Adaptive AIMD Congestion Control *

Alex Kesselman[†] akessel@mpi-sb.mpg.de

Yishay Mansour[‡] mansour@cs.tau.ac.il

Abstract

The main objectives of a congestion control algorithm are high bandwidth utilization, fairness and responsiveness in changing environment. However, these objectives are contradicting in particular situations since the algorithm has to constantly probe available bandwidth, which may affect its stability. This paper proposes a novel congestion control algorithm that achieves high bandwidth utilization providing fairness among competing connections and, on the other hand, is sufficiently responsive to changes of available bandwidth. The main idea of the algorithm is to use *adaptive* setting for the Additive Increase/Multiplicative Decrease (AIMD) congestion control scheme, where parameters may change dynamically, with respect to the current network conditions.

1 Introduction

With its explosive growth, the Internet backbone faces the challenge of operating at its capacity. When TCP acquires information about the current network conditions, there exists a strong tension between needing to keep a large amount of data in flight in order to fill the bandwidth delay product versus having to wait lengthy periods of time to attain feedback regarding a changing environment, especially at the onset of congestion. Moreover, in most cases the Internet does not provide any explicit information about the network state, and it is up to the transport protocol to form its own estimates, and then to use them to adapt as efficiently as possible. For these reasons congestion avoidance and control have become critical to the use of the Internet. TCP congestion control based on additive increase/multiplicative decrease (AIMD) has been introduced by Jacobson [10]. This scheme helped to avoid the congestion collapse, as shown by Floyd and Fall [6]. The basic congestion control has been later augmented with fast retransmission and fast recovery mechanisms due to Jacobson [11] and Stevens [28].

In recent years, significant research efforts have been devoted to modeling such a complex protocol as TCP. Under certain assumptions of synchronized feedback, Chiu and Jain [5] demonstrated that an AIMD control scheme converges to a stable and fair operating point efficiently utilizing the network resources (see also [16]). Yang and Lam [29] investigate tuning of the AIMD congestion control parameters. Lakshman and Madhow [20] and Kumar [17] use Markovian analysis to express the throughput of a TCP connection. Mathis et al. [24] and Padhye et al. [25] study analytic models of the TCP behavior. Kelly [15] and Low et al. [18, 21, 19] consider

* A preliminary version of this paper appeared in Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC), 2003.

† Max Planck Institut fur Informatik, Stuhlsatzenhausweg 85, 66123, Saarbrucken, Germany.

‡ School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.

TCP congestion control as an algorithm seeking the global optimum under a certain utility (cost) function and infer the functions implicit in the existing protocols. Katabi et al. [14] design a number of mechanisms for providing more fine-grained congestion related feedback from routers.

While TCP congestion control is appropriate for elastic applications such as bulk data transfer, some real-time applications find halving the sending rate in response to a single packet loss to be unnecessarily severe because it results in drastic degradation in the user-perceived quality. As a result, a number of alternative TCP congestion control algorithms have been proposed. Bansal and Balakrishnan [2] study binomial congestion control algorithms, which are a nonlinear generalization of AIMD. Floyd et al. [7] introduce the TFRC algorithm that adjusts its sending rate as a function of the measured loss rate. Bansal et al. [3] investigate the behavior of slowly-responsive congestion control algorithms. However, for safe deployment in the Internet these algorithms must be stable and interact well with TCP. Mahdavi and Floyd [22] introduce the TCP-friendly paradigm, where a congestion control mechanism is said to be TCP-friendly if its bandwidth usage, in the presence of a constant loss rate, is the same as that of TCP.

Our work. We suggest an *adaptive* AIMD scheme, in which the parameters may change dynamically with respect to the obtained feedback* . In a nutshell, each positive feedback increases the additive increase term (while resetting the multiplicative decrease factor), and each negative feedback increases the multiplicative decrease factor (while resetting the additive increase term). Basically, our scheme has an inherent multiplicative increase rule of the additive increase term. This strategy, on the one hand, achieves high bandwidth utilization providing fairness among competing connections and, on the other hand, is sufficiently responsive on changes of available bandwidth in a dynamic environment. In particular, we demonstrate that (1) the length of a convergence period of our algorithm depends *logarithmically* on the available bandwidth (compared to the linear time of the existing AIMD algorithms); (2) in a steady state the bandwidth utilization is close to the optimal; (3) the time it takes for a connection to reach its fair share is proportional to the square of the logarithm of the available bandwidth. This algorithm, as we will show, is not TCP-friendly, i.e., when competing with TCP for a bottleneck resource, it will consume a larger fraction of bandwidth. However, following [2], we believe that the primary threat to the Internet stability comes not from the flows using some form of TCP-incompatible congestion control but from flows that do not use any congestion control at all. Since our algorithm demonstrates good theoretical characteristics, the proposed scheme may suggest new lines for further research in TCP congestion control.

Related work. The idea to make TCP congestion control adaptive is reinforced by some recent works. Floyd et al. [8] propose a modification of TCP congestion control that adapts the increase strategy and makes it more aggressive for high bandwidth links. Maor and Mansour [23] introduce AdaVegas, an adaptive version of TCP Vegas that updates the increase rate dynamically, and using simulations compare it to TCP Vegas.

Jin et al. [12] derive a TCP-friendly congestion control algorithm called SIMD (Square-Increase/Multiplicative-Decrease), which utilizes history information. Namely, the increase in window size is proportional to the square of the time elapsed since the detection of the last loss. Our scheme is similar to SIMD in sense that the increase rate also depends on the history, but contrary to [12], the decrease rate depends on the history as well. That allows our algorithm to achieve better bandwidth utilization in a stable environment.

Karp et al. [13] and Arora and Brinkman [1] study a model in which an adversary controls the available bandwidth and present almost optimal policies for regulating the rate of a single connection using multiplica-

* The feedback indicates whether any packets in the prior window encountered congestion in the network.

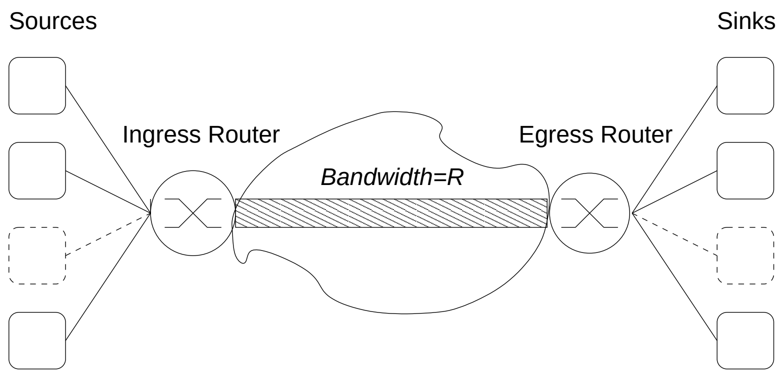

Figure 1: The network model.

tive increase/multiplicative decrease (MIMD) approach. Garg and Young [9] analyze another MIMD-based protocol and demonstrate that under certain assumptions the network throughput is near the maximum possible. However, it is well known that the MIMD scheme is unfair while our adaptive AIMD scheme is able to achieve high utilization as MIMD does without sacrificing fairness.

The congestion control algorithm has been often modeled as a system with a binary feedback function indicating congestion. Shenker [27] introduces a simple flow control scheme in which sources make synchronous rate adjustments, and applies it to a network of Poisson sources and exponential servers. Unlike [27], we make no assumptions regarding the traffic generation process. Piccolboni and Schindelhauer [26] investigate the problem of predicting the available bandwidth. In contrast to [26], we compare our algorithm with an optimal strategy rather than the best constant prediction strategy.

Bar-Noy et al. [4] study dynamic bandwidth allocation that takes into account latency and utilization and the goal is to minimize the number of bandwidth allocation changes. In this work we do not care about the number of bandwidth allocation changes, but instead we concentrate on the other objectives.

The rest of the paper is organized as follows. In Section 2 we present our model. Section 3 describes the adaptive congestion control algorithm. In Section 4 we analyze the performance of our algorithm. Section 5 contains analysis of long connections. In Section 6 we study the interaction of our algorithm with TCP. The concluding remarks appear in Section 7.

2 Model Description

In this section we describe our model. We start with the system model and then proceed to the congestion control objectives.

2.1 System Model

We consider multiple connections sharing a network path (see Figure 1). We assume that all connections run the same congestion control algorithm and have the same round trip time (RTT).

Time is divided into successive rounds, i.e., time is discrete and events happen only at time $t = 1, 2, 3, \ldots$.

At the start of a *round*, each connection transmits a window of packets and at the end of a round, each connection receives a feedback. We assume that there is always data pending at the source while the connection is active. Next we introduce a few useful definitions.

Definition 2.1 *We denote by S_t the set of active connections in the system at time t.*

The connections can *join* and later *leave* the system, i.e., the set of active connections may change.

Definition 2.2 *We denote by $w_t(i)$ the window of the i-th connection at time t. We denote by W_t the* cumulative *window at time t, that is $W_t = \sum_{i \in S_t} w_t(i)$.*

Next we relate the cumulative window size to the transmission rate. Note that since we assume identical RTT's, the total sending rate is proportional to the cumulative window size.

Definition 2.3 *We denote by R_t the total* transmission rate *at time t, which equals W_t/RTT.*

Let R be the bandwidth available to the connections. We define the target cumulative window at which the available bandwidth is fully utilized and no packets are dropped.

Definition 2.4 *We denote by B the* target *cumulative window for which $R_t = R$, i.e., $B = R \cdot RTT$.*

By the end of a round t, all connections receive a binary feedback from the network F_t. If $W_t \leq B$ then $F_t = 0$, which indicates a successful transmission of all packets sent. Otherwise, if $W_t > B$, then $F_t = 1$, which indicates dropping of one or more packets.

2.2 Congestion Control Objectives

The *congestion control algorithm* of connection i has to determine the connection window $w_t(i)$ at time t given the history up to this time, that is $w_0(i), \ldots, w_{t-1}(i)$ and $F_0, \ldots, F_{t-1}$. Now we turn to define the objectives of a congestion control algorithm.

First we consider bandwidth utilization. In case the set of active connections does not change during sufficiently long period of time, we would like the congestion control algorithm to reach a steady state in which the throughput of the algorithm is close to the available bandwidth R.

Definition 2.5 *The congestion control algorithm A is said to* stabilize *if there exist constants M_R and C_R such that if the set of connections does not change during $[t_0, t']$ then $|R_{t'} - R|/R \leq C_R$, where $t' > t_0 + M_R$. We call to C_R the* stability constant.

Next we introduce a relative fairness function for a congestion control algorithm, which states that after a pre-defined period of time in the system each connection must reach at least the average window size with respect to the other active connections minus a fairness index.

Definition 2.6 *The congestion control algorithm is said to be* fair *if there exist constants M_F and C_F such that if connection i is active during $[t_0, t']$ then $W_{t'}/|S_{t'}| - w_{t'}(i) \leq C_F$, where $t' > t_0 + M_F$. We call to C_F the* fairness constant.

3 Adaptive Congestion Control

In this section we present our congestion control algorithm. The algorithm uses a TCP-like additive increase/multiplicative decrease scheme. The novel idea of the algorithm is to update the additive increase term γ and the decrease factor β with respect to the received feedbacks. We initialize γ to 1 and multiply it by $\phi > 1$ each time after a successful transmission. Similarly, we start with β equal to $\psi > 1$ and multiply it by ψ after obtaining a negative feedback. The adaptive congestion control algorithm is presented on Figure 2.

Round 0: /* Initialization. */

$w_0 = 1;$

$\beta_0 = \psi;$

$\gamma_0 = 1;$

Round t:

$send(\lceil w_t \rceil)$; /* Send window of $\lceil w_t \rceil$ packets. */

If $F_t = 0$ **Then** /* No loss. */

$w_{t+1} = w_t + \gamma_t;$

$\gamma_{t+1} = \gamma_t \cdot \phi;$

$\beta_{t+1} = \psi;$

Else /* Loss occurred. */

$w_{t+1} = \max(w_t/\beta_t, 1)$; /* The minimal window size is 1. */

$\beta_{t+1} = \beta_t \cdot \psi;$

$\gamma_{t+1} = 1;$

Figure 2: Adaptive Congestion Control – $ACC(\phi, \psi)$.

It will be convenient for us to divide the run of the ACC algorithm into phases, as follows.

Definition 3.1 *We denote by a* phase *a maximal sequence of consecutive window increases followed by a sequence of decreases. We call the former sequence an* increase sub-phase *and the latter sequence a* decrease sub-phase.

To complete the description of $ACC(\phi, \psi)$ it remains to find proper ϕ and ψ values that will ensure stability in a steady state and at the same time allow the algorithm to respond quickly on bandwidth changes. If we assign ϕ and ψ values that are close to 1, that would guarantee stability since in a steady state an increase or decrease sub-phase continues only a few rounds and thus the amplitude of changes of the cumulative window size is limited. On the other hand, when the available bandwidth changes drastically, exponential growth of γ and β would provide quick adaptation to the new state. Intuitively, AIMD rule has inherent fairness characteristics and the algorithm indeed guarantees fairness, as we prove below.

Now we define a property that is necessary for convergence of the ACC algorithm. This property implies that for a proper selection of ϕ and ψ values, the number of rounds in a decrease sub-phase decreases from

$D(\phi,\psi)$ to 1, which allows us to show that the stability constant is small. (The exact meaning of the following definition will become clear from the analysis.)

Definition 3.2 *Let $D(\phi,\psi) = \lceil \sqrt{2\log_\psi \phi + 1/4} - 1/2 \rceil$. We say that ϕ and ψ are* well-behaved *if $\phi > \psi$ and for all $1 < k \leq D(\phi,\psi)$ the following holds*

$$k > \lceil \sqrt{2\log_\psi(\phi - (\phi-1)/\psi^{k(k+1)/2}) + 1/4} - 1/2 \rceil.$$

We also define a constant

$$C(\phi,\psi) = \max((1-1/\psi), (\phi-1)(1-1/\psi)).$$

We will show that $C(\phi,\psi)$ is in fact the stability constant. For example, the values $\phi = 1.4$ and $\psi = 1.1$ are well-behaved, and the corresponding constant is $C(1.4, 1.1) \approx 0.09$.

4 Analysis of the ACC Algorithm

In this section we analyze convergence and fairness of the ACC algorithm. We start from an arbitrary state and assume that the set of connections in the system remains unchanged during the convergence period. We also assume that there are N concurrent connections and $B \gg N$.

4.1 General Properties of the ACC Algorithm

In this section we derive some general properties of the ACC algorithm.

Observation 1 *All connections have the same values of γ and β after receiving the first negative feedback.*

In our analysis we assume that the connections have already received at least one negative feedback and thus are being synchronized (as we show later, it takes at most $\log_\phi(B/N)$ rounds).

We aim to show that the amplitude of changes of the cumulative window size always decreases until we reach a steady state. Thus, we derive an upper bound on the cumulative window size at the end of an increase sub-phase and a lower bound on the cumulative window size at the end of a decrease sub-phase.

The following claim bounds from above the cumulative window size at the end of an increase sub-phase as a function of the initial cumulative window size.

Claim 4.1 *For an increase sub-phase $\mathcal{I} = [t_1, t_2]$ the following holds:*

$$W_{t_2+1} \leq B + (\phi-1)(B - W_{t_1}) + N.$$

Proof: The claim trivially holds for $t_1 = t_2$. In what follows we assume that $t_2 > t_1$. Observe that for a round $t \in \mathcal{I}$ s.t. $t > t_1$, the increase term of connection i is $\gamma_t(i) = \phi^{t-t_1}$. Therefore,

$$w_t(i) - w_{t_1}(i) = \sum_{j=0}^{t-1-t_1} \phi^j = \frac{\phi^{t-t_1+1} - 1}{\phi - 1}.$$

Thus, we obtain that:

$$\gamma_t(i) = (\phi-1)(w_{t-1}(i) - w_{t_1}(i)) + 1.$$

After summing over all connections we get:

$$\begin{aligned}\sum_{i=1}^{N}\gamma_t(i) &\leq \sum_{i=1}^{N}((\phi-1)(w_{t-1}(i)-w_{t_1}(i))+1)\\ &= (\phi-1)(W_{t-1}-W_{t_1})+N.\end{aligned}$$

Notice that there is no loss at time t_2, i.e., $W_{t_2} \leq B$. Hence,

$$\begin{aligned}W_{t_2+1} &= W_{t_2}+\sum_{i=1}^{N}\gamma_{t_2}(i)\\ &\leq B+(\phi-1)(W_{t_2-1}-W_{t_1})+N\\ &\leq B+(\phi-1)(B-W_{t_1})+N.\end{aligned}$$

▮

The next corollary follows immediately from Claim 4.1.

Corollary 4.2 *The maximal cumulative window size is bounded from above by ϕB, for $W_{t1} \geq N/(\phi-1)$.*

Since the number of connections is negligible in comparison with the window size, in the sequel we will ignore the additive term of N in the bound of the above claim.

The following claim bounds the length of a decrease sub-phase as a function of the initial cumulative window size, which implies a lower bound on the cumulative window size.

Claim 4.3 *The length of a decrease sub-phase $\mathcal{I}=[t_1,t_2]$ is at most $\lceil\sqrt{2\log_\psi c+1/4}-1/2\rceil$ if $W_{t_1} \leq cB$ for a constant $c>1$.*

Proof: Let us define $k=t_2-t_1+1$ to be the length of $\mathcal{I}$. During $\mathcal{I}$ the cumulative window W_t consequently decreases by a factor of $\psi(i)$, $i=1,\ldots,k$. Notice that the total decrease factor up to time t_2+1 is $\prod_{i=1}^{k}\psi(i)=\psi^{k(k+1)/2}$, that is $W_{t_2+1}=\frac{W_{t_1}}{\psi^{k(k+1)/2}}$. The claim follows since $W_{t_2+1}\leq B$ and $W_{t_1}\leq cB$. ▮

Corollary 4.2 and Claim 4.3 imply the next corollary.

Corollary 4.4 *The maximal length of a decrease sub-phase is $D(\phi,\psi)$ rounds.*

4.2 Convergence

The next theorem shows that $ACC(\phi,\psi)$ converges to a steady state and has a reasonably small stability constant for proper values of ϕ and ψ.

Theorem 4.5 *$ACC(\phi,\psi)$ converges to a steady state starting from an arbitrary state after*

$$M_R = D(\phi,\psi)(\log_\phi(B/N)+D(\phi,\psi))$$

rounds and has a stability constant of

$$C_R = C(\phi,\psi).$$

for well-behaved ϕ and ψ.

Proof: We first prove that the ACC algorithm converges to a steady state and then present a bound on the duration of the convergence period M_R. The proof of convergence is by induction on the phase number. We introduce the following notation. Let

$$f(n) = \begin{cases} D(\phi,\psi) - \min(n, (D(\phi,\psi) - 1)) & n \geq 0, \\ \infty & n = -1; \end{cases}$$

and let $g(x) = x(x+1)/2$.

Induction Hypothesis. For the n-th phase, the window size at the end of the increase and the decrease sub-phases is at most

$$B(1 + (\phi - 1)(1 - \frac{1}{\psi^{g(f(n-1))}}))$$

and at least $B/\psi^{g(f(n))}$, respectively.

Induction Basis $(n = 0)$**.** By Corollary 4.2, the maximal window size is bounded by ϕB. The corresponding decrease sub-phase continues at most $D(\phi,\psi)$ rounds according to Corollary 4.4. Hence, the window size at the end of the first decrease sub-phase cannot drop below $B/\phi^{g(D(\phi,\psi))}$.

Induction Step. Suppose that the induction hypothesis is satisfied for all $n' \leq n$ and let us prove that it is also satisfied for the $(n+1)$-th phase. By the induction hypothesis, the window size at the end of n-th phase is at least $B/\psi^{g(f(n))}$. Claim 4.1 implies that the window size at the end of the increase sub-phase is bounded by

$$B(1 + (\phi - 1)(1 - \frac{1}{\psi^{g(f(n))}})).$$

From Claim 4.3 and the fact that ϕ and ψ are well-behaved it follows that the number of rounds in the decrease sub-phase is at most $f(n+1)$. Thus, the window size at the end of the $(n+1)$-th phase is bounded from below by $B/\psi^{g(f(n+1))}$.

We have shown that after $D(\phi,\psi)$ phases the ACC algorithm converges to a steady state in which the window size changes between B/ψ and $B(1 + (\phi-1)(1 - 1/\psi))$, which establishes the stability constant.

Now we derive an upper bound on the length of the convergence period. The duration of an increase sub-phase is bounded by $O(\log_\phi(B/N))$ rounds due to the fact that there are N connections. Corollary 4.4 implies that the duration of a decrease sub-phase is at most $D(\phi,\psi)$ rounds, which yields the theorem. ▮

For $\phi = 1.4$ and $\psi = 1.1$, in a steady state the total transmission rate will vary between $0.9 \cdot R$ and $1.04 \cdot R$, which is a fairly good bandwidth utilization.

4.3 Fairness

Now we proceed to study fairness of the ACC algorithm. First we define so called "excess" window of a connection, that is the difference between the connection window and the fair share if the connection window is above the fair share, and zero otherwise. The cumulative excess window of the system is the sum of the excess windows of all individual connections.

Definition 4.1 *We define the excess window of the i-th connection at time t to be* $e_t(i) = \max(w_t(i) - W_t/|S_t|, 0)$. *We denote by* E_t *the* cumulative *excess window at time t, that is* $E_t = \sum_{i \in S_t} e_t(i)$.

We present an upper bound on the number of rounds needed for a connection to reach a window size that is at least the fair share minus the fairness constant. We concentrate on the dynamics of the cumulative excess window in the system and consider increase and decrease sub-phases separately.

The next claim shows that the cumulative excess window does not change during an increase sub-phase.

Claim 4.6 *For any increase sub-phase $\mathcal{I} = [t_1, t_2]$, $E_{t_1} = E_{t_2+1}$.*

Proof: Consider a round $t \in \mathcal{I}$. Notice that the windows of all connections are increased by the same term γ_t. We show that the cumulative excess window remains constant:

$$\begin{aligned}
E_{t+1} &= \sum_{i=1}^{N} e_{t+1}(i) = \sum_{i=1}^{N} \max(w_{t+1}(i) - W_{t+1}/N, 0) \\
&= \sum_{i=1}^{N} \max((w_t(i) + \gamma_t) - (W_t + \gamma_t N)/N, 0) \\
&= \sum_{i=1}^{N} \max(w_t(i) - W_t/N, 0) \\
&= E_t.
\end{aligned}$$

∎

The following claim demonstrates that the cumulative excess window is decreased by a factor of at least ψ during a decrease sub-phase.

Claim 4.7 *For any decrease sub-phase $\mathcal{I} = [t_1, t_2]$, $E_{t_2+1} \le E_{t_1}/\psi$.*

Proof: We argue that the cumulative excess window is decreased by a factor of ψ during the first round:

$$\begin{aligned}
E_{t_1+1} &= \sum_{i=1}^{N} e_{t_1+1}(i) = \sum_{i=1}^{N} \max(w_{t_1+1}(i) - W_{t_1+1}/N, 0) \\
&= \sum_{i=1}^{N} \max(w_{t_1}(i)/\psi - W_{t_1}/(\psi N), 0) \\
&= \frac{1}{\psi} \sum_{i=1}^{N} \max(w_{t_1}(i) - W_{t_1}/N, 0) \\
&= E_{t_1}/\psi.
\end{aligned}$$

∎

The next observation states that the system is guaranteed to get to a fair operating point when the cumulative excess window drops below C_F.

Observation 2 *If $E_t \le C_F$ then for all $i \in S_t$, $w_t(i) - W_t/|S_t| \le C_F$.*

Now we are ready to prove the theorem that establishes fairness of the ACC algorithm.

Theorem 4.8 *$ACC(\phi, \psi)$ converges to the fair share starting from an arbitrary state after*

$$M_F = \log_\psi(B/C_F)(\log_\phi(B/N) + D(\phi,\psi))$$

rounds and has a fairness constant of C_F.

Proof: By Claim 4.6, the cumulative excess window does not change during an increase sub-phase. Moreover, according to Claim 4.7 each decrease sub-phase reduces the cumulative excess window by a factor of ψ. Thus, after $\log_\psi(B/C_F)$ phases the total excess window will fall below C_F. The theorem follows from Observation 2 and the fact that the duration of a phase is bounded by $\log_\phi(B/N) + D(\phi,\psi)$ rounds. ▮

5 Long Connections

In this section we study the bandwidth utilization of the ACC algorithm when each connection remains in the system for al least $L \gg M_F$ rounds, where M_F is the convergence-to-fairness period of Theorem 4.8. We assume that the maximal number of concurrent connections is bounded by N while the set of connections in the system can change dynamically. We introduce the notion of utilization ratio analogous to the stability constant for the dynamic case. Then we derive the utilization ratio of the ACC algorithm.

5.1 Utilization Ratio

The goal of a congestion control algorithm is to maximize the number of packets transmitted so as to minimize the packet loss. Next we define a cost function associated with dropped packets.

Definition 5.1 *In case $F_t = 1$, the* overshooting cost *at time t is defined as $L_t = \alpha(W_t - B)$, where $\alpha \geq 1$ is a parameter that reflects the cost of losing packets* .*

Next we define the notion of so called "effective" gain of a congestion control algorithm that takes into account dropped packets.

Definition 5.2 *We define the* effective gain *of the congestion control algorithm A as:*

$$G_t^A = \begin{cases} W_t & F_t = 0, \\ W_t - L_t & F_t = 1. \end{cases}$$

Notice that in a steady state during every round the effective gain of a congestion control algorithm with a stability constant C_R is at least $B(1 - \alpha C_R)$.

We define an *input sequence* as a sequence of connection arrivals and departures. Given an input sequence σ, we define the value function of the congestion control algorithm A as:

$$V^A(\sigma) = \sum_{t=t_s}^{t_f} G_t^A,$$

where t_s is the arrival time of the first connection and t_f is the departure time of the last connection. Notice that the value function may even be negative for some policies. However, as we will demonstrate, the value of our algorithm is always positive.

Now we define the utilization ratio of a congestion control algorithm A as the worst-case ratio between the value obtained by A and the value obtained by an optimal algorithm OPT taken over all input sequences. If

* Our analysis can be extended to deal with other cost functions. For example, a natural cost function may be a linear function with an additive constant term, that is $L_t = \beta + \alpha(W_t - B)$.

the worst case ratio is negative, then we define the utilization ratio as zero. (Notice that a trivial policy that transmits no packets has a utilization ratio equal to zero.)

Definition 5.3 *The* utilization ratio U_A *of the congestion control algorithm A is the minimum non-negative ratio, over all input sequences σ, between the value of A and the value obtained by an optimal algorithm OPT, that is* $\max\{\min_\sigma(V^A(\sigma)/V^{OPT}(\sigma)), 0\}$.

Observe that in our model the optimal value of the input sequence σ equals to the sum of the bandwidth-delay product over all periods in which at least one connection in the system is active, i.e.,

$$V^{OPT}(\sigma) = B \cdot |\{t \,:\, |S_t| > 0\}|.$$

That is due to the fact that even a single connection in the system can utilize the bandwidth completely.

5.2 Utilization Ratio of the ACC Algorithm

We show that the utilization ratio of $ACC(\phi,\psi)$ is $1 - O(\frac{\ln N \cdot M_R}{L}) - \alpha C(\phi,\psi)$, where M_R is the convergence-to-utilization period of Theorem 4.5. (Recall that α is a parameter that reflects the cost of losing packets.) We note that the established bound depends *logarithmically* rather than *linearly* on the number of connections.

Theorem 5.1 *The utilization ratio of $ACC(\phi,\psi)$ is at least*

$$U_{ACC(\phi,\psi)} = 1 - \tfrac{(\alpha\cdot(\phi-1)+1)(2\ln(N+1)+1)\cdot M_R}{L} - \alpha C(\phi,\psi)$$

assuming that all connections have duration of at least $L \geq 2M_F$ rounds.

Proof: We divide the schedule of the ACC algorithm into *periods*. A period starts upon arrival of a connection to an empty system and terminates when the system is empty again. Each period is further partitioned into *phases* of length L. We assume that such a partition is possible.

By Theorem 4.5, the ACC algorithm has a stability constant of $C(\phi,\psi)$. Thus, in a steady state the effective gain of $ACC(\phi,\psi)$ during a phase is at least $B \cdot L - \alpha \cdot C(\phi,\psi) \cdot B \cdot L$. In addition, some extra cost can be paid in case of overshooting or some bandwidth may be lost (with respect to OPT) in case of underutilization. We call the former the *overshooting cost* and the latter the *opportunity cost*. We analyze separately two "bad" events leading to a transient state that may incur additional overshooting or opportunity cost, namely connection arrival and departure. According to Theorem 4.5, a bad event can affect the system for at most M_R rounds. We will show that during a phase the total overshooting and opportunity cost is bounded by

$$(\alpha \cdot (\phi - 1) + 1)(2\ln(N+1) + 1) \cdot B \cdot M_R.$$

That would imply that the utilization ratio of the ACC algorithm is at least

$$1 - \frac{(\alpha \cdot (\phi - 1) + 1)(2\ln(N+1) + 1) \cdot M_R}{L} - \alpha C(\phi,\psi).$$

(Note that in every phase OPT sends $B \cdot L$ packets.)

Connection arrival. Observe that when a new connection arrives to the system, it starts with the initial window size of 1. The next observation is based on this fact and follows from the analysis of Theorem 4.5.

Observation 3 *A connection arriving to a non-empty system does not affect the amplitude of changes of the cumulative window size.*

Now consider a connection arrival. If the system is empty, Corollary 4.2 implies that the overshooting cost is bounded by $\alpha(\phi - 1) \cdot B \cdot M_R$. Clearly, the opportunity cost is bounded by $B \cdot M_R$. Observe that a connection can enter an empty system only once during the first phase of a period. Otherwise, if the system is in a transient or in a steady state, then by Observation 3 an arriving connection incurs no additional overshooting or opportunity cost. Therefore, the total cost incurred by the first arriving connection during a phase is bounded by $(\alpha(\phi - 1) + 1) \cdot B \cdot M_R$.

Connection departure. Consider a departing connection. If it is the last active connection, then the phase ends and the incurred overshooting and opportunity cost are zero. Otherwise, if there remain some active connections, the system may enter a transient state. Suppose that connection i departs at time t_d. By Claim 4.1, the overshooting cost due to the created instability is at most $\alpha \cdot (\phi - 1) \cdot w_{t_d}(i) \cdot M_R$. We also have that the opportunity cost is bounded by $w_{t_d}(i) \cdot M_R$. Hence, the total cost incurred by the departure of connection i is bounded by $(\alpha(\phi - 1) + 1) \cdot w_{t_d}(i) \cdot M_R$. Next we derive an upper bound on the total overshooting and opportunity cost incurred by *all* departing connections during a phase.

Let us consider the k-th phase $\mathcal{I}_k = [t, t + L)$. Notice that the connections leaving the system throughout $\mathcal{I}_k$ should have arrived during $\mathcal{I}_{k-1}$ or earlier. Let $S^d \subseteq S_t$ be the set of connections departing during the k-th phase and let $S' \subseteq S^d$ be the subset of these connections that arrived before time $t - M_F$. We denote $N' = |S'|$. Analysis of Theorem 4.8 implies that all connections in S' at time t have window sizes that differ by at most 1, which are bounded by B/N'. Observe also that all connections in S' always update their windows synchronously. Therefore, the window size of the j-th departing connection from S' is at most B/j. Since $N' \leq N$, the total cost incurred by connections in S' is bounded by

$$\sum_{j=1}^{N}((\alpha \cdot (\phi - 1) + 1) \cdot \frac{B}{j} \cdot M_R) < (\alpha \cdot (\phi - 1) + 1) \cdot \ln(N + 1) \cdot B \cdot M_R.$$

It remains to bound the total cost incurred by connections in $S'' = S^d \setminus S'$. The earliest possible departure time for these connections is $t + L - M_F$. We have that similarly to the previous case the total cost incurred by these connections is also bounded by $(\alpha \cdot (\phi - 1) + 1) \cdot \ln(N + 1) \cdot B \cdot M_R$ since by our assumption $L - M_F \geq M_F$. The theorem follows. ∎

6 Interaction with TCP

In this section we study the case in which an ACC connection competes with a regular TCP Reno connection. We analyze the relative fairness of the ACC algorithm. We assume that the TCP Reno connection is in the *congestion avoidance phase* (steady state of TCP) having the increase term of 1 and the decrease factor of 2. As we show in the next theorem, an ACC connection could consume a significantly larger fraction of the available bandwidth compared to the regular TCP Reno connection due to the fact that its window grows exponentially while the TCP window grows linearly.

Theorem 6.1 *The fairness constant of a system with an $ACC(\phi, \psi)$ connection and a TCP Reno connection is at least $C_F \geq B - 2\log B + 1$ for the proper selection of ϕ and ψ values.*

Proof: Consider the following scenario in which the ACC connection and the TCP connection are synchronized in their increase/decrease sub-phases. Each increase sub-phase finishes when the maximum unfairness is reached. Then the ACC connection receives a negative feedback and one packet of the TCP connection is lost. The following decrease sub-phase continues just one round. Let $B - \log B + 1$ and $\log B$ be the window size of the ACC connection and that of the TCP connection at the end of the increase sub-phase, respectively.

Notice that after the decrease sub-phase, it takes $\log_\phi((B - \log B + 1)(1 - 1/\psi))$ rounds to the ACC connection to increase its window from $(B - \log B + 1)/\psi$ up to $B - \log B + 1$ and it takes $(\log B)/2$ rounds to the TCP connection to increase it window from $(\log B)/2$ up to $\log B$. If we have that

$$\log_\phi((B - \log B + 1)(1 - 1/\psi)) = (\log B)/2,$$

then the same sequence of window sizes of both connections is repeated in all increase/decrease sub-phases. Thus, for the proper selection of ϕ and ψ values satisfying the equation above, the fairness constant is at least $B - 2\log B + 1$. ∎

7 Concluding Remarks

In this paper we propose a novel end-to-end congestion control algorithm, based on *adaptive* AIMD scheme. The proposed algorithm attains almost optimal utilization in a steady state providing fairness between competing connections and at the same time responds quickly on bandwidth changes. Even though our algorithm cannot be implemented without additional work, we believe that the proposed scheme may suggest new directions for further improvement of the current TCP congestion control.

Some challenging future directions are modifying our algorithm to make it TCP-compatible, studying more complex network topologies and performing simulations with real traffic to find the optimal ACC parameters. Possible extensions of our model are: connections receiving distinct feedbacks, connections having different RTT's and a fluid model rather than discrete rounds.

References

[1] S. Arora and W. Brinkman, "A Randomized Online Algorithm for Bandwidth Utilization," *Proceedings of The Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 535-539, 2002.

[2] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms," *Proceedings of IEEE INFOCOM 2001*, pp. 631-640.

[3] D. Bansal, H. Balakrishnan, S. Floyd and Scott Shenker, "Dynamic Behavior of Slowly Responsive Congestion Control Algorithms," *Proceedings of ACM SIGCOMM 2001*, pp. 263-274.

[4] A. Bar-Noy, Y. Mansour and B. Schieber, "Competitive Dynamic Bandwidth Allocation," *Proceedings of The Seventeenth Annual ACM symposium on Principles of Distributed Computing*, pp. 31-39, 1998.

[5] D. M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, Vol. 17, No. 1, pp. 1-14, June 1989.

[6] 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, Aug. 1999.

[7] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-Based Congestion Control for Unicast Applications,” *Proceedings of ACM SIGCOMM 2000*, pp. 43-56.

[8] S. Floyd, S. Ratnasamy, and S. Shenker, “Modifying TCP’s Congestion Control for High Speeds,” *Internet draft*, May 2002.

[9] N. Garg and N. E. Young, “On-Line End-to-End Congestion Control,” *Proceedings of The 43rd Symposium on Foundations of Computer Science*, pp. 303-312, 2002.

[10] V. Jacobson, “Congestion Avoidance and Control,” *Proceedings of ACM SIGCOMM 1988*, pp. 314-329.

[11] V. Jacobson, “Modified TCP congestion avoidance algorithm,” *end2end-interest mailing list*, Apr. 30, 1990.

[12] S. Jin, L. Guo, I. Matta and A. Bestavros, “TCP-friendly SIMD Congestion Control and Its Convergence Behavior,” *Proceedings of The 9th IEEE International Conference on Network Protocols*, 2001.

[13] R. M. Karp, E. Koutsoupias, C. H. Papadimitriou and S. Shenker “Optimization problems in congestion control,” *Proceedings of The 41st Symposium on Foundations of Computer Science*, pp. 66-74, 2000.

[14] D. Katabi, M. Handley, and C. Rohrs, “Congestion control for high bandwidth-delay product networks,” *Proceedings of ACM Sigcomm 2002*, pp. 89-102.

[15] F. P. Kelly, “Mathematical modelling of the Internet,” *Proceedings of The 4th International Congress on Industrial and Applied Mathematics*, pp. 685-702, 1999.

[16] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, Vol. 49, pp. 237-252, 1998.

[17] A. Kumar, “Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link,” *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, pp. 485-498, Aug.1998.

[18] S. H. Low, “A duality model of TCP flow controls,” *Proceedings of The 2000 ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*.

[19] S. H. Low, F. Paganini and J. C. Doyle, “Internet Congestion Control,” *IEEE Control Systems Magazine*, Vol. 22, No. 1, pp. 28-43, Feb. 2002.

[20] 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*, Vol. 3, No. 3, pp. 336-350, June 1997.

[21] S. H. Low, L. Peterson, and L. Wang, “Understanding Vegas: a duality model,” *Journal of ACM*, Vol. 49, No. 2, pp. 207-235, March 2002.

[22] J. Mahdavi and S. Floyd, “TCP-Friendly Unicast Rate-Based Flow Control,” *Available from http://www.psc.edu/ networking/papers/tcp_friendly.html*, Jan. 1997.

[23] A. Maor and Y. Mansour, "AdaVegas: Adaptive Control for TCP Vegas," *Proceedings of IEEE Globecom 2003*.

[24] M. Mathis, J. Semske, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, Vol. 27, No. 3, pp. 67-82, July 1997.

[25] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," *Proceedings of ACM SIGCOMM 1998*, pp. 303-314.

[26] A. Piccolboni and C. Schindelhauer, "Discrete Prediction Games with arbitrary Feedback and Loss," *Proceedings of The 14th Annual Conference on Computational Learning Theory*, pp. 208-223, 2001.

[27] S. Shenker, "A Theoretical Analysis of Feedback Flow Control," *Proceedings of ACM SIGCOMM 1990*, pp. 156-165.

[28] W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *RFC2001*.

[29] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," *Proceedings of The 2000 International Conference on Network Protocols*, pp. 187-198.