Theoretical Computer Science 452 (2012) 107–116

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Game-theoretic analysis of Internet switching with selfish users[☆]

Alex Kesselman[a], Stefano Leonardi[b,*]

[a] Google Inc., USA

[b] Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti, Sapienza University of Rome, Roma, Italy

ARTICLE INFO

Article history:
Received 4 February 2012
Received in revised form 12 May 2012
Accepted 23 May 2012
Communicated by X. Deng

Keywords:
Selfish users
Internet switching
Drop policy

ABSTRACT

We consider the problem of Internet switching, where traffic is generated by selfish users. We study a packetized (TCP-like) traffic model, which is more accurate than the widely used fluid model. We assume that routers have First-In-First-Out (FIFO) buffers of bounded capacity managed by the drop-tail policy. The utility of each user depends on its goodput and the congestion level. Since selfish users try to maximize their own utility disregarding the system objectives, we study Nash equilibria that correspond to a steady state of the system. We quantify the degradation in the network performance called the price of anarchy resulting from such selfish behavior. We show that for a single bottleneck buffer, the price of anarchy is proportional to the number of users. Then we propose a simple modification of the Random Early Detection (RED) drop policy, which reduces the price of anarchy to a constant. We demonstrate that a Nash equilibrium can be reached if all users deploy TCP Vegas as their transport protocol under the drop-tail policy. We also consider some natural extensions of our model including the case of multiple Quality of Service (QoS) requirements, routing on parallel links and general networks with multiple bottlenecks.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

If all Internet users voluntarily deploy a congestion-responsive transport protocol (e.g. TCP [20]), one can design this protocol so that the resulting network would achieve certain performance goals such as high utilization or low delay. However, with the fast growth of the Internet users population, the assumption about cooperative behavior may not remain valid. Users are likely to behave "selfishly", that is each user makes decisions so as to optimize its own utility, without coordination with the other users. Buffer sharing and bandwidth allocation problems are prime candidates for such a selfish behavior.

If a user does not reduce its sending rate upon congestion detection, it can get a better share of the network bandwidth. On the other hand, all users suffer during congestion collapse, since the network delay and the packet loss increase drastically. Therefore, it is important to understand the nature of congestion resulting from selfish behavior. A natural framework to analyze this class of problems is that of non-cooperative games, and an appropriate solution concept is that of Nash equilibrium [25]. Strategies of the users are at a Nash equilibrium if no user can gain by unilaterally deviating from its current policy.

The subject of this paper is a game-theoretic analysis of the Internet switching problem. We consider a bottleneck First-In-First-Out (FIFO) buffer shared by many users. The users compete for the buffer share in order to maximize their throughput,

☆ A preliminary version of this paper appeared in Proceedings of WINE'05.

* Corresponding author.
E-mail addresses: alex.kesselman@gmail.com (A. Kesselman), Stefano.Leonardi@dis.uniroma1.it, leon@dis.uniroma1.it (S. Leonardi).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.05.029

but they suffer from the created congestion. We assume that each user knows its buffer usage, the queue length and the buffer size. In reality, these parameters can be estimated only approximately (e.g. such a mechanism exists in TCP Vegas). However, in this work we rather study *lack of coordination* and not *lack of information*. The goal of a user is to maximize its utility. We assume that the utility function of a user increases when its goodput increases and decreases when the network congestion increases.

We assume that there are n users and the buffer capacity is B while $B \gg n$. In our model all packets have a constant size. Time is slotted. Every time step each user may or may not send a packet to the buffer.[1] These packets arrive in an arbitrary order and are processed by the buffer management policy one by one. Then the first packet in the FIFO order is transmitted on the output link.

The drop policy has to decide at each time step which of the packets to drop and which to accept. It can also preempt (drop) accepted packets from the buffer. Under the *drop-tail* policy, all arriving packets are accepted if the buffer is not full and dropped otherwise (when the buffer is full).

We denote by w_i^t the number of packets of user i in the buffer at the beginning of time step t. We call it the *buffer usage* of user i. We denote by $W^t = \sum_i w_i^t$ the *queue length* at time t. We also denote by W_{-i}^t the buffer usage of all users but user i, i.e., $W_{-i}^t = W^t - w_i^t$. We assume that the users are greedy, i.e., they always have data to send and the queue is never empty.

We denote by $r_i^t = w_i^t / W^t$ the *instant transmission rate* of user i at time t. We say that the system is in a *steady state* if the queue length remains constant. Intuitively, when a packet of user i is transmitted at time step t, at time step $t+1$ user i will send a new packet to the queue. In a steady state, the *average transmission rate* of user i is $r_i = w_i/W$, where w_i is the average buffer usage of user i and W is the average queue length. Note that in our model transmission rate is essentially equal to the goodput of the user.

Now we proceed to define a utility function, which falls into a general family of utility functions increasing in the sending rate and decreasing in the network congestion (see [9,30]). However, the rate and the delay (congestion) are rather incomparable values and thus it is natural to normalize one of these parameters. In this work we introduce a novel notion of *congestion level*, that is the congestion level at time t is $L^t = \frac{W^t}{B}$. Note that the congestion level is zero when the buffer is empty and one when the buffer is full.

The utility of user i at time t is

$$u_i(w_i^t, W_{-i}^t) = r_i^t \cdot (1 - L^t),$$

i.e., it is proportional to the goodput while decreasing in the congestion level. We assume that user i sends a packet to the buffer at time t if its utility increases, i.e.,

$$u_i(w_i^t + 1, W_{-i}^t) > u_i(w_i^t, W_{-i}^t).$$

Note that users maximize their instantaneous utility, and not the long-term one adapting to the current network conditions. In a steady state, the utility of user i is

$$u_i(w_i, W_{-i}) = r_i \cdot (1 - L),$$

where $L = W/B$. Observe that when the buffer is almost empty, the utility of each user approximately equals its goodput. On the other hand, when the buffer is nearly full, all users have utility close to zero. The latter situation can be viewed as congestion collapse.

The *strategy* of each user is its buffer usage while the strategies of the other users define the background buffer backlog. Now we define a Nash equilibrium.

Definition 1. The system is said to be in a *Nash equilibrium* if no user can benefit by changing its instantaneous buffer usage.

The total utility of the users in a Nash equilibrium is $\sum_{i=1}^n u_i(w_i, W_{-i})$. Observe that under an optimal fair centralized policy, all users have equal sending rates of $1/n$ and experience zero delay, which results in the total utility of 1. For example, users may send one packet in turn every time step and in this case the link would be fully utilized with zero congestion level. We define the *price of anarchy* in a Nash equilibrium to be $1/\sum_{i=1}^n u_i(w_i, W_{-i})$. We are also concerned with the fairness of a Nash equilibrium. A Nash equilibrium is said to be *fair* if all users have the same buffer usage.

A Nash equilibrium in a networking environment is interesting only if it can be reached efficiently (in polynomial time). We define the *convergence time* to a Nash equilibrium as the maximum number of time steps required to reach a Nash equilibrium starting from an arbitrary state of the buffer.

We demonstrate that the drop-tail buffering policy imposes a fair Nash equilibrium, where all users have identical buffer usage. However, the price of anarchy is proportional to the number of users. We also show that the system converges to a Nash equilibrium in polynomial time, namely after $O(B^2)$ time steps. Then we propose a simple modification of the Random Early Detection (RED) policy [14] called Preemptive RED (PRED) that achieves a constant price of anarchy. We note that PRED is in the spirit of CHOKe [27] (see Section 4). We demonstrate that a Nash Equilibrium can be reached if all users deploy TCP Vegas as their transport protocol.

[1] We note that our results can be extended to the case in which each user can send an arbitrary number of packets.

Finally, we consider some natural extensions of our model including the case of multiple QoS requirements, routing on parallel links and general networks with multiple bottlenecks. We show that if users have different QoS requirements, the buffer usage of each user in a Nash equilibrium depends on the requested QoS. We demonstrate that in the case m identical links, the price of anarchy drops to n/m. We also establish that a max–min fair rate allocation is a Nash Equilibrium for general networks (under some restricting assumptions). We complement our theoretical analysis with simulations.

Paper organization. The rest of the paper is organized as follows. In Section 2 we discuss the related work. The analysis of a single buffer appears in Section 3. The PRED policy is presented in Section 4. Section 5 establishes a connection to TCP Vegas. In Section 6 we study some extensions of our model. We conclude with Section 7.

2. Related work

Shenker [30] analyzes switch service disciplines with a $M/M/1$ model and Markovian arrival rates. The utility of each user is increasing in its rate and decreasing in the network congestion. Shenker shows that the traditional FIFO policy does not guarantee efficiency and fairness and proposes a policy called Fair Share that guarantees both of them.

Garg et al. [18] study a switching problem using a continuous fluid-flow based traffic model, which is amenable to analysis of an arbitrary network. The user's utility is an increasing function of its goodput only. Garg et al. show that selfish behavior leads to congestion collapse and propose a rate inverse scheduling service discipline under which a max–min fair rate allocation is a Nash equilibrium. Contrary to [18], we consider only a simple FIFO scheduling policy.

Unfortunately, the complexity of the policies proposed in [30,18] is too high to implement them in the core of the Internet since they have to maintain per-flow state. Dutta et al. [10] analyze simple state-less buffer management policies under the assumption that the traffic sources are Poisson and the utility of each user is proportional to its goodput. Dutta et al. demonstrate that drop-tail and RED do not impose Nash equilibria and present a modification of RED that enforces an efficient Nash equilibrium. Differently from [10], we assume that the utility of each user depends on the network congestion as well.

In a recent paper, Gao et al. [17] propose an efficient drop policy that in case of congestion drops packets of the highest-rate sender. Gao et al. [17] show that if all sources are Poisson, this policy results in a Nash equilibrium which is a max–min fair rate allocation. In addition, it is demonstrated that the throughput of a TCP source is a constant factor of its max–min-fairness value when competing with Poisson sources. In contrast to [17], we do not make any assumptions regarding the traffic pattern.

Unlike the works mentioned above, in this paper we study *packetized* traffic model in which sources do not control the sending rate explicitly, but rather make it on per-packet basis. This model is implicit in the TCP protocol.

Nash equilibria for network games have been extensively studied in the literature. Douligeris and Mazumdar [9] determine conditions for a Nash equilibrium for an $M/M/1$ system when the utility of a user is a function of the throughput and the delay. Orda et al. [26] investigate the uniqueness of Nash equilibria in communication networks with selfish users. Korilis and Lazar [22] study Nash equilibria of a non-cooperative flow control game. Congestion control schemes using pricing based on explicit feedback are proposed by Kelly et al. [21]. Gibbens and Kelly [19] explore the implementation of network pricing in which users are charged for marks that routers place on packets in order to achieve congestion control. Akella et al. [1] present a game-theoretic analysis of TCP congestion control. Qiu et al. [28] consider selfish routing in intra-domain network environments.

Traditionally, in Computer Science research has been focused on finding a global optimum. With the emerging interest in computational issues in game theory, the *price of anarchy* introduced by Koutsoupias and Papadimitriou [23] has received considerable attention [7,8,15,29,12]. The price of anarchy is the ratio between the cost of the worst possible Nash equilibrium (the one with the maximum social cost) and the cost of the social optimum (an optimal solution with the minimum social cost). In some cases the price of anarchy is small, and thus good performance can be achieved even without a centralized control.

If a Nash equilibrium imposed by selfish users is not satisfying, one can deploy resource allocation policies to improve the situation. Recently, Christodoulou et al. [6] have introduced the notion of coordination mechanism, which is a resource allocation policy whose goal is to enforce a better Nash equilibrium. They show that even simple local coordination mechanisms can significantly reduce the price of anarchy.

Efficient convergence to a Nash equilibrium is especially important in the network environment, which is highly variable. The question of convergence to a Nash equilibrium has received significant attention in the game theory literature [16]. Altman et al. [2] and Boulogne et al. [4] analyze the convergence to a Nash equilibrium in the limit for a routing and a scheduling game, respectively. Even-Dar et al. [11] consider deterministic convergence time to a pure Nash equilibrium for a load balancing game (routing on parallel links).

3. Single buffer

In this section we consider the basic case of a single buffer. We first characterize Nash equilibria and derive the price of anarchy. Then we establish an upper bound on the convergence time. Finally, we study the Nash traffic model, that is traffic resulting in a Nash equilibrium.

3.1. Price of anarchy

The intuition explaining why congestion collapse does not happen is that at some point users start to suffer from the delay due to their own packets in the bottleneck buffer and would not further increase their sending rates. The next theorem shows the existence of a unique Nash equilibrium and derives the price of anarchy.

Theorem 1. *In a single buffer system, there is a unique fair Nash equilibrium, where all users have identical buffer usage, and the price of anarchy is n.*

Proof. By the definition of a Nash equilibrium, no user can increase its utility by changing the buffer usage. Therefore, a Nash equilibrium is a local maximum point for the utility function of each user i. We have that the first derivative $(u_i(w_i, W_{-i}))'$ must be equal to zero:

$$\frac{-w_i^2 - 2W_{-i}w_i - W_{-i}^2 + BW_{-i}}{B(w_i + W_{-i})^2} = 0,$$

and thus

$$w_i = \sqrt{BW_{-i}} - W_{-i}. \tag{1}$$

It is easy to verify that the second derivative $(u_i(w_i, W_{-i}))''$ is negative:

$$\frac{-2W_{-i}}{B(w_i + W_{-i})^3} < 0.$$

We obtain that a Nash equilibrium is fair since for each user i, $W = \sqrt{BW_{-i}}$, where W is constant. Summing equation (1) for all users, we get $W = B(n-1)/n$. Therefore, the unique Nash equilibrium is $w_i = B(n-1)/n^2$ for each user i. The price of anarchy in the Nash equilibrium is

$$1/\sum_{i=1}^{n}\left(\frac{1}{n}\cdot\frac{1}{n}\right) = n. \quad \square$$

For simplicity, we assume that $B(n-1)/n^2$ is an integer. Otherwise, there would be an approximate Nash equilibrium in which the buffer usage of each user varies between $\lfloor B(n-1)/n^2 \rfloor$ and $\lceil B(n-1)/n^2 \rceil$, as our simulations indeed show.

3.2. Convergence

Now we analyze convergence to a Nash equilibrium. The next theorem demonstrates that the convergence time to a Nash equilibrium is proportional to the square of the buffer size.

Theorem 2. *The convergence time to a Nash equilibrium is $O(B^2)$.*

Proof. Let $w^* = B(n-1)/n^2$ be the buffer usage of each user in a Nash equilibrium. Observe that until a Nash equilibrium is reached, each time step either a user with the maximum buffer usage (greater than w^*) can benefit from decreasing its buffer usage or a user with the minimum buffer usage (smaller than w^*) can benefit from increasing its buffer usage. Note that a user can increase its buffer usage at any time if the buffer is not full, but can decrease its buffer usage only when one of its packets is transmitted out of the buffer.

Initially, we let the system run for B time steps so that the users would be able to fill the buffer. Note that at this point no user with the maximum buffer usage wishes to increase its buffer usage. In the sequel, the maximum buffer usage will only decrease and the minimum buffer usage will only increase until they become equal to w^*.

We divide time into *phases* of B time steps. Consider a phase that starts at time step t. We argue that if a Nash equilibrium is not reached by time $t + B$, then either the maximum buffer usage decreases or the minimum buffer usage increases. Otherwise, there would be a Nash equilibrium contradicting Theorem 1, which states that in a Nash equilibrium all users have the same buffer usage. Therefore, the system reaches a Nash equilibrium after B phases or in B^2 time steps. $\square$

We also perform a convergence simulation. We consider different settings, where initially the buffer is filled randomly with packets of different users. Then we trace the system for a number of time steps and monitor the buffer usage. The dynamics of the minimum and the maximum buffer usage for 8 and 16 users appears in Fig. 1. The simulation results show that the actual convergence time is roughly proportional to the buffer size.

3.3. Nash traffic model

The traffic model of any QoS architecture is the regulation of the rate at which a user is allowed to inject packets into the network. There are two important policing criteria: *average rate* – limit of the long-term average rate at which packets can be injected into the network and *burst size* – the limit of the maximum number of packets that can be sent into the network over an extremely short time interval. The leaky bucket is an algorithm used in networks to check that data transmissions conform to defined limits on bandwidth and burstiness.

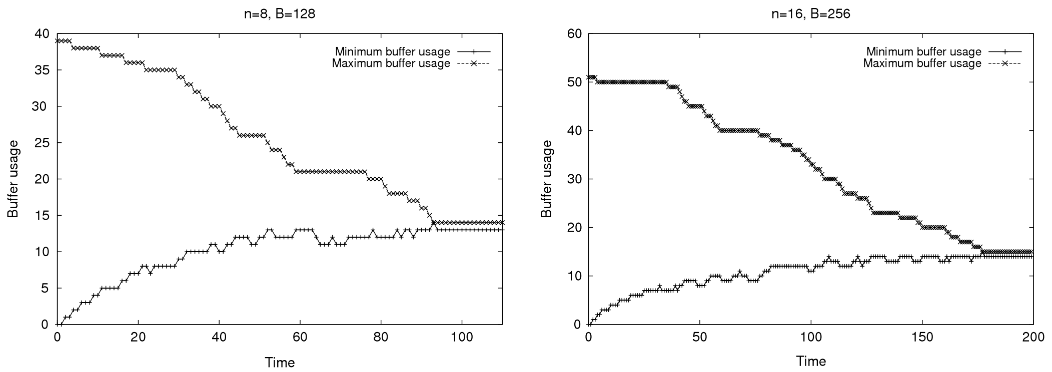

Fig. 1. Convergence to a Nash equilibrium.

(1) When a new packet arrives, apply the RED policy (**regular** drop).
(2) If the packet is dropped by RED, preempt (drop) the earliest packet of the same user from the buffer, if any (**extra** drop).

Fig. 2. The Preemptive RED (PRED) drop policy.

Definition 2. The (σ, ρ) *leaky bucket* mechanism is an abstraction modeling a source with an average rate of ρ and burst size of at most σ. Therefore, at any time interval of length l there are at most $\rho \cdot l + \sigma$ packets entering the system.

We show that the traffic of a user in a Nash equilibrium conforms to the leaky bucket model.

Theorem 3. *In a Nash equilibrium the traffic of each user conforms to the leaky bucket model with* $\sigma = B(n-1)/n^2$ *and* $\rho = 1/n$.

The theorem follows due to the fact that in order to maintain the buffer usage of $B(n-1)/n^2$, each user must send a burst of $B(n-1)/n^2$ packets and keep sending packets at the average rate of $1/n$.

4. PRED policy

In this section we consider the problem of designing a coordination mechanism (drop policy) that will improve the price of anarchy. The main goal of a drop policy is to control the average queue length. This can be done by dropping arriving packets probabilistically when the average queue length exceeds some pre-defined threshold. Such a policy called Random Early Detection (RED) has been introduced by Floyd and Jacobson [14]. We will show that a simple modification of RED reduces the price of anarchy to a constant. Again, we assume that the users are aware of the drop policy.

The RED policy calculates the average queue length, using a low-pass filter with an exponential weighted moving average. The average queue length is compared to two thresholds, a minimum threshold and a maximum threshold. When the average queue length is less than the minimum threshold T_l, no packets are dropped. When the average queue length is greater than the maximum threshold T_h, every arriving packet is dropped. When the average queue length is between the minimum and the maximum threshold, each arriving packet is dropped with probability p, which is a function of the average queue length.

Pan et al. [27] devise a simple packet dropping scheme, called CHOKe, that discriminates against unresponsive or misbehaving flows aiming to approximate the fair queuing policy. When a packet arrives at a congested router, CHOKe picks up a packet at random from the buffer and compares it with the arriving packet. If they both belong to the same flow, then they are both dropped. Otherwise, the randomly chosen packet is left intact and the arriving packet is admitted into the buffer with the same probability as in RED.

Our goal is to ensure a small queue length. Unfortunately, CHOKe is not aggressive enough in penalizing misbehaving flows since the probability that the buffer usage of a non-responsive user decreases is inversely proportional to the queue length. We propose a modification of RED in the spirit of CHOKe, called Preemptive RED (PRED). The PRED policy is presented in Fig. 2. The main feature of PRED is extra drop mechanism that drops an additional packet of the same user from the buffer when its packet is dropped by RED. Intuitively, we try to penalize users that do not respond to congestion signals. Note that if there is no penalty associated with dropped packets, users will greedily send new packets as long as they can benefit from increasing their buffer usage.

The next theorem analyzes Nash equilibria imposed by PRED.

Theorem 4. *Under the PRED policy, the price of anarchy in a Nash equilibrium is at most* $\frac{B}{B-T_l}$ *and there exists a fair Nash equilibrium.*

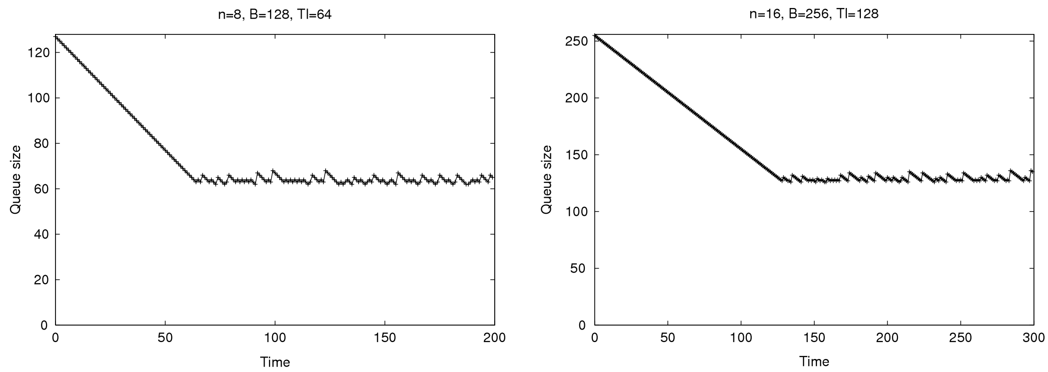

Fig. 3. The dynamics of the queue length under PRED.

Proof. First we show how to select the drop probability function so that in a Nash equilibrium the queue length is bounded by T_l. Consider the expected utility of user i after sending a new packet when the queue length is $X \geq T_l$. The buffer usage of user i will increase and decrease by one with probability $1 - p$ and p, respectively. Therefore, the expected utility of user i is

$$(1 - p(X)) \cdot u_i^t(w_i + 1, X_{-i}) + p(X) \cdot u_i^t(\max(w_i - 1, 0), X_{-i}).$$

User i would refrain from sending additional packets if the expected utility is less than its current utility $u_i(w_i, X_{-i})$, and we get that $p(X)$ must be greater than

$$\frac{u_i(w_i + 1, X_{-i}) - u_i(w_i, X_{-i})}{u_i(w_i + 1, X_{-i}) - u_i(\max(w_i - 1, 0), X_{-i})}. \tag{2}$$

Suppose that the drop probability function satisfies inequality (2) for each user i and for each $w_i \geq T_l/n$ (e.g. $p \approx 1/2$ satisfies the above requirements).

If $T_l \geq B(n - 1)/n$ then clearly there is a unique Nash equilibrium identical to that imposed by the drop-tail policy. If $T_l < B(n - 1)/n$, then we argue that $w^* = T_l/n$ is a fair Nash equilibrium. We have that no user can benefit from increasing its buffer usage above T_l/n by the selection of the drop probability function and no user can benefit from decreasing its buffer usage below T_l/n since in the Nash equilibrium imposed by the drop-tail policy the buffer usage of each user is greater than T_l/n.

We also claim that in a Nash equilibrium the queue length is bounded by T_l. If it is not the case, at least one user has buffer usage greater than T_l/n. However, the selection of the drop probability function implies that its buffer usage will eventually drop to or below T_l/n, which contradicts to the stability of a Nash equilibrium. Therefore, the price of anarchy is at most $\frac{B}{B-T_l}$. □

Note that if the queue length exceeds T_l, new users with zero buffer usage would still keep sending new packets since they can only increase their utility. That allows to avoid Denial of Service (DoS) when the buffer is completely monopolized by old users.

Next we present a simulation that illustrates the effect of PRED on the queue length. The setting is similar to that of Section 3. The dynamics of the queue length for 8 and 16 users is presented in Fig. 3. It turns out that the queue length in a steady state is very close to T_l and has a small variance, as our analysis indeed shows.

5. Connection to TCP Vegas

In this section we demonstrate that a Nash equilibrium can be achieved if the users deploy TCP Vegas [5] as their transport protocol. To detect network congestion, once every round trip time (*RTT*), TCP Vegas uses the current window size (*WIN*), the most recent *RTT* and the minimum *RTT* observed so far (*minRTT*) to compute:

$$\Delta = WIN\frac{RTT - minRTT}{RTT}.$$

Since $RTT - minRTT$ is the total path queuing delay and WIN/RTT is an estimate of the current throughput, the product of these two values is an estimate of the number of packets from this flow that are backlogged in the network. The goal of the Vegas congestion avoidance algorithm is to keep this number within a fixed range defined by two thresholds, α and β. Thus, TCP Vegas increases *WIN* if $\Delta < \alpha$ and decreases it if $\Delta > \beta$. Otherwise, the window size remains unchanged. The next observation shows that a steady state of TCP Vegas with proper parameters is a Nash equilibrium.

Observation 1. *A steady state of TCP Vegas corresponds to a Nash equilibrium if* $\alpha = \beta = w^*$, *where* w^* *is the buffer usage from which users have no incentive to deviate.*

Note that in a steady state, the *RTT* observed by the users does not change and the sending window is solely a function of the minimum observed *RTT* and the current *RTT*. We have that the buffer usage of each user in the steady state is w^*,

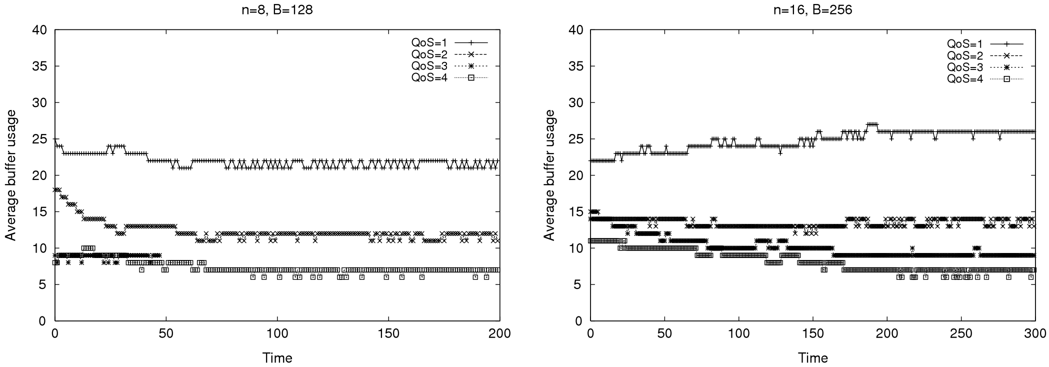

Fig. 4. The buffer usage distribution according to the requested QoS.

provided that the minimum observed *RTT* equals $(n-1)w^*$, i.e., the current *RTT* equals the minimum observed *RTT*. Clearly, it is a Nash equilibrium.

Unfortunately, in reality it is hard to guarantee that all users see the "right" minimum *RTT*.

6. Extensions

In this section we study some natural extensions of our model. In particular, we study the case of multiple QoS requirements, the case of parallel links and the case of general networks.

6.1. Multiple QoS requirements

First we consider multiple QoS requirements. Note that users running real-time and multimedia applications are more sensitive to the delay than users running file transfer or web surfing applications (so called elastic traffic). We add an additional QoS parameter to the utility function: the utility function of user i at time t is

$$u_i(w_i^t, W_{-i}^t) = r_i^t \cdot \left(1 - L^t\right)^{k_i},$$

where k_i is the QoS requested by user i. Note that the larger k_i, the smaller delay is required by the application of user i. For instance, $k = 1$ may correspond to elastic traffic. The proof of the following theorem is analogous to that of Theorem 1.

Theorem 5. *There exists a fair Nash equilibrium in which the buffer usage of each user i depends on k_i, where all users with the same QoS requirement have the same buffer usage.*

We perform a simulation in which users with different QoS requirements share the same buffer. The setting is similar to that of Section 3. The dynamics of the buffer usage with respect to the requested QoS for 8 and 16 users appears in Fig. 4. Note that in a steady state the users with stringent delay requirements have a smaller buffer usage compared to that of the users with more relaxed delay requirements.

6.2. Parallel links

Now we study the case of m parallel links. The *capacity* of a link is defined as the number of packets that can be transmitted at each time step. Link j has capacity R_j and a buffer of size B_j. The model is the same as in the previous section, with the exception that the routes are not fixed and consist of only one link, i.e., each user must select exactly one of the links. The utility function of each user is defined as the utility for the buffer corresponding to the selected link. Note that the timing may be different for each link depending on its capacity. For example, a large enterprise may have multiple outbound links leading to different Internet Service Providers (ISPs). We assume that $R_1 \geq \cdots \geq R_m$ and all buffers have a uniform size B.

The next theorem derives the price of anarchy as a function of the link rates.

Theorem 6. *The price of anarchy in a Nash equilibrium is* $n\frac{\sum_{j=1}^m R_j}{\left(\sum_{j=1}^m \sqrt{R_j}\right)^2}$.

Proof. Consider a Nash equilibrium. According to Theorem 1, in a local Nash equilibrium imposed by the drop-tail policy on link j, the utility of each user is R_j/n_j^2. Suppose that $n >> m$ is sufficiently large so that all links are populated. Since in a Nash equilibrium no user can benefit by switching to another link, it must be the case that for any two links j and j' we have that: $R_j/n_j^2 = R_{j'}/n_{j'}^2$. We obtain that for $j > 1$,

$$n_j = n_1\sqrt{R_j/R_1}. \tag{3}$$

Summing equation (3) over all users, we get that

$$n = n_1 \sum_{j=1}^{m} \sqrt{R_j/R_1}.$$

Observe that the optimal utility is $\sum_{j=1}^{n} R_j$. Hence, the price of anarchy is

$$\frac{\sum_{j=1}^{m} R_j}{\sum_{j=1}^{m} \frac{R_j}{n_j}} = n \frac{\sum_{j=1}^{m} R_j}{\left(\sum_{j=1}^{m} \sqrt{R_j}\right)^2}. \quad \square$$

Corollary 1. *If all links are identical, the price of anarchy is n/m.*

Another interesting problem related to network design is how to fix the buffer sizes in order to achieve at the Nash equilibrium the same delay for all parallel links. This question is related to selfish routing introduced by Roughgarden and Tardos [29], where in a Nash equilibrium the delay on all paths between the source and the destination is the same.

Theorem 7. *In the Nash equilibrium all users experience the same delay if we have that*

$$B_l = B_1 \cdot \sqrt{\frac{R_l}{R_1}} \frac{\frac{n}{\sum_{j=1}^{m} \sqrt{\frac{R_j}{R_1}}} - 1}{\sqrt{\frac{R_l}{R_1}} \frac{n}{\sum_{j=1}^{m} \sqrt{\frac{R_j}{R_1}}} - 1}$$

for each link $l \neq 1$.

6.3. General networks

Finally, we consider the case of a general network. Each user i sends its flow via a fixed route $p_i = j_1^i, \ldots, j_{l_i}^i$ consisting of l links. We denote by n_j the number of users passing through link j. We add a subscript/superscript j to the notation in order to refer to the corresponding parameter of link j. We assume that the utility function of user i is

$$u_i = r_i \cdot \prod_{j \in p_i} (1 - L_j),$$

where

$$r_i = \min_{j \in p_i} r_i^j.$$

We say that a link j is a *bottleneck* for user i if $r_i = r_i^j$. The rationale is that the throughput of a user is determined by a bottleneck link with the lowest throughput while delay is a function of all links on the selected route. Note that in our objective function the congestion level is a *multiplicative* metric, although the delay is an *additive* metric. However, if we take the logarithm of the utility function, we obtain that

$$\log u_i = \sum_{j \in p_i} \log({r_i}^{1/l_i}(1 - L_j)).$$

Observe also that if the traffic of each user passes through only one non-empty buffer, this utility function coincides with the utility function for the case of a single buffer.

Max–min fairness is a simple, well-recognized approach to define fairness in networks [3], where the minimum data rate that a dataflow achieves is maximized. The aim is to allocate as much as possible to poor users without wasting resources. Observe that if j is a bottleneck link for user i then it is fully utilized and user i has the maximal rate among all users that share this link. An algorithm for obtaining a max–min fair rate allocation is as follows. We increase rates of flows at the same pace, until a link is saturated. Then we fix the rates of flows passing through this saturated link and keep increasing others. The procedure is repeated until there are no more flows whose rates can be increased.

Definition 3. We call a rate allocation max–min fair if one cannot increase the rate of a user without decreasing the rate of another user with smaller transmission rate.

In the next theorem we demonstrate that a max–min fair rate allocation is a Nash Equilibrium and derive the price of anarchy. In our analysis we will make a simplifying assumption that the effect of an individual user on the queue lengths of non-bottleneck links on its route is negligible.[1]

Theorem 8. *A max–min fair rate allocation is a Nash Equilibrium and the price of anarchy is at most n^l, where l is the length of the longest route, that is* $\max_i l_i$.

Proof. Consider a max–min fair rate allocation. Let j_i be the bottleneck link of user i and let S_j be the set of users for which link j is the bottleneck link.

Note that user i affects the queue length of its bottleneck link j_i and by our assumption its influence on the queue length of any other link $j' \in p_i - \{j_i\}$ is negligible. We get that the utility of user i is approximately

$$u_i \approx C \cdot r_i^{j_i} \cdot (1 - L_{j_i}),$$

where

$$C = \prod_{j \in p_i, j \neq j_i} (1 - L_j).$$

Theorem 1 implies that if all users in S_{j_i} share the bandwidth equally and the queue length is

$$T_{j_i} = B_{j_i}(|S_{j_i}| - 1)/|S_{j_i}|,$$

it is a Nash Equilibrium for these users. However, there is also background traffic generated by users not in S_{j_i} passing through j_i, each of which has a buffer usage that guarantees its bottleneck rate. Thus, users in S_{j_i} will be in a Nash Equilibrium if they share the bandwidth equally while the queue length may be greater than T_{j_i} due to this background traffic.

Now we will bound the price of anarchy. Observe that under an optimal centralized policy, all users experience zero delay and the utility of user i is simply r_i. On the other hand, the utility of user i in a Nash Equilibrium is

$$r_i \cdot \prod_{j \in p_i} (1 - L_j) \geq r_i / n^l,$$

which yields the theorem. □

The performance limitations in the current Internet arise primarily from constraints at the edges of the network, i.e., last mile connectivity to users. Narrow-band access links (e.g., dial-up or DSL) limit the ability of applications to utilize the bandwidth available in the backbone and result in negligible queuing in the interior of the network. Thus, in the Internet a user typically has two potential bottlenecks, one at the entry and one at the exit point from the core of the network.

Corollary 2. *For Internet-like networks, the price of anarchy is at most n^2.*

Note that if we deploy the PRED drop policy, the price of anarchy is reduced to a constant, as shown in the following theorem.

Theorem 9. *Under the PRED policy, there exists a max–min fair Nash equilibrium and the price of anarchy is at most* $\left(\frac{B}{B - T_l}\right)^l$.

Proof sketch. The proof of this theorem is similar to that of Theorem 8. The only difference is that each bottleneck link is controlled by PRED and according to Theorem 4, $T_{j_i} = T_l$. Hence, the utility of user i in a Nash Equilibrium is

$$r_i \cdot \prod_{j \in p_i} (1 - L_j) \geq \left(\frac{B}{B - T_l}\right)^l,$$

which establishes the theorem. □

7. Concluding remarks

The Internet is stable today mostly due to the fact that the majority of the users voluntarily use a congestion-responsive TCP protocol. However, some users can benefit from not reducing their transmission rate during congestion. Thus, if users behave selfishly, the assumption about cooperative behavior may not remain valid. Therefore, it is important to understand the nature of congestion resulting from selfish behavior.

We analyze the users' behavior by means of game theory. We consider a single bottleneck buffer under a packetized traffic model, which makes our approach more close to real networks such as the Internet. We show that there exist efficient and fair Nash equilibria imposed by a simple FIFO buffering policy. Moreover, such an equilibrium can be reached if the users deploy TCP Vegas as their transport protocol. However, the congestion created by the users is rather high. Then we propose a simple modification of RED policy, which decreases the congestion at a Nash equilibrium. Finally, we consider some natural extensions of our model.

We believe that our results can help to shed more light on the stability of the existing Internet infrastructure in the presence of selfish users. Some interesting open problems include analysis of routing in general networks, alternative utility functions and interaction of greedy and non-greedy users (e.g. VBR and CBR applications).

[1] We note that this assumption is not necessarily correct for arbitrary networks with multiple bottlenecks.

For further reading

[13,24]

Acknowledgments

The work of the first author is supported by AvH-Stiftung. A part of this work has been done while visiting the Dipartimento di Scienze dell'Informazione at Universita' di Roma ''La Sapienza'', Italy. The first author is also supported in part by EYES project and MIUR grant FIRB WebMinds.

References

[1] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, S. Shenker, Selfish behavior and stability of the internet: a game-theoretic analysis of TCP, Proceedings of ACM SIGCOMM'02.
[2] E. Altman, T. Basar, T. Jimenez, N. Shimkin, Routing into two parallel links: game-theoretic distributed algorithms, Journal of Parallel and Distributed Computing on ''Routing in Computer and Communication Networks'' 61 (9) (2001) 1367–1381. (special issue).
[3] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, 1987.
[4] T. Boulogne, E. Altman, O. Pourtallier, On the convergence to Nash equilibrium in problems of distributed computing, Annals of Operation research, 2002.
[5] L.S. Brakmo, L.L. Peterson, TCP Vegas: end to end congestion avoidance on a global internet, IEEE Jounal on Selected Areas in Communications 13 (1995) 1465–1480.
[6] G. Christodoulou, E. Koutsoupias, A. Nanavati, Coordination mechanisms, in: Proceedings of ICALP'04.
[7] A. Czumaj, P. Krysta, B. Vocking, Selfish traffic allocation for server farms, in: Proceedings of STOC'02.
[8] A. Czumaj, B. Vocking, Tight bounds on worst case equilibria, in: Proceedings of SODA'02.
[9] C. Douligeris, R. Mazumdar, On Pareto optimal flow control in an integrated environment, in: Proceedings of the 25th Allerton Conference on Communication, Control and Computing, 1987.
[10] D. Dutta, A. Goel, J. Heidemann, Oblivious AQM and Nash equilibria, in: Proceedings of INFOCOM'03.
[11] E. Even-Dar, A. Kesselman, Y. Mansour, Convergence time to Nash equilibria, in: Proceedings of ICALP'03, pp. 502–513.
[12] R. Feldmann, M. Gairing, T. Lücking, B. Monien, M. Rode, Nashification and the coordination ratio for a selfish routing game, in: Proceedings of ICALP'03, pp. 514–526.
[13] M. Florian, D. Hearn, Network equilibrium models and algorithms, in: M.O. Ball, et al. (Eds.), Network Routing. Handbooks in RO and MS, Elsevier, 1995, pp. 485–550.
[14] S. Floyd, V. Jacobson, Random early detection for congestion avoidance, in: IEEE/ACM Transactions on Networking, August 1993.
[15] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, P. Spirakis, The structure and complexity of nash equilibria for a selfish routing Game, in: Proceedings of ICALP'02.
[16] D. Fudenberg, D. Levine, The theory of learning in games, MIT Press, 1998.
[17] X. Gao, K. Jain, L.J. Schulman, Fair and efficient router congestion control, in: Proceedings of SODA'04, pp. 1050–1059.
[18] R. Garg, A. Kamra, Varun Khurana, A game-theoretic approach towards congestion control in communication networks, ACM SIGCOMM Computer Communications Review 32 (3) (2002) 47–61.
[19] R. Gibbens, F. Kelly, Resource pricing and the evolution of congestion control, Automatica 35 (1999) 1969–1985.
[20] V. Jacobson, Congestion avoidance and control, in: Proceedings of ACM SIGCOMM'88.
[21] F. Kelly, A. Maulloo, D. Tan, Rate control in communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society 49 (1998) 237–252.
[22] Y.A. Korilis, A.A. Lazar, On the existence of equilibria in noncooperative optimal flow control, Journal of the ACM 42 (1995) 584–613.
[23] E. Koutsoupias, C.H. Papadimitriou, Worst-case equilibria, in: Proceedings of STACS'99.
[24] R.J. La, V. Anantharam, Optimal routing control: game theoretic approach, in: Proceedings of the 36rd IEEE Conference on Decision and Control, 1997, pp. 2910–2915.
[25] J.F. Nash, Non-cooperative games, Annals of Mathematics 54 (1951) 286–295.
[26] A. Orda, N. Rom, N. Shimkin, Competitive routing in multi-user communication networks, IEEE/ACM Transaction on Networking 1 (1993) 614–627.
[27] R. Pan, B. Prabhakar, K. Psounis, CHOKe – a stateless active queue management scheme for approximating fair bandwidth allocation, in: Proceedings of INFOCOM00, pp. 942–951.
[28] L. Qiu, Y.R. Yang, Y. Zhang, S. Shenker, On selfish routing in internet-like environments, in: Proceedings of ACM SIGCOMM'03.
[29] T. Roughgarden, E. Tardos, How bad is selfish routing?, Journal of the ACM 49 (2) (2002) 236–259.
[30] S. Shenker, Making greed work in networks a game-theoretic analysis of switch service disciplines, IEEE/ACM Transactions on Networking 3 (1995) 819–831.