Dynamic Internet Congestion with Bursts*

Stefan Schmid[1] and Roger Wattenhofer[2]

[1] Computer Engineering and Networks Laboratory (TIK), ETH Zurich, CH-8092 Zurich, Switzerland
schmiste@tik.ee.ethz.ch
http://dcg.ethz.ch/members/stefan.html
[2] Computer Engineering and Networks Laboratory (TIK), ETH Zurich, CH-8092 Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch
http://www.distcomp.ethz.ch/members/roger.html

Abstract. This paper studies throughput maximization in networks with dynamically changing congestion. First, we give a new and simple analysis of an existing model where the bandwidth available to a flow varies multiplicatively over time. The main contribution however is the introduction of a novel model for dynamics based on concepts of network calculus. This model features a limited form of amortization: After quiet times where the available bandwidth was roughly constant, the congestion may change more abruptly. We present a competitive algorithm for this model and also derive a lower bound.

1 Introduction

The problem of avoiding congestion in the Internet has been studied with zeal for many years. The TCP congestion control mechanism of todays Internet successfully employs a window-based scheme to prevent the Internet from being overloaded. Thereby, the size of the so-called *TCP congestion window* is an approximation of the available network capacity. When TCP suffers a packet loss, it assumes that the network is congested and reduces the window's size. Consequently, the sending rate is cut down, and the Internet hosts collaboratively alleviate the load.

In the past, the transport layer and in particular the congestion problem was first studied empirically, and later embraced by the queuing theory and control theory communities. In order to analyze and compare protocols theoretically, a traffic model is needed. Queuing and control theory researchers have refined their early Poisson traffic models to an astonishing level of detail. However, probabilistic models are intricate to analyze. Probabilistic models that are simple enough to be analytically tractable might never model traffic accurately enough, as the nature of network traffic is self-similar and *bursty* [18].

In their seminal paper Karp, Koutsoupias, Papadimitriou, and Shenker [10] have proposed to study congestion control from a worst-case perspective instead. Karp et al. model congestion control as an online game between a flow and an adversarial network. In particular, the available bandwidth of the network changes over time and the

* Research supported by the Swiss National Science Foundation.

Y. Roberts et al. (Eds.): HiPC 2006, LNCS 4297, pp. 159–170, 2006.

flow gets only a limited feedback—namely, whether packets have been lost or not—about the currently available bandwidth.

In this paper, we follow the algorithmic online approach proposed by Karp et al. [10] to broaden our understanding of congestion control. We build upon [10] by focusing on the dynamics of congestion. In particular, we integrate a notion of bursts happening in a worst-case manner. Although we do not claim that our models accurately represent what happens in the Internet, we believe that they are interesting and may ignite a further discussion on future variants of congestion control.

Concretely, after a new analysis of a model by Karp et al., we introduce the burst model: Instead of considering an adversary which always changes the bandwidth similarly each round, our adversary may accumulate some power in quiet rounds and then change the congestion more abruptly in later rounds. For this adversary, a lower as well as an upper bound are derived for the competitive ratio.

The paper is organized as follows. Section 2 reviews related work and also gives a short overview of the relevant network calculus concepts. After setting the stage in the model section (Section 3), we study the case of multiplicatively changing congestion in Section 4. In Section 5 our new model is presented in detail and analyzed. We state open problems in Section 6 and conclude the paper in Section 7.

2 Related Work

TCP lies at the heart of today's Internet, and many aspects of TCP are still subject to active research. For a reference on TCP, we refer the reader to [17]. TCP congestion control has been studied intensively, both from an empirical and from a theoretical perspective. Due to space constraints, we are bound to concentrate on the closest related work only.

In our work, we analyze congestion control from a *worst-case perspective* using competitive analysis. Generally, we think that a better algorithmic (worst-case) understanding of the transport layer is necessary. Whereas all other layers have received quite a lot of attention in the past (e.g., cf. [2] for the link layer, and [15] for the network layer), there has been comparatively little algorithmic networking research about the transport layer. Some notable exceptions are for instance adversarial queuing theory [6], the study of the TCP ACK problem [9], or mechanism design [7].

Our model is due to Karp et al. [10] who define several optimization problems related to congestion control. The authors investigate the issue of regulating the rate of a single unicast flow when the bandwidth available to it is unknown and changes over time. In our paper, we extend [10] in two respects: First, we provide a new analysis of a model where the bandwidth changes multiplicatively; our analysis is simpler and gives *strict* competitive bounds. Second, we enhance their model with *bursts*: Thereby, the congestion may change more after a time of quiescence.

The work by Karp et al. has already had an interesting follow-up by Arora and Brinkman [4] who study *randomized* algorithms for a dynamically changing congestion. In particular, they propose an asymptotically optimal randomized online algorithm against an adversary which may change the congestion by a constant factor in every round. Unfortunately, they assume a fairly weak oblivious adversary (see also the discussion in Section 6): Their algorithm uses randomization only in the first round, while the sending rate of all other rounds is computed deterministically. The adversary however is not allowed to be adaptive in these deterministic rounds.

The idea that an adversary may accumulate power over time has already appeared in the area of packet routing and is related to the adversarial queuing theory by Borodin et al. [6]. The problem considered there is as follows: Given a packet switched network and an adversary which continuously injects packets that have to be routed from a source to a destination node, how much buffer space is needed at the nodes, and what is the delivery time? In the paper by Aiello et al. [1], the adversary is allowed to inject any sequence of packets into the network, as long as in any w consecutive rounds, the total load created by the paths associated with the packets inserted in this time period is at most wr on any edge, for some $w \geq 1, r \leq 1$. The adversary studied by Andrews et al. [3] is similar to our adversary. Given two parameters $b \geq 1, r \leq 1$, for any $T \geq 1$ consecutive time steps, the adversary may inject as many packets as it wants, as long as the total load created by the paths associated with these packets is at most $Tr + b$ on any edge. These two adversary models have been compared by Rosén in [16]. A contribution of our paper is to introduce a modified version of the adversary in [3] on the *transport layer*.

Short Overview of Network Calculus. We now give a short introduction to those concepts of network calculus which are relevant to our work. Network calculus is a relatively new technique to analyze deterministic queuing systems found in communication networks. For a detailed introduction to network calculus, see [14].

In network calculus, there exists the notion of *arrival curves* which provide deterministic limitations to the network traffic sent by sources. Given that the data flows indeed correspond to these limitations, it is possible to make statements about the deterministic behavior of the network (maximal delays, maximal queue lengths, etc.).

Arrival curves are defined as follows. Let R be a data flow, and let $R(t)$ be the total number of bits R has sent until time t. Let α be an increasing function defined for all times $t \geq 0$. We say that R has an arrival curve α if and only if for all $s \leq t$:

$$R(t) - R(s) \leq \alpha(t - s)$$

In other words, the total number of bits sent until time t by flow R may never exceed the bits sent by R until some time s plus $\alpha(t-s)$. In this paper, we look at a so-called *leaky bucket arrival curve* defined as $\alpha(t) = c_1 t + c_2$ for some non-negative constants c_1, c_2. Figure 1 visualizes the constraints imposed upon a flow R by such an arrival curve: The total number of bits sent may increase by c_2 at once and with a rate c_1 over time, unless there is a conflict with a constraint from a previous round. Informally, the total number of bits must always be less or equal the minimum constraint that arises if the curve α is attached to all points of $R(t)$.

Note that such an arrival curve incorporates a limited form of amortization: If flow R only sends a few bits for several rounds, the constraints of earlier rounds get weaker and allow R to send up to c_2 bits at once in some later round.

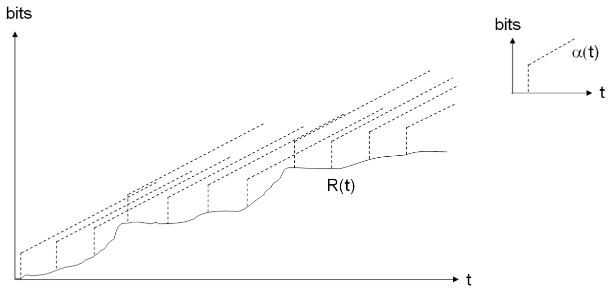

Fig. 1. Leaky bucket arrival curve: The number of bits sent by flow R may never exceed the constraints from earlier times (dashed lines), i.e., $\forall s \leq t : R(t) \leq R(s) + \alpha(t-s)$

3 Model

In the Internet, there is no central authority allocating bandwidth to hosts. On the contrary, individual hosts are responsible for setting their sending rate.[1] In this paper, we consider the problem of regulating the rate of a unicast flow from one host to another such that the throughput is maximized. The bandwidth available to the flow thereby fluctuates according to the varying requirements for bandwidth of other competing flows. A host is not provided direct information about the competing demands for bandwidth or the Internet topology, but does receive some limited information as to whether the flow is experiencing packet drops, and must determine its transmission rate solely on the basis of this information.

We assume that time is divided into infinitely many successive *rounds* and consider a *worst-case model* where in every round t, an adversary ADV selects the available bandwidth u_t. Thereby, u_t represents the maximum rate at which a host can transmit without experiencing packet drops. The host on the other hand runs an algorithm ALG which decides the sending rate x_t of round t, and receives immediate feedback as to whether packet drops have occurred, i.e., whether $x_t > u_t$. ALG can then choose the rate x_{t+1}.

We assume a *severe cost model* [10] where a host cannot transmit anything in round t if $x_t > u_t$, but can transmit at a rate x_t if $x_t \leq u_t$. Formally, the *gain* of ALG in round t is defined as follows:

$$gain_{ALG}(x_t, u_t) := \begin{cases} x_t & \text{, if } x_t \leq u_t \\ 0 & \text{, otherwise} \end{cases}$$

[1] Usually, this is done automatically by TCP. However, by using the User Datagram Protocol (UDP), selfish programs can try to maximize their own throughput and may have no incentive to reduce congestion collaboratively. Although it is generally believed that routers are configured to give priority to TCP packets [8]—with the consequence that UDP packets are dropped first if the Internet gets congested—at least in theory it is possible to design networking software from scratch that circumvents this restriction by sending UDP packets which look like TCP packets.

An optimal offline algorithm OPT knows the sequence $\{u_t\}$ in advance and achieves a gain of

$$gain_{OPT}(x_t, u_t) = u_t$$

in round t. These gains reflect two major issues: The online algorithm experiences an *opportunity cost* if its sending rate is smaller than the available bandwidth (case $x_t < u_t$), and a *retransmission overhead* if its packets are dropped due to congestion (case $x_t > u_t$).

We are in the realm of *competitive analysis* [5] and define the (strict) *competitive ratio* ρ achieved by ALG as the total amount of data (over all rounds) sent by OPT divided by the total amount of data sent by ALG (cf. Definition 3.1).

Definition 3.1. *[ρ-competitive] We say that an algorithm ALG is* (strictly) ρ-competitive *compared to an optimal offline algorithm OPT if for all input sequences I, it holds that*

$$gain_{OPT}(I) \leq \rho \cdot gain_{ALG}(I).$$

The goal of the online algorithm designer is to minimize ρ. Henceforth, we will assume that ALG knows the initial bandwidth, i.e., $x_0 = u_0$.

Observe that an unrestricted adversary could frustrate every online algorithm by always selecting $u_t := x_t - \varepsilon$ for some arbitrary small $\varepsilon > 0$. The natural way out proposed by Karp et al. [10] is to assume that the available bandwidth does not change too drastically over time. In this paper, we study different ways to restrict the adversary. In Section 4, we consider the multiplicative model proposed by Karp et al. In Section 5, we extend this model to allow for changes with *bursts*.

We will call rounds t where the online algorithm successfully transmits its packets without loss *good rounds*, and rounds t where $x_t > u_t$ *bad rounds*, cf. Definition 3.2.

Definition 3.2 (Good and Bad Rounds). *A round t where $x_t \leq u_t$ is called* good*, a round t where $x_t > u_t$ is called* bad.

We defer the description of the different adversaries to the corresponding sections. However, we now define the following class of online algorithms.

Definition 3.3 ($\mathcal{ALG}(G, B)$). *Let $\mathcal{ALG}(G, B)$ be the online algorithm which chooses*

$$x_{t+1} := \begin{cases} G \cdot x_t & \text{, if } x_t \leq u_t \\ B \cdot x_t & \text{, otherwise} \end{cases}$$

for some $G \geq 1$ and $B \leq 1$. That is, the algorithm $\mathcal{ALG}(G, B)$ increases the rate by a factor G after a good round, and decreases it by a factor B after a bad round.

The sending rate x_{t+1} of an algorithm $\mathcal{ALG}(G, B)$ depends solely on the binary feedback whether its probing rate x_t was larger than the available bandwidth u_t in the previous round or not.

4 Multiplicative Adversaries

In this section, we look at multiplicative changes of the available bandwidth. We first consider a model where the adversary can increase the bandwidth at most by a factor $\mu \geq 1$ per round and can decrease it arbitrarily (cf. Definition 4.1). Later, we will study a model where also the reduction is constrained multiplicatively (cf. Definition 4.2).

So let's look at the adversary $\mathcal{ADV}_{mult}$ (cf. Definition 4.1) proposed by Karp et al.

Definition 4.1 **($\mathcal{ADV}_{mult}$).** *$\mathcal{ADV}_{mult}$ may choose the new bandwidth u_{t+1} in the interval $[0, u_t \cdot \mu]$, i.e.,*

$$\mathcal{ADV}_{mult} : u_{t+1} \in [0, u_t \cdot \mu],$$

for some given $\mu \geq 1$.

First, we restate the lower bound given in [10].

Theorem 4.1. *[10] Against $\mathcal{ADV}_{mult}$, no online algorithm can achieve a competitive ratio smaller than μ.*

Proof. Consider the following adversary ADV: In every round t, it chooses

$$u_t := \begin{cases} \mu & \text{, if } x_t \leq 1 \\ 1 & \text{, otherwise} \end{cases}$$

Thus, whenever an online algorithm ALG sends at a rate larger than one, all its packets are dropped because of congestion. On the other hand, if ALG transmits at a rate of 1 or less, the rate of OPT is at least a factor μ larger. Moreover, since ADV changes the available bandwidth at most by a factor of μ per round, it is indeed of type $\mathcal{ADV}_{mult}$. □

In [10], it is shown that the algorithm $\mathcal{ALG}(\mu, \frac{\sqrt{\mu}}{\sqrt{\mu}+\sqrt{\mu-1}})$ yields a competitive ratio of

$$\rho = (\sqrt{\mu} + \sqrt{\mu - 1})^2$$

against $\mathcal{ADV}_{mult}$. However, [10] uses a different definition for the competitive ratio which allows for (possibly large) additive constants. By our strict definition (cf. Definition 3.1), the ratio can be much larger. To see this, assume an adversary which reduces the available bandwidth in every round by a factor slightly larger than $\frac{\sqrt{\mu}+\sqrt{\mu-1}}{\sqrt{\mu}}$. In this case, $\mathcal{ALG}(\mu, \frac{\sqrt{\mu}}{\sqrt{\mu}+\sqrt{\mu-1}})$ is only successful in the first round, and hence $gain_{ALG} = u_0$, while

$$gain_{OPT} \approx u_0 \cdot \sum_{i=0}^{\infty} (\frac{\sqrt{\mu}}{\sqrt{\mu} + \sqrt{\mu - 1}})^i.$$

Therefore, the (strict) competitive ratio is

$$\rho = \frac{gain_{OPT}}{gain_{ALG}} \approx \frac{\sqrt{\mu} + \sqrt{\mu - 1}}{\sqrt{\mu - 1}}.$$

For small μ, ρ can get very large (for instance $\rho > 100$ if $\mu = 1.0001$).

In the following, we give a simple proof that the algorithm $\mathcal{ALG}(\mu, 1/2)$ has a strict competitive ratio 4μ. According to Theorem 4.1, this is asymptotically optimal.

Theorem 4.2. *$\mathcal{ALG}(\mu, 1/2)$ is 4μ-competitive against $\mathcal{ADV}_{mult}$.*

Proof. First, we show by induction that in every good round t, $u_t \leq 2\mu x_t$. For $t = 0$, $u_0 = x_0$ and the claim holds. For the induction step, consider the round $t-1$ before the good round t. There are two possibilities: either round $t-1$ has been bad ($x_{t-1} > u_{t-1}$) or good ($x_{t-1} \leq u_{t-1}$). If round $t-1$ has been bad, we have $x_t = x_{t-1}/2$ and $u_t \leq u_{t-1}\mu < x_{t-1}\mu = 2\mu x_t$, hence $u_t/x_t < 2\mu$, and the claim holds. If on the other hand round $t-1$ was good, the algorithm increases the bandwidth at least as much as the adversary. Together with the induction hypothesis, the claim also follows in this case.

Having studied the gain in good rounds, we now consider bad rounds. We show that in the bad rounds following a good round t, the adversary can increase its gain at most by $2\mu x_t$. So let t be the good round preceding a sequence of bad rounds, i.e., $x_t \leq u_t$, $x_{t+1} > u_{t+1}$, $x_{t+2} > u_{t+2}$, etc. We know that $x_{t+1} = \mu x_t$, so—because it is a bad round—u_{t+1} must be smaller than μx_t. Furthermore, we have $x_{t+2} = x_{t+1}/2 = \mu x_t/2$ and hence $u_{t+2} < \mu x_t/2$, $x_{t+3} = \mu x_t/4$ and hence $u_{t+3} < \mu x_t/8$, and so on. By a geometric series argument, the gain of the adversary in the bad rounds is upper bounded by $2\mu x_t$.

Therefore,

$$\begin{aligned}\rho &= \frac{gain_{OPT}(good) + gain_{OPT}(bad)}{gain_{ALG}(good)} \\ &< \frac{2\mu \cdot gain_{ALG}(good) + 2\mu \cdot gain_{ALG}(good)}{gain_{ALG}(good)} \\ &< 4\mu.\end{aligned}$$

□

To conclude this section, we give another kind of proof to show that the algorithm $\mathcal{ALG}(\mu, 1/\mu^3)$ has a good competitive ratio for small μ. For our analysis, we assume a slightly more restricted adversary $\mathcal{ADV}^*_{mult}$ (cf. Definiton 4.2).

Definition 4.2 ($\mathcal{ADV}^*_{mult}$). *$\mathcal{ADV}^*_{mult}$ chooses the new bandwidth u_{t+1} from the interval $[u_t/\mu, u_t \cdot \mu]$, i.e.,*

$$\mathcal{ADV}^*_{mult} : u_{t+1} \in [u_t/\mu, u_t \cdot \mu].$$

Theorem 4.3. *$\mathcal{ALG}(\mu, 1/\mu^3)$ is $(\mu^4 + \mu)$-competitive against $\mathcal{ADV}^*_{mult}$.*

Proof. The fact that ALG reduces its rate by a factor μ^3 after a bad round implies that the next round is always good: Assume, for the sake of contradiction, that round $t+1$ is the first bad round following another *bad* round t, which—by the induction hypothesis—follows a good round $t-1$. Hence, $x_{t-1} \leq u_{t-1}$. Moreover, observe that $u_{t+1} \geq u_t/\mu \geq u_{t-1}/\mu^2$, but on the other hand, $x_{t+1} = x_t/\mu^3 = \mu x_{t-1}/\mu^3 = x_{t-1}/\mu^2$. Therefore, $x_{t+1} \leq u_{t+1}$. Contradiction!

We now first analyze the gain of a good round t and show that $u_t < \mu^4 x_t$. There are two cases: Either round $t-1$ has also been good, or not. If it has been a good round, then round t is at least as competitive as round $t-1$ because $x_t = \mu x_{t-1}$. If on the other hand round $t-1$ has not been good, we have $u_{t-1} < x_{t-1}$, $x_t = x_{t-1}/\mu^3$ and $u_t \leq \mu u_{t-1}$. Therefore, $x_t = x_{t-1}/\mu^3 > u_{t-1}/\mu^3 \geq u_t/\mu^4$, and the claim follows.

Next, we study the gains in a bad round t. In this case, it holds that $u_t < \mu x_{t-1}$: Since $x_{t-1} \leq u_{t-1}$, $x_t = \mu x_{t-1}$ and $u_t < x_t$, and hence $u_t < \mu x_{t-1}$.

Therefore,

$$\begin{aligned}\rho &= \frac{gain_{OPT}(good) + gain_{OPT}(bad)}{gain_{ALG}(good)} \\ &< \frac{\mu^4 \cdot gain_{ALG}(good) + \mu \cdot gain_{ALG}(good)}{gain_{ALG}(good)} = \mu^4 + \mu.\end{aligned}$$

□

Since $\mathcal{ADV}^*_{mult}$ is a special case of $\mathcal{ADV}_{mult}$, Theorem 4.2 also applies for $\mathcal{ADV}^*_{mult}$. Hence, it is possible to run $\mathcal{ALG}(\mu, 1/\mu^3)$ against $\mathcal{ADV}^*_{mult}$ if μ is small, and $\mathcal{ALG}(\mu, 1/2)$ otherwise, which yields the following corollary.

Corollary 4.4. *There is a deterministic online algorithm which is* $\min\{\mu^4 + \mu, 4\mu\}$*-competitive against* $\mathcal{ADV}^*_{mult}$.

5 Network Calculus Adversary

5.1 Description of $\mathcal{ADV}_{nc}$

In this section, we introduce the adversary $\mathcal{ADV}_{nc}$ which is based on *network calculus* [14] concepts. We will extend the model introduced in Section 4 by a form of limited amortization which allows for more drastic bandwidth changes after times of quiescence.

$\mathcal{ADV}_{nc}$ has two parameters: A *rate* $\mu \geq 1$ and *maximum burst factor* $B \geq 1$. In every round, the available bandwidth u_t varies according to these parameters in a multiplicative manner. More precisely, $\mathcal{ADV}_{nc}$ can select the new bandwidth u_{t+1} from the interval

$$\mathcal{ADV}_{nc} : u_{t+1} \in [\frac{u_t}{\beta_t \mu}, u_t \cdot \beta_t \cdot \mu],$$

that is, the available bandwidth can change by a factor of at most $\beta_t \mu$. Thereby, β_t is the *burst factor at time* t. This burst factor is explained next.

On average, the available bandwidth can change by a factor μ per round. However, there can be times of only small changes, but then the bandwidth changes by factors larger than μ in later rounds. This is modeled with the burst factor β_t: At the beginning, β_t equals B, i.e., $\beta_0 = B$. For $t > 0$, the burst factor β_t is computed depending on β_{t-1} and the actual bandwidth change c_{t-1} that has happened in round $t-1$. More precisely,

$$\beta_t = \min\{B, \beta_{t-1} \frac{\mu}{c_{t-1}}\}$$

where

$$c_t := \begin{cases} \frac{u_{t+1}}{u_t} & \text{, if } u_{t+1} > u_t \\ \frac{u_t}{u_{t+1}} & \text{, otherwise} \end{cases}$$

This means that if the available bandwidth has changed by a factor less than μ in round t, i.e., $c_t < \mu$, the burst factor *increases* by a factor μ/c_t, and hence the bandwidth can change more in the next round—and vice versa if $c_t > \mu$.

In other words, the adversary can save adversarial power for forthcoming rounds. However, this amortization is limited as β_t never becomes larger than B for all rounds t. Also note that $\forall t : \beta_t \geq 1$, as $c_t \leq \mu\beta_t$ by the definition of $\mathcal{ADV}_{nc}$.

5.2 Analysis

At first sight, it seems that $\mathcal{ADV}_{nc}$ has roughly the same power as $\mathcal{ADV}^*_{mult}$: In order to change the bandwidth with a factor larger than μ, $\mathcal{ADV}_{nc}$ must have changed the bandwidth by a factor less than μ in previous rounds.[2] However, as we will see in the following, an online algorithm cannot exploit these quiet rounds sufficiently, and the competitive ratio does depend on B.

Theorem 5.1. *The competitive ratio is at least* $\Omega\left(\mu\sqrt{B}/\log B\right)$ *against* $\mathcal{ADV}_{nc}$.

Proof. Consider the following adversary ADV. ADV will select $u_t = 1$ whenever the burst factor β_t is not maximal in a round t, i.e., if $\beta_t < B$. If $\beta_t = B$, ADV continues choosing $u_t = 1$ until $x_t \leq 1$ for the first time. Then, if $x_t \leq 1$ and $\beta_t = B$, it selects $u_t = \mu\sqrt{B}$ but immediately sets the available bandwidth back to $u_{t+1} = 1$ in the next round. Therefore, no online algorithm can ever transmit at a rate larger than 1. Since ADV must be of type $\mathcal{ADV}_{nc}$, it can do this trick at most every $\lceil \log B/\log\mu \rceil$ rounds: After these two bursts (from 1 to $\mu\sqrt{B}$ and from $\mu\sqrt{B}$ back to 1), the burst factor becomes 1, and it takes $\lceil \log B/\log\mu \rceil$ rounds to increase it again to B: $\mu^i \geq B \Leftrightarrow i \geq \log B/\log\mu$.

Let us call the time period between two rounds where ADV raises the bandwidth from 1 to $\mu\sqrt{B}$ a *phase*. In every phase, ALG has a gain of at most

$$gain_{ALG} \leq 2 + \lceil \log B/\log\mu \rceil .$$

On the other hand, the optimal algorithm's gain is at least

$$gain_{OPT} \geq 1 + \lceil \log B/\log\mu \rceil + \mu\sqrt{B}.$$

Hence,

$$\rho = \frac{gain_{OPT}}{gain_{ALG}} \geq \frac{1 + \lceil \log B/\log\mu \rceil + \mu\sqrt{B}}{2 + \lceil \log B/\log\mu \rceil} \in \Omega\left(\mu\frac{\sqrt{B}}{\log B}\right).$$

□

Note that the lower bound given in Theorem 5.1 even holds for online algorithms which get perfect (instead of only binary) feedback about the bandwidth of the previous round.

Although we were not able to find an algorithm which yields a tight upper bound, it can be shown that $\mathcal{ALG}(\mu\sqrt[3]{B}, 1/2)$ comes close to the bound of Theorem 5.1.

[2] Except for the first rounds of course, where a burst B comes "for free". However, as mentioned in Section 3, we consider infinite games only.

Theorem 5.2. *The competitive ratio of $\mathcal{ALG}(\mu\sqrt[3]{B}, 1/2)$ is $\mathcal{O}\left(\mu^{3/2}B^{2/3}\right)$ against $\mathcal{ADV}_{nc}$.*

Proof. We use again the proof technique of Section 4. First, we analyze the missed gain in bad rounds:

$$\begin{aligned} gain_{OPT}(bad) &\leq \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i \cdot \mu\sqrt[3]{B} \cdot gain_{ALG}(good) \\ &\leq 2\mu\sqrt[3]{B} \cdot gain_{ALG}(good) \in \mathcal{O}\left(\mu\sqrt[3]{B}\right) \cdot gain_{ALG}(good) \end{aligned}$$

Next, the good rounds are tackled. Let t be the last bad round before a good round $t+1$. Hence, $x_t > u_t$, $x_{t+1} = x_t/2 \leq u_{t+1}$, and $x_{t+2} = \mu\sqrt[3]{B}x_t/2$.

There are two cases: Either round $t+2$ is also good, or not. If round $t+2$ is good, $u_{t+2} \leq \mu^2 B x_t$. We have

$$\rho \leq \frac{u_{t+1} + u_{t+2}}{x_{t+1} + x_{t+2}} \leq \frac{\mu B + \mu^2 B}{1/2 + \mu\sqrt[3]{B}/2} \in \mathcal{O}\left(\mu B^{2/3}\right)$$

More good rounds would reduce this ratio, because ALG grows faster than ADV.

If round $t+2$ is not good, it holds that $x_t > u_t$ and $x_{t+2} = \mu\sqrt[3]{B}x_t/2 > u_{t+2}$. Now observe that $u_{t+1} < \mu^{3/2}B^{2/3}x_t$. Assume, for the sake of contradiction, that $u_{t+1} \geq \mu^{3/2}B^{2/3}x_t$. Then the burst factor in round $t+1$ is at most $\beta_{t+1} \leq \sqrt[3]{B}/\sqrt{\mu}$, and thus

$$u_{t+2} \geq \frac{u_{t+1}}{\mu\beta_{t+1}} \geq \frac{\mu^{3/2}B^{2/3} \cdot \sqrt{\mu}}{\sqrt[3]{B} \cdot \mu} x_t = \mu\sqrt[3]{B}x_t > x_{t+2}.$$

Contradiction. Hence,

$$\rho \leq \frac{u_{t+1}}{x_{t+1}} \leq \frac{\mu^{3/2}B^{2/3}x_t}{x_t/2} \in \mathcal{O}\left(\mu^{3/2}B^{2/3}\right)$$

Thus, in conclusion,

$$\begin{aligned} \rho &= \frac{gain_{OPT}(good) + gain_{OPT}(bad)}{gain_{ALG}(good)} \\ &\leq \frac{\mathcal{O}\left(\mu^{3/2}B^{2/3}\right) \cdot gain_{ALG}(good) + \mathcal{O}\left(\mu\sqrt[3]{B}\right) \cdot gain_{ALG}(good)}{gain_{ALG}(good)} \\ &\in \mathcal{O}\left(\mu^{3/2}B^{2/3}\right) \end{aligned}$$

□

6 Open Research Questions

Karp et al. have already pointed out several future research directions, for instance the study of different cost models. In this paper, we have extended their work by a novel model for the dynamics of the available bandwidth.

We believe that our network calculus model opens up many exciting questions. For example, the lower bound and the upper bound we presented are not tight. It would be interesting to know if there are asymptotically better online algorithms, or whether our lower bound is too pessimistic. Another challenge is the design of *randomized* online algorithms. In fact, Arora and Brinkman [4] have addressed this problem for the multiplicative adversary $\mathcal{ADV}_{mult}$ and presented an algorithm with competitive ratio $\mathcal{O}(\log \mu)$. By using Yao's minimax principle [5], it can be shown that this is asymptotically optimal. However, the authors assume a weak oblivious adversary: Their scheme uses randomization only in the first round, while all later rounds are deterministic. But the adversary is not allowed to be adaptive even in these deterministic rounds! The case of a stronger adversary is still an open problem. It is straight-forward to extend the algorithm by Arora and Brinkman for $\mathcal{ADV}_{nc}$: Over-pessimistically, we can assume that $\mathcal{ADV}_{nc}$ changes the bandwidth by a factor $B \cdot \mu$ in every round, which yields a competitive ratio of $\mathcal{O}(\log(B\mu))$. However, also here, it would be interesting to study a more powerful adversary which can be adaptive in deterministic rounds.

Finally, we believe that our network calculus adversary is an interesting model for dynamics in completely different fields of research.

7 Conclusion

This paper has studied online algorithms which aim at maximizing throughput in the presence of dynamic bandwidth changes. We have derived an asymptotically optimal algorithm for a multiplicative model. Moreover, a novel model for the congestion dynamics has been presented together with a lower and an upper bound for the competitive ratio. We hope that our models will give an impetus for future research. Generally, we believe that a better algorithmic (worst-case) understanding of the transport layer is necessary. Whereas all other layers have received quite a lot of attention in the past, the transport layer has always been a step-child of algorithmic networking research.

References

1. W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosén. Adaptive Packet Routing for Bursty Adversarial Traffic. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC)*, pages 359–368, New York, NY, USA, 1998.
2. D. Aldous. Ultimate Instability of Exponential Back-off Protocol for Acknowledgement Based Transmission Control of Random Access Communication Channels. *IEEE Transactions on Information Theory*, 1987.
3. M. Andrews, B. Awerbuch, A. Fernández, T. Leighton, Z. Liu, and J. Kleinberg. Universal-Stability Results and Performance Bounds for Greedy Contention-Resolution Protocols. *Jounal of the ACM*, 48(1):39–69, 2001.
4. S. Arora and B. Brinkman. A Randomized Online Algorithm for Bandwidth Utilization. In *Proceedings of the 13th Annual ACM Symposium on Discrete Algorithms (SODA)*, pages 535–539, Philadelphia, PA, USA, 2002.
5. A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

6. A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. Adversarial Queuing Theory. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (STOC)*, 1996.
7. J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the Cost of Multicast Transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.
8. K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall. Improving the Reliability of Internet Paths with One-hop Source Routing. In *Symposium on Operating Systems Design & Implementation (OSDI)*, 2004.
9. A. Karlin, C. Kenyon, and D. Randall. Dynamic TCP Acknowledgement and Other Stories about $e/(e-1)$. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (STOC)*, 2001.
10. R. M. Karp, E. Koutsoupias, C. H. Papadimitriou, and S. Shenker. Optimization Problems in Congestion Control. In *Proceedings of Symposium on Foundations of Computer Science (FOCS)*, pages 66–74, 2000.
11. F. Kelly. Mathematical Modelling of the Internet. In *Bjorn Engquist and Wilfried Schmid: Mathematics Unlimited*. springer, 2001.
12. F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49, 1998.
13. T. V. Lakshman and U. Madhow. The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, 1997.
14. J.-Y. Le Boudec and P. Thiran. *Network Calculus*. Springer LNCS 2050 Tutorial, 2001.
15. F. Leighton, B. Maggs, and S. Rao. Universal Packet Routing Algorithms. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 256–269, 1988.
16. A. Rosén. A Note on Models for Non-Probabilistic Analysis of Packet Switching Networks. *Inf. Process. Lett.*, 84(5):237–240, 2002.
17. R. Stevens and G. R. Wright. *TCP/IP Illustrated Vol. 2 (The Implementation)*. Addison-Wesley, 1995.
18. W. Willinger, W. Leland, M. Taqqu, and D. Wilson. On the Self-Similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking*, pages 1–15, 1994.