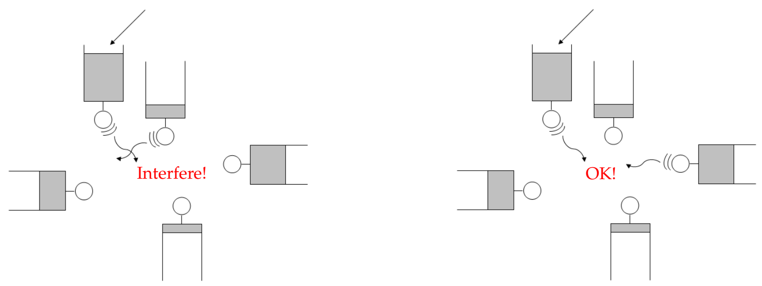
# Delay Optimal Queue-based CSMA

DEVAVRAT SHAH AND JINWOO SHIN†
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139. USA.
†{devavrat,jinwoos}@mit.edu

## Contention Resolution in Wireless Network



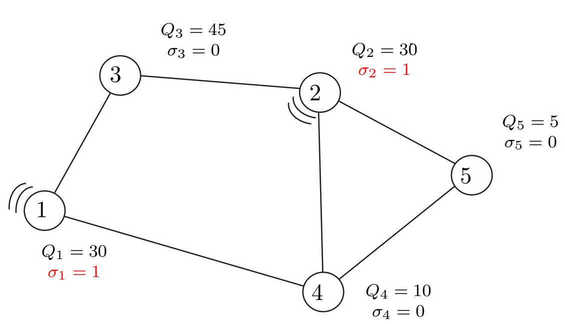

Constraints

- Two simultaneously transmitting queues may interfere with each other.

Goal

- Design a scheduling algorithm : at each time instance
  - Select non-interfering queues to transmit using local information
  - So that queues keep smallest under the largest possible arrival traffic.

## Mathematical Model




- Network interference graph $G = (V, E)$ of $n$ queues.
- Packets arrive with rate $\lambda_i$ for queue $i$.
  - $A_i(t) = \#$ packets arrive at time $t$ and $\mathbb{E}[A_i(t)] = \lambda_i$.
- $\boldsymbol{\sigma}(t) = [\sigma_i(t)] \in \{0,1\}^n$ be the schedule at time $t \in \mathbb{N}$.
  - $\sigma_i(t) = 1$ means the queue $i$ is transmitting at time $t$.
  - $\boldsymbol{\sigma}(t) \in \mathcal{I}(G) := \{\boldsymbol{\sigma} \in \{0,1\}^n : \sigma_i + \sigma_j \leq 1$ for all $(i,j) \in E\}$.

Hence, the queueing evolution becomes $Q_i(t+1) = Q_i(t) + A_i(t) - \sigma_i(t)$.

## MW (Max-Weight) Algorithm

MW algorithm suggests to choose $\boldsymbol{\sigma}(t)$ as MW independent set w.r.t. queues $Q$.

$$\boldsymbol{\sigma}(t) \;=\; \max_{\rho \in \mathcal{I}(G)} \sum_i \rho_i \cdot Q_i(t).$$

**Theorem 1 (Tassiulas and Ephremide 1992)** *The max-weight algorithm guarantees*

$$\limsup_{t\to\infty} \mathbb{E}\left[\sum_i Q_i(t)\right] = O(n^2) \qquad \text{if } \boldsymbol{\lambda} \in conv(\mathcal{I}(G)).$$

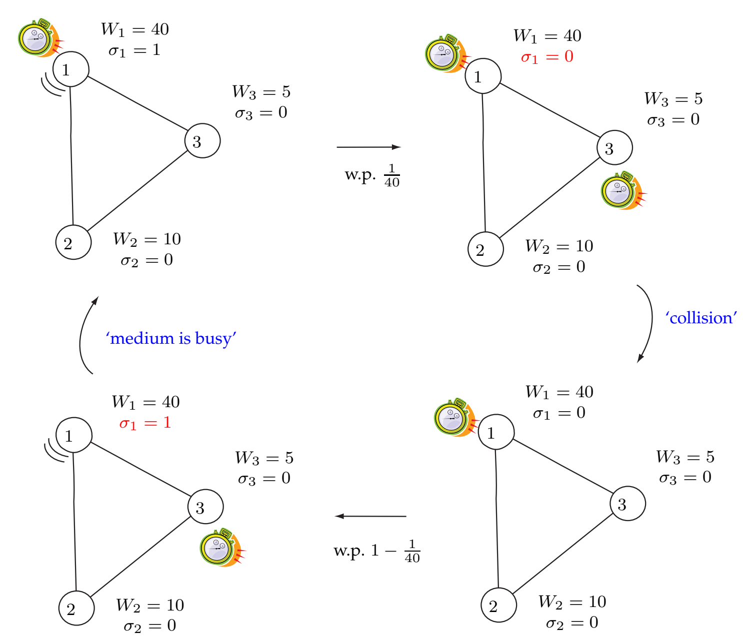
## Our Algorithm

- Each queue has an independent Bernoulli clock of rate 1/2.
- When the clock of the queue $i$ ticks at time $t$,
  - $i$ checks whether any neighbor of $i$ is transmitting at time $t-1$.
  - If no,

$$\sigma_i(t) = \begin{cases} 0 & \text{with probability } \frac{1}{W_i(t)} \\ 1 & \text{otherwise \& no collision} \end{cases},$$

  where $W_i(t)$ will be decided later.

## Example




## Recent Related Works

**Theorem 2 (Shah and Shin 2010)** *The algorithm keeps queues finite if*

$$W_i(t) \text{ is 'essentially' equal to } \log Q_i(t) \text{ and } \boldsymbol{\lambda} \in conv(\mathcal{I}(G)).$$

**Theorem 3 (Shah, Tse and Tsitsiklis 2009)** *If $\boldsymbol{\lambda} > 0$, any (centralized or distributed) poly-complexity algorithm cannot keep queues as poly-size unless NP $\subset$ BPP.*

- *However, there is some hope since G has some structure!*

## Wireless Network has Polynomial Growth

**Definition 1 (Graphs with Polynomial Growth)** *$G$ is a polynomial growth graph if for any node $i$,*

$$\#\text{ nodes within distance } k \;<\; \mathbf{poly}(k).$$

- Wireless interference graph $G$ always have polynomial growth since nodes place in some geographic area, for example $\mathbb{R}^3$.

## Main Result : Choice of $W$ using Graph Partitioning Scheme

**Theorem 4 (Jung and Shah 2008)** *If $G$ has polynomial growth, there exists a random partition $G_1, \ldots, G_k$ of $G$ such that $|G_i| = O(1)$ and*

$$\text{MW Independent-set of } G \;\approx\; \bigcup_i \text{MW Independent-set of } G_i$$

---

We suggest to run the graph partitioning scheme at time $L, 2L, 3L, \ldots$.

$$W_i(t) = \begin{cases} 1 & \text{if } i \text{ is in the boundary of the most recent partition} \\ C \cdot \frac{Q_i(t)}{Q_{\max}(t)} & \text{otherwise} \end{cases},$$

where $L, C > 0$ is some constant.

---

**Theorem 5** *There exists $L, C$ such that the algorithm using $W$ described above guarantees*

$$\limsup_{t\to\infty} \mathbb{E}\left[\sum_i Q_i(t)\right] = O(n) \qquad \text{if } \boldsymbol{\lambda} \in conv(\mathcal{I}(G)).$$

## Proof Intuition

1. If $i$ is in the boundary of partition, $i$ keeps silent.
   - Non-silent nodes inside partitions update their schedules.
2. In time $L$, each partition $G_i$ 'learns' MW independent set of $G_i$
   - Due to our choice of
   $$W_i(t) = C \cdot \frac{Q_i(t)}{Q_{\max}(t)}.$$
   - Learning time $L$ should be exponentially large w.r.t. $|G_i|$.
   - However, $L = O(1)$ since $|G_i| = O(1)$.
3. Hence, the entire network $G$ 'learns' MW independent set of $G$ in time $L$.

$$\text{Delay of Our Algorithm} \;\leq\; \text{Delay of MW Algorithm} \;+\; L.$$

## Discussion and Simulation

- $Q_{\max}(t)$ and the graph partitioning scheme are maintainable in a distributed manner only using minimal (1-bit) message passings per each time.
- In the simulation result, RSS denotes the algorithm using $W_i(t) = \log Q_i(t)$.