds
- Efficient: $\mathcal{O}(\log \log n)$ bits of memory/bandwidth
- Completely distributed: no identifiers, no global information
- Works with any strongly connected communication topology
- Can be used to decide rather than simply stabilize

# Fin

Thank you

# Logarithmic rounding

$$r_\beta(x) := (1+\beta)^{\lfloor \log_{1+\beta} x \rfloor}$$

# Logarithmic rounding

$$r_\beta(x) := (1+\beta)^{\left\lfloor \log_{1+\beta} x \right\rfloor}$$

$$\left.\begin{array}{c} X_1 \sim \mathsf{Exp}(\lambda), \\ \left(\vdots\right) \\ X_k \sim \mathsf{Exp}(\lambda) \end{array}\right\} \quad \text{i.i.d.}$$

$$\mathsf{Pr}\left[\left|\frac{r_\beta(X_1) + \cdots + r_\beta(X_\ell)}{\ell} - \frac{1}{\lambda}\right| \geq \frac{\alpha + \beta + \alpha\beta}{\lambda}\right]$$

$$\leq 2\,\mathsf{exp}\left(-\frac{\ell\alpha^2}{3}\right)$$

# Convergence of the algorithm $\overline{\mathcal{R}}$

$$\beta = \varepsilon/(6+\varepsilon)(b-a+1)$$
$$\ell = \left\lceil 3(4+\varepsilon)^2(\ln 8 - \ln \eta)(b-a+1)^2/\varepsilon^2 \right\rceil$$

# Convergence of the algorithm $\overline{\mathcal{R}}$

$$\beta = \varepsilon/(6+\varepsilon)(b-a+1)$$
$$\ell = \left\lceil 3(4+\varepsilon)^2(\ln 8 - \ln \eta)(b-a+1)^2/\varepsilon^2 \right\rceil$$

Theorem:

- $\forall t \geq n-1, x_u = x^*$
- $\Pr\left[x^* \notin [\theta-\varepsilon, \theta+\varepsilon]\right] \leq \eta/2$

# Quantization levels

Each $\sigma_u^{(i)}$, $\nu_u^{(i)}$ can be represented over $Q = \mathcal{O}\left(\frac{1}{\varepsilon}\left(\log n - \log \eta - \log \varepsilon\right)\right)$ quantization levels, with probability at least $1 - \eta/2$.

# Quantization levels

Each $\sigma_u^{(i)}$, $\nu_u^{(i)}$ can be represented over $Q = \mathcal{O}\left(\frac{1}{\varepsilon}\left(\log n - \log \eta - \log \varepsilon\right)\right)$ quantization levels, with probability at least $1 - \eta/2$.

$\rightsquigarrow$ memory/bandwidth in $\mathcal{O}\left((-\log \eta/\varepsilon^2)(\log\left(\log n - \log \eta\right) - \log \varepsilon)\right)$ bits.