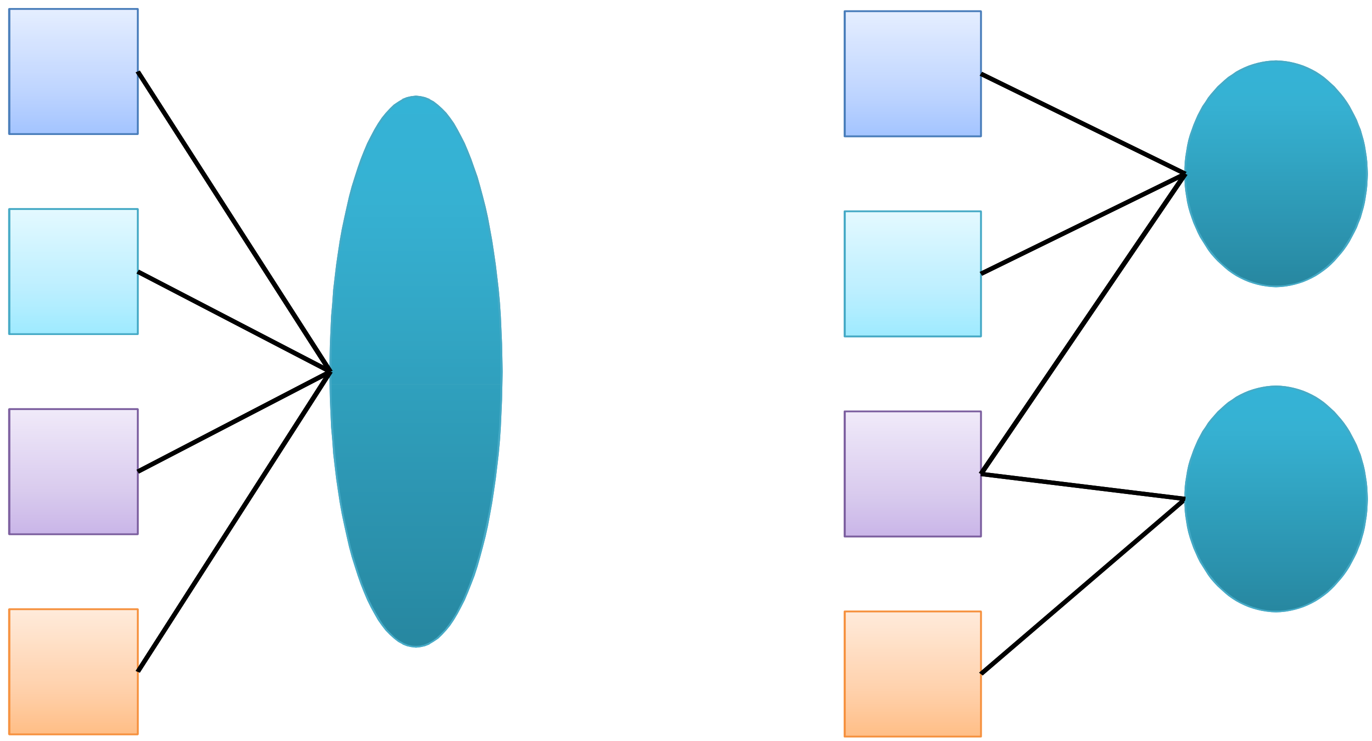
Medium Access using Queues

Devavrat Shah Jinwoo Shin

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

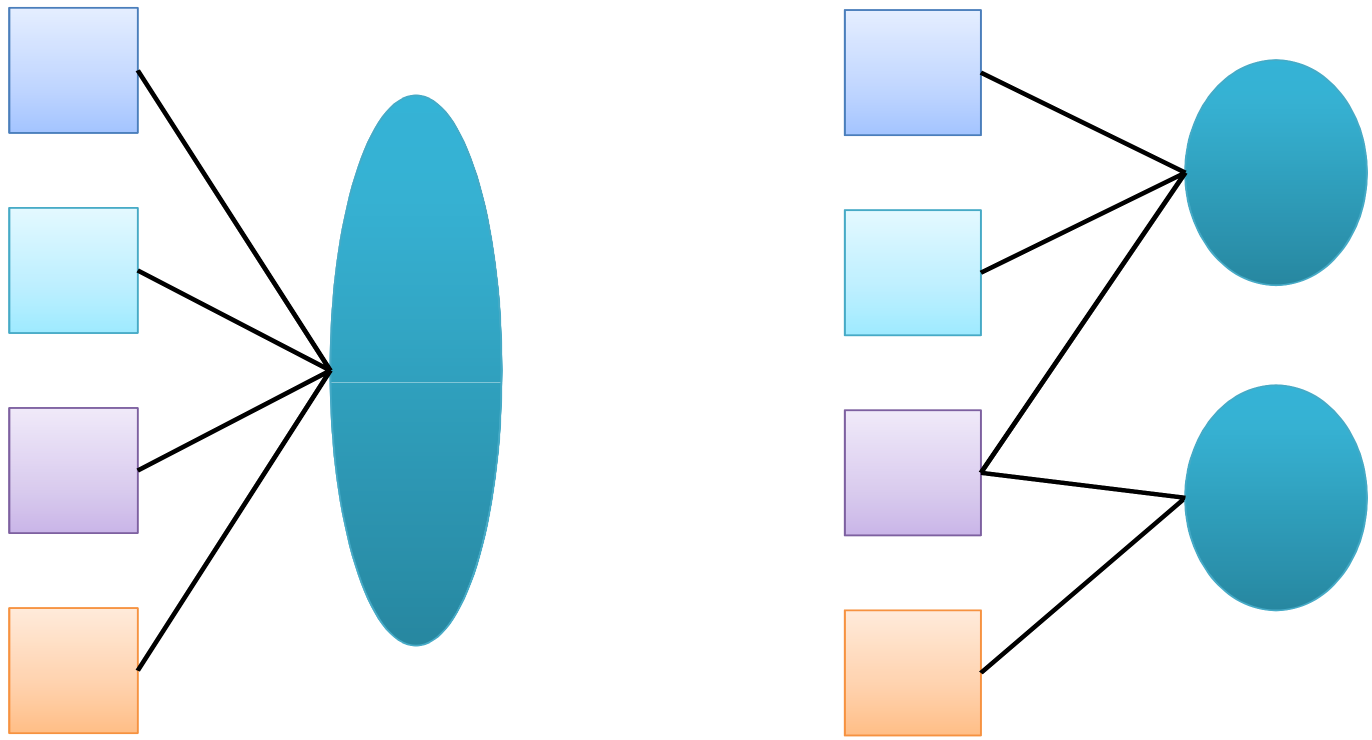
DARPA ITMANET Project Meeting
Austin 2010

Contention resolution



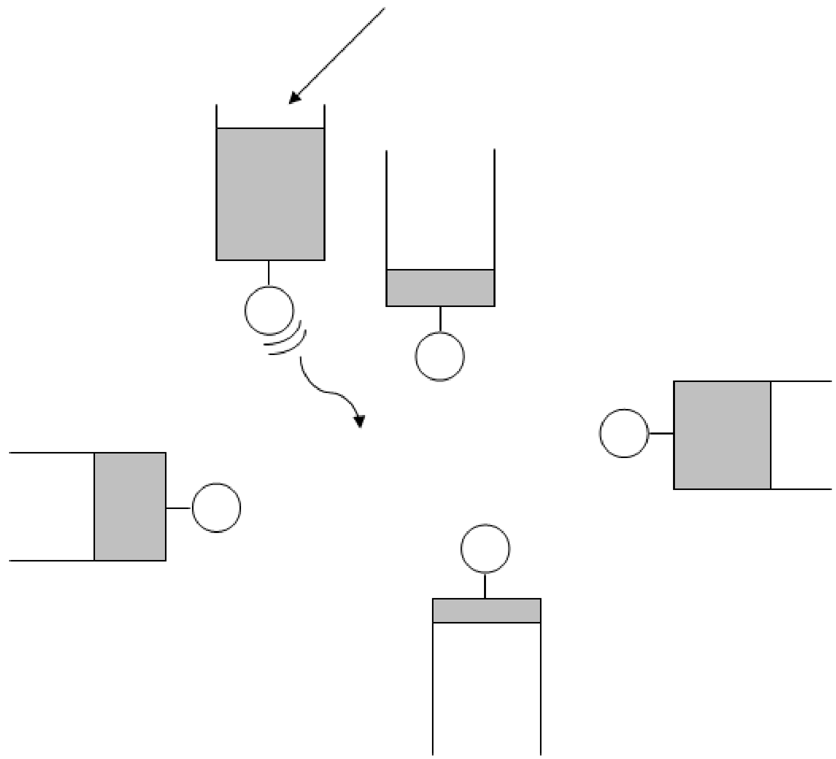
- Examples
  - (Old) Ethernet, wireless network, large software systems, parallel computation, distributed database system,...

Contention resolution

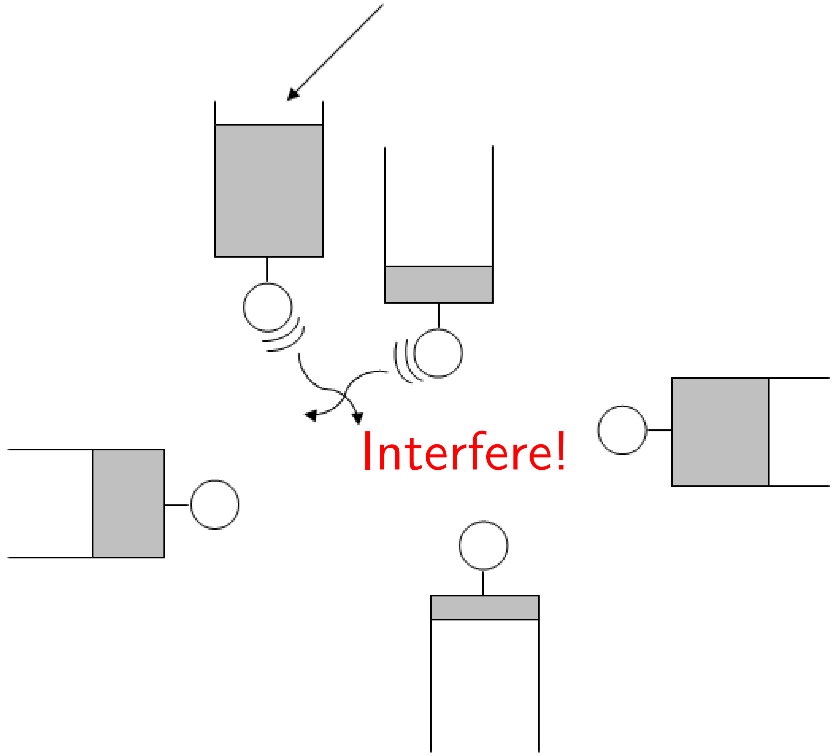


- Key challenge: efficient algorithm design under stringent constraint
  - Minimal co-operation to reduce 'protocol overhead', e.g.
    - nodes know if resource is BUSY or FREE
    - or, their attempt to access was SUCCESS or FAILURE

Model

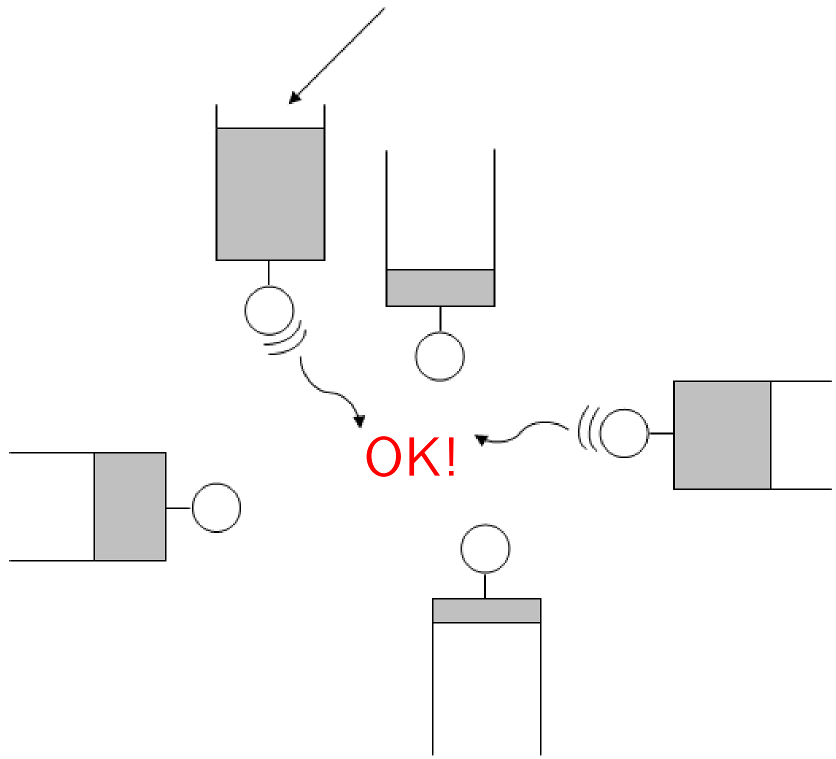


Model



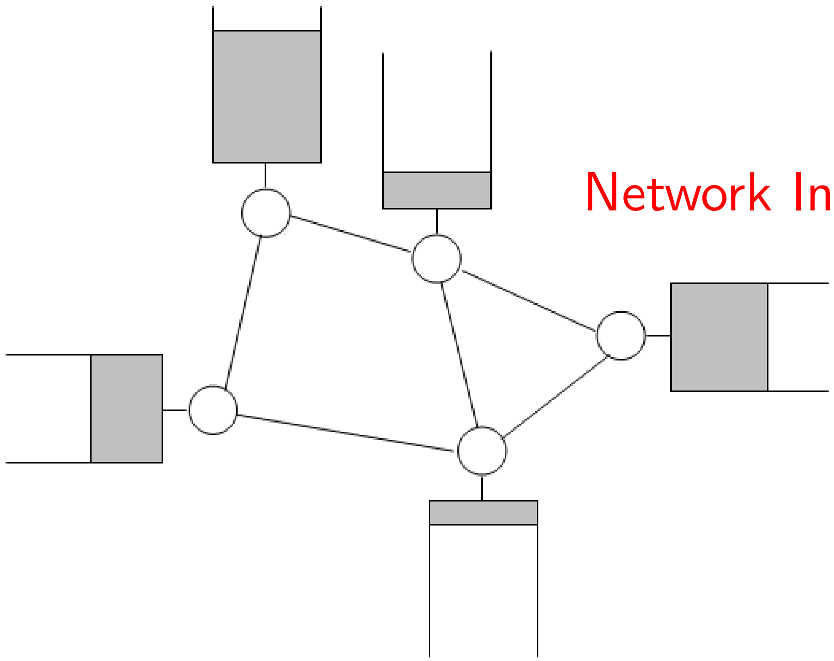
- Constraints
  - Interfering nodes can not transmit simultaneously

Model



- Constraints
  - Interfering nodes can not transmit simultaneously

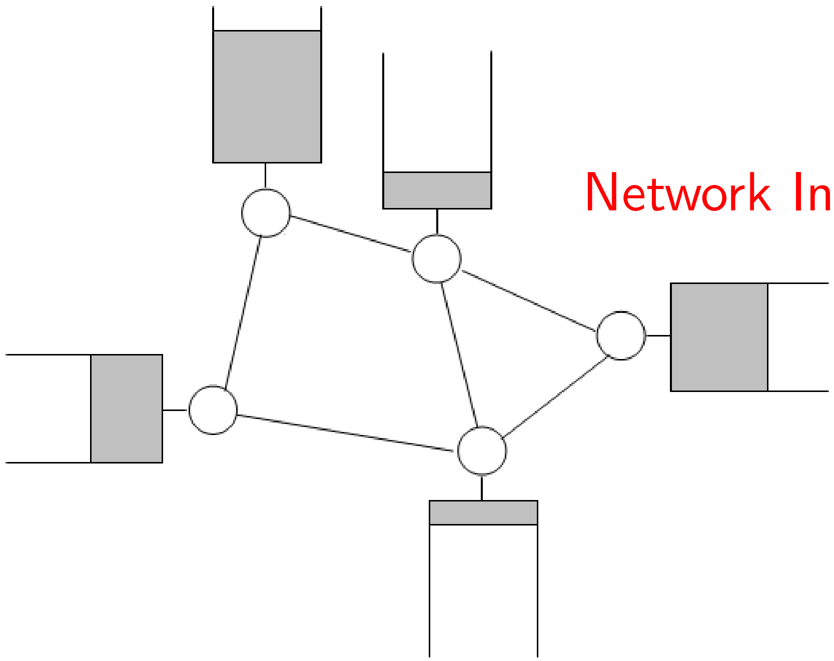
Model



Network Interference Graph

- Constraints
  - Interfering nodes can not transmit simultaneously
  - Nodes have only local information
    - Contending simultaneous transmissions

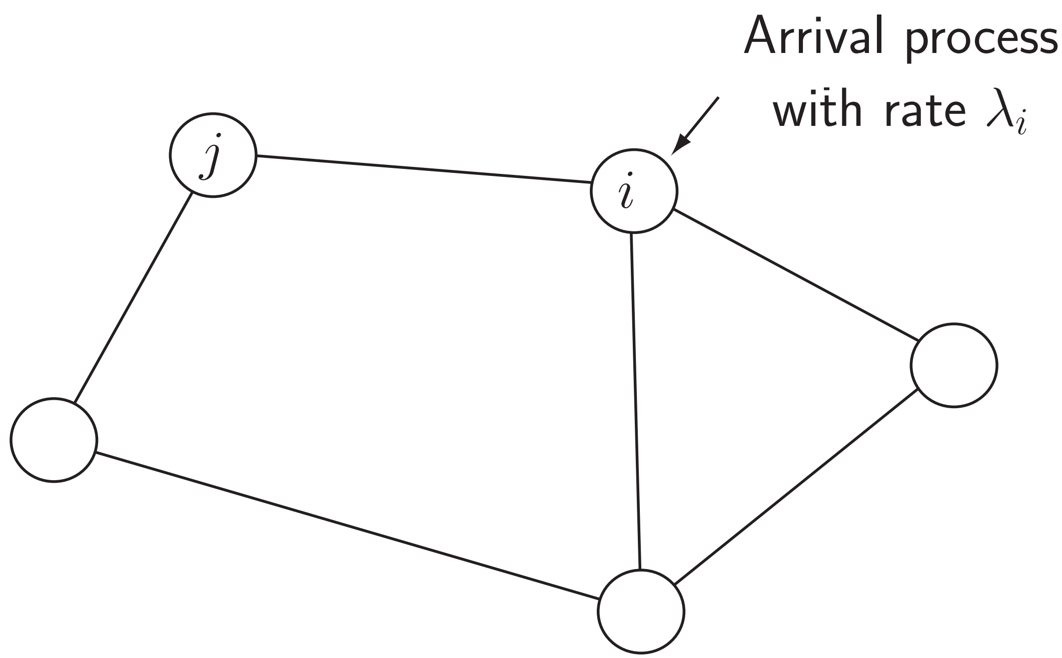
Model



Network Interference Graph

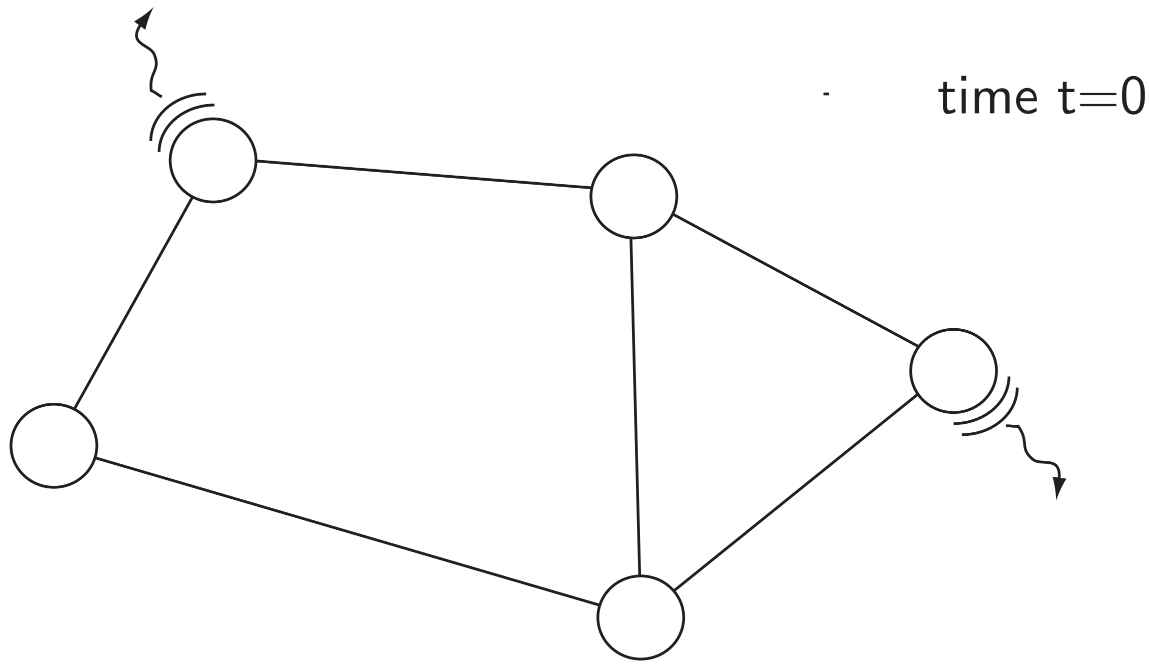
- Medium access
  - When to transmit subject to inference constraints
    - using local information
    - with an aim to maximize utilization of wireless medium

Model



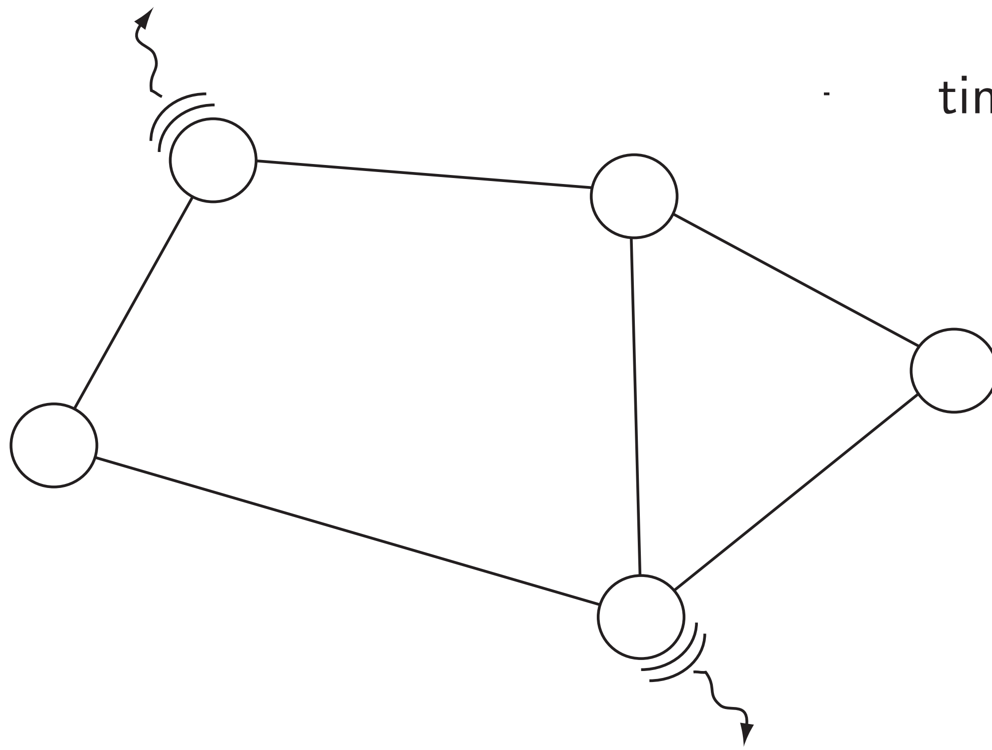
- Network interference graph $G = (V, E)$ with n queues
  - $E = \{(i, j) : \ i$ and j can't tx simultaneously$\}$
  - Packets arrive at rate λ_i for queue i
- Medium access: at each time instance
  - Selects non-interefering queues (to tx), i.e. independent set of G

Model



- Network interference graph $G = (V, E)$ with n queues
  - $E = \{(i, j) :\ i$ and j can't tx simultaneously$\}$
  - Packets arrive at rate λ_i for queue i
- Medium access: at each time instance
  - Selects non-interefering queues (to tx), i.e. independent set of G

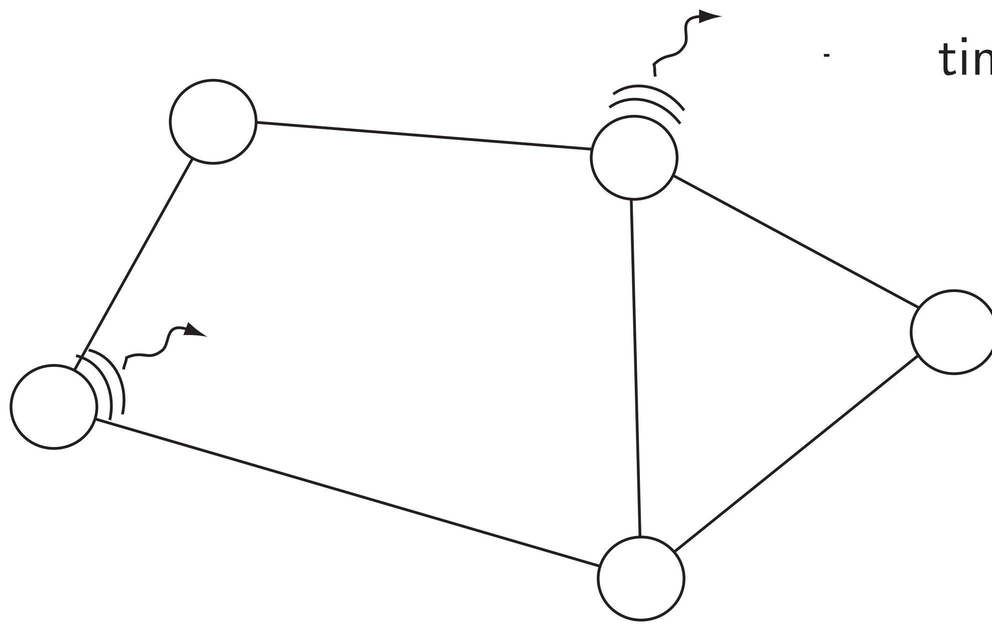
Model



time t=1

- Network interference graph $G = (V, E)$ with n queues
  - $E = \{(i, j) :\ i$ and j can't tx simultaneously$\}$
  - Packets arrive at rate λ_i for queue i
- Medium access: at each time instance
  - Selects non-interefering queues (to tx), i.e. independent set of G

Model



- Network interference graph $G = (V, E)$ with n queues
  - $E = \{(i, j) :\ i$ and j can't tx simultaneously$\}$
  - Packets arrive at rate λ_i for queue i
- Medium access: at each time instance
  - Selects non-interefering queues (to tx), i.e. independent set of G

Model

- Let $\mathcal{I}(G)$ be set of independent sets of G
  - That is, $\mathcal{I}(G) = \{\boldsymbol{\sigma} \in \{0,1\}^n : \sigma_i + \sigma_j \leq 1$ for all $(i,j) \in E\}$

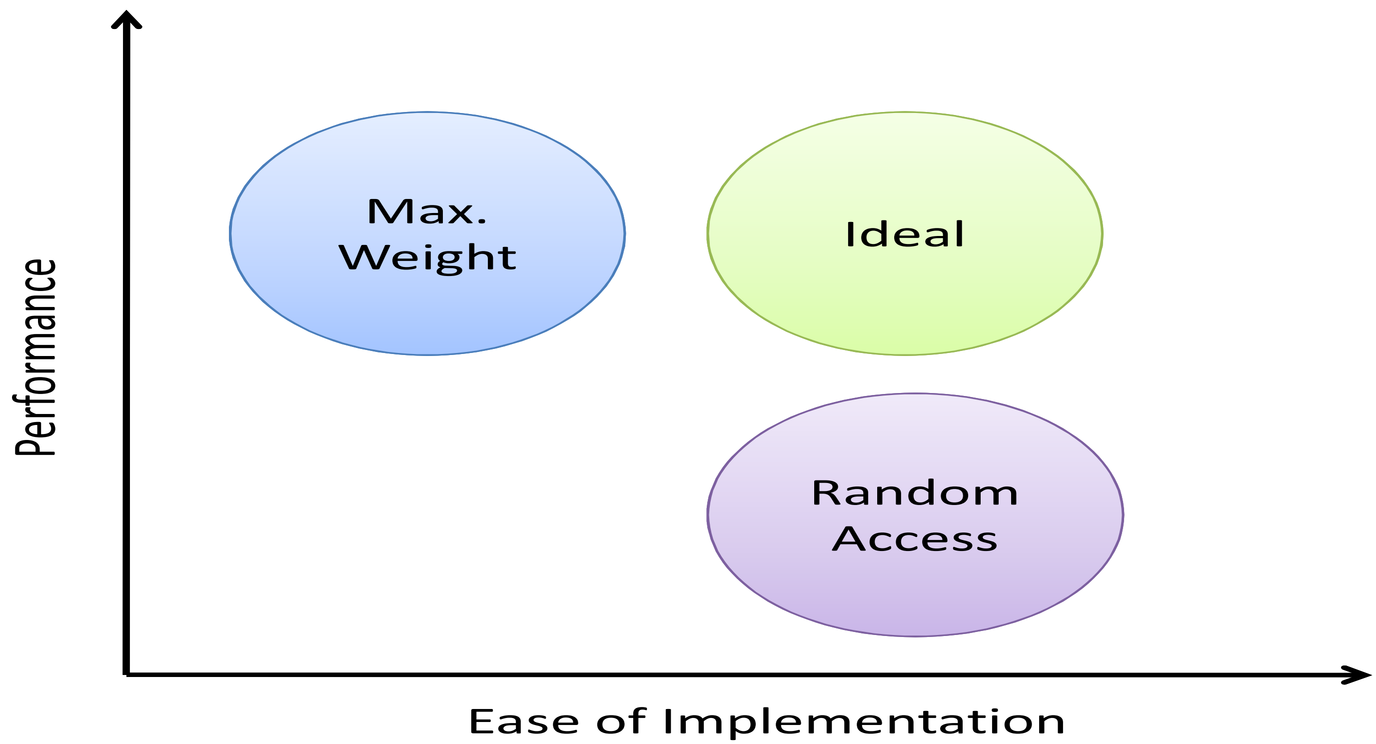
- Effective service rate vector $\mu = [\mu_i]$ is s.t.
  - $\mu = \sum_{\boldsymbol{\sigma} \in \mathcal{I}(G)} \alpha_{\boldsymbol{\sigma}} \boldsymbol{\sigma}$, with $\alpha_{\boldsymbol{\sigma}} \geq 0$
    - $\sum_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}} \leq 1$

- Therefore, effective resource or 'capacity region'
  - Convex hull of $\mathcal{I}(G)$, say conv$(\mathcal{I}(G))$

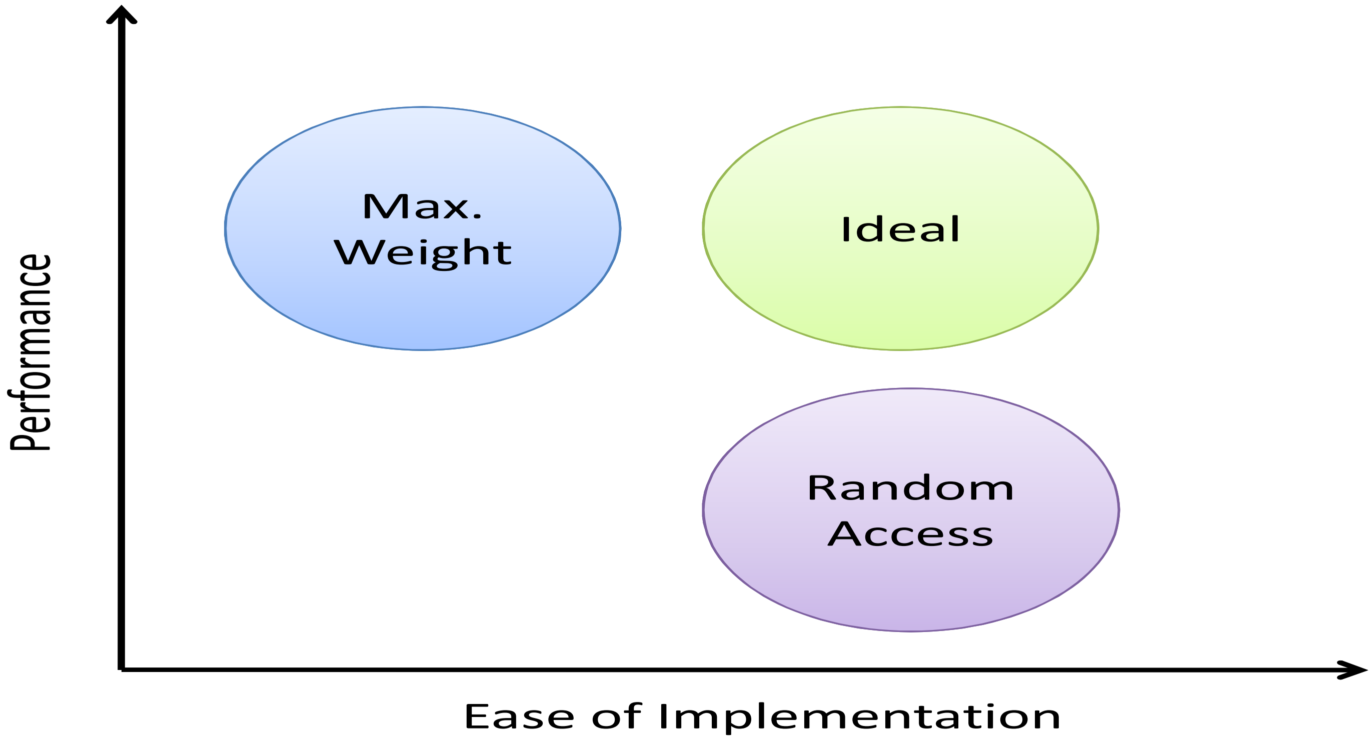
Performance metric

- Throughput optimal medium access
  - Queues remain finite for any $\lambda \in \mathsf{conv}(\mathcal{I}(G))^o$

Status quo


Performance
Max.
Weight
Ideal
Random
Access
Ease of Implementation

Status quo



- Our algorithm
  - Adaptive random access based on queue-size
  - ‘Simulates’ maximum weight algorithm

Our algorithm

- Each queue i checks medium 'regularly'
  - Whether any 'neighboring' node is txing or not
  - If medium is free, attempts transmission with prob. p_i
    - upon being successful, tx for time duration W_i
  - Else
    - do nothing

- Our choice
  - $p_i = 1$ and $\mathbb{E}[W_i] = f(Q_i)$
    - choice of f determines performance crucially
    - a reasonable choice of f is $\log$

Our algorithm: continuous time

- Each queue has an independent Exponential clock of rate $1/2$

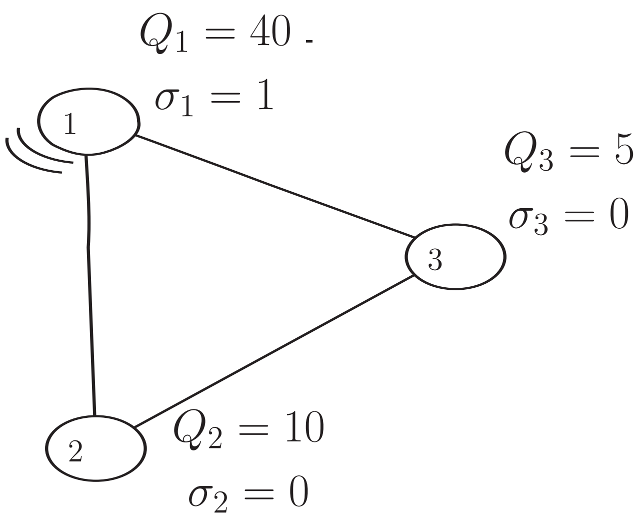
- When clock of queue i ticks, say at time t
  - If $\sigma_i(t^-) = 1$,

$$\sigma_i(t) = \begin{cases} 0 & \text{with probability } \frac{1}{f(Q_i(\lfloor t \rfloor))} \\ 1 & \text{otherwise} \end{cases}$$

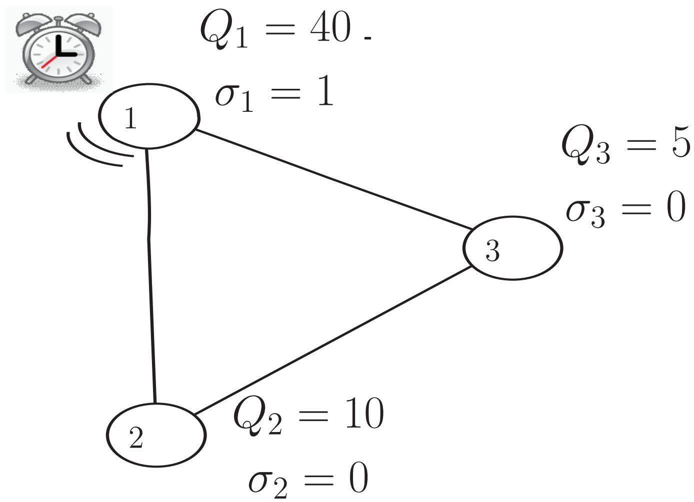
  - Else, i check if medium is free at time t^- and if so,

$$\sigma_i(t) = \begin{cases} 1 & \text{with probability } 1 \\ 0 & \text{otherwise} \end{cases}$$

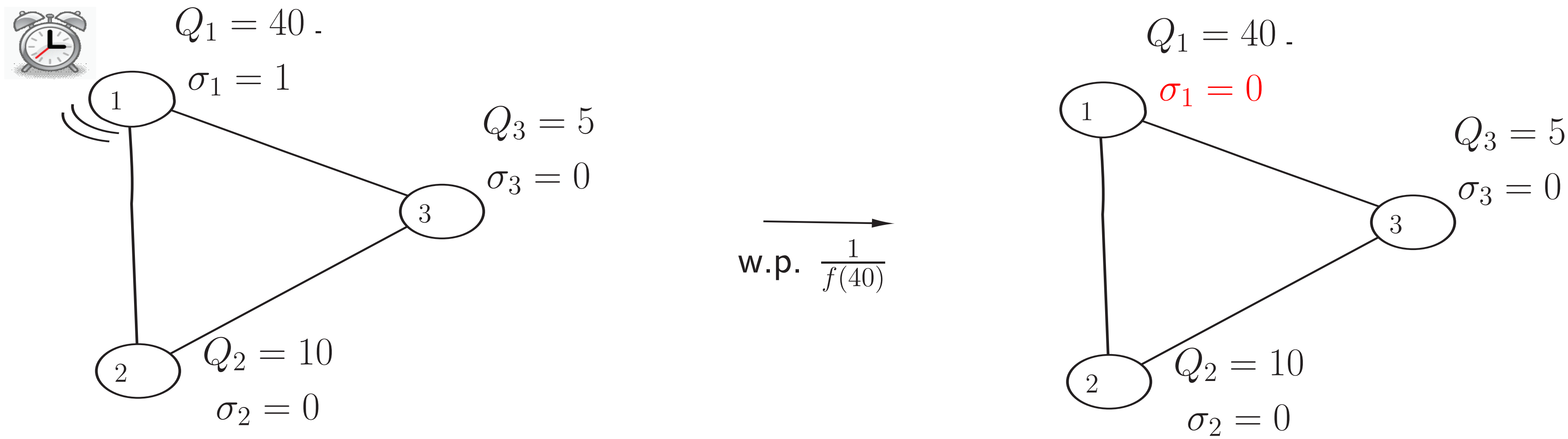
Our algorithm: example (cont time)



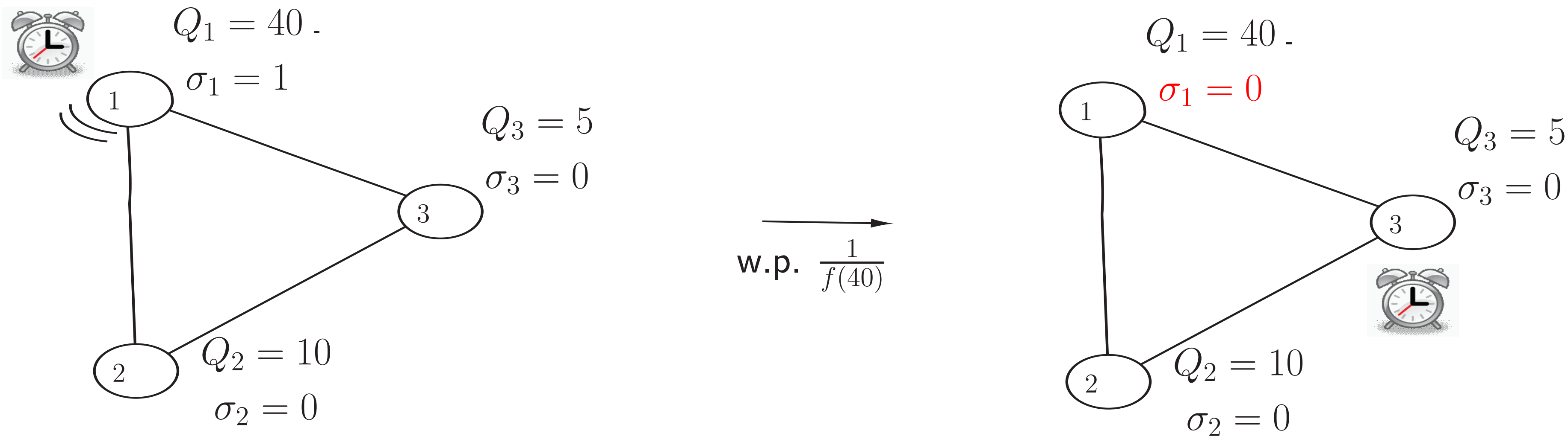
Our algorithm: example (cont time)



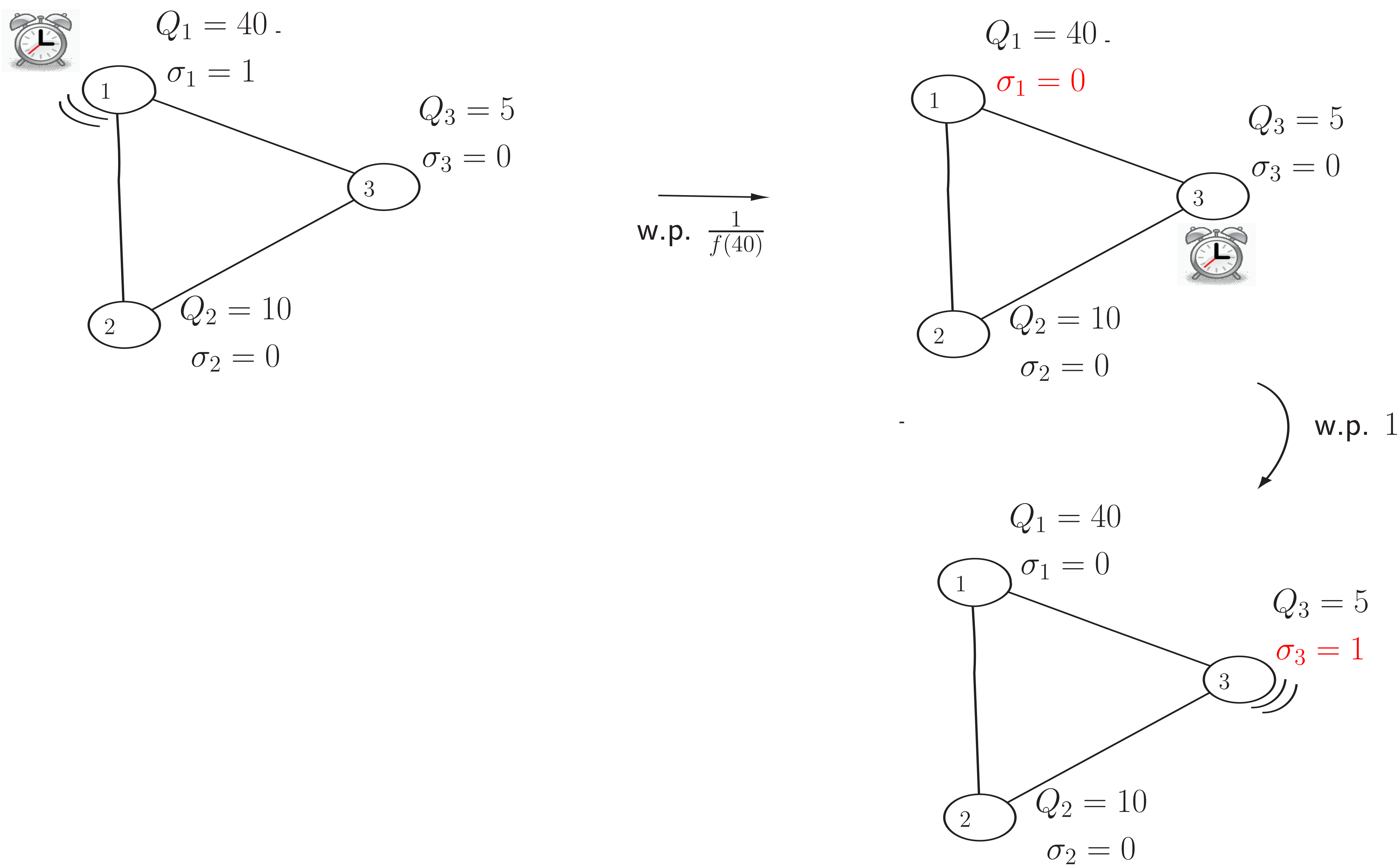
Our algorithm: example (cont time)



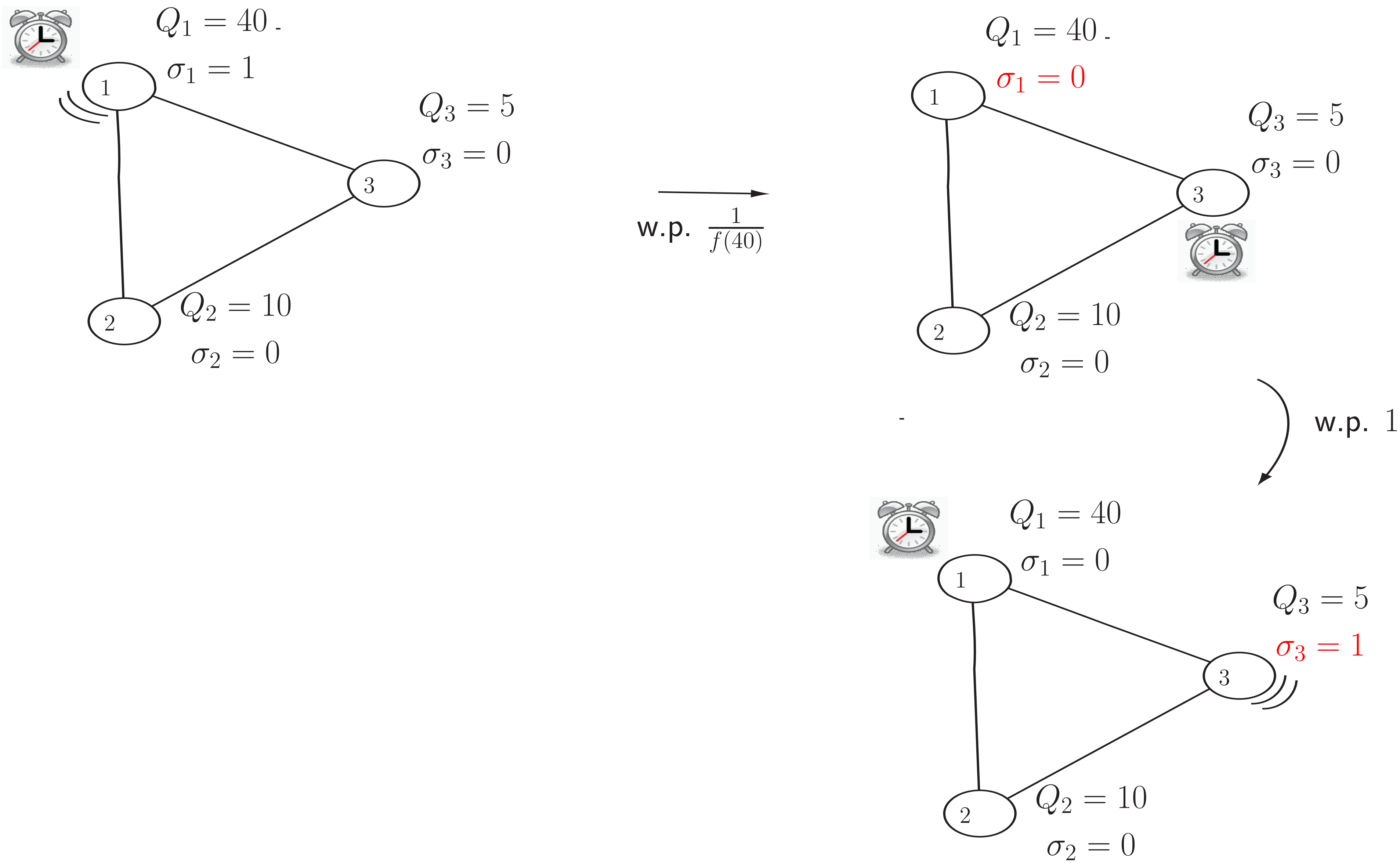
Our algorithm: example (cont time)



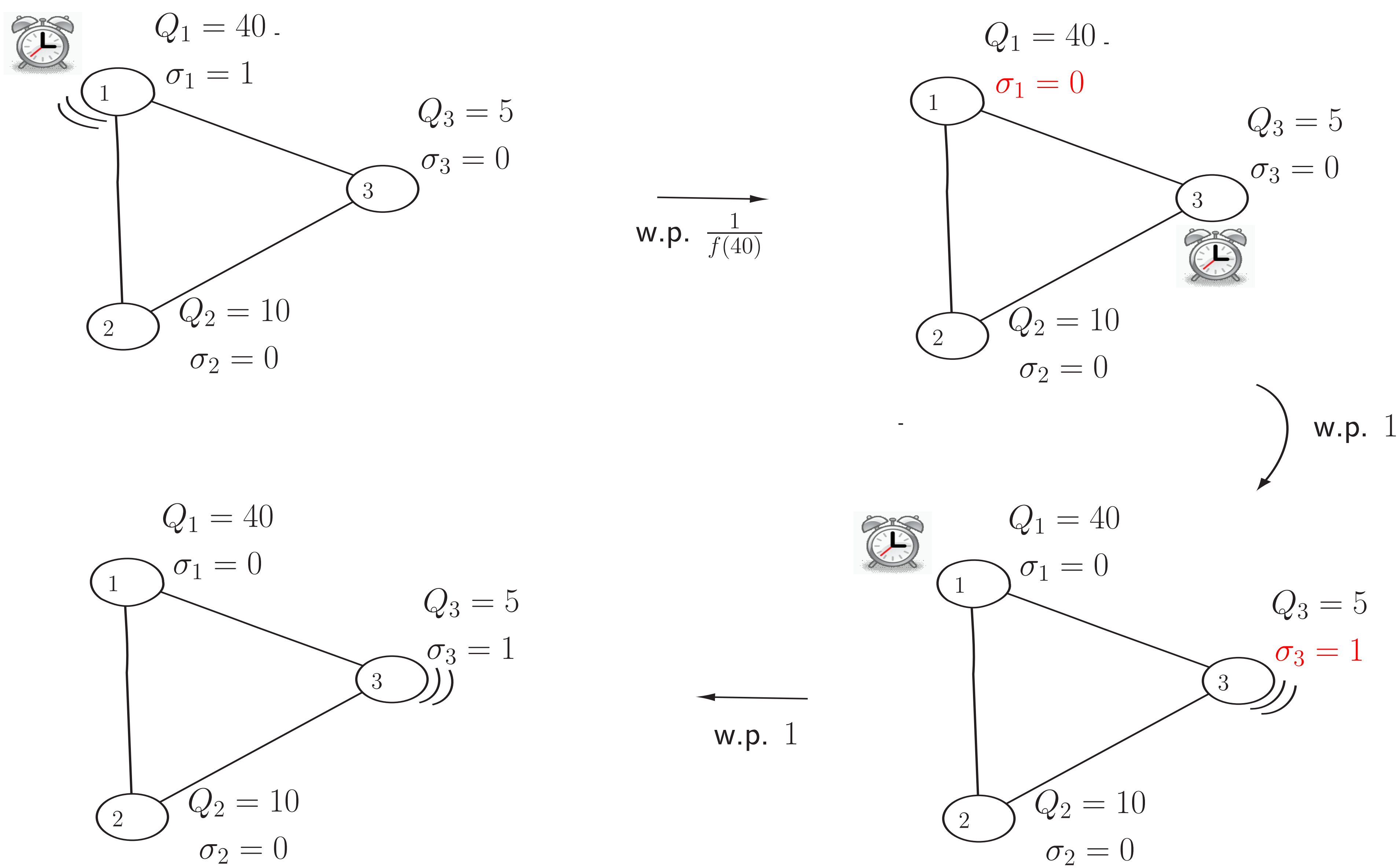
Our algorithm: example (cont time)



Our algorithm: example (cont time)



Our algorithm: example (cont time)



Our algorithm: discrete time

- Each queue has an independent Bernoulli clock of rate $1/2$

- If clock of queue i ticks at time t, then

  ◦ If $\sigma_i(t-1) = 1$,

$$\sigma_i(t) = \begin{cases} 0 & \text{with probability } \frac{1}{f(Q_i(t))} \\ 1 & \text{otherwise} \end{cases}$$

  ◦ Else, i check if medium free at time $t-1$

    - if so, it attempts to transmit with probability 1

$$\sigma_i(t) = \begin{cases} 1 & \text{if no collision} \\ 0 & \text{otherwise} \end{cases}$$

Our algorithm: throughput optimality

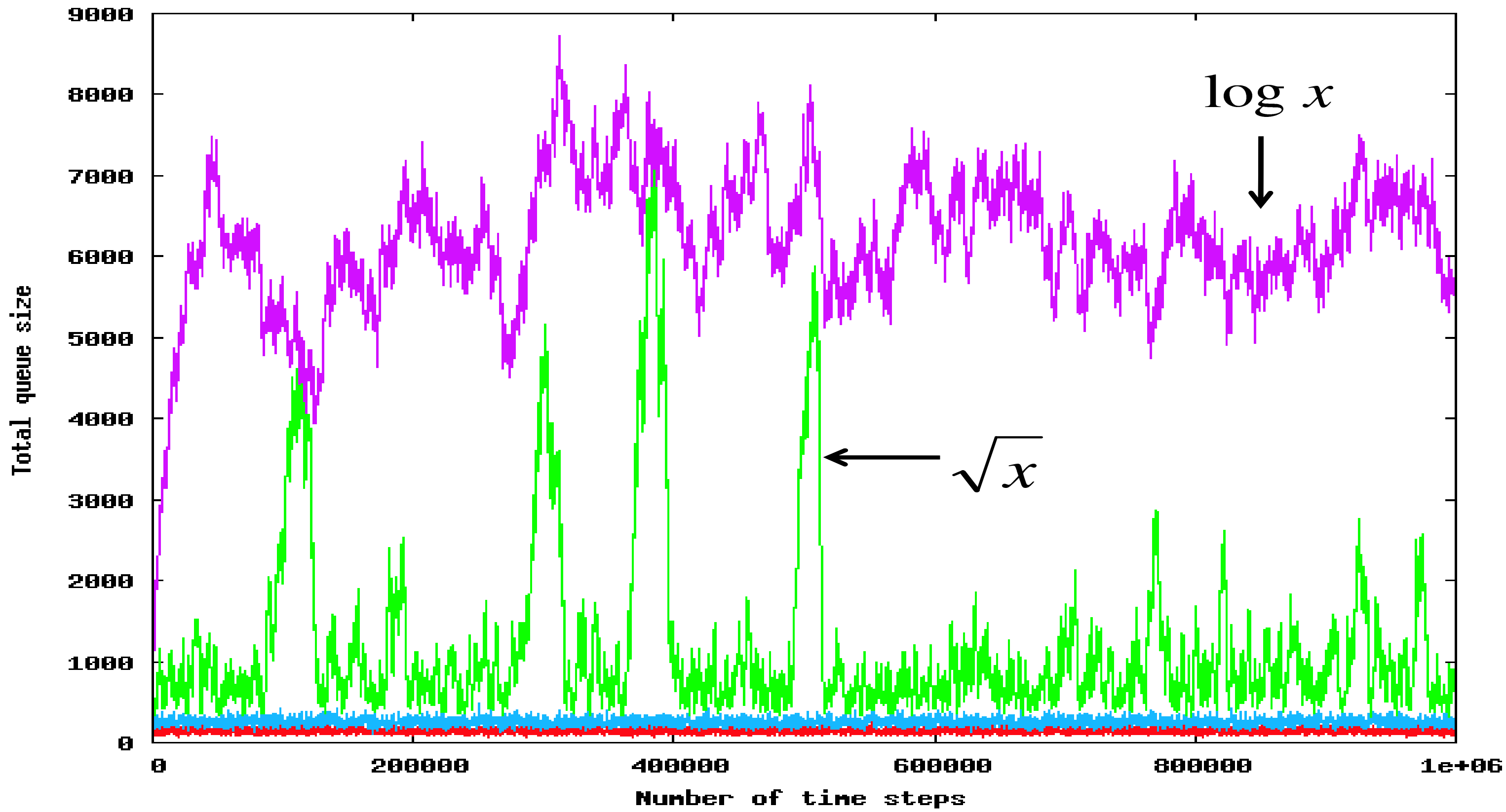
- **Theorem.** [Ragagopalan-**S**-Shin 09, **S**-Shin 09, 10] The algorithm is throughput optimal.
    - For both continuous and discrete time
    - Weight of queue i

$$W_i(t) = \max\Big(f(Q_i(\lfloor t \rfloor)), \sqrt{f(Q_{\mathsf{max}}(\lfloor t \rfloor))}\Big).$$

    - With any $f(x) = \exp(o(\log x))$, like $\log x$, poly$(\log x)$,...

- Specifically, we establish that
    - The network Markov process is positive (Harris) recurrent

Best choice of f?



- Slower f leads to
  - Small ‘variance’ in queue-sizes
  - At the cost of higher ‘average’ queue-sizes

Beyond throughput

- What about queue-sizes (on avg., with high prob.) ?
  - For algorithm described, queue-sizes depend on
    - mixing time of random walk on space of schedules
    - could scale exponentially in number of nodes
  - But, for maximum weight schedule
    - Queue-sizes scale polynomially in n
- Basic question: what are the tradeoffs between
  - Throughput, queue-sizes and complexity of algorithm

Beyond throughput

- Basic question: what are the tradeoffs between
  - Throughput, queue-sizes and complexity of algorithm

- If an algorithm achieves at least 50% throughput, then
  - What is possible
    - Poly queue-size, but Exp complexity – maximum weight
    - Poly complexity, but Exp queue-size – our algorithm
  - What is *not* possible
    - Poly queue-size and poly complexity [**S**-Tse-Tsitsiklis 09]

Beyond throughput

- If an algorithm achieves at least 50% throughput, then
  - What is possible
    - Poly queue-size, but Exp complexity – maximum weight
    - Poly complexity, but Exp queue-size – our algorithm
  - What is *not* possible
    - Poly queue-size and poly complexity [**S**-Tse-Tsitsiklis 09]
- Going forward, is it possible to design
  - Random access for practical networks
    - with Poly queue-size ?
  - Initial attempt [**S**-Shin 10]
    - for network graphs with polynomial growth
    - requires localized co-operation

Beyond throughput

- If an algorithm achieves at least 50% throughput, then
  - What is possible
    - Poly queue-size, but Exp complexity – maximum weight
    - Poly complexity, but Exp queue-size – our algorithm
  - What is *not* possible
    - Poly queue-size and poly complexity [**S**-Tse-Tsitsiklis 09]
- Going forward, is it possible to design
  - Random access for practical networks
    - with Poly queue-size ?
  - Initial attempt [**S**-Shin 10]
  - Random access with interference cancellation
    - and dealing with *hidden terminals*

Related works

- Some of the recent related works
    - Modiano-Shah-Zussman 06
    - Gupta-Stolyar 06, Marbach 06
    - Duvry-Dousse-Thiran 07
    - Bordenave-McDonald-Proutiere 08
    - Jiang-Walrand 08, Rajagopalan-Shah 08
    - Liu-Yi-Proutiere-Chiang-Poor 09
    - Leconte-Ni-Srikant 09
    - Jiang-Shah-Shin-Walrand 09
    - Jiang-Walrand 10
    - Shah-Shin 10
    - van de Ven-van Leeuwaarden-Denteneer-Janssen 10
    - ...