Oblivious AQM and Nash Equilibria

Dutta, Goal and Heidmann

In Proceedings of the IEEE Infocom, pages 106-113, San Francisco, California, USA, March 2003. IEEE.

Presented by ZHOU Zhen cszz

COMP 670O

Today's Internet

- There are indications that the amount of non-congestion-reactive traffic is on the rise.
  Most of this misbehaving traffic does not use TCP. e.g. Real-time multi-media, netork games.
- The unresponsive behavior can result in both unfairness and congestion collapse for the Internet.
- The network itself must now participate in controlling its own resource utilization

Active Queue Management

A congestion control protocol (e.g. TCP) operates at the end-points and uses the drops or marks received from the Active Queue Management policies (e.g. Drop-tail, RED) at routers as feedback signals to adaptively modify the sending rate in order to maximize its own goodput.

- Oblivious (stateless) AQM: a router strategy that does not differentiate between packets belonging to different flows.
  Easier to implement
- Stateful schemes: e.g. Fair Queuing

  Gateways maintain separate queues for packets from each individual source. The queues are serviced in a round-robin manner.

Oblivious AQM Scheme – *Drop Tail*

Buffers as many packets as it can and drops the ones it can't buffer

- Distributes buffer space unfairly among traffic flows.
- Can lead to global synchronization as all TCP connections "hold back" simultaneously, hence networks become under-utilized.

Oblivious AQM Scheme – *Random Early Detection*

Monitors the average queue size and drops packets based on statistical probabilities

- If the buffer is almost empty, all incoming packets are accepted; As the queue grows, the probability for dropping an incoming packet grows; When the buffer is full, the probability has reached 1 and all incoming packets are dropped.
- Considered more fair than tail drop – The more a host transmits, the more likely it is that its packets are dropped.
- Prevents global synchronization and achieves lower average buffer occupancies.

Oblivious AQM and Nash Equilibria

The paper studies the existence and quality of Nash equilibria imposed by oblivious AQM schemes on selfish agents:

- Motivation
- Markovian Internet Game Model
- Existence
- Efficiency
- Achievability
- Summary

Game Setting

- Players: n selfish end-point traffic agents.
  Model player i's traffic arrival by Poison process (λ_i).
- Strategy: increase or decrease the average sending rate λ_i.
- Utility: U_i = goodput $\mu_i = \frac{\text{successful rate}}{\text{total rate}}$.
- Rules: oblivious AQM policy with dropping probability p.
  Model the system as M/M/1/K queue.

Poisson arrivals/Exponentially distributed service/one server/finite capacity buffer

Symmetric Nash Equilibrium Condition

- No selfish agent has any incentive to unilaterally deviate from its current state.

$$\forall i, \quad \frac{\partial U_i}{\partial \lambda_i} = 0$$

Symmetric Nash Equilibrium Condition

- No selfish agent has any incentive to unilaterally deviate from its current state.

$$\forall i, \quad \frac{\partial U_i}{\partial \lambda_i} = 0$$

- Every agent has the same goodput at equilibrium.

$$\forall i, j \quad \mu_i = \mu_j \text{ and } \lambda_i = \lambda_j = \frac{\lambda}{n}$$

Symmetric Nash Equilibrium Condition

- No selfish agent has any incentive to unilaterally deviate from its current state.

$$\forall i, \quad \frac{\partial U_i}{\partial \lambda_i} = 0$$

- Every agent has the same goodput at equilibrium.

$$\forall i, j \quad \mu_i = \mu_j \text{ and } \lambda_i = \lambda_j = \frac{\lambda}{n}$$

- Hence functions of router state (drop probability, queue length) are independent in i.

$$\forall i, \quad \frac{\partial}{\partial \lambda_i} = \frac{d}{d\lambda}$$

Symmetric Nash Equilibrium Condition

- No selfish agent has any incentive to unilaterally deviate from its current state.

$$\forall i, \quad \frac{\partial U_i}{\partial \lambda_i} = 0$$

- Every agent has the same goodput at equilibrium.

$$\forall i, j \quad \mu_i = \mu_j \text{ and } \lambda_i = \lambda_j = \frac{\lambda}{n}$$

- Hence functions of router state (drop probability, queue length) are independent in i.

$$\forall i, \quad \frac{\partial}{\partial \lambda_i} = \frac{d}{d\lambda}$$

- Utility fucntion for each player at N.E.

$$U_i = \mu_i = \lambda_i(1-p).$$

Symmetric Nash Equilibrium Condition

- No selfish agent has any incentive to unilaterally deviate from its current state.

$$\forall i, \quad \frac{\partial U_i}{\partial \lambda_i} = 0$$

- Every agent has the same goodput at equilibrium.

$$\forall i, j \quad \mu_i = \mu_j \text{ and } \lambda_i = \lambda_j = \frac{\lambda}{n}$$

- Hence functions of router state (drop probability, queue length) are independent in i.

$$\forall i, \quad \frac{\partial}{\partial \lambda_i} = \frac{d}{d\lambda}$$

- Utility fucntion for each player at N.E.

$$U_i = \mu_i = \lambda_i(1-p).$$

Nash condition: $\frac{dp}{1-p} = \frac{n d\lambda}{\lambda}$

Efficient Nash Equilibrium Condition

- Denote the aggregate throughput $\tilde{\lambda}_n$, goodput $\tilde{\mu}_n$, and drop probability $\tilde{p}_n$ at N.E..
- **Efficient** if the goodput of any selfish agent is *bounded below* when the throughput of the same agent is *bounded above*.

  1. $\tilde{\mu}_n = \tilde{\lambda}_n(1 - \tilde{p}_n) \geq c_1$
  2. $\tilde{\lambda}_n \leq c_2$

  where c_1, c_2 are some constants.
- Therefore, $\tilde{p}_n$ is also bounded.

Outline

- Motivation
- Markovian Internet Game Model
- Existence

  Are there oblivious AQM schemes that impose Nash equilibria on selfish users?
- Efficiency
- Achievability
- Summary

Drop-Tail Queuing

- Drop probability (from queuing theory)

$$p = \frac{\lambda^K(1-\lambda)}{1-\lambda^{K+1}}$$

Theorem 1: There is NO Nash Equilibrium for selfish agents and routes implementing Drop-Tail queuing.

Drop-Tail Queuing

- Drop probability (from queuing theory)

$$p = \frac{\lambda^K(1-\lambda)}{1-\lambda^{K+1}}$$

Theorem 1: There is NO Nash Equilibrium for selfish agents and routes implementing Drop-Tail queuing.

Proof:

$$\mu_i = \lambda_i(1-p) = (\frac{\lambda_i}{\lambda})\lambda(1-p) = (\frac{\lambda_i}{\lambda})\mu$$

$$\frac{\partial \mu_i}{\partial \lambda_i} = \mu \frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) + (\frac{\lambda_i}{\lambda})\frac{d\mu}{d\lambda} > 0$$

Drop-Tail Queuing

- Drop probability (from queuing theory)

$$p = \frac{\lambda^K(1-\lambda)}{1-\lambda^{K+1}}$$

Theorem 1: There is NO Nash Equilibrium for selfish agents and routes implementing Drop-Tail queuing.

Proof:

$$\mu_i = \lambda_i(1-p) = (\frac{\lambda_i}{\lambda})\lambda(1-p) = (\frac{\lambda_i}{\lambda})\mu$$

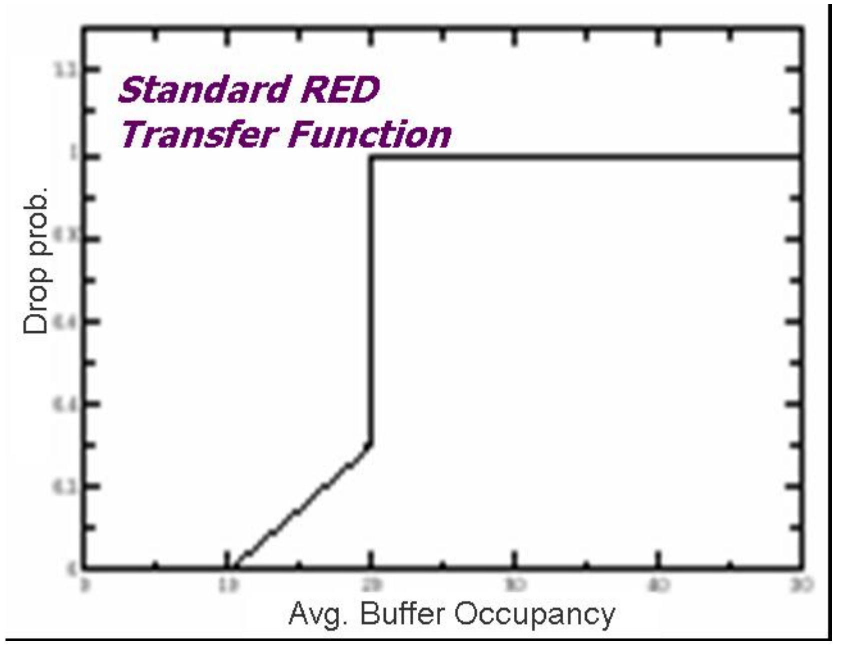
$$\frac{\partial \mu_i}{\partial \lambda_i} = \mu \frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) + (\frac{\lambda_i}{\lambda})\frac{d\mu}{d\lambda} > 0$$

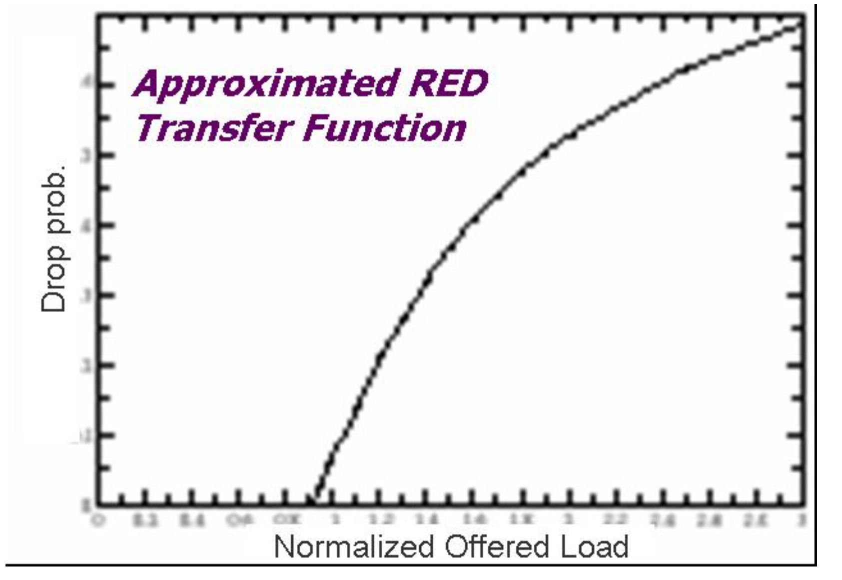
$$\frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) = \frac{\lambda-\lambda_i}{\lambda^2}$$

$$\mu = \frac{\lambda(1-\lambda^K)}{1-\lambda^{K+1}} = 1 - \frac{1}{1+\lambda+\lambda^2+\ldots+\lambda^K}$$

RED

- Drop probability (approximate steady state model [Dutta *et al*])

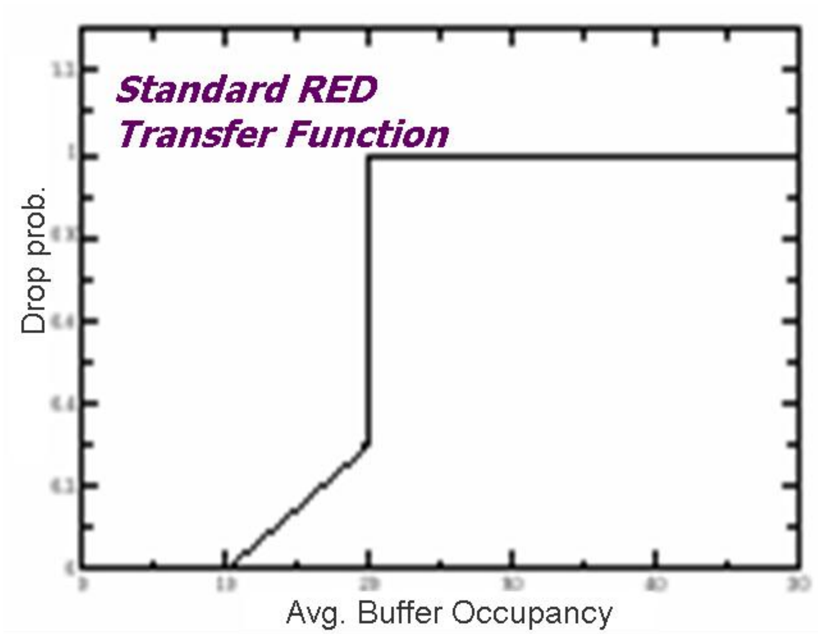


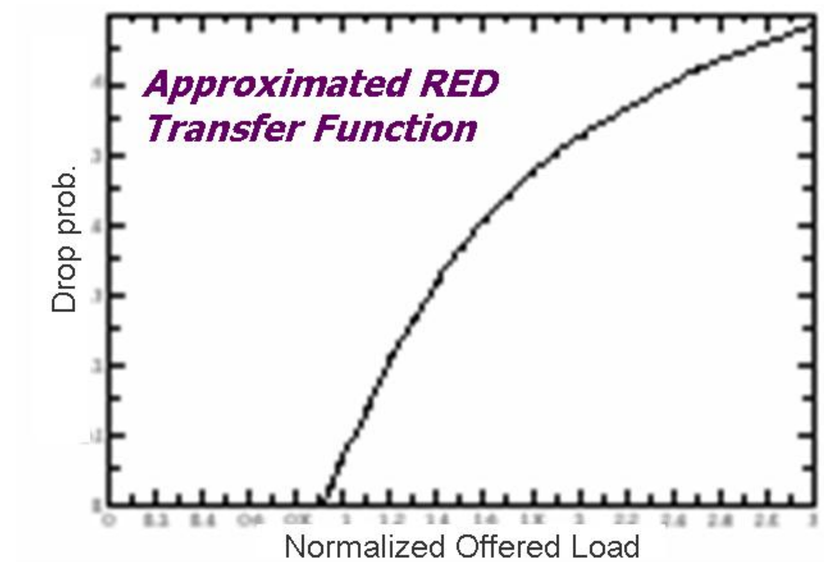


$$p = \begin{cases} 0 & \text{if } l_q < \min_{th} \\ (l_q - \min_{th}) \times \frac{p_{\max}}{\max_{th} - \min_{th}} & \text{if } \min_{th} \le l_q \le \max_{th} \\ 1 & \text{otherwise} \end{cases}$$

RED

- Drop probability (approximate steady state model [Dutta *et al*])





$$p = \begin{cases} 0 & \text{if } l_q < \min_{th} \\ (l_q - \min_{th}) \times \frac{p_{\max}}{\max_{th} - \min_{th}} & \text{if } \min_{th} \le l_q \le \max_{th} \\ 1 & \text{otherwise} \end{cases}$$

- Queue length at steady state (from queuing theory)

$$l_q = \frac{\lambda(1-p)}{1-\lambda(1-p)} \le \max_{th}$$

RED

Theorem 2: RED Does NOT impose a Nash equilibrium on uncontrolled selfish agents.

Proof:

$$\left.\begin{aligned} 1-p &= \left(\frac{l_q}{1+l_q}\right)\left(\frac{1}{\lambda}\right) \\ \mu_i &= \lambda_i(1-p) \end{aligned}\right\} \frac{\partial \mu_i}{\partial \lambda_i} = \frac{l_q}{1+l_q}\frac{\partial}{\partial \lambda_i}\left(\frac{\lambda_i}{\lambda}\right) + \left(\frac{\lambda_i}{\lambda}\right)\frac{\partial \mu}{\partial \lambda}\left(\frac{l_q}{1+l_q}\right) > 0$$

RED

Theorem 2: RED Does NOT impose a Nash equilibrium on uncontrolled selfish agents.

Proof:

$$\left.\begin{array}{r} 1 - p = (\frac{l_q}{1+l_q})(\frac{1}{\lambda}) \\ \mu_i = \lambda_i(1-p) \end{array}\right\} \frac{\partial \mu_i}{\partial \lambda_i} = \frac{l_q}{1+l_q}\frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) + (\frac{\lambda_i}{\lambda})\frac{\partial \mu}{\partial \lambda}(\frac{l_q}{1+l_q}) > 0$$

- RED punishes all flows with the same drop probability.
- Misbehaving flows can push more traffic and get less hurt (marginally).
- There is no incentive for any source to stop pushing packets.

Virtual Load RED

- Drop probability

$$p = \begin{cases} 0 & \text{if } l_{vq} < \min_{th} \\ \frac{l_{vq} - \min_{th}}{\max_{th} - \min_{th}} & \text{if } \min_{th} < l_{vq} < \max_{th} \\ 1 & \text{otherwise} \end{cases}$$

where $l_{vq} = \frac{\lambda}{1-\lambda}$ is the M/M/1 queue length.

Virtual Load RED

- Drop probability

$$p = \begin{cases} 0 & \text{if } l_{vq} < \min_{th} \\ \frac{l_{vq} - \min_{th}}{\max_{th} - \min_{th}} & \text{if } \min_{th} < l_{vq} < \max_{th} \\ 1 & \text{otherwise} \end{cases}$$

where $l_{vq} = \frac{\lambda}{1-\lambda}$ is the M/M/1 queue length.

Theorem 3: VLRED imposes a Nash Equilibrium on selfish agents if $\min_{th} \leq \sqrt{1 + \max_{th}} - 1$.

Proof:

$$\lambda \frac{dp}{d\lambda} = \frac{l_{vq} + l_{vq}^2}{\max_{th} - \min_{th}}$$

By *Nash condition*, $l_{vq}^2 + (n+1)l_{vq} - n \max_{th} = 0$.

$$\tilde{l}_{vq} = \frac{\sqrt{(n+1)^2 + 4n \max_{th}}}{2} - \frac{n+1}{2}$$

The positive root is independent of $\min_{th}$.

Given that $\tilde{l}_{vq} \geq \min_{th}$, we have $\min_{th} \leq \sqrt{1 + \max_{th}} - 1$.

Outline

- Motivation
- Markovian Internet Game Model
- Existence
- Efficiency

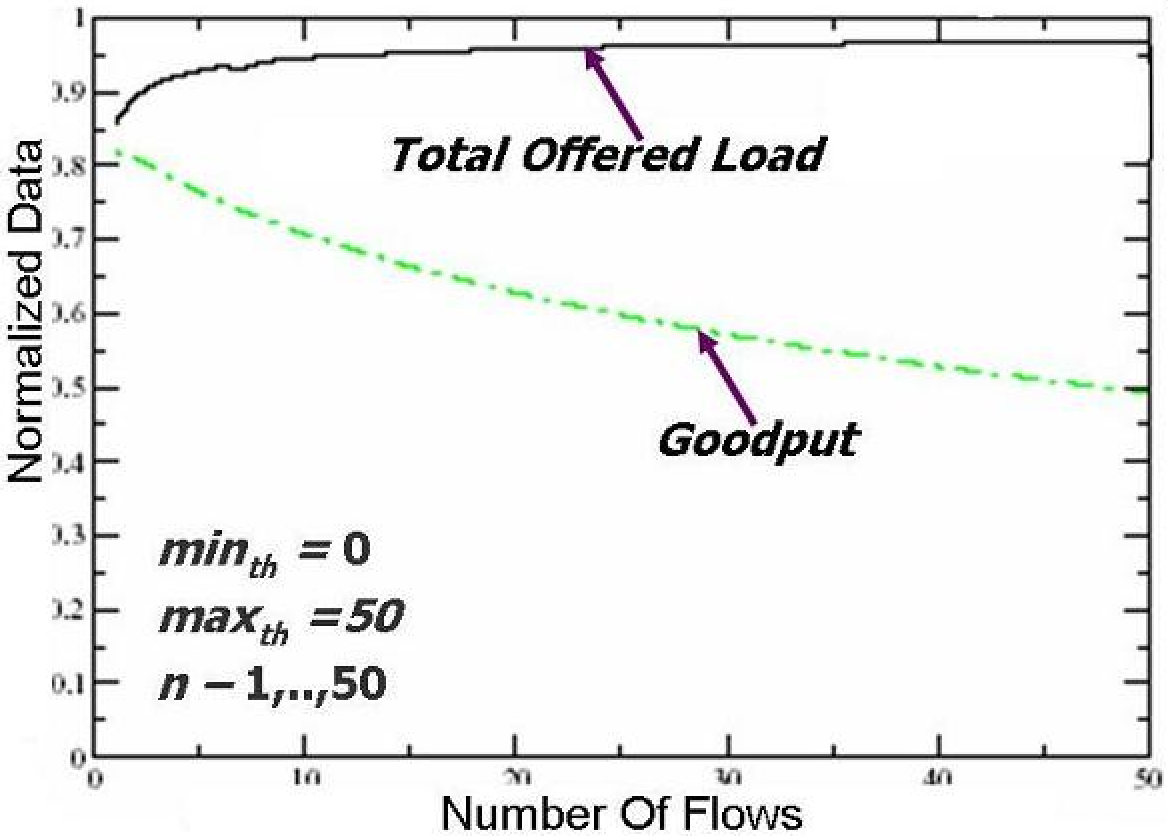
  If an Oblivious AQM scheme can impose a Nash equilibria, is that equilibria efficient, in terms of achieving high goodput and low drop probability.
- Achievability
- Summary

VLRED is not Efficient

- The total throughput is bounded above.

$$\tilde{l}_{vq} = \frac{\tilde{\lambda}_n}{1-\tilde{\lambda}_n}$$

$$\Rightarrow \tilde{\lambda}_n = \frac{\tilde{l}_{vq}}{1+\tilde{l}_{vq}} < 1.$$



VLRED is not Efficient

- The total throughput is bounded above.

$$\tilde{l}_{vq} = \frac{\tilde{\lambda}_n}{1-\tilde{\lambda}_n}$$

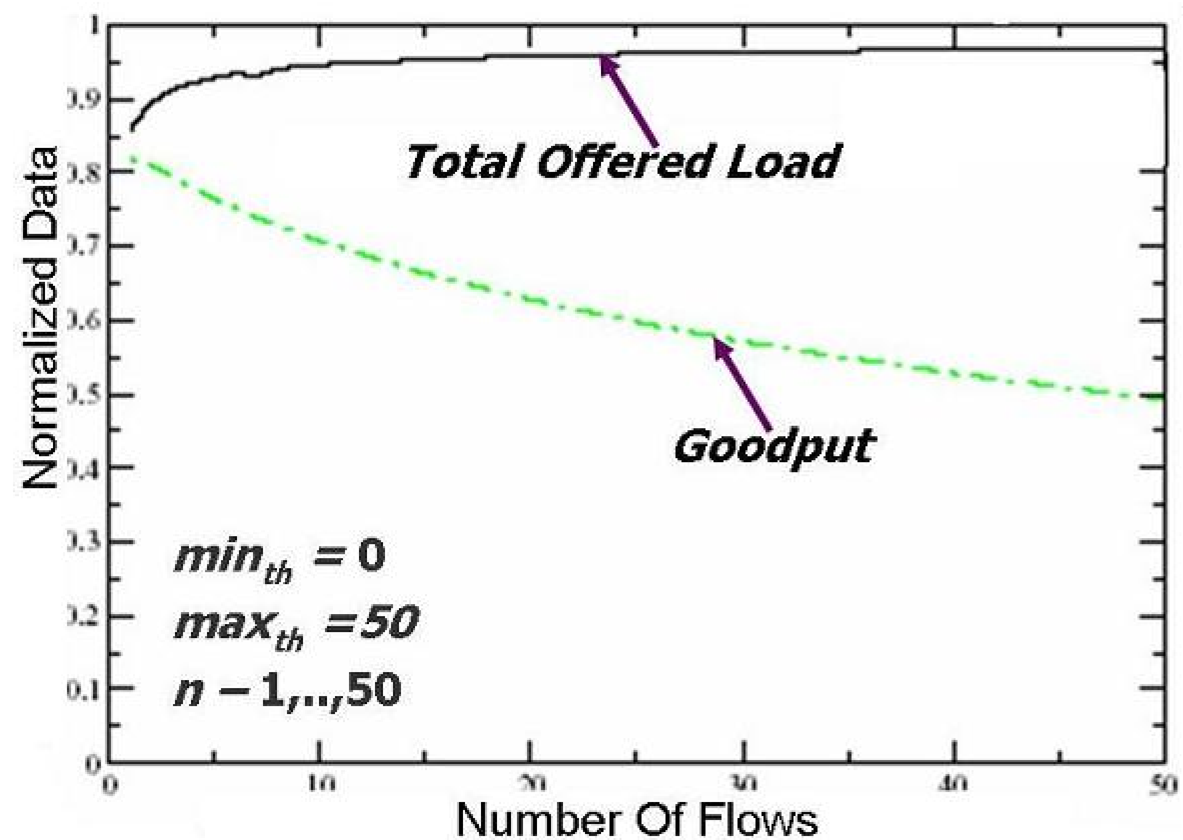
$$\Rightarrow \tilde{\lambda}_n = \frac{\tilde{l}_{vq}}{1+\tilde{l}_{vq}} < 1.$$

- At N.E., $\tilde{l}_{vq}^2 = \alpha n \tilde{\mu}_n$.

  where $\tilde{\mu}_n = \tilde{\lambda}_n(1 - \tilde{p}_n)$, and $\alpha = \max_{th} - \min_{th}$.

- The total goodput falls to 0 asymptotically.

$$\tilde{\mu}_n = \Theta(\tilde{l}_{vq}^2 / n)$$



Efficient Nash AQM

- Assume the total load at N.E. $\tilde{\lambda}_n = 1 - 1/(4n^2)$.
- By *Nash condition*, assuming n continuous

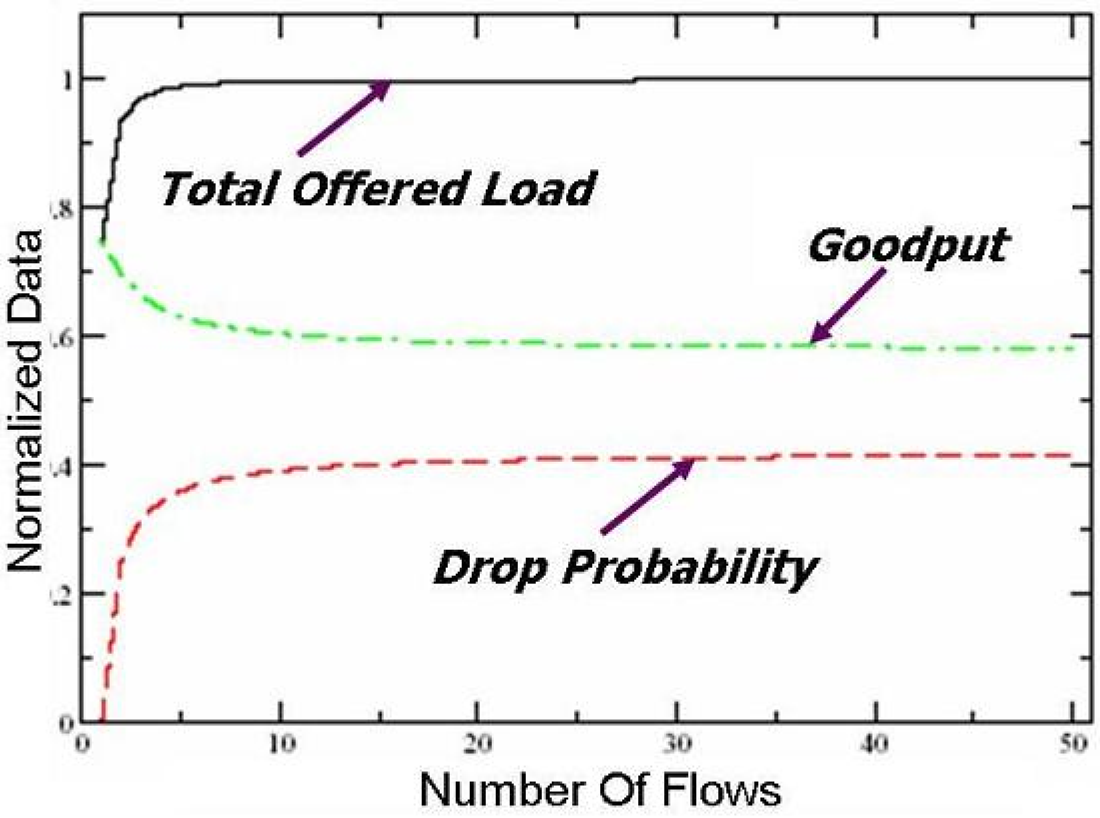
$$\frac{dp}{1-p} = \frac{d\lambda}{2\lambda\sqrt{1-\lambda}} \Rightarrow \tilde{p}_n = 1 - \frac{1}{\sqrt{3}}\sqrt{\frac{1+\sqrt{1-\lambda}}{1-\sqrt{1-\lambda}}}$$

Efficient Nash AQM

- Assume the total load at N.E. $\tilde{\lambda}_n = 1 - 1/(4n^2)$.
- By *Nash condition*, assuming n continuous

$$\frac{dp}{1-p} = \frac{d\lambda}{2\lambda\sqrt{1-\lambda}} \Rightarrow \tilde{p}_n = 1 - \frac{1}{\sqrt{3}}\sqrt{\frac{1+\sqrt{1-\lambda}}{1-\sqrt{1-\lambda}}}$$

- $\tilde{\lambda}_n$ is bounded above, and $\tilde{\mu}_n$ is bounded below.



Outline

- Motivation
- Markovian Internet Game Model
- Existence
- Efficiency
- Achievability

  How easy is it for players (users) to reach the equilibrium point? or How can we ensure that agents actually reach the Nash equilibrium state?
- Summary

Achievability

- $\tilde{\lambda}_i$ – i agents' throughput at N.E.
- $p = f(\tilde{\lambda}_i)$ – drop probability (non-decreasing and convex)
- $\Delta_i = \tilde{\lambda}_i - \tilde{\lambda}_{i-1}$ – sensitivity coefficient

By the *Nash condition* and the efficient condition

Assume $\Delta_i = i^\alpha \quad \Rightarrow \quad \Delta_i = i^{-(2+\epsilon)}$.

The sensitivity coefficient falls faster than the inverse quadric.

The equilibrium imposed by any oblivious AQM strategy is (very) sensitive to the number of agents, thus making it *impractical* to deploy in the Internet.

Outline

- Motivation
- Markovian Internet Game Model
- Existence
- Efficiency
- Achievability
- Summary

Summary

- The Markovian (M/M/1/K) Game
- Existence – Drop tail and RED cannot impose a Nash equilibra. VLRED imposes a Nash equilibra, but the equilibrium points do not have a very high utilization.
- Efficiency – ENAQM imposes an efficient Nash equilibra.
- Achievability – Equilibrium points in oblivious AQM strategies are very sensitive to the change in the number of users.

Summary

- The Markovian (M/M/1/K) Game
- Existence – Drop tail and RED cannot impose a Nash equilibra. VLRED imposes a Nash equilibra, but the equilibrium points do not have a very high utilization.
- Efficiency – ENAQM imposes an efficient Nash equilibra.
- Achievability – Equilibrium points in oblivious AQM strategies are very sensitive to the change in the number of users.

Protocol Equilibrium: A protocol which leads to an efficient utilization and a somewhat fair distribution of network resources (like TCP does), and also ensure that no user can obtain better performance by deviating from the protocol.