Extended Analysis of Binary Adjustment Algorithms

Sergey Gorinsky Harrick Vin

Technical Report TR2002-39
Department of Computer Sciences
The University of Texas at Austin
Taylor Hall 2.124, Austin, TX 78712, USA
{*gorinsky,vin*}*@cs.utexas.edu*

August 11, 2002

Abstract – **Congestion control in the Internet relies on binary adjustment algorithms. For example, Transmission Control Protocol (TCP) in its congestion avoidance mode behaves similarly to Additive-Increase Multiplicative-Decrease (AIMD) algorithm. Chiu and Jain offer a theoretical justification for choosing AIMD: among stable linear algorithms, AIMD ensures the quickest convergence to maxmin-fair states. Whereas Chiu-Jain model rests on a well-known unrealistic assumption of uniform feedback, more precise analytic characterizations of TCP behavior are developed and validated. In particular, the advanced theory and experiments agree that TCP congestion control does not converge to maxmin fairness. However, despite the recent progress in TCP feedback modeling, it is still common to use Chiu-Jain model for comparison of binary adjustment algorithms. This paper argues against such practice. We provide evidence that due to the incorrect assumption of uniform feedback, Chiu-Jain model is not suitable for trustworthy conclusions about properties of an adjustment algorithm. We emphasize that until algorithms are analyzed with a more realistic feedback model, optimal choice of a binary adjustment algorithm will remain an open problem.**

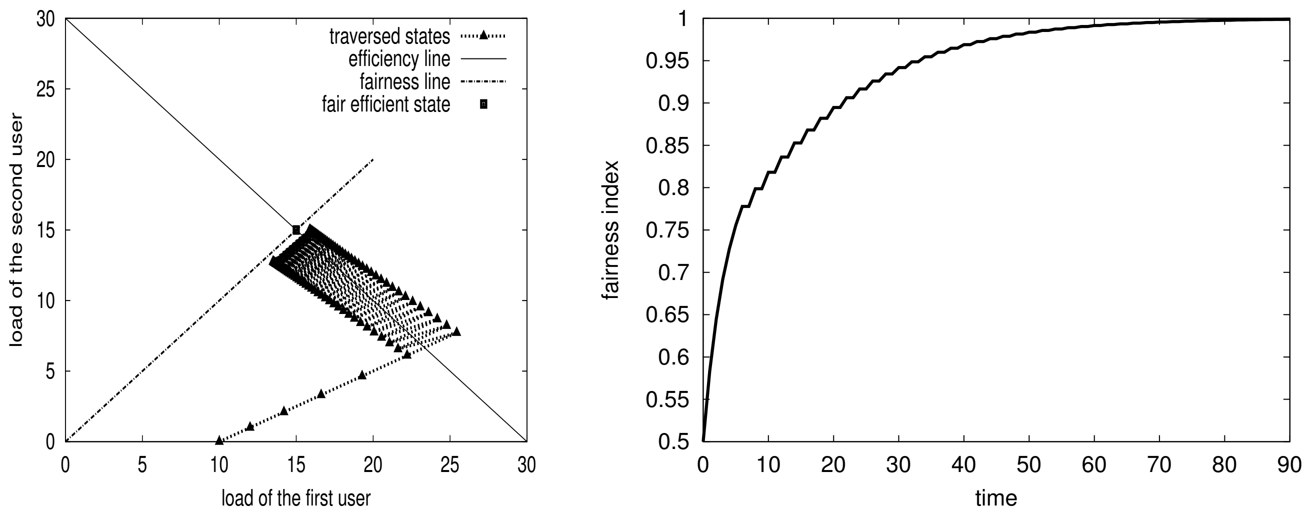
I. INTRODUCTION

In such a complex distributed system as the Internet, it is all but impossible to provide each user with an exact up-to-date value for its fair and efficient load on the network. Instead, users control congestion with *binary adjustment algorithms*: a user adjusts its load in response to binary signals that indicate whether the user must decrease or can increase the load. For example, Transmission Control Protocol (TCP) exercises binary congestion control – the TCP sender steps up its transmission after receiving a new acknowledgment; the sender reduces its load upon a retransmission timeout or after receiving three duplicate acknowledgments [1], [5]. Until the first indication of congestion, each TCP connection raises its load in a manner resembling the Multiplicative-Increase (MI) algorithm [4]. This reliance on MI is supposed to enable quick convergence to efficient states. Once efficiency is achieved, the TCP connection switches to the congestion avoidance mode and adjusts the load similarly to Additive-Increase Multiplicative-Decrease (AIMD) algorithm [4]. The choice of AIMD is supposed to provide stability, i.e., convergence to fair efficient states.

Chiu and Jain provide a theoretical justification for favoring AIMD: according to their analysis of linear adjustment algorithms for a simple feedback model, AIMD yields the quickest convergence to maxmin-fair states [4]. For simplicity, Chiu-Jain model assumes *uniform feedback* – all users receive identical feedback. In reality however, the probability to receive a congestion indication is higher for the user with a larger load. Subsequent analytical studies of TCP congestion control represent feedback more realistically and predict the transmission rate for a TCP connection more accurately [2], [10], [12], [13]. Furthermore, experiments and more realistic models with *proportional negative feedback* agree that bandwidth allocation under TCP does not converge to maxmin fairness [13], [16].

While reliance of TCP on AIMD does not attain the original goal of convergence to maxmin fairness, it is logical to reexamine the presumed superiority of AIMD over alternative algorithms. In fact, new algorithms have been proposed to improve upon various features of AIMD congestion control [3], [7], [8], [9]. However, despite the recent advances in TCP feedback modeling, it is still common to use Chiu-Jain model for comparison of binary adjustment algorithms. This paper argues against such practice. We provide evidence that due to the incorrect assumption of uniform feedback, Chiu-Jain model is not suitable for trustworthy conclusions about properties of an adjustment algorithm. In particular, we show that albeit the scalable MIMD (Multiplicative-Increase Multiplicative-Decrease) algorithm is not stable under uniform feedback, MIMD does converge to fair states under the more realistic assumption of proportional negative feedback. Our findings suggest that until algorithms are analyzed with a realistic feedback model, optimal choice of a binary adjustment algorithm will remain an open problem.

The rest of our paper is structured as follows. Section II discusses the issue of stability. Section III examines the speed of convergence to fair states. Section IV studies the impact of different RTTs. Finally, Section V sums up our conclusions.



(a) state diagram (b) fairness index

Figure 1: Stability of LIMD in Chiu-Jain model where $a_I = 1$, $b_I = 1.1$, $b_D = 0.85$, $X_{goal} = 30$, $n = 2$, $x_1(0) = 10$, and $x_2(0) = 0$.

II. Stability

In Chiu-Jain model, the network is a single bottleneck link shared by cooperative users [4]. The model assumes that all users have the same RTT and adjust their loads simultaneously. Consequently, the model employs a discrete timescale where every instant t corresponds to the moment when each user i adjusts its load to $x_i(t)$. The network provides the users with a binary feedback $y(t)$ which indicates whether the total load $X(t-1)$ after the previous adjustment exceeds an optimal value X_{goal}:

$$y(t) = \begin{cases} 1 & \text{if } X(t-1) > X_{goal}, \\ 0 & \text{if } X(t-1) \leq X_{goal} \end{cases} \quad (1)$$

where $X(t)$ is the combined load of all n users at time t:

$$X(t) = \sum_{i=1}^{n} x_i(t). \quad (2)$$

Note that the model assumes *uniform feedback* – all the users receive the same feedback $y(t)$. The users have no access to other external information including n, X_{goal}, or $X(t-1)$.

Chiu and Jain perform a static analysis for the following class of linear adjustment algorithms:

$$\forall i \;\; x_i(t) = \begin{cases} a_I + b_I x_i(t-1) & \text{if } y(t) = 0, \\ a_D + b_D x_i(t-1) & \text{if } y(t) = 1 \end{cases} \quad (3)$$

where a_I, b_I, a_D, and b_D are real constants.

The criteria for choosing an appropriate algorithm include its *stability*: from any initial state, load $x_i(t)$ of each user i must converge towards the efficient and fair amount of X_{goal}/n. To quantify fairness, Chiu and Jain use index $F(t)$ from [6]:

$$F(t) = \frac{(X(t))^2}{n \sum_{i=1}^{n} (x_i(t))^2} \quad (4)$$

According to Chiu-Jain analysis, a linear algorithm for binary adjustments should belong to the following LIMD (Linear-Increase Multiplicative-Decrease) class in order to converge to fair efficient states:

PROPOSITION 1. *To make a linear adjustment algorithm stable, its decrease policy* should *be multiplicative, and its increase policy* should *have an additive component and* may *have a multiplicative component:*

$$a_I > 0, \;\; b_I \geq 1, \;\; a_D = 0, \;\; 0 \leq b_D < 1. \quad (5)$$

After convergence to efficiency, the total load under LIMD oscillates within the following tight bounds:

$$b_D X_{goal} < X(t) \leq n a_I + b_I X_{goal}. \quad (6)$$

Proposition 1 has an intuitive explanation: multiplicative adjustments (both decreases and increases) do not affect the fairness index while additive increases enhance it. Consequently, LIMD algorithms monotonically raise the fairness index to its optimal value of 1.

EXAMPLE 1. *Figure 1 illustrates the behavior of the* LIMD *algorithm with $a_I = 1$, $b_I = 1.1$, and $b_D = 0.85$ in the network that has the optimal load of $X_{goal} = 30$ and serves two users with the initial loads of $x_1(0) = 10$ and $x_2(0) = 0$. The state diagram in Figure 1a represents concurrent loads of the users as points in a 2-dimensional space: the efficiency line corresponds to the efficient states where $X(t) = X_{goal}$, and the fairness line denotes the fair states where $x_1(t) = x_2(t)$. The*

LIMD *algorithm converges to the intersection of these two lines, i.e., to the fair efficient state characterized by $x_1(t) = x_2(t) = X_{goal}/2$. Figure 1b tracks the fairness index for the traversed states.*

In Chiu-Jain model, AIAD (Additive-Increase Additive-Decrease) and MIMD (Multiplicative-Increase Multiplicative-Decrease) algorithms do not converge to fair states. Their instability however is due to the assumption of uniform feedback: since such feedback does not contain information about fairness, convergence to fair states requires different functions for increase and decrease.

Under the more realistic assumption of proportional negative feedback, a user with a larger load receives congestion indications more frequently. Thus, proportional negative feedback reflects fairness and thereby facilitates convergence to fair states. In comparison to uniform feedback, proportional negative feedback yields a wider class of stable adjustment algorithms. For example, Venkitaraman et al argue that AIAD converges to fair states under proportional negative feedback [15]. Below, we show that MIMD has the same property.

Consider a synchronous MIMD-controlled network with proportional negative feedback. Let r_i and r_j be the number of congestion indications received respectively by users i and j during interval $[0,t]$. At time t, loads of the users become:

$$x_i(t) = b_D^{r_i} b_I^{t-r_i} x_i(0) \text{ and } x_j(t) = b_D^{r_j} b_I^{t-r_j} x_j(0). \quad (7)$$

If $x_i(t) > x_j(t)$ during interval $[0,t]$, then due to proportional negative feedback we have $r_i > r_j$ and:

$$\begin{aligned} & \frac{x_i(t)}{x_j(t)} \\ = \quad & \{ (7) \} \\ & \frac{b_D^{r_i} b_I^{t-r_i} x_i(0)}{b_D^{r_j} b_I^{t-r_j} x_j(0)} \\ = \quad & \\ & \left(\frac{b_D}{b_I}\right)^{r_i - r_j} \cdot \frac{x_i(0)}{x_j(0)} \\ < \quad & \{ 0 < b_D < 1,\ b_I > 1, \text{ and } r_i > r_j \} \\ & \frac{x_i(0)}{x_j(0)}. \end{aligned}$$

Therefore, the MIMD-controlled network converges toward the fair states where $x_i(t) = x_j(t)$.

Stability of MIMD under proportional negative feedback makes this algorithm a viable alternative to AIMD because AIMD and other algorithms with additive components are not scalable. As (6) shows for Chiu-Jain model, oscillations of the total load after convergence under AIMD grow linearly in size as the number of users increases. Morris confirms experimentally that the average loss rate grows linearly with the number of competing TCP connections and thereby worsens TCP performance [11]. On the other hand, MIMD congestion control is scalable since the size of the total load oscillations after convergence under MIMD does not depend on the number of users. Consequently, it is possible to design such an MIMD-controlled network with ECN-style packet marking [14] that traffic in steady loaded states always oscillates within the buffer of a bottleneck link and keeps the link fully utilized without losing packets.

III. Speed of Convergence to Fair States

Chiu and Jain show that *smoothness* (measured as the size of oscillations in the total load) and *responsiveness* (measured as time of convergence to efficient states) conflict: any attempt to change a_I, b_I, or b_D to improve smoothness worsens responsiveness, and vice versa. Conceding this trade-off between responsiveness and smoothness, [4] examines the *speed of convergence to fair states*. After proving that a single increase under LIMD boosts the fairness index the most when $b_I = 1$, [4] concludes with:

PROPOSITION 2. *To provide the quickest convergence to fair states, an* LIMD *algorithm should have an additive increase policy and a multiplicative decrease policy:*

$$a_I > 0, \;\; b_I = 1, \;\; 0 \le b_D < 1. \quad (8)$$

As the following example shows, Proposition 2 is not completely correct. After a sequence of adjustments, LIMD with $b_I > 1$ can raise the fairness index higher than AIMD.

EXAMPLE 2. *Let the network have $X_{goal} = 10$ and two users with the initial loads of $x_1(0) = 7$ and $x_2(0) = 0$. Consider* AIMD *with $a_I = 1.1$, $b_D = 0.5$ and* LIMD *with $a_I = 1$, $b_I = 1.02$, $b_D = 0.5$. According to (6), these algorithms have the same smoothness. Due to the larger a_I/b_I ratio,* AIMD *raises the fairness index higher than* LIMD *after the first adjustment (as Chiu and Jain prove). However as Figure 2 shows,* LIMD *raises the fairness index higher than* AIMD *after the sequence of six adjustments.*

When LIMD improves the fairness index quicker, either the gained advantage is not significant or AIMD reaches the same level of fairness after a small number of additional adjustments. Thus, despite the occasionally suboptimal speed, AIMD does yield the quickest, or at least compatible, overall convergence to fair states in Chiu-Jain model.

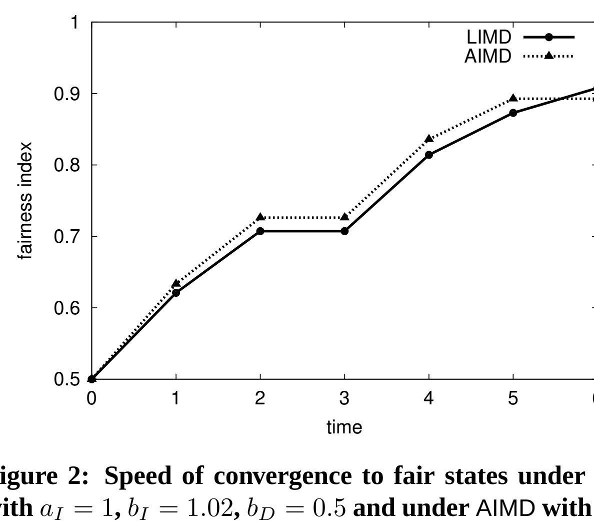


Figure 2: Speed of convergence to fair states under LIMD with $a_I = 1$, $b_I = 1.02$, $b_D = 0.5$ and under AIMD with $a_I = 1.1$, $b_D = 0.5$ in Chiu-Jain model where $n = 2$, $X_{goal} = 10$, $x_1(0) = 7$, and $x_2(0) = 0$.

As Section II shows, the unrealistic assumption of uniform feedback affects conclusions about stability. Let us now examine whether Chiu-Jain model yields reliable conclusions about the speed of convergence to fair states. Consider two TCP connections that have RTT of 100 ms, use 500 B packets and share a 100 Mb/s bottleneck link. Let the second connection start when the first one utilizes the entire bottleneck. Then, the new connection surpasses its slow-start phase and switches immediately to the congestion avoidance mode. Chiu-Jain model represents this scenario as an AIMD-controlled network with two users, $a_I = 1$, $b_D = 0.5$, $X_{goal} = 2500$, $x_1(0) = X_{goal}$, and $x_2(0) = 0$. In this model, the second user reaches its fair share of $X_{goal}/2$ only after 27546 adjustments which translates into 45 minutes of real time.

However according to experiments with TCP in the same settings, the new TCP connection receives congestion indications less frequently and reaches its fair bandwidth share substantially quicker than predicted by Chiu-Jain model with its uniform feedback. Thus, proportionality of negative feedback accelerates the convergence to fair states more significantly than the choice of AIMD over other LIMD algorithms.

IV. Different Round-Trip Times

Since bandwidth allocation under TCP depends on round-trip times [13], [16], it is interesting to examine a version of Chiu-Jain model where users can have different RTTs. Our asynchronous extension adds two parameters d_i and f_i to Chiu-Jain model: d_i represents delay from user i to the bottleneck link while f_i denotes delay from the bottleneck link to the user. Thus, RTT for user i equals $d_i + f_i$.

Taking the feedback delays into account, the asynchronous model redefines the total load on the network at time t as:

$$X(t) = \sum_{i=1}^{n} x_i(t - d_i) \tag{9}$$

where $x_i(t - d_i)$ refers to the load of user i at time $t - d_i$.

Similarly to the original model, the total load alone determines feedback at time t:

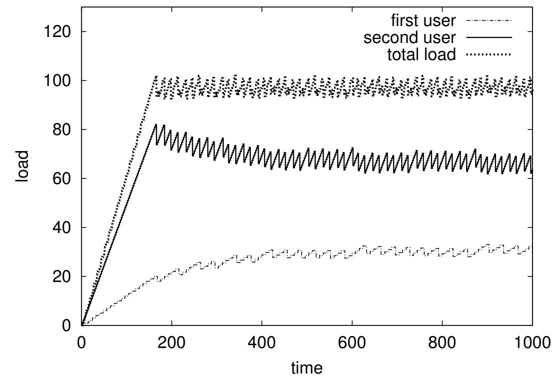
$$y(t) = \begin{cases} 1 & \text{if } X(t) > X_{goal}, \\ 0 & \text{if } X(t) \leq X_{goal}. \end{cases} \tag{10}$$

Let us now examine the network behavior when each user employs AIMD to adjust its load once per its RTT:

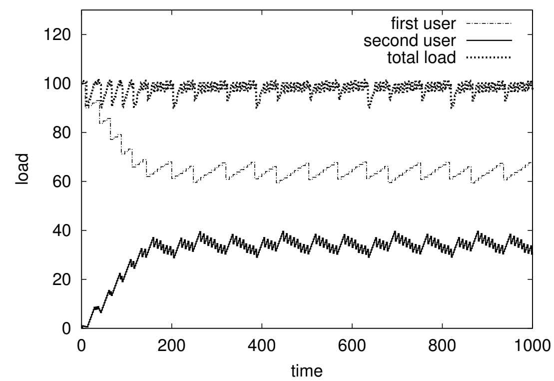
$$\forall i \; x_i(t) = \begin{cases} a_I + x_i(t - d_i - f_i) & \text{if } y(t - f_i) = 0, \\ b_D x_i(t - d_i - f_i) & \text{if } y(t - f_i) = 1. \end{cases} \tag{11}$$

EXAMPLE 3. *Consider an AIMD-controlled network with two users, $a_I = 1$, $b_D = 0.9$, $X_{goal} = 100$, and different RTTs. The first user has RTT of 8 (formed by d_1 = 4 and $f_1 = 4$). RTT for the second user is four times shorter and equals 2 (consisting of d_2 = 1 and $f_2 = 1$). Figure 3 presents the network behavior for two sets of initial conditions.*

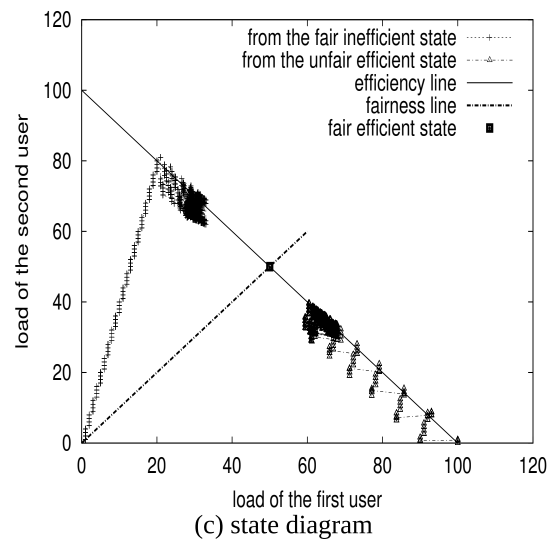
When the users have the initial loads of $x_1(0) = 0$ and $x_2(0) = 0$, the network converges from this fair inefficient state to steady states where the second user imposes a larger load than the first user. If the network starts at the unfair efficient state with the initial loads of $x_1(0) = 100$ and $x_2(0) = 0$, the relationship is reverse: after convergence to steady states, the second user settles at a smaller load than the first user.



(a) convergence from the fair inefficient state



(b) convergence from the unfair efficient state



(c) state diagram

Figure 3: AIMD behavior in the asynchronous version of Chiu-Jain model where $a_I = 1$, $b_D = 0.9$, $n = 2$, $X_{goal} = 100$, d_1 = 4, $f_1 = 4$, d_2 = 1, and $f_2 = 1$.

In addition to the lack of convergence to maxmin fairness, the above example illustrates the following two properties of AIMD in this asynchronous model:

1) AIMD is sensitive to initial conditions and has multiple attractors.
2) After convergence under AIMD to steady states, users with *longer* RTTs can maintain *larger* loads.

These properties contradict both Chiu-Jain model and more realistic models with proportional negative feedback. To understand the reason for the contradictions, let us return to Example 3 and compare frequencies of negative feedback for different users after the users converge to steady states. In the scenario when the first user (i.e., the user with the longer RTT) receives a smaller steady load, 24% of feedback to this user is negative whereas only 13% of feedback to the second user indicates congestion. However, when the first user acquires a larger steady load, the frequencies of negative feedback are reverse, 13% and 22% for the first and second users respectively.

In contrast, users in Chiu-Jain model always receive negative feedback with the same frequency. In models with proportional negative feedback, users with larger load always receive congestion indications more frequently. In both cases, AIMD has a single attractor.

The above discussion not only shows that the asynchronous version of Chiu-Jain model is unrealistic but also pinpoints the source of the flaws – incorrect frequencies of negative feedback to different users. Thus, this section reinforces our position that a realistic model for negative feedback is essential for trustworthy conclusions about properties of binary adjustment algorithms.

V. Conclusion

In this paper, we revisited the issue of choosing a binary adjustment algorithm for congestion control. Chiu and Jain pioneered analysis of this problem and offered a justification for favoring AIMD: among stable linear algorithms, AIMD ensures the quickest convergence to maxmin-fair states. Chiu-Jain model however made a simplifying assumption of uniform feedback for all users. In reality, the user with a larger load has a higher probability to receive a congestion indication. Subsequent analyses of TCP congestion control represented feedback more realistically and predicted the transmission rate for a TCP connection more accurately. Furthermore, the advanced theory and experiments agreed that reliance on AIMD does not enable TCP to converge to maxmin fairness.

Despite the recent progress in TCP feedback modeling, it is still common to use Chiu-Jain model for comparison of binary adjustment algorithms. This paper argued against such practice. We provided evidence that due to the incorrect assumption of uniform feedback, Chiu-Jain model is not suitable for trustworthy conclusions about properties of an adjustment algorithm.

First, we showed that although MIMD is not stable in Chiu-Jain model, MIMD does converge to fair states under the more realistic assumption of proportional negative feedback. This finding is significant because in contrast to AIMD, MIMD congestion control is scalable: the size of the total load oscillations under MIMD in steady states does not depend on the number of users. Second, we observed that proportionality of negative feedback accelerated convergence to fairness under LIMD more significantly than the choice of AIMD over other LIMD algorithms. Third, after showing that AIMD has multiple unfair attractors in an asynchronous version of Chiu-Jain model, we pinpointed the incorrect frequencies of negative feedback in this asynchronous model as the source of the contradiction with stability of AIMD.

The aggregate of the above arguments leads us to the conclusion that until algorithms are analyzed with a more realistic feedback model, optimal choice of a binary adjustment algorithm will remain an open problem.

References

[1] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, April 1999.

[2] E. Altman, K. Avrachenkov, and C. Barakat. A Stochastic Model of TCP/IP with Stationary Random Loss. In *Proceedings ACM SIGCOMM 2000*, August 2000.

[3] D. Bansal and H. Balakrishnan. Binomial Congestion Control Algorithms. In *Proceedings IEEE INFOCOM 2001*, April 2001.

[4] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, 17(1):1–14, June 1989.

[5] V. Jacobson. Congestion Avoidance and Control. In *Proceedings ACM SIGCOMM'88*, August 1988.

[6] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Technical Report TR-301, DEC, September 1984.

[7] S. Jin, L. Guo, I. Matta, and A. Bestavros. TCP-friendly SIMD Congestion Control and Its Convergence Behavior. In *Proceedings IEEE ICNP 2001*, November 2001.

[8] A. Lahanas and V. Tsaoussidis. Additive Increase Multiplicative Decrease - Fast Convergence (AIMD-FC). In *Proceedings IEEE Networks 2002*, August 2002.

[9] D. Loguinov and H. Radha. Increase-Decrease Congestion Control for Real-time Streaming: Scalability. In *Proceedings IEEE INFOCOM 2002*, June 2002.

[10] J. Mahdavi and S. Floyd. TCP-Friendly Unicast Rate-Based Flow Control. End2end-interest mailing list, January 1997.

[11] R. Morris. Scalable TCP Congestion Control. In *Proceedings IEEE INFOCOM 2000*, March 2000.

[12] A. Mukherjee and J. Strikwerda. Analysis of Dynamic Congestion Control Protocols - A Fokker-Planck Approximation. In *Proceedings ACM SIGCOMM'91*, September 1991.

[13] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings ACM SIGCOMM'98*, September 1998.

[14] K.K. Ramakrishnan and S. Floyd. A Proposal to Add Explicit Congestion Notification (ECN) to IP. RFC 2481, January 1999.

[15] N. Venkitaraman, T. Kim, K. Lee, S. Lu, and V. Bharghavan. Design and Evaluation of a Suite of Congestion Control Algorithms for the Future Internet. In *Proceedings ACM SIGMETRICS'99*, May 1999.

[16] M. Vojnovic, J. Y. Le Boudec, and C. Boutremans. Global Fairness of Additive-Increase and Multiplicative-Decrease with Heterogeneous Round-Trip Times. In *Proceedings IEEE INFOCOM 2000*, March 2000.