Oblivious AQM and Nash Equilibria

Debojyoti Dutta
USC/ISI
4676 Admiralty Way
Marina Del Rey, CA 90292
Email: ddutta@isi.edu

Ashish Goel
Dept. of Computer Science
Univ. of Southern California
941 W. 37th Place
Los Angeles, CA 90089-0781
Email: agoel@usc.edu

John Heidemann
USC/ISI
4676 Admiralty Way
Marina Del Rey, CA 90292
Email: johnh@isi.edu

***Abstract*— An oblivious Active Queue Management scheme is one which does not differentiate between packets belonging to different flows. In this paper, we study the existence and the quality of Nash equilibria imposed by oblivious AQM schemes on selfish agents. Oblivious AQM schemes are of obvious importance because of the ease of implementation and deployment, and Nash equilibrium offers valuable clues into network performance under non-cooperative user behavior. Specifically, we ask the following three questions:**

1) **Do there exist oblivious AQM schemes that impose Nash equilibria on selfish agents?**
2) **Are the imposed equilibria, if they exist, efficient in terms of the goodput obtained and the drop probability experienced at the equilibrium?**
3) **How easy is it for selfish users to reach the Nash equilibrium state?**

We assume that the traffic sources are Poisson but the users can control the average rate. We show that drop-tail and RED do not impose Nash equilibria. We modify RED slightly to obtain an oblivious scheme, VLRED, that imposes a Nash equilibrium, but is not efficient. We then present another AQM policy, EN-AQM, that can impose an efficient Nash equilibrium. Finally, we show that for any oblivious AQM, the Nash equilibrium imposed on selfish agents is highly sensitive as the number of agents increases, thus making it hard for the users to converge to the Nash equilibrium, and motivating the need for equilibria-aware protocols.

I. Introduction

Several transport layer congestion control algorithms and router strategies (Active Queue Management schemes) have evolved over the last fifteen years. Congestion reactive protocols such as TCP [24], [11] and AQM strategies such as RED [7] have catalyzed a lot of interesting research during the last decade. For our purposes, AQM strategies can be classified into two types: *oblivious* (stateless) and stateful. An oblivious AQM scheme does not inspect packets to determine which flow they belong to. Hence it cannot perform differential marking or scheduling for different flows. Stateful schemes such as fair queueing [2] offer good performance on a variety of metrics. On the other hand, oblivious schemes such as drop-tail and RED are easier to implement, and have enjoyed wider deployment.

The current Internet is dominated by TCP traffic [23]. The TCP protocol is well defined, robust, and congestion-reactive [19]. However, there are indications that the amount of non-congestion-reactive traffic is on the rise [8]. Most of this misbehaving traffic does not use TCP. Thus, it seems important to study scenarios where end-points are greedy and selfish, and do not follow socially accepted congestion control mechanisms. TCP (and in fact, any transport protocol that we are aware of) does not guarantee good performance in the face of aggressive, greedy users who are willing to violate the protocol to obtain better performance. It would be quite useful to have protocols which lead 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. We use the term *protocol equilibrium* to describe this phenomenon. If protocol equilibrium is achievable, then it would be a useful tool in designing robust networks.

Of course, one could use stateful schemes such as fair queueing to guard against selfish users. However, in this paper, we would like to explore the limits of what can be achieved using oblivious AQM schemes that are stateless and easily deployable. The existence and practicality of protocol equilibrium (even assuming that the routers adopt oblivious AQM schemes) appear to be very hard questions to answer, and an incremental approach for tackling this problem seems appropriate.

Selfish users can be modeled using tools from game theory [18]. In a game there are rules and players. In the Internet game, the rules are set by the AQM policies and the players are the end-point selfish traffic agents. A fundamental solution concept in game theory is the Nash equilibrium [18]. In the context of our problem, a Nash equilibrium is a scenario where no selfish agent has any incentive to unilaterally deviate from its current state. Thus the existence of a Nash equilibrium implies a stable state of the network in the presence of selfish users, but does not provide any clues as to how this state should be achieved.

It is easy to see that an oblivious AQM strategy cannot lead to a protocol equilibrium unless it also imposes a Nash equilibrium on selfish users. As a first step, in this paper, we explore the existence and the quality of the Nash equilibria that can be imposed on selfish users by oblivious AQM strategies. We will assume that goodput is the performance metric of interest to the selfish users, and each user controls its own offered load. Although our work was motivated by the broader question of protocol equilibria, we believe that our results are interesting in their own right. Oblivious AQM schemes are of obvious importance because of the ease of implementation and deployment, and Nash equilibrium offers valuable clues into network performance under non-cooperative user behavior.

A. Our Contribution

In this paper, we study the Nash equilibria imposed by oblivious AQM schemes on selfish agents which generate Poisson traffic but can control the average rate. Even though Poisson traffic does not accurately model Internet traffic, it is a reasonable first step. We will restrict ourselves to AQM strategies that guarantee bounded average buffer occupancy, regardless of the total arrival rate. Specifically, we address the following questions:

1) **Existence**: Are there oblivious AQM schemes that impose Nash equilibria on selfish users?
2) **Efficiency**: If an oblivious AQM scheme can impose a Nash equilibrium, is that equilibrium good or *efficient*, in terms of achieving high goodput and low drop probability?
3) **Reachability/Achievability**: How easy is it for the users / players to reach the equilibrium point?

We first derive a necessary and sufficient condition, that we call the Nash condition, for oblivious AQM schemes to impose Nash equilibria on selfish users. We show that the popular oblivious AQM strategies, Drop-tail and RED, do not impose Nash equilibria on selfish users. Then we propose a variation of RED, *VLRED*, that can impose a Nash equilibrium. But we note that the utilization at the equilibrium point drops to 0 asymptotically as the number of users increases. This motivates us to develop another AQM scheme (EN-AQM) which can impose a Nash equilibrium and guarantee strict bounds on equilibrium performance, that is, provide lower bounds on goodput, bounded average delays, and upper bounds on drop probability at equilibrium. It is surprising that oblivious schemes can have such strong properties.

We also observe that for any oblivious AQM scheme, the Nash equilibrium imposed on selfish agents is highly sensitive to the increase in the number of users, making it hard to deploy and difficult for users to converge to. This further motivates the need for equilibrium aware protocols.

B. Related Work

Game theory [18] is a very mature topic. The current challenges of game theory applied to computer networks are summarized by Papadimitrou [21]. Several papers (for example [6], [22], [15], [16], [3], [27], [1], [25], [26], [17], [12], [5], [17]) have applied tools from microeconomics and game theory to computer networks over the last fifteen years. A thorough literature survey is beyond the scope of the paper. We now consider some of the most directly relevant related work, and compare them with our approach.

In [3], Douligeris et. al. determine conditions for Nash and Stackelberg equilibrium for a M/M/1 system when the utility function is a composite function of the throughput and the delay. Our work models routers using more realistic models. For example, we use the M/M/1/K queue [14] to model Drop-tail routers. Thus our game is quite different from theirs. Also, they do not model Drop-tail and RED while we do. Besides, we present AQM schemes that impose Nash equilibria on selfish users with bounded average buffer occupancies.

Korilis et. al. [16] model the problem of decentralized control as a *constrained game* in which the strategy of any player is dependent on other players. The model includes a primitive acknowledgment based Markovian traffic arrival with an upper bound on the arrival rate and service time. They show that Nash equilibria can exist in a general product-form network under the condition guaranteeing that no one user can be forced out of the system by other users. Unfortunately, that assumption may not be valid at all times in the Internet. This is true especially with routers having limited buffers. We, on the other hand, look at equilibria imposed by AQM schemes at routers using realistic models with bounded average buffer occupancies. Besides, we look at oblivious AQM schemes.

Shenker [27] takes another approach and views the Internet game from a switch scheduling perspective. The authors prove that with Markovian arrival rates, the *fair share* allocation scheme is the only scheme within a class of buffer allocation functions (called MAC) that can guarantee a Nash equilibrium (on selfish agents), and, is also Pareto efficient. The authors use a simple M/M/1 model for their switch service discipline and they study equilibria imposed by the buffer allocation algorithm. We show, in the appendix, that oblivious AQM schemes with bounded average buffer occupancy cannot be in MAC. Hence, the MAC framework is not applicable to our problem. Also, they do not consider existence of Nash equilibria for newer schemes like RED as we do.

Several other papers have investigated related problems. For example, Park et. al. [22] study Nash equilibrium properties of the QoS game. However their utility functions are much different from ours due to their multilevel service model and threshold based step functions for modeling utility. They claim that for a single service level, their model boils down to that in [27]. Gibbens et. al [9] have studied the effect of selfish users in the context of the user optimization problem defined in [13]. Their model assumes that the selfishness arises due to the user's disregard for the effect of its own action on prices. They model the Internet game differently using a AIMD like protocol with the congestion marks from the routers. Thus, our assumptions and our problem is different from theirs. Note that we have not looked at feedback signals from the network and pricing issues.

In a recent paper Akkela et. al. [1] model greedy users with a TCP like algorithm where selfish users can control the traffic by varying the parameters (α, β) of the AIMD congestion control algorithm. They show that RED does not have a Nash equilibria using empirical models and through simulation. Also they have a restricted notion of selfish traffic. They also propose a variation of CHOKe [20] and show, by simulation, that a *good* Nash equilibrium is reached by TCP like greedy flows. Unlike their work, this paper makes formal arguments to prove results for a more general class of selfish users. Also, unlike them, we consider *oblivious AQM techniques*. We are not aware of any other work that formally analyzes the Nash equilibrium imposed on selfish users by oblivious AQM strategies.

The format of this paper is as follows. Section II defines the Internet game. In Section III, we show that the current mechanisms present in the Internet, Drop-tail and RED, cannot impose Nash equilibria on selfish agents. Then, we propose a new variant of RED that achieves Nash equilibrium in Section III-C. We note that this proposed scheme is not efficient. In Section IV, we present a simple router mechanism that imposes a Nash equilibrium and, at the same time, bounds the network performance at equilibrium. Then we discuss the sensitivity of the Nash equilibrium imposed by oblivious AQM schemes in Section VI and argue that selfish users would find it hard to reach the equilibrium state. Finally we discuss our future directions and conclude in Section VI.

II. The Markovian Internet Game: Preliminaries

The Markovian Internet game can be defined as follows. The players are end-point traffic agents. These agents are selfish, that is, they are only concerned about their own good. Each player has a strategy which is to control the average rate of traffic that the player tries to push through the network. We model the players' traffic arrival rate by a Poisson process with an average rate of λ_i. Each player i has a simple utility function U_i equal to its goodput μ_i. The rules of the game are determined by the AQM schemes in routers. In this work, we only consider oblivious AQM schemes. An oblivious router has a drop probability p due to an average aggregate load of λ and an average service time of unity. Now, oblivious routers may or may not impose *symmetric Nash equilibria* on selfish agents. A symmetric Nash equilibrium ensures that every agent has the same goodput at equilibrium. We only consider *symmetric Nash equilibrium* in this paper and we drop the symmetric adjective throughout the paper. Also, unless mentioned otherwise, quantities such as the rates, goodput and throughput are averages as we assume Poisson traffic sources.

For Nash equilibrium to hold, we have the following conditions:

- No agent can increase their goodput, at Nash equilibrium, by either increasing or decreasing their throughput. This can be written down as
$$\forall\ i,\ \frac{\partial U_i}{\partial \lambda_i} = 0. \tag{1}$$
Since $U_i = \mu_i$, $\frac{\partial \mu_i}{\partial \lambda_i} \leq 0.$ for all i.
- At Nash equilibrium, all flows have the same utility or goodput. That is, $\forall i, j\ \{\mu_i = \mu_j \text{ and } \lambda_i = \lambda_j\}$.
- For oblivious AQM strategies and functions of router states like drop probability and queue length,
$$\forall i,\ \frac{\partial}{\partial \lambda_i} = \frac{d}{d\lambda}$$
.

The above conditions can be used to derive an interesting condition that must be true at Nash equilibrium. The utility function for each agent can be written down as

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

Taking partial derivatives we get

$$\frac{\partial \mu_i}{\partial \lambda_i} \ = \ 1 - p - \lambda_i \frac{\partial p}{\partial \lambda_i} \ = \ 0.$$

Since we consider only oblivious AQM schemes, we have

$$\frac{\partial p}{\partial \lambda_i} = \frac{dp}{d\lambda}$$

. Since we consider symmetric Nash equilibrium, we must also have.

$$\lambda_i \ = \ \frac{\lambda}{n}.$$

Thus, we have the following **Nash condition** which must be satisfied at Nash equilibrium:

$$\frac{dp}{1-p} \ = \ \frac{nd\lambda}{\lambda}. \tag{2}$$

To evaluate whether the Nash equilibrium imposed by an AQM scheme is good, let us define a term, *efficiency*. Let the the aggregate throughput, goodput, drop probability be denoted by $\tilde{\lambda}_N$, $\tilde{\mu}_N$ and $\tilde{p}_N$ respectively. The Nash equilibria imposed by an AQM is efficient, if the goodput of any selfish agent is bounded below when the throughput (offered load) of that same agent is bounded above. The conditions for efficiency are:

1) $\tilde{\lambda}_N(1-\tilde{p}_N) \geq c_1$
2) $\tilde{\lambda}_N \leq c_2$

where c_1, c_2 are some constants. Thus is easy to see that even the drop probability at equilibrium is also bounded.

TCP Traffic and Equilibria

Before we delve into the technical details, it is interesting to see how TCP traffic fits into the picture. TCP traffic is the dominant traffic in the Internet. However, TCP does not exhibit selfish behavior unless one changes the underlying protocol. It is known ([10]) that there exists a unique equilibrium point in a network with long lived TCP flows resulting in a steady state. Akkela et.al [1] model selfish traffic that behaves like TCP on loss indications, uses an AIMD protocol, and allows the level of selfishness to be controlled by the two AIMD parameters (α, β) [1]. We believe that their model covers only a subset of selfish users. In our work, we do not make any assumption of the nature of the selfish protocol. In the following subsections, we will consider the interaction of routers and greedy Poisson traffic with variable rates.

III. Existence

In this section, we answer the *existence question* as outlined in Section I-A, that is, whether there exist oblivious router strategies that impose Nash equilibria on selfish agents. First, we consider the existence of equilibria imposed by popular AQM schemes like Drop-tail and RED. Then, we propose a simple scheme of our own. We shall use simple router models with the help of queueing theory [14]. Recall that, for the sake of simplicity, we assume that the average service time of our queueing systems is unity.

A. Drop-tail Queueing

In this section, we consider Drop-tail routers prevalent in the Internet and we model them with a M/M/1/K queue. From queueing theory, we know that the drop probability of a Drop-tail router with a buffer size B and offered load λ is given by

$$p \;=\; \frac{\lambda^B(1-\lambda)}{1-\lambda^{B+1}}.$$

Then we have the following theorem, the proof of which is simple and intuitive, and, is presented here for completeness:

Theorem 1: For selfish agents and routers implementing Drop-tail queuing, there is no Nash equilibrium.

Proof: Consider a single link with a maximum buffer size of B and several selfish flows traversing this link. Let λ_i, μ_i be the throughput, goodput of each flow respectively. Assume p to be the drop probability of the router. Then, μ_i is given by

$$\mu_i \;=\; \lambda_i(1-p) \;=\; \left(\frac{\lambda_i}{\lambda}\right)\lambda(1-p) \;=\; \left(\frac{\lambda_i}{\lambda}\right)\mu.$$

Applying the condition for Nash equilibrium, and noting that Drop-tail is oblivious, we have

$$\frac{\partial \mu_i}{\partial \lambda_i} \;=\; \mu\frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) + \left(\frac{\lambda_i}{\lambda}\right)\frac{d\mu}{d\lambda}.$$

The first term in the above equation is positive as

$$\frac{\partial}{\partial \lambda_i}\left(\frac{\lambda_i}{\lambda}\right) \;=\; \frac{\lambda-\lambda_i}{\lambda^2}.$$

The proof of the second term being positive is shown in Section B of the appendix. Thus, $\frac{\partial \mu_i}{\partial \lambda_i}$ is positive for all $\lambda_i > 0$, and there can be no Nash equilibrium. ■

B. RED

RED [7] (Random Early Detection) is an AQM scheme that prevents global synchronization and achieves lower average buffer occupancies. To analyze whether this scheme can impose a Nash equilibrium using our approach, we need an analytical model of RED that yields the drop probability as a function of the offered load. Thus, we borrow an approximate steady state model of RED from [4] for our analysis. The standard RED transfer function and that along with our transformation is shown in Figure 1.

Let the average queue length at the router be l_q and the drop probability at the router be p. The RED characteristics can be expressed by

$$p \;=\; \begin{cases} 0 & \text{if } l_q < \min_{th} \\ (l_q - \min_{th}) \times \frac{p_{\max}}{\max_{th} - \min_{th}} & \text{if } \min_{th} \le l_q \le \max_{th} \\ 1 & \text{otherwise.} \end{cases} \qquad (3)$$

As long as the queue length is below $\min_{th}$, the drop probability is zero. If queue length is between $\min_{th}$ and $\max_{th}$, drop probability increases linearly between 0 and $p_{\max}$. After this limit is crossed, drop probability becomes unity.

As defined earlier, let λ be the aggregate offered load, and let p be the drop probability. The RED router sees a thinned arrival process with a rate of $\lambda(1-p)$. Thus, we have the following lemma:

Lemma 1: The steady state average value of the queue-length with utilization λ and drop probability p is given by

$$l_q \;=\; \frac{\lambda(1-p)}{1-\lambda(1-p)}. \qquad (4)$$

In the steady state, the average queue length of a RED router is never larger than $\max_{th}$. Thus we have:

Lemma 2: In the steady state, the average queue length of a RED router is never larger than $\max_{th}$

If the average drop probability seen at this router is p and the steady state queue length is between the interval $\min_{th}$ and $\max_{th}$, the drop probability p at this router is given by

$$p \;=\; \frac{(\frac{\lambda(1-p)}{1-\lambda(1-p)} - \min_{th}) \times p_{\max}}{\max_{th} - \min_{th}}.$$

With the above model, the following theorem is easy to prove:

Theorem 2: RED does not impose a Nash equilibrium on uncontrolled selfish agents.

Proof: From Equation 4, we solve for $1-p$ and get

$$1-p \;=\; (\frac{l_q}{1+l_q})(\frac{1}{\lambda}).$$

Now the goodput, μ_i, in terms of the offered load λ_i as

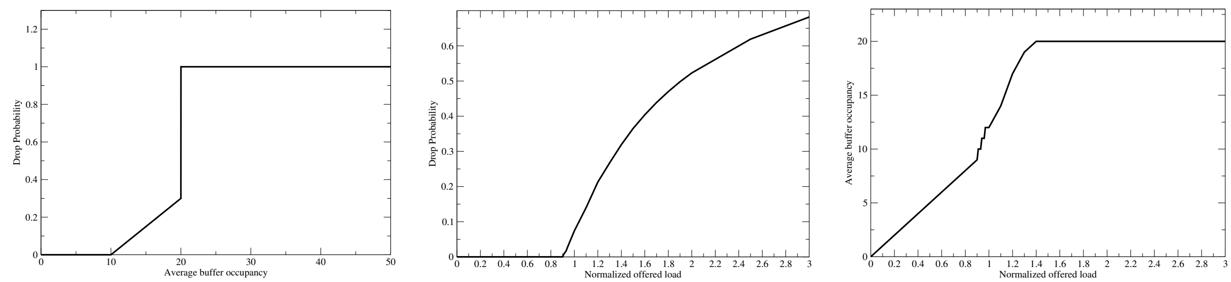
$$\mu_i \;=\; \lambda_i(1-p) \;=\; (\frac{\lambda_i}{\lambda})(\frac{l_q}{1+l_q}).$$

Now taking partial derivatives, we have

$$\frac{\partial \mu_i}{\partial \lambda_i} \;=\; (\frac{l_q}{1+l_q})\frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) + (\frac{\lambda_i}{\lambda})\frac{\partial}{\partial \lambda_i}(\frac{l_q}{1+l_q}).$$

Now, $\frac{\partial}{\partial \lambda_i}(\frac{\lambda_i}{\lambda}) > 0$. Since RED is oblivious, the second term in the above equation is $(\frac{\lambda_i}{\lambda})\frac{1}{(1+l_q)^2}\frac{dl_q}{d\lambda}$. It is easy to prove that $\frac{dl_q}{d\lambda} \ge 0$. The proof is very simple, and has been omitted. Thus $\frac{\partial \mu_i}{\partial \lambda_i} > 0$. ■

The intuition behind the above proof is as follows. RED punishes all flows with the same drop probability. The nature of the drop function is considerably gentle. Thus, misbehaving flows can push more traffic and get less hurt (marginally). Hence there is no incentive for any source to stop pushing packets. This implies the non-existence of an equilibrium.



Standard RED transfer function | Transfer function from our model | Buffer occupancy from our model

Fig. 1. The different models of RED. In this example, the maximum buffer size = 50, $\min_{th}$ = 10, $\max_{th}$ = 20 and $p_{\max}$ = 0.3

C. Virtual Load RED (VLRED)

In the previous sub-sections, we saw that drop-tail and RED do not impose Nash equilibria on selfish users. In this section, we answer the question of the existence of Nash equilibria as defined in Section I-A. We show that Nash equilibrium can be imposed by changing the RED transfer function. In this section, we present a new AQM technique, *Virtual Load RED (VLRED)*, where the transfer function for the drop probability can be written down as

$$p = \begin{cases} 0 & l_{vq} < \min_{th} \\ \frac{l_{vq} - \min_{th}}{\max_{th} - \min_{th}} & \min_{th} < l_{vq} < \max_{th} \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

Here $l_{vq} = \frac{\lambda}{1-\lambda}$. Note that l_{vq} is the length of the infinite buffer *M/M/1* queue when faced with the same load. Instead of using the measured queue length, as in RED, we have used virtual queue lengths, hence the name VLRED. Now, we ask whether this AQM strategy has a Nash equilibrium.

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

Proof: Suppose there are n flows with each source i offering a load of λ_i. Let us assume that the drop probability can be written down as a continuous function for all $l_{vq} < \max_{th}$ given by

$$p = \frac{\frac{\lambda}{1-\lambda} - \min_{th}}{\max_{th} - \min_{th}}.$$

Then we have

$$\lambda \frac{dp}{d\lambda} = \left(\frac{1}{\max_{th} - \min_{th}} \right) \left(\frac{\lambda}{1-\lambda} + \left(\frac{\lambda}{1-\lambda} \right)^2 \right).$$

Now using the Nash condition in Equation 2, we have

$$\left(\frac{1}{\max_{th} - \min_{th}} \right) \left(\frac{\lambda}{1-\lambda} + \left(\frac{\lambda}{1-\lambda} \right)^2 \right) = n \left(1 - \frac{\frac{\lambda}{1-\lambda} - \min_{th}}{\max_{th} - \min_{th}} \right).$$

$$\Rightarrow \frac{\lambda}{1-\lambda} + \left(\frac{\lambda}{1-\lambda} \right)^2 = n \left((\max_{th} - \min_{th}) - \left(\frac{\lambda}{1-\lambda} - \min_{th} \right) \right)$$

Substituting $t = \frac{\lambda}{1-\lambda}$ and simplifying, we have

$$t^2 + (n+1)t - n.\max_{th} = 0 \quad (6)$$

The positive root of the the above equation is

$$t = \frac{\sqrt{(n+1)^2 + 4n.\max_{th}}}{2} - \frac{n+1}{2}$$

For this solution to be valid, $t \geq \min_{th}$. It is easy to show that this is true if $\min_{th} \leq \sqrt{1 + \max_t h} - 1$. The proof is very simple and has been omitted. Thus, the existence of the solution under the given condition proves that the Nash equilibrium exists. Note that we can get the throughput at the Nash equilibrium by $\tilde{\lambda}_n = \frac{t}{1+t} < 1$. A surprising feature of our solution is that the root, when valid, is independent of $\min_{th}$. ■

D. Discussion

We now illustrate the properties of this Nash equilibrium with an example. We take $\min_{th} = 0$, $\max_{th} = 50$, $n = 1, .., 50$ and solve Equation 6. As n increases, the goodput of the system starts to decrease due to increasing drop probability due to the average buffer occupancy approaching $\max_{th}$ and the l_{vq} rising very steeply. This effect is shown in Figure 2 where the y-axis plots both the throughput and the goodput as we increase the number of flows, shown in the x-axis. Also, we can formally prove the following:

Theorem 4: VLRED is not efficient.

Proof: Assuming a unit service rate at a router, the drop probability p can be written in terms of the total arrival rate λ as

$$p = \frac{\alpha \lambda}{1-\lambda} + \beta.$$

where α and β are some constants. Differentiating the above equation we have

$$\frac{dp}{d\lambda} = \frac{\alpha}{(1-\lambda)^2}.$$

Now rearranging the terms after using Equation 2 and equating with the above equation, we have

$$\left(\frac{\lambda}{1-\lambda} \right)^2 = \frac{n\lambda(1-p)}{\alpha}.$$

Now, $\frac{\lambda}{(1-\lambda)} = l_{vq}$ and $\lambda(1-p)$ is the total goodput μ. Assume that VLRED is efficient. Then, by the definition of efficiency, the goodput is bounded above and below by constants, and,

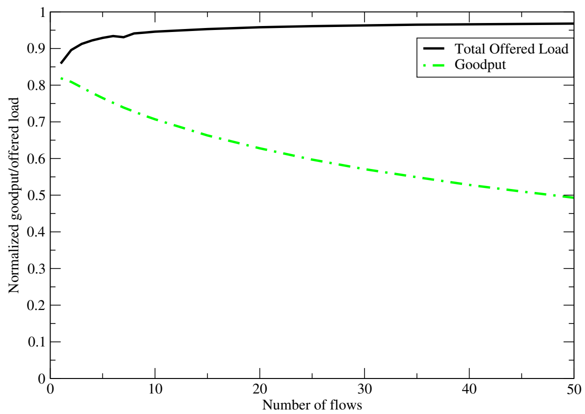


Fig. 2. The throughput and the goodput at a VLRED router as the number of flows changes

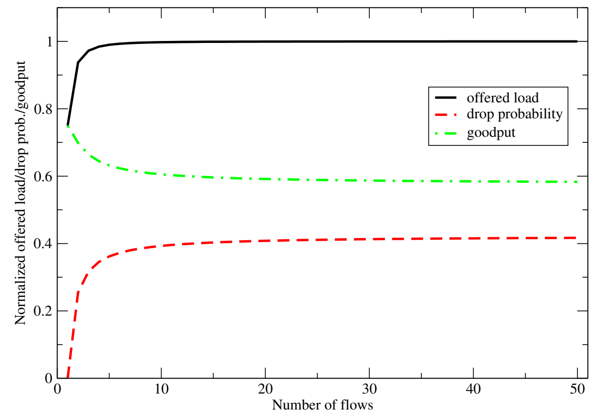


Fig. 3. The offered load (throughput), drop probability and the goodput at the Nash equilibrium imposed by EN-AQM as the number of flows passing through our router mechanism changes

hence, $n\mu = \theta(l_{vq}^2)$. Now, in VLRED, l_{vq} is bounded by $\frac{\max_{th}}{1+\max_{th}}$. Thus, $n\mu$ is bounded by a constant. Hence, the goodput falls to 0 asymptotically. ■

Thus, even though VLRED has a Nash equilibrium, it is clear that the equilibrium points do not have very high utilization. Hence, there is a need to design oblivious AQM strategies that result in efficient equilibria.

IV. Efficiency

In the previous section, we have seen that even though VLRED can impose a Nash equilibria on selfish users, it is not efficient. In this section, we ask whether there is a router strategy that can impose an efficient Nash equilibrium. This is the second question as defined in Section I-A. We present an oblivious AQM strategy, namely *Efficient Nash AQM (EN-AQM)*, which ensures the existence of an efficient Nash equilibrium.

Consider a single bottleneck link. From the previous sections, we have seen that in order for a router queue management technique to enforce a Nash equilibrium Equation 2 must be satisfied. Now, assume that the total desirable offered load at Nash equilibrium is related to the number of greedy agents in the following way

$$\tilde{\lambda}_n = 1 - \frac{1}{4n^2}. \quad (7)$$

The intuition behind this relation is presented in Section V. Using the above equation, and the Nash condition, after allowing n to be continuous for the time being, we can write down

$$\frac{dp}{1-p} = \frac{d\lambda}{2\lambda\sqrt{1-\lambda}}.$$

Now, substituting $y^2 = 1 - \lambda$, we have

$$\frac{-dp}{1-p} = \frac{1}{2}[\frac{dy}{1+y} - \frac{-dy}{1-y}].$$

Integrating the above equation, we have

$$\log 1 - p = \log\sqrt{\frac{1+y}{1-y}} + \log k.$$

where k is an arbitrary constant to be determined. The solution can be written down as

$$p = 1 - k\sqrt{\frac{1+y}{1-y}}.$$

To choose k, we assume that when there is one user, we would like to ensure that the drop probability is zero as long as his offered load is less than unity. This gives $k = \frac{1}{\sqrt{3}}$. Thus the transfer function of EN-AQM is given by

$$p = 1 - \frac{1}{\sqrt{3}}\sqrt{\frac{1+\sqrt{1-\lambda}}{1-\sqrt{1-\lambda}}}. \quad (8)$$

By our assumption, $\tilde{\lambda}_n$ is bounded. It is simple, albeit tedious, to prove that $\tilde{p}_n$ is also bounded. We omit the proof, and, instead, use Figure 3 to illustrate the efficiency of EN-AQM.

To see how EN-AQM works, let us look at Figure 3. We see that even though the number of flows through the router increases, the drop probability and the goodput at the Nash equilibrium are bounded. This illustrates that EN-AQM can ensure an efficient Nash equilibrium. Hence, with very minimal information, we can ensure that our oblivious router mechanism achieves the objective of ensuring a Nash equilibrium that yields good performance under selfish behavior. The constants which the drop probability and the throughput depend on can be fine tuned by modifying the constant k and the constant 4 in Equation 7.

V. Achievability

In the previous section, we have seen that EN-AQM can impose an efficient Nash equilibrium on selfish agents with bounds on performance and drop probability. Now we move on to answer the third question in Section I-A: How can we ensure that agents actually reach the Nash equilibrium state? In this section, we show that the equilibrium imposed by any oblivious router strategy is very sensitive to the number of agents, thus making it impractical to deploy in the Internet.

Let us define $\tilde{\lambda}_i$ to be the offered load at equilibrium when the number of agents is i. We can now rewrite the drop probability p, as a function of the aggregate offered load $\tilde{\lambda}_i$, as $p = f(\tilde{\lambda}_i)$. Let us also assume that f is non-decreasing and convex. From Equation 2 and the fact that an efficient AQM will bound the drop probability, we get

$$\alpha_1 \frac{i}{\tilde{\lambda}_i} \leq f'(\tilde{\lambda}_i) \leq \alpha_2 \frac{i}{\tilde{\lambda}_i}.$$

where p_1, p_2 are some constants. Let us also define a *sensitivity coefficient* Δ_i, when the number of agents in the system is i, such that

$$\Delta_i \ = \ \tilde{\lambda}_i - \tilde{\lambda}_{i-1}.$$

Now, the following must be true due to the convexity of $\tilde{p}_i$

$$\tilde{p}_i \ \geq \ \tilde{p}_1 + \Delta_1 f'(\tilde{\lambda}_1) + \Delta_2 f'(\tilde{\lambda}_2) + ... + \Delta_{i-1} f'(\tilde{\lambda}_{i-1}).$$

By efficiency, we get the following equation

$$\sum_{i=1}^{\infty} i\Delta_i \ \leq \ c \text{ where } c \text{ is a constant.}$$

Consider $\Delta_i = i^{\alpha}$. Then, using the above equation, we have

$$\sum_{i=1}^{\infty} i^{\alpha+1} \leq c.$$

Now the left hand side is bounded iff $\alpha + 1 < -1$, or $\alpha < -2$. This gives us $\Delta_i = i^{-(2+\epsilon)}$. Hence, we can see that as i increases, the sensitivity coefficient becomes smaller and smaller, that is, it falls faster than an inverse quadratic. Thus, it is hard for agents to reach the equilibrium point without any help from the router.

Note that the above analysis points out the relation between the number of users and the total offered load in Equation 2. We see that relation must have a term in $\frac{1}{n^{1+\epsilon}}$. Choosing $\epsilon = 1$, and setting some realistic boundary conditions, we get the strategy EN-AQM; this is the intuition behind Equation 7 in Section IV.

VI. Discussions, Conclusion and Future Work

In this paper, we have investigated the following three questions pertaining to Nash equilibria imposed by oblivious AQM techniques on selfish users: existence, efficiency and achievability. We have shown that Drop-tail and RED, cannot impose a Nash equilibrium. We have also shown there are simple mechanisms, such as VLRED, that do impose Nash equilibria. Then we showed that there are AQM schemes, such as EN-AQM, that can impose efficient Nash equilibria. Finally, we showed that the equilibrium points in oblivious AQM strategies are very sensitive to the change in the number of users. Thus it may be hard to deploy oblivious schemes that do have Nash equilibria without the explicit help of a protocol.

Previous work [27] has shown that only the *fair share* mechanism can have Pareto efficient Nash equilibria. Also, Douligeris et. al [3] show that it is possible to to reach Nash equilibria with the power utility function. They assume a M/M/1 system with infinite buffer capacity and they use a very different composite utility function involving the goodput and the delay. In contrast, we have considered more realistic models of router mechanisms with bounded buffers for Drop-tail queueing and we also analyze RED. We have noticed that the results are significantly different due to our requirement of bounded average buffer occupancies.

We need to explore each of the above proposed policies in detail. For VLRED, we need to explore why the Nash equilibrium point does not result in good network utilization. Our conjecture is that VLRED's drop function becomes very harsh as we reach equilibrium. Thus, there is a need to study gentler versions of VLRED and determine whether such modifications can still impose Nash equilibria.

We should note that the it is difficult for users to reach the equilibrium point as the equilibrium point is too sensitive. This motivates the question of **Protocol Equilibrium**: *Can we design protocols which lead to efficient network operation, such that no user has any incentive to unilaterally deviate from the protocol?*

References

[1] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, and S. Shenker, "Selfish behavior and stability of the Internet: A game-theoretic analysis of TCP," *ACM SIGCOMM*, 2002.

[2] A. Demers, S. Keshav, and S.J. Shenker, "Analysis and simulation of a fair queueing algorithm," *ACM SIGCOMM*, 1989.

[3] C. Douligeris and R. Mazumdar, "On Pareto optimal flow control in an integrated environment," *Proc. of the 25th Allerton Conference on Communication, Control and Computing*, Oct 1987.

[4] D. Dutta, A. Goel, and J. Heidemann, "Faster network simulation with scenario pre-filtering," Tech. Rep., USC/ISI Tech Report 550, November 2001.

[5] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker, "Sharing the cost of multicast transmissions," *Journal of Computer and System Sciences*, vol. 63, no. 1, pp. 21–41, 2001.

[6] D. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini, "Economic models for allocating resources in computer systems," *In Scott Clearwater, editor, Market-Based Control: A Paradigm for Distributed Resource Allocation, Scott Clearwater. World Scientific, Hong Kong, 1996.*, 1996.

[7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance.," *IEEE/ACM Transactions on Networking, V.1 N.4*, 1993.

[8] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.

[9] R. Gibbens and F. Kelly, "Resource pricing and the evolution of congestion control," 1998.

[10] R.J. Gibbens, S.K. Sargood, C. Van Eijl, F.P. Kelly, H. Azmoodeh, R.N. Macfadyen, and N.W. Macfadyen, "Fixed-point models for the end-to-end performance analysis of IP networks," in *Proceedings of the 13th ITC Specialist Seminar: IP Traffic Measurement, Modeling and Management, Sept 2000, Monterey, California*, 2000.

[11] V. Jacobson, "Congestion avoidance and control," *ACM Computer Communication Review; Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988*, vol. 18, 4, pp. 314–329, 1988.

[12] K. Jain and V. V. Vazirani, "Applications of approximation algorithms to cooperative games," in *ACM Symposium on Theory of Computing*, 2001, pp. 364–372.

[13] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, 1998, vol. 49.

[14] L. Kleinrock, *Queueing Theory Volume 1*, Wiley, 1967.
[15] Y. A. Korilis, A. A. Lazar, and A. Orda, "Architecting noncooperative networks," *IEEE Journal of Selected Areas in Communications*, vol. 13, no. 7, pp. 1241–1251, 1995.
[16] Y. A. Korilis and A. A. Lazar, "On the existence of equilibria in noncooperative optimal flow control," *Journal of the ACM*, vol. 42, no. 3, pp. 584–613, 1995.
[17] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," *Lecture Notes in Computer Science*, vol. 1563, pp. 404–413, 1999.
[18] M. J. Osborne and A. Rubenstein, *A Course in Game Theory*, The MIT Press, Cambridge, Massachusetts, 1994.
[19] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," in *ACM SIGCOMM '98*, Vancouver, CA, 1998, pp. 303–314.
[20] R. Pan, B. Prabhakar, and K. Psounis, ""CHOKe - A stateless queue management scheme for approximating fair bandwidth allocation"," in *INFOCOM*, 2000.
[21] C. H. Papadimitrou, "Algorithms, games, and the Internet," in *ACM Symposium on Theory of Computing*, 2001.
[22] K. Park, M. Sitharam, and S. Chen, "Quality of Service provision in noncooperative networks: heterogeneous preferences, multi-dimensional qos vectors, and burstiness," *Proc. 1st International Conference on Information and Computation Economies , 1998.*, 1998.
[23] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, 1999.
[24] J. Postel, "Transmission Control Protocol," Internet Request for Comments RFC 793, September 1981.
[25] T. Roughgarden and E. Tardos, "How bad is selfish routing?," in *IEEE Symposium on Foundations of Computer Science*, 2000, pp. 93–102.
[26] T. Roughgarden, "Designing networks for selfish users is hard," in *IEEE Symposium on Foundations of Computer Science*, 2001, pp. 472–481.
[27] S. Shenker, "Making greed work in networks: A game-theoretic analysis of switch service disciplines," *IEEE/ACM Transactions on Networking*, vol. 3, 1995.

APPENDIX

A. Oblivious AQM and MAC

Shenker [27] defines a class of allocation of functions called *MAC* that forms the basis of the analysis. Consider rates of agents r_i and their buffer occupancy c_i. Now any allocation is in MAC must have

1) $\frac{\partial c_i}{\partial r_j} \geq 0$ for all i and j.
2) $\frac{\partial c_i}{\partial r_i} > 0$ for all i.

We have the following observation:

Theorem 5: Any oblivious router mechanism with average bounded buffer occupancy cannot in MAC.

Proof: Suppose an oblivious strategy satisfying the above is in MAC. Let $R = \sum_{i=0}^{n} r_i$ and L be the average queue length. Since we assume Poisson traffic sources, we can write c_i as

$$c_i \;=\; \frac{r_i}{R} L.$$

Now taking partial derivatives and noting that our router strategy is oblivious, we have

$$\frac{\partial c_i}{\partial r_j} \;=\; \frac{-r_i L}{R^2} \;+\; \frac{r_i}{R}\left(\frac{dL}{dr_j}\right) \;=\; \frac{r_i}{R}\left(\frac{-L}{R} + \frac{dL}{dR}\right). \quad (9)$$

Now, for the above condition (1) to hold, we have have $\frac{\partial c_i}{\partial r_j} \geq 0$. That implies

$$\frac{dL}{L} \;\geq\; \frac{dR}{R}.$$

On solving the above, we have

$$L \;\geq\; kR \text{ , } k \text{ is a constant.}$$

Now, R is unbounded in the MAC framework. Hence L must be unbounded. But we assume that L is bounded. Hence, this is a contradiction. Thus, this AQM cannot be in MAC ■

Since drop-tail and RED are oblivious, and ensure bounded average buffer occupancy, they cannot be in MAC.

B. Drop Tail

Lemma 3: In a M/M/1/K queue, the goodput increases as the offered load increases.

Proof: Assume the service rate is unity. The goodput μ is given by

$$\mu \;=\; \lambda(1-p) \;=\; \frac{\lambda(1-\lambda^B)}{1-\lambda^{B+1}}.$$

Thus,

$$\mu \;=\; 1 - \frac{1}{1+\lambda+\lambda^2+\ldots+\lambda^B}.$$

Thus, we see that μ has always a positive derivative. Thus, goodput increases with offered load ■