Two Problems of TCP AIMD Congestion Control *

Yang Richard Yang, Min Sik Kim, Xincheng Zhang, Simon S. Lam
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712–1188
{yangyang,minskim,zxc,lam}@cs.utexas.edu
TR2000-13
June, 2000

Abstract

AIMD (additive increase and multiplicative decrease) algorithm has been used in many congestion control protocols, including TCP in the Internet. In this paper, we compare AIMD and MAIMD (multiplicative additive increase and multiplicative decrease). Our evaluation shows that the convergence speeds to fair states of AIMD and MAIMD are close to each other. However, MAIMD has some advantages. For example, its speed to use available network bandwidth can be much faster than AIMD. We have also investigated AIMD behaviors under a more realistic asynchronous system model. Under this model, AIMD system can have more than one attractors, which can be another contributor to the fairness problem of TCP.

1 Introduction

In a shared network such as the Internet, end systems should react to congestion by adapting their transmission rates to avoid congestion collapse and keep network utilization high [3]. The robustness of the current Internet is due in large part to the end-to-end congestion control mechanisms of TCP [5]. In particular, TCP uses an *additive increase multiplicative decrease* (AIMD) algorithm [2]; the TCP sending rate in congestion avoidance state is controlled by a congestion window which is halved for every window of data containing a packet drop, and increased by one packet per window of data acknowledged. Recently, many new congestion control protocols have been proposed and investigated [4, 6, 13, 11, 1, 14, 7, 10, 12, 8, 15]. The objective of these new congestion protocols is to address the needs of new multimedia applications. We notice that, like TCP, many of these proposals are based on the AIMD principle. Further, there is even a common belief that AIMD is optimal and is a necessary condition for a congestion control mechanism to be stable [9].

*Research sponsored in part by National Science Foundation grant No. ANI–9977267 and grant No. ANI–9506048. Experiments were performed on equipment procured with NSF grant No. CDA–9624082.

AIMD congestion control was first studied by Chiu and Jain [2]. Figure 1 shows the system model they used to analyze a congestion control system. In this model, x_i denotes

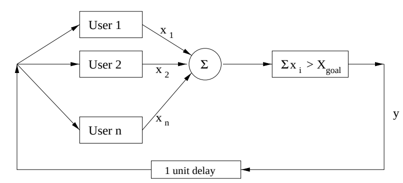

Figure 1: Chiu/Jain congestion system model

the load generated by user i. The congestion status of the network at time t is measured by $X(t) = \sum x_i(t)$. When $X(t) > X_{goal}$, the network is considered to be congested and the network sends a signal $y = 1$ to ask all users to slow down; otherwise, the network indicates no congestion by sending $y = 0$. In this case, all users increase their load.

Formally, the dynamics of the system can be specified as:

$$\begin{aligned} X(t) &= \sum x_i(t) \\ y(t) &= \begin{cases} 0 & \text{if } X(t) \le X_{goal} \\ 1 & \text{otherwise} \end{cases} \\ x_i(t+1) &= \begin{cases} a_I + b_I x_i(t) & \text{if y(t) = 0} \\ a_D + b_D x_i(t) & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

It is important to notice that Equation (1) assumes homogeneous delay for all users, with a unit delay on each feedback link. In other words, what the authors have analyzed is a synchronous system.

Denote $\mathbf{x}$ as a vector of all users load x_i. Define a fairness index function $F(\mathbf{x})$ as:

$$F(\mathbf{x}) = \frac{(\sum x_i)^2}{n(\sum x_i^2)} \tag{2}$$

With the above system model in Equation (1) and the definition for fairness index function F in Equation (2), the authors derived the following results:

- If the parameters a_I, b_I, a_D, and b_D satisfy the following condition:

$$\left\{ \begin{array}{lcl} a_I & > & 0, \quad b_I \geq 1 \\ a_D & = & 0, \quad 0 \leq b_D < 1 \end{array} \right. \tag{3}$$

  the system will converge to the efficient and fair states. We should point out that the condition in Equation (3) is weaker than what we commonly refer to as AIMD, which specifies $b_I = 1$. To distinguish these two cases, we refer to the case when $b_I > 1$ as MAIMD (multiplicative and additive increase and multiplicative decrease), and the case of $b_I = 1$ as AIMD (additive increase and multiplicative decrease). We will discuss AIMD in the next item.

  As for the evolution of fairness index F, the authors show that F satisfies the following recurrence relation:

$$\begin{aligned} F(\mathbf{x}(t+1)) = F(\mathbf{x}(t)) + & \quad (1 - F(\mathbf{x}(t))) \\ & \times \left(1 - \frac{\sum x_i^2(t)}{\sum (c + x_i(t))^2}\right) \end{aligned} \tag{4}$$

  where $c = a_I / b_I$ when $y(t) = 0$, and $c = a_D / b_D = 0$ when $y(t) = 1$. Since $c = 0$ when users reduce loads, F will not change. When the users increase their loads, we can see that $c > 0$, and unless $F(\mathbf{x}(t))$ is 1, F will increase. Overall, the fairness index non-decreasingly approaches the maximum value, 1.

- Further, based on Equation (4), the authors assert that for a given a_I, setting $b_I = 1$ will have the fastest convergence speed to the fair states. Therefore, the best choice should be additive increase (because $b_I = 1$, instead of the general case of $b_I \geq 1$) and multiplicative decrease (because $a_D = 0$). To summarize, AIMD means the following stronger condition than in Equation (3):

$$\left\{ \begin{array}{lcl} a_I & > & 0, \quad b_I = 1 \\ a_D & = & 0, \quad 0 \leq b_D < 1 \end{array} \right. \tag{5}$$

To gain some intuition, the left sub-figure in Figure 2 shows the state trace (x_1, x_2) of a two user system. We observe that the system does converge to the fair states of $x_1 = x_2$ line and then oscillates along this line. The right sub-figure in Figure 2 shows the evolution trace of the fairness index F. We observe that F non-decreasingly approaches the maximum value, 1.

However, the above results are based on the analysis of a synchronous system model, and some other decisions also need to be justified. In particular, we are interested in the following questions:

- First, from Equation (4), for a given a_I, we know that setting $b_I = 1$ will generate the largest single step increase from $F(\mathbf{x}(t))$ to $F(\mathbf{x}(t+1))$. However, this choice is a local optimization. What we are interested is the total number of steps from an initial state to the final fair states. In other words, what we are interested is the global convergence speed. It is not clear that choosing a local optimal value of $b_I = 1$ will have the global optimal effect.

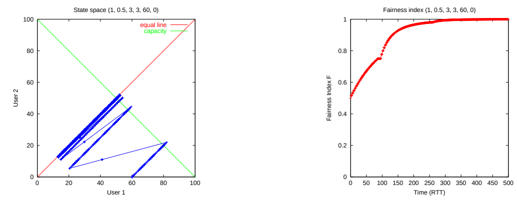

Figure 2: State and fairness index F traces

- Second, define $\mathcal{F}$ as the set of fair states, i.e. for a state $\mathbf{x}$ such that $F(\mathbf{x}) = 1$. For a two user system, we know that $\mathcal{F}$ is the $x_1 = x_2$ line. Define $\mathcal{S}$ as the set of steady states. Formally, we say a state $\mathbf{x}$ is a steady state if the system visits the state infinite times[1] starting from any initial state. For a two user system, we know that $\mathcal{S}$ will be a subsegment of the line $x_1 = x_2$. For system with n users, the oscillation range of any x_i will be:

$$b_D b_I \frac{X_{goal}}{n} + a_I \leq x_i \leq b_I \frac{X_{goal}}{n} + a_I \tag{6}$$

  Therefore, another metric to describe the behavior of a system is to consider the time when the system enters the steady states.

- Third, what the authors have analyzed is a synchronous system model. What about the system behaviors for an asynchronous system model, e.g., a system where users have different round-trip times? It is straightforward to see that under this system model the fairness index F will not non-decreasingly approaches 1. We are interested in whether there are any other unexpected behaviors.

The rest of this paper is organized as follows. In Section 2, we discuss AIMD congestion control, and compare AIMD with MAIMD in terms of convergence speeds to both fair states and steady states. In Section 3, we consider a more general system model where users can have different round-trip times. We show that under AIMD, the system can have more than one attractors and, therefore, exhibits chaotic behaviors. In Section 5, we discuss the implications of the findings to real world network congestion control. We conclude in Section 5.

[1] We assume computations have a finite precision.

2 Comparison of AIMD and MAIMD

In this section, we compare AIMD with MAIMD. What we are interested is their convergence speeds. The converged states are fair states $\mathcal{F}$ and steady states $\mathcal{S}$.

2.1 Convergence speeds to fair states $\mathcal{F}$

According to [2], for a given a_I, AIMD will have the optimal convergence speed to fair states. However, as we discussed in the previous section, this optimization is a local optimization.

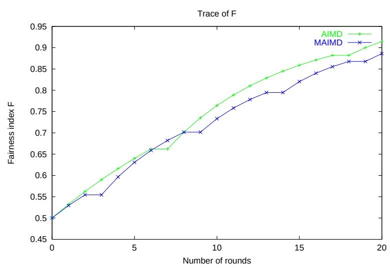

Figure 3: AIMD and MAIMD Convergence to fair states

Figure 3 shows the trace of F for a two user system. The system starts with an initial state of $(30, 0)$, and X_{goal} is 40. The a_I for both AIMD and MAIMD is set to be 1. The b_I for MAIMD is set to be 1.1. We observe from this figure that the convergence speeds of AIMD and MAIMD is close to each other. It is particular interesting to notice that at time 7, fairness index of AIMD is 0.66, and the fairness index of MAIMD is 0.68. This shows that MAIMD can have faster convergence speed in a global manner. Without enumerating more experimental results, we find that in most cases, the fairness convergence speeds of AIMD and MAIMD are close to each other. Therefore, in our opinion, choosing AIMD over MAIMD based on convergence speed to fair states may not be the best decision criterion. Some other factors should be considered. In particular, we next consider the convergence speeds of AIMD and MAIMD to steady states.

2.2 Convergence speeds to steady states $\mathcal{S}$

In the previous subsection we discussed AIMD and MAIMD convergence speeds to fair states. In this subsection, we study their convergence speed to the steady state. We consider the following metrics:

- Response time T: is the number of steps for the system load $X(t)$ to increase from an initial load $X(0)$ to the steady state load, which is the following range:

$$\begin{cases} X_{steady}(t) & \geq X_{min} = b_D X_{goal} \\ X_{steady}(t) & \leq X_{max} = b_I X_{goal} + n a_I \end{cases} \tag{7}$$

- Overload δ: is the load imposed on the system when the load exceeds X_{goal}. We define δ as:

$$\delta = \max \frac{X(t) - X_{goal}}{X_{goal}} \tag{8}$$

- Rate oscillation η: is the oscillation range of a receiver at the steady states. Since all users will have the same load at steady state, we define rate oscillation in terms of system load. We define η as:

$$\eta = \frac{X_{max} - X_{min}}{X_{min}} \tag{9}$$

2.2.1 Response time T

We know that if the load $X(0)$ is greater than X_{max}, both AIMD and MAIMD will reduce their load exponentially. And we have $T = \log_{b_D} \frac{X_{max}}{X(0)}$. Therefore, AIMD and MAIMD have the same performance. Next, we consider the case when $X(0)$ is less than X_{min}. In this case, we have the following expression for T:

$$T = \begin{cases} \frac{X_{min} - X(0)}{n a_I} & \text{if AIMD} \\ \log_{b_I} \frac{(b_I - 1) X_{min} + n a_I}{(b_I - 1) X(0) + n a_I} & \text{if MAIMD} \end{cases} \tag{10}$$

Therefore, we see that MAIMD can increase its load in exponential speed, and therefore can be much faster than AIMD.

2.2.2 Overload δ

From Equations (7) and (8), we have the following expression for δ:

$$\delta = (b_I - 1) + \frac{n a_I}{X_{goal}} \tag{11}$$

From (11), we can make the following observations. First, MAIMD will generate a lighter overhead than AIMD if they have under the same a_I. The extra overload in terms of percentage over X_{goal} is $b_I - 1$. Therefore, when $b_I = 1.1$, MAIMD will generate 10% higher overload. Second, we observe that the second term of overhead $\frac{n a_I}{X_{goal}}$ depends on the number of receivers. Therefore, when n is large, the overload can be very high, this shows the undesired effect of additive increase.

2.2.3 Rate oscillation η

From Equations (7) and (9), we have the following expression for η:

$$\eta = \frac{b_I}{b_D} - 1 + \frac{n a_I}{b_D X_{goal}} \tag{12}$$

From (12), we observe that for a given system MAIMD increases rate fluctuation. The magnitude of the fluctuation is proportional to $\frac{b_I}{b_D}$. Therefore, if $b_I = 1.1$, the increased oscillation will be $\frac{0.1}{b_D}$ in terms of percentage increase.

3 Chaos behaviors of AIMD with different RTTs

In the previous section, we considered both AIMD and MAIMD. However, the analysis is based on a synchronous system model. In this section, we consider a system model that users can have different round-trip times. The system diagram is shown in Figure 4.

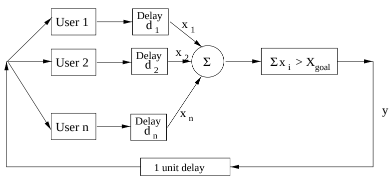

Figure 4: System Model with Different RTTs

In this model, x_i is again the load generated by user i. The congestion status of the network at time t is measured by $X(t) = \sum x_i(t - d_i)$. When $X(t) > X_{goal}$, the network is considered to be congested and the network sends a signal $y = 1$ to ask all users to slow down; otherwise, the network indicates no congestion by sending $y = 0$, and the users increase their load.

Formally, the dynamics of the system can be specified as:

$$\begin{aligned} X(t) &= \sum x_i(t - d_i) \\ y(t) &= \begin{cases} 0 & \text{if } X(t) \leq X_{goal} \\ 1 & \text{otherwise} \end{cases} \\ x_i(t+1) &= \begin{cases} a_I + b_I x_i(t) & \text{if } \sum_{k=1}^{d_i - 1} y(t-k) = 0 \\ a_D + b_D x_i(t) & \text{otherwise} \end{cases} \end{aligned} \qquad (13)$$

We notice that other models to capture round-trip are also possible.

Next, we extend the fairness index function F to the weighted fairness index function:

$$F(\mathbf{x}, \mathbf{d}) = \frac{(\sum d_i x_i)^2}{n(\sum (d_i x_i)^2)} \qquad (14)$$

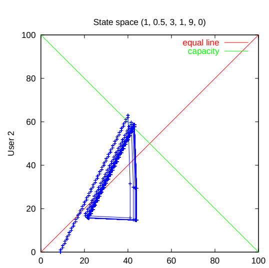

Figure 5: State trace with different RTTs

Figures 5 and 6 show the state and fairness index traces of two users with different initial conditions. The first user has a delay d_i of 3 and starts with a window size 9; the second user has a delay d_i of 1 and starts with window size 0. We observe from Figure 5 that in this case, the fair states will not be the $x_1 = x_2$ line. From Figure 6, we notice that the fairness index will not increase non-decreasingly towards 1.

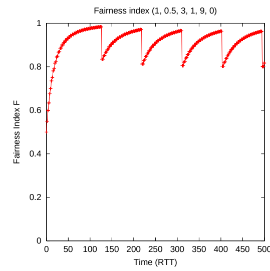

Figure 6: Fairness index trace with different RTTs

Figures 7 and 8 show state and fairness index traces for another initial conditions. In these two figures, user 1 starts with window size 10, and user 2 starts with window size 0. Notice that the only difference of Figure 5 and Figure 7 is the starting condition of user 1. In Figure 5, user 1 starting window is 9, and in Figure 7 it is 10.

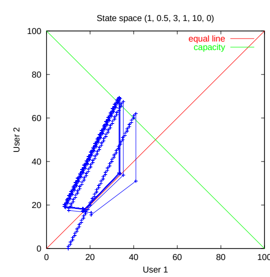

Figure 7: State trace with different RTTs

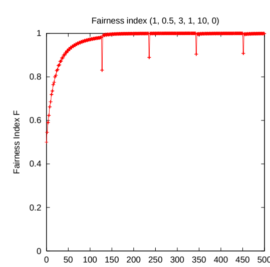

Figure 8: Fairness index trace with different RTTs

Figures 9 and 10 show state and fairness index traces of two users starting at yet another initial condition. In this figure, user 1 starts with a window size of 60.

Comparing these three state traces, we can make the following observations:

- Instead of converging to the equal window size line,

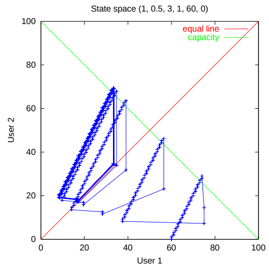

Figure 9: State trace with different RTTs

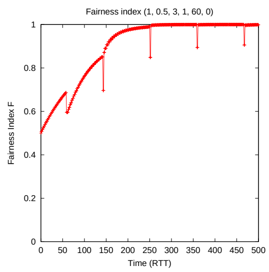

Figure 10: Fairness index trace with different RTTs

systems with different user RTTs converge to cyclic states, i.e., an attractor.

- Instead of non-decreasingly increasing to 1, the fairness index F may drop.

- We notice that Figure 5 converges to a different attractor from that of Figure 7 and Figure 9. Figure 5 starts very close (user 1 initial window 9) to Figure 7 (user 1 initial window 10) but they converge to different attractors. This is a typical behavior of a chaotic system.

- We have tried to start the system in various initial states. Overall, we have observed two attractors: the one in Figure 5 and the one in Figure 7 and Figure 9. Most of the initial states converge to the second attractor. Also, we found the first attractor to be unstable. For example, when we disturb it (by omitting a drop for one user), the attractor converges to the other attractor. Further, we notice that the attractor in Figure 5 has a worse fairness property than that in Figure 7.

4 Implications to the Internet

- AIMD is widely considered to be the optimal control strategy. However, from our investigation in Section 2, we observe that MAIMD also has some advantages, especially in its faster speed to use network available bandwidth. We notice that MAIMD can be considered as a combination of AIMD additive increase and TCP slow start multiplicative increase. We believe further investigation of MAIMD is needed.

- We have observed that TCP flows under similar conditions can achieve different throughput. The traditional explanation is that this is a reflection of the random events in the network. From the investigation in this paper, we observe that the chaotic behavior of AIMD congestion control is another contributor.

5 Conclusion

We compared AIMD and MAIMD in this paper. We found that the convergence speeds to fair states of AIMD and MAIMD are close to each other. However, we observe that MAIMD has some advantages. For example, its speed to use network available bandwidth can be much faster than that of AIMD. Therefore, it points out that further research of MAIMD is needed. We also investigated AIMD behaviors under a more realistic asynchronous system model. We found that under this model, AIMD system can have more than one attractors, and therefore can be another contributor to the fairness problem of TCP.

References

[1] Shanwei Cen, Calton Pu, and Jonathan Walpole. Flow and congestion control for Internet streaming applicaitons. In *Proceedings of Multimedia Computing and Networking 1998*, January 1998.

[2] Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17, 1989.

[3] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4), August 1999.

[4] Stephen Jacobs and Alexandros Eleftheriadis. Providing video services over networks without quality of service guarantees. In *Proceedings of World Wide Web Consortium Workshop on Real-time Multimedia and the Web*, October 1996.

[5] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM '88*, August 1988.

[6] J. Mahdavi and S. Floyd. TCP-friendly unicast rate-based flow control. Note sent to the end2end-interest mailing list, 1997.

[7] V. Ozdemir and I. Rhee. TCP emulation at the receivers (TEAR), presentation at the rm meeting, November 1999.

[8] Jitendra Padhye, Jim Kurose, Don Towsley, and Rajeev Koodli. A model based TCP-friendly rate control protocol. In *Proceedings of NOSSDAV '99*, June 1999.

[9] Larry L. Peterson and Bruce S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers, 2nd edition, October 1999.

[10] Reza Rejaie, Mark Handley, and Deborah Estrin. RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet. In *Proceedings of IEEE INFOCOM '99*, volume 3, March 1999.

[11] Dorgham Sisalem and Henning Schulzrinne. The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. In *Proceedings of NOSS-DAV '98*, July 1998.

[12] Wai-Tian Tan and Avideh Zakhor. Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Trans. on Multimedia*, 1, June 1999.

[13] Thierry Turletti, Sacha Fosse Parisis, and Jean-Chrysostome Bolot. Experiments with a layered transmission scheme over the Internet. Research Report No 3296, INRIA, November 1997.

[14] Lorenzo Vicisano, Luigi Rizzo, and Jon Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of IEEE INFOCOM '99*, volume 3, March 1999.

[15] Yang Richard Yang and Simon S. Lam. General AIMD congestion control. In *Proceedings of ICNP 2000*, November 2000.