Distributed Scheduling using Reversible Dynamics

JINWOO SHIN AND DEVAVRAT SHAH[†]

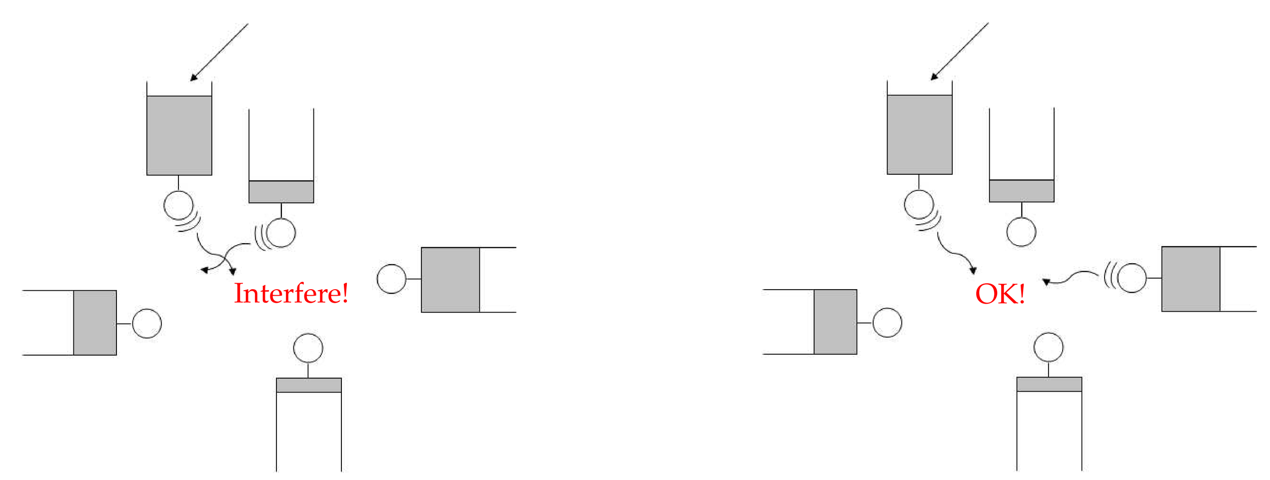
Laboratory for Information and Decision Systems

Massachusetts Institute of Technology

Cambridge, MA 02139. USA.

[†]{jinwoos,devavrat}@mit.edu

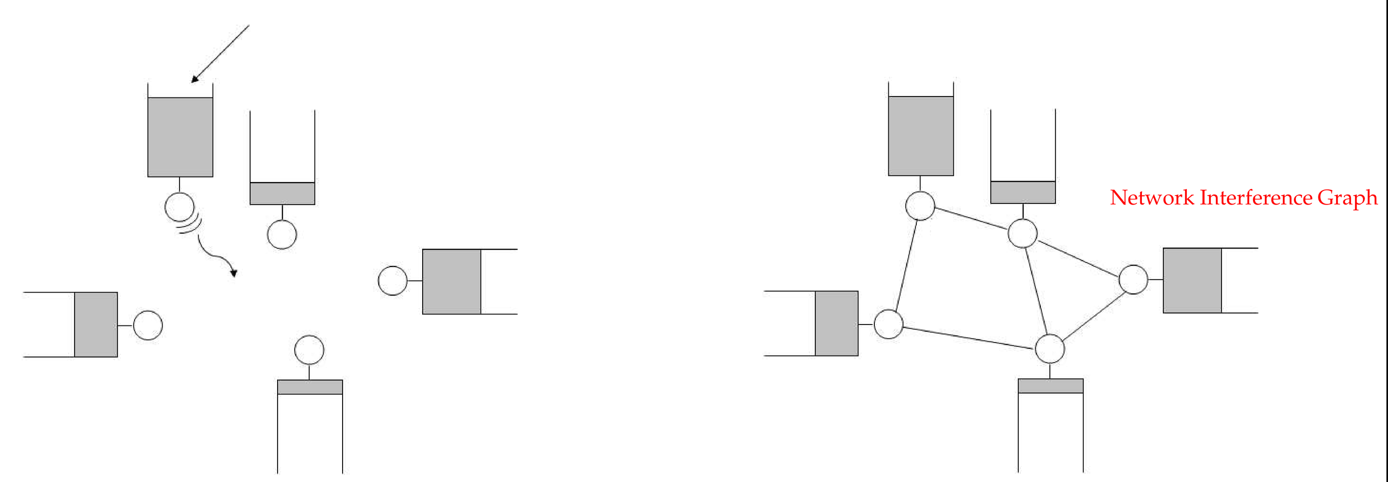
Wireless Network



Constraints:

- Two simultaneously transmitting nodes interfere with each other.

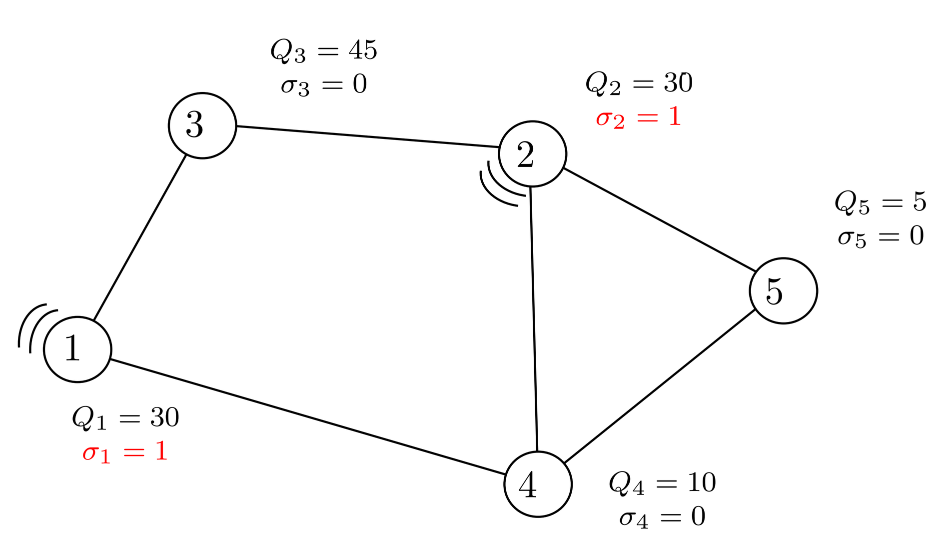
Question : Scheduling



Question:

- Which nodes should transmit simultaneously using "local information".
  - CSMA Information: Each node can sense whether the medium is busy or not.
- So that it is throughput-optimal.
  - It keeps queues finite when the network is underloaded.

Mathematical Model



- Network interference graph $G = (V, E)$ of N queues.
- Independent Poisson packet-arriving process with rate λ_i for queue i.
- $\boldsymbol{Q}(t) = [Q_i(t)] \in \mathbb{R}_+^N$ be the queue-sizes at time t.
- $\boldsymbol{\sigma}(t) = [\sigma_i(t)] \in \{0,1\}^N$ be the schedule at time t.
  - $\sigma_i(t) = 1$ means the queue i is transmitting at time t.
  - $\sigma_i + \sigma_j \leq 1$ if $(i, j) \in E$.

Main Result : Our Algorithm and Its Stability

- Each queue has an independent Exponential clock of rate 1.
- When the clock of the queue i ticks at time t,
  - i checks whether the medium is free i.e. no neighbor of i is transmitting.
  - If yes,

$$\sigma_i(t^+) = \begin{cases} 1 & \text{with probability } \frac{\exp[W_i(t)]}{1+\exp[W_i(t)]} \\ 0 & \text{otherwise.} \end{cases}$$

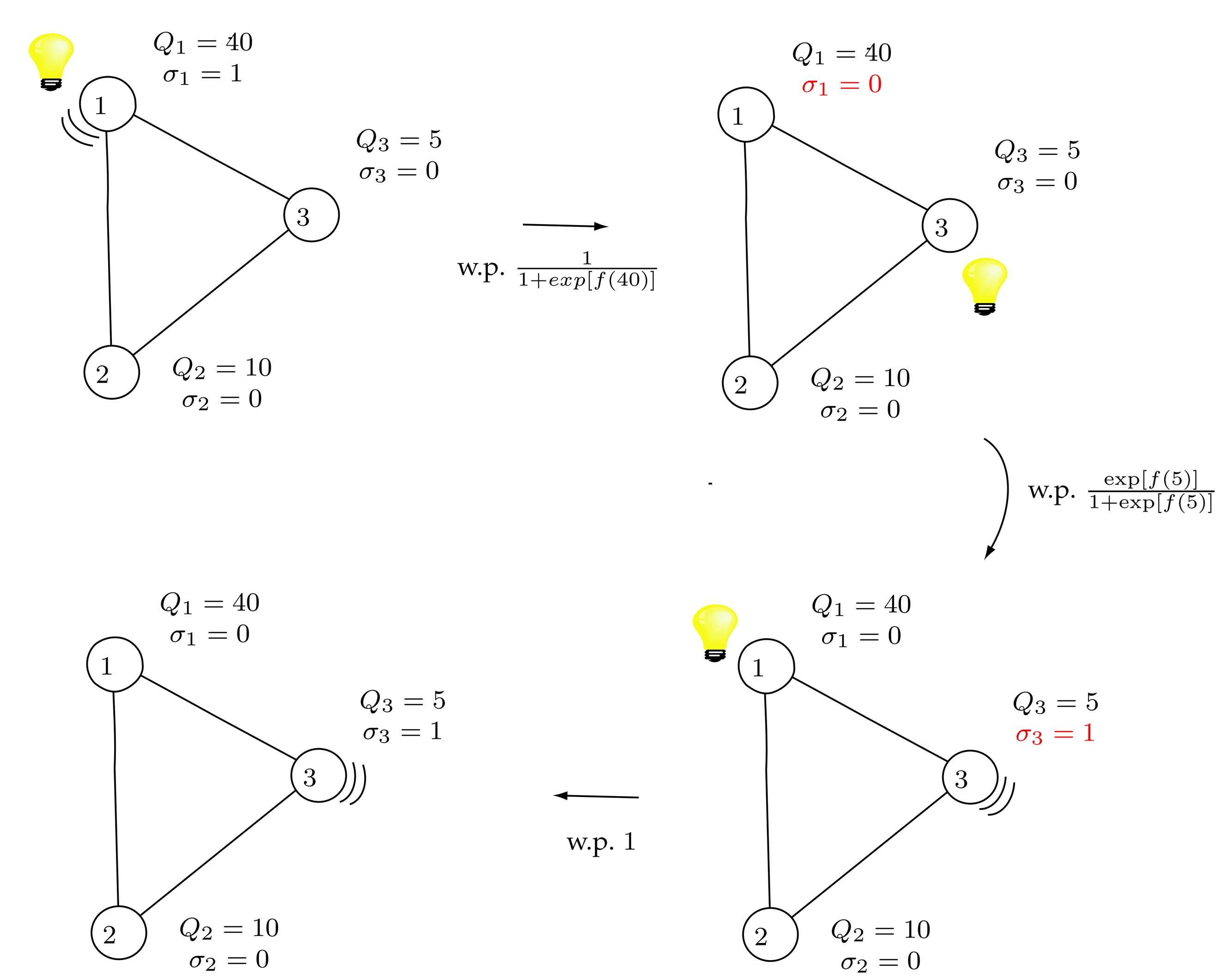
  - Else, do nothing.

Theorem 1 *The algorithm is throughput-optimal with*

$$W_i(t) = \max\left\{f(Q_i(t)), \sqrt{f(Q_{\max}(t))}\right\} \qquad \textit{and} \qquad f(\cdot) = \log\log(\cdot).$$

Example

For simplicity, consider $W_i(t) = f(Q_i(t))$.



Proof Intuition I : Time-varying Glauber Dynamics

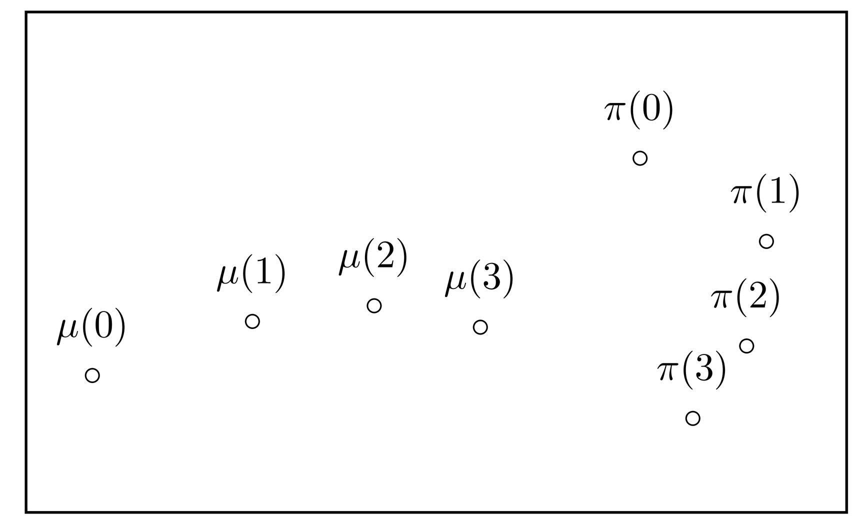
- Our algorithm runs Glauber Dynamics with time-varying weight $\boldsymbol{W}(t)$.
- The stationary distribution π of Glauber Dynamics with weight $\boldsymbol{W}$ satisfies

$$\pi(\boldsymbol{\sigma}) \propto \exp\left[\sum_i W_i \cdot \sigma_i\right].$$

  - High mass on large weighted schedules.
- Therefore, sampling $\boldsymbol{\sigma}$ w.r.t π is essentially a MW (maximum weight) choice!
  - With respect to weight $\boldsymbol{W} \approx f(\boldsymbol{Q})$.

Proof Intuition II : $\mu(t) \approx \pi(t)$

- Let $\mu(t)$ be the actual distribution of $\boldsymbol{\sigma}(t)$ under our algorithm.
- Assume $\mu(t) \approx \pi(t)$.
  - Our algorithm samples essentially the f-MW schedule.
  - The f-MW choice leads to throughput-optimality [Tassiulas and Ephremides 92].
- Main Question: $\mu(t) \approx \pi(t)$?



- If $\pi(t)$ moves slower than $\mu(t)$, $\mu(t)$ eventually catch up $\pi(t)$!

Proof Intuition III : Choice of f

Speed of π and μ:

- $\Delta\pi \approx \Delta W \approx \Delta f(Q) \approx f'(Q)$.
- $\Delta\mu \approx \frac{1}{\exp[W]} \approx \frac{1}{\exp[f(Q)]}$.

Therefore,

$$\Delta\pi < \Delta\mu \qquad \text{if } f(\cdot) = \log\log(\cdot).$$

Discussions and Simulation

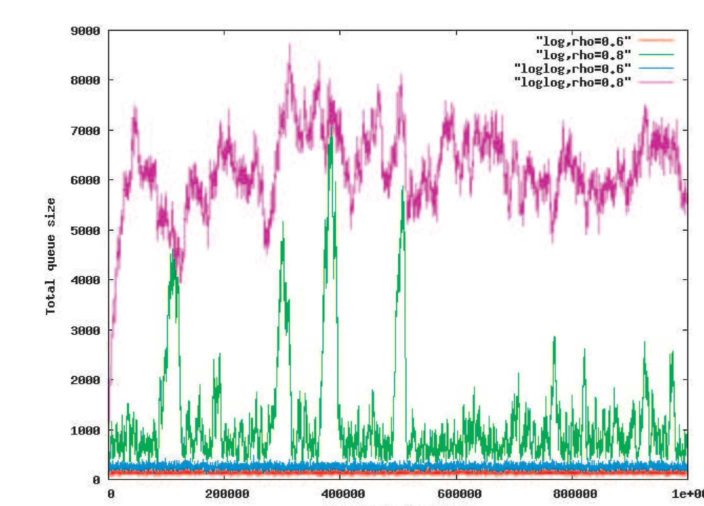
Q1: Why we need $Q_{\max}$ in the weight?

- Due to some technical reasons.
- We believe that it is not necessary.

Q2: How each node know the global information $Q_{\max}$ in a distributed manner?

- Its estimation can be maintained via 1-bit message-passing per unit time.
- Throughput-optimal property does not change under the estimation.

Q3: How about other choices of f?



Comparison between log and log log