Fairness in MIMD Congestion Control Algorithms

E. Altman K. E. Avrachenkov B. J. Prabhu
INRIA Sophia Antipolis
2004 route des lucioles, 06902 Sophia Antipolis, France.
Email: {altman, k.avrachenkov, bprabhu}@sophia.inria.fr

Abstract— **The Mulitplicative Increase Multiplicative Decrease (MIMD) congestion control algorithm in the form of Scalable TCP has been proposed for high speed networks. We study fairness among sessions sharing a common bottleneck link, where one or more sessions use the MIMD algorithm. Losses, or congestion signals, occur when the capacity is reached but could also be initiated before that. Both synchronous as well as asynchronous losses are considered. In the asynchronous case, only one session suffers a loss at a loss instant. Two models are then considered to determine which source looses a packet: a *rate dependent* model in which the packet loss probability of a session is proportional to its rate at the congestion instant, and the *independent loss rate* model. We first study how two MIMD sessions share the capacity in the presence of general combinations of synchronous and asynchronous losses. We show that, in the presence of rate dependent losses, the capacity is fairly shared whereas rate independent losses provide high unfairness. We then study inter protocol fairness: how the capacity is shared in the presence of synchronous losses among sessions some of which use additive increase multiplicative decrease (AIMD) protocols whereas the others use MIMD protocols.**

I. Introduction

In the Internet, data transfer protocols use various congestion control algorithms to achieve rate control. Until now the AIMD algorithm was found to provide satisfactory performance. However, in high speed networks , the additive increase in the sender's rate may lead to inefficient link utilization [1]. To overcome this drawback in high speed networks, the MIMD algorithm has been proposed as an alternative to the AIMD algorithm (e.g., [2], [3]). Therefore, in the future, situations may arise when sessions using these two algorithms would compete for the same network resource. The share of the capacity obtained by each of these sessions will depend on the various parameters specific to the algorithms. The sharing of a resource gives rise to the question of how fairly is this resource shared among the sessions. Fairness issues have been addressed in several previous works. In [4], the authors considered a class of rate control algorithms in the presence of synchronous control signals. They showed that the AIMD algorithm converged to fairness whereas the MIMD algorithm did not converge. In [5], the author studied the MIMD algorithm under a more realistic assumption of rate dependent losses and argued that MIMD algorithm also converges to fairness. In [6] the convergence to fairness of the different flavours of TCP was studied both analytically and using simulations. The monotonic convergence to fairness for algorithms in rate-based TCP-friendly applications was studied in [7]. In [8], the authors remarked that for sessions with different round trip times (RTT), Scalable TCP (which uses MIMD algorithm) is extremely unfair. They proposed a new algorithm which improved the efficiency as well as the fairness.

Most of the congestion control algorithms rely on binary feedback (i.e., presence or absence of congestion) from the network to adapt the sending rate and reduce congestion. The presence of congestion in the network is signalled by either dropping or marking packets. Congestion signals occur not only when the link capacity is achieved but could also be initiated before that. For example, a congestion signal is sent as a consequence of a packet drop when the link buffer is full (i.e., when the capacity is achieved), or when a router using AQM schemes such as RED or ECN decides to respectively drop or mark a packet (i.e., a congestion signal is initiated before the capacity is achieved). In the rest of the paper, we shall use the terms losses and congestion signals interchangeably. The term losses does not only refer to packet drops. It is used to describe a decrease in the sending rate as a result of a congestion signal which could be either due to a packet drop or due to a packet mark. Losses are said to be synchronous when all the sessions sharing the link suffer a loss at the same time instant. In the asynchronous case, only one session suffers a loss. We use two models to determine which of the sources receives the congestion signal: (1) the *rate dependent loss model* in which the loss probability of a session is proportional to its rate at the congestion instant [9], and (2) the *independent loss rate model*, in which probability with which a congestion signal is sent to a session is independent of the session's rate.

The rest of the paper is organised as follows. In Section II we present a brief overview of the model and mention the contribution of this work. In the first part of this paper (Sections III and IV), we study how two MIMD sessions (with either the same RTT or with different RTTs) share the capacity in the presence of general combinations of synchronous and asynchronous losses. We show that, in the presence of rate dependent losses, the capacity is fairly shared between the two sessions whereas rate independent asynchronous losses result in high unfairness even when sessions are symmetric. In the second part of this paper (Section V and VI), we consider a heterogenous scenario with synchronous losses only. We study how the capacity is shared among several sessions, each of which uses either the AIMD or the MIMD algorithm. We show that the AIMD session obtains a share which is independent of the link capacity, and that the rest of the capacity is utilized by the MIMD session. Finally, we present the conclusions in Section VII.

II. OVERVIEW

We use the following notation. A function from $\mathbb{R}$ to $\mathbb{R}$ will be denoted using sans serif font, such as $\mathsf{x}()$. For example, $\mathsf{x}(t)$ denotes a function defined for all real values of t. A function from $\mathbb{Z}$ to $\mathbb{R}$ will be denoted using italic fonts as $x()$. Usually, $x(n)$ would be the value of $\mathsf{x}(t)$ at the n^{th} sampling instant. A vector a will denote a row vector. Its transpose will be denoted by a'. Also, we use the term session to mean an instance of a given algorithm. The term user will be emplyed to describe someone who initiates one or more instances (i.e., sessions) of the same algorithm.

The following model is mainly based on the model in [4]. Consider two flows which share a link of capacity C. Let $\mathsf{x}(t) \equiv (\mathsf{x}_1(t), \mathsf{x}_2(t))$ be the rate vector at time t, where $\mathsf{x}_1(t)$ and $\mathsf{x}_2(t)$ denote the instantaneous rates of session 1 and of session 2, respectively. The set of feasible rate vectors, $\{(\mathsf{x}_1, \mathsf{x}_2) | \mathsf{x}_1 + \mathsf{x}_2 \leq C; \mathsf{x}_1, \mathsf{x}_2 \geq 0\}$ is shown in Fig. 1. The line

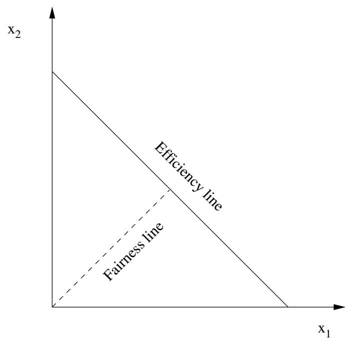


Fig. 1. The rate allocation vector

$\mathsf{x}_1(t) + \mathsf{x}_2(t) = C$ is called the efficiency line. On this line the available capacity is fully utilized. The line $\mathsf{x}_1(t) = \mathsf{x}_2(t)$ is called the fairness line. On this line both the sessions obtain the same rates, and hence the bandwidth sharing is said to be fair. The sessions react to control signals by adapting their rates in the following way.

$$x(n+1) = \begin{cases} b_I x(n) + a_I, & \text{for an increase signal,} \\ b_D x(n) + a_D, & \text{for a decrease signal,} \end{cases}$$

where a_I, b_I, a_D and b_D are constants and the sampling is done just after the reception of control signals. In TCP Reno, for example, arrival of ACKs can be considered as increase signals, and arrival of duplicated ACKs or a time-out is considered to be a decrease signal. In [4] it has been argued that for convergence to the fairness line, the increase algorithm has to be multiplicative and additive (i.e., $a_I > 0$ and $b_I \geq 1$) and the decrease algorithm has to be multiplicative ($b_D < 1$ and $a_D = 0$). The authors assume that the control signals are synchronous for both the sessions. That is, both the sessions receive control signals at the same instant and both of them either increase, or decrease their rates simultaneously[1]. Indeed, under these two assumptions, the rate vector for the MIMD algorithm stays on a line joining the origin to the initial rate vector, and hence does not converge to fairness. In [5], the author argues that under a more realistic assumption of rate dependent congestion signals, the MIMD algorithm also converges to the fairness line. In the first part of the paper, we first show that, for sessions with the same RTT, the MIMD algorithm converges to fairness when the congestion signals are asynchronous and rate dependent. We obtain the expressions for the long term fairness index, the rate of convergence to the steady state distribution and the mean time to achieve fairness. In [8] it was argued that for sessions with different RTTs, Scalable TCP (or, the MIMD algorithm) is extremely unfair. We show that, even for sessions with different RTTs, a certain degree of fairness can be achieved by introducing sufficient number of rate dependent losses. We then show, through simulations, that the results obtained for two sessions also hold for n sessions.

In the second part of this paper, we consider several sessions sharing a common link on which losses are due to buffer overflow and are, therefore, synchronous. In [6] such a scenario was considered for sessions using the AIMD algorithm only. In [7], fairness and packet-loss scalability analysis and simulations for session using more general binomial algorithms was provided . In [8], fairness issues were considered for high-speed networks. The RTT-fairness was compared for different proposals of TCP in high-speed networks. Here, too, the fairness was studied among sessions using the same algorithm. However, we consider a heterogeneous scenario where different sessions may use different algorithms. This type of scenarios may be of interest in the future when, for example, sessions using Scalable TCP and standard TCP will share the same link. We analyze the equilibrium throughput and the window size of the sessions and compare them with simulations.

As was pointed out in [7], the window-based notation can be converted to a rate-based notation using the relation $x(n) = w(n)\frac{MTU}{RTT}$, where $w(n)$ is the window at the n^{th} sampling instant, M is the packet size in bits, and RTT is the round trip time in seconds. We shall use the rate-based notation in the first part of the paper, and the window-based notation in the second part of the paper.

III. FAIRNESS IN MIMD SESSIONS (EQUAL RTTS)

We consider two sessions which share a link of capacity C. At time t, the rates obtained by the two sessions are denoted by $\mathsf{x}(t) \equiv (\mathsf{x}_1(t), \mathsf{x}_2(t))$. At each control instant, the controller sends a control signal to each source. This control signal either informs on no congestion (a 0 signal) or of congestion (a 1 signal). In the absence of congestion, the sources increase their

[1]These assumptions are validated in [10] by simulations for the AIMD versions of TCP, with approximately the same RTTs, and we validate them later for the MIMD and the AIMD versions of TCP (see, e.g., Fig. 5 and its discussion).

TABLE I

REACTION TO CONTROL SIGNALS

control vector	$\mathsf{x}_1(t_j+)$	$\mathsf{x}_2(t_j+)$	
$(0,0)$	$\mathsf{x}_1(t_j)$	$\mathsf{x}_2(t_j)$	
$(0,1)$	$\mathsf{x}_1(t_j)$	$\beta \cdot \mathsf{x}(t_j)$	async. cong. signal
$(1,0)$	$\beta \cdot \mathsf{x}_1(t_j)$	$\mathsf{x}_2(t_j)$	async. cong. signal
$(1,1)$	$\beta \cdot \mathsf{x}_1(t_j)$	$\beta \cdot \mathsf{x}_2(t_j)$	sync. cong. signal

rate exponentially, i.e.,

$$\mathsf{x}_\mathsf{k}(t+\tau) = \alpha^{\tau/\tau_0} \cdot \mathsf{x}_\mathsf{k}(t), \quad k = 1, 2,$$

where τ_0 is the time constant (for example, the RTT) for the sessions, and $\alpha > 1$ is the increase factor. The above formulation is a continuous time equivalent of a multiplicative algorithm in which, for every RTT without congestion signals, the sender multiplies the window by a factor of α. This can be seen by substituting $t = n\tau_0$. We assume that the two sources receive the control signals at the same instant. However, unlike the model in [4], the two sources can receive different control signals. That is, a congestion signal need not be sent to both the sources at the same instant. Hence, the congestion signals could be asynchronous. Let $\beta < 1$ be the decrease factor. Let the j^{th} control signal be received at time t_j. Then, the four possibilities for the rate vector, $\mathsf{x}(t_j+)$, just after t_j, are given in Table I.

The source continues with the increase algorithm on the reception of 0 signal. On the other hand, when a source receives a 1 signal, it instantaneously reduces its rate. We assume that whenever the link capacity is attained then either a synchronous or an asynchronous loss occurs. Furthermore, asynchronous losses may also occur before attaining the capacity.

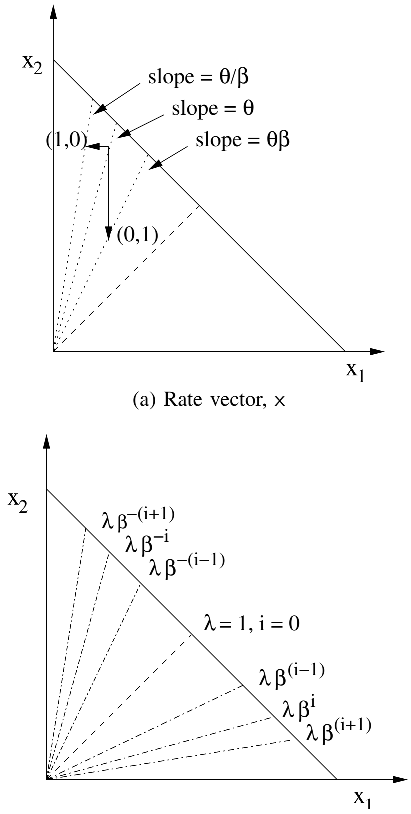
A. *Instantaneous throughput ratio process*

We now study the instantaneous throughput ratio process. It is shown to be a Markov process with a countable state space; we shall show that this process could be stable or unstable depending on the asynchronous loss process.

In Figure 2(a), we show the geometric interpretation of the response to the different control signals. Let $\theta(t)$ be the slope of the line joining the origin and the current vector, $\mathsf{x}(t)$. That is,

$$\theta(t) = \frac{\mathsf{x}_2(t)}{\mathsf{x}_1(t)}.$$

If there are no congestion signals in the interval $(t, t+\tau)$ then the rate vector at time $(t+\tau)$ will be $\alpha^{\tau/\tau_0}(\mathsf{x}_1(t), \mathsf{x}_2(t))$. This vector also lies along the line with slope $\theta(t)$. If a $(1,1)$ signal was generated at t, then the vector after the response, $\mathsf{x}(t+)$, is $\beta(\mathsf{x}_1(t-), \mathsf{x}_2(t-))$. This vector also lies along the line with slope $\theta(t)$. Therefore, the rate vector remains along a line with the same slope as long as the control vector is either $(0,0)$ or $(1,1)$, or there are no control signals. However, it can be seen that the rate vector moves to the line with slope $\theta(t)/\beta$ when a control signal of $(1,0)$ is generated. Similarly, the rate vector moves to the line with slope $\theta(t) \cdot \beta$ when a



(a) Rate vector, x

(b) State space, $\mathbb{S}$, of x

Fig. 2. Geometric interpretation.

control vector of $(0,1)$ is generated. Therefore, if there were only synchronous congestion signals (i.e., (1,1) signals) then $\theta(t) = \theta(0)$ for all t, and any initial unfair sharing would remain forever. Therefore, if MIMD protocols are used then it is essential to provide a stream of asynchronous congestion signals using, for example, some queue management scheme.

Let t_n denote the time instant when the n^{th} control signal arrives. Let $\theta(n) = \theta(t_n+)$ be the slope of the rate vector sampled just after the n^{th} control signal. From Fig. 2(a), it can be seen that, given an initial slope of $\theta(0)$, the slope of the line along which the rate vector lies just after the n^{th} control signal can be written as $\theta(n) = \theta(0)\beta^i$ for some $i \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of all integers. For any given initial slope, $\theta(0)$, we can find a unique $\lambda \in (\beta^{1/2}, \beta^{-1/2})$ and $j \in \mathbb{Z}$ such that $\theta(0)$ can be expressed in terms of λ as $\theta(0) = \lambda\beta^j$. For convenience, we will define θ_n in terms of λ. The state space of $\theta(n)$ is a countably infinite state space defined by

$$\mathbb{S} = \{\lambda\beta^i, \forall i \in \mathbb{Z}\}.$$

A geometric interpretation of $\mathbb{S}$ is shown in Fig. 2(b). The line $\lambda = 1, i = 0$ is the fairness line. The continuous time increase and instantaneous decrease of the algorithm allows us to obtain the above formulation.

In the rest of this section, we assume that $\lambda = 1$. This assumption is equivalent to saying that the initial vector has a slope of β^j for some $j \in \mathbb{Z}$. If λ is not equal to 1 then, by any combination of control signals, the instantaneous rate vector can only get close to the fairness line by getting to λ. The rate vector cannot, however, be on the fairness line.

Let $\{\mathsf{s}(t) \in \mathbb{Z}\}$ be the process which indicates that the rate vector at time t lies on a line with slope $\beta^{\mathsf{s}(t)}$. We embed this process at instants of arrival of the control signals. Then, the sampled process $s(n) = \mathsf{s}(t_n+)$. We will show that $s(n)$ is a Markov chain, and it is stable or unstable depending upon the asynchronous loss process.

B. Rate dependent loss model

Next, we consider the rate dependent loss model in which the loss probability of a session is proportional to the session's rate at the congestion instant. In particular, if at the n^{th} control instant a congestion signal is sent, then we assume that the probability of loss for session k is given by $\frac{x_k(n)}{x_1(n)+x_2(n)}$.

Proposition 3.1: $\{s_n, n \geq 0\}$, is a discrete state-space Markov chain with transition probabilities given by

$$\begin{aligned}
s_{n+1}|(s_n = i) &= \begin{cases} i+1 & \text{w.p.} \quad \epsilon\frac{\beta^{|i|}}{1+\beta^{|i|}}, \\ i-1 & \text{w.p.} \quad \epsilon\frac{1}{1+\beta^{|i|}}, \quad i > 0, \\ i & \text{w.p.} \quad 1-\epsilon, \end{cases} \\
s_{n+1}|(s_n = 0) &= \begin{cases} +1 & \text{w.p.} \quad \epsilon/2, \\ -1 & \text{w.p.} \quad \epsilon/2, \qquad i = 0, \\ 0 & \text{w.p.} \quad 1-\epsilon, \end{cases} \\
s_{n+1}|(s_n = i) &= \begin{cases} i-1 & \text{w.p.} \quad \epsilon\frac{\beta^{|i|}}{1+\beta^{|i|}}, \\ i+1 & \text{w.p.} \quad \epsilon\frac{1}{1+\beta^{|i|}}, \quad i < 0. \\ i & \text{w.p.} \quad 1-\epsilon, \end{cases}
\end{aligned} \tag{1}$$

Remark 3.1: The probability of asynchronous loss at each control instant is ϵ. The asynchronous loss could be due to buffer overflow, some queue management scheme, or packet loss because of an erroneous transmission channel. In the absence of asynchronous losses (i.e., when there are only increase signals and synchronous losses), the instantaneous throughput ratio remains the same. Also, we model the instantaneous throughput ratio, and not the instantaneous throughputs themselves. Therefore, the capacity of the link does not appear in the calculations.

We are interested in finding the steady state distribution of the Markov chain, and the mean first passage time to the state $i = 0$ starting from a random state. The state $i = 0$ corresponds to the fairness line. Therefore, the mean first passage time from a random state to the state $i = 0$ gives an indication of the mean time before the rate vector reaches the fairness line. In general, the first passage time to state $i = 0$ gives the first passage time to λ. If $\lambda \neq 1$, s_n cannot be on the fairness line, and so the above performance measure corresponds to first passage time to the state closest to the fairness line for this particular process.

The fairness index at the n^{th} control instant, F_n, is defined as follows

$$F_n = \frac{1}{2}\frac{(x_1(n) + x_2(n))^2}{x_1(n)^2 + x_2(n)^2}.$$

We can write F_n in terms of β^i as follows

$$F_n = \frac{1}{2}\sum_{i=-\infty}^{\infty} P(s_n = i)\frac{(1+\beta^i)^2}{1+\beta^{2i}}.$$

The long term fairness index, F_∞, can be obtained by taking the limit $n \to \infty$. We assume that the process s_n converges to its stationary limit, s_∞.[2] Then, F_∞ can also be expressed as

$$F_\infty = \frac{1}{2} + \sum_{i=-\infty}^{\infty} P(s_\infty = i)\frac{\beta^i}{1+\beta^{2i}}. \tag{2}$$

Stability. The existence of the limiting distribution, s_∞, can be ensured by proving that the Markov chain s is positive recurrent. A Markov chain is positive recurrent if it satisfies the Foster's criterion [11] which is stated below.

Theorem 3.1 (Foster): An irreducible Markov chain s, on a countable state $\mathbb{Z}$, is ergodic if and only if there exists a positive function $f(\alpha)$, $\alpha \in \mathbb{Z}$, a number $\mu > 0$ and a finite set A such that

$$E[f(s_{n+1}) - f(s_n)|s_n = i] \leq -\mu, \quad i \notin A, \tag{3a}$$

$$E[f(s_{n+1})|s_n = i] < \infty, \quad i \in A. \tag{3b}$$

Let $f(i) = |i|, i \in \mathbb{Z}$ and $A = \{0\}$. Let Δf_i be define as $\Delta f_i := E[f(s_{n+1}) - f(s_n)|s_n = i]$. First, we show that condition (3a) is satisfied. For $i \neq 0$, from (1), we have

$$\begin{aligned}
\Delta f_i &= \epsilon\frac{\beta^{|i|}}{1+\beta^{|i|}}(|i|+1) + \epsilon\left(1 - \frac{\beta^{|i|}}{1+\beta^{|i|}}\right)(|i|-1) \\
&\quad +(1-\epsilon)|i| - |i| \\
&= \epsilon\left(2\frac{\beta^{|i|}}{1+\beta^{|i|}} - 1\right) \\
&\leq -\epsilon\left(1 - 2\frac{\beta}{1+\beta}\right).
\end{aligned}$$

which is strictly negative for any $\beta \in [0, 1)$. Therefore, condition (3a) is satisfied. To check for (3b), for $i = 0$ we have

$$E[f(s_{n+1})|s_n = i] = \frac{\epsilon}{2}|-1| + \frac{\epsilon}{2}|1| = \epsilon < \infty.$$

Therefore, the Markov chain s satisfies the conditions of Theorem 3.1 and, hence, is positive recurrent.

C. The independent loss rate model

If the session to which a congestion signal is sent is independent of the rates then the transition probabilities become

$$s_{n+1}|(s_n = i) = \begin{cases} i-1 & \text{w.p.} \quad \frac{\epsilon}{2}, \\ i+1 & \text{w.p.} \quad \frac{\epsilon}{2}, \qquad \forall i. \\ i & \text{w.p.} \quad 1-\epsilon, \end{cases} \tag{4}$$

We show that this results in instability. The Markov chain s is symmetric on $\mathbb{Z}$ and, therefore, is null recurrent. Since s is null

[2] We establish this later. In cases that the Markov chain is null recurrent (Subsection III-C) we shall understand the steady state distribution to correspond to the compactification of the state space in which $-\infty$ and ∞ are added to the state space and each will have a stationary probability of 0.5.

recurrent, and the discrete state space has two accumulation points, $(C, 0)$ and $(0, C)$, on the line $\mathsf{x}_1 + \mathsf{x}_2 = C$, the probability of being in any small vicinity of each point is $\frac{1}{2}$. The mean time to go from one extreme to another will be ∞ and, therefore, one connection will get the whole capacity. This suggests that rate independent losses are not sufficient to improve the fairness whereas rate dependent losses can indeed provide a fair share of the capacity.

In the rest of this section, we thus focus on the rate dependent loss model.

D. Steady state distribution

The Markov chain s is positive recurrent, and, therefore, the steady state distribution $P(s_\infty = i)$ exists. Since s_n is symmetric about the state 0, we can consider a Markov chain, $\{y_n, n \geq 0\}$, on the state space $\{0, 1, 2, ...\}$, in order to obtain the steady state distribution of s. The transition probabilities at the n^{th} control instant for this random walk are given by

$$y_{n+1}|(y_n = i) = \begin{cases} i+1 & \text{w.p. } \epsilon q_i, \\ i-1 & \text{w.p. } \epsilon(1-q_i), \quad i > 0, \\ i & \text{w.p. } 1-\epsilon, \end{cases}$$

$$y_{n+1}|(y_n = i) = \begin{cases} i+1 & \text{w.p. } \epsilon, \\ i & \text{w.p. } 1-\epsilon, \end{cases} \quad i = 0,$$

where $q_i = \frac{\beta^i}{1+\beta^i}$. Let y_∞ denote the steady state process to which y_n converges. Let p_i denote the probability of y_∞ being in state i. Then

Proposition 3.2:

$$p_i = p_0 \prod_{j=0}^{i-1} \frac{q_j}{1 - q_{j+1}}. \tag{5}$$

p_0 can be obtained from the equation $\sum_{i=0}^{\infty} p_i = 1$.

Proof: Let $p_n = 0, \quad \forall n < 0$. The balance equation for this walk can be written as

$$\begin{aligned} p_i &= (1-\epsilon)p_i + \epsilon q_{i-1}p_{i-1} + \epsilon(1-q_{i+1})p_{i+1} \\ &= q_{i-1}p_{i-1} + (1-q_{i+1})p_{i+1} \end{aligned} \tag{6}$$

These are the balance equations for a birth-death process with state dependent transition probabilities. The solution for this type of process is known to be of the form in (5) [12]. ■

From (5), p_i is given by

$$p_i = p_0 \beta^{i(i-1)/2}(1+\beta^i).$$

Since $\beta^i \to 0$ as $i \to \infty$, the tail of p_i can be seen to decrease as β^{i^2} which is a very fast decrease. In particular, this means that the process is around the fairness line most of the time.

The steady state distribution of s_∞ can be obtained from the following relations

$$P(s_\infty = i) = \begin{cases} \frac{1}{2}P(y_\infty = |i|), & \text{for } i \neq 0, \\ P(y_\infty = |i|), & \text{for } i = 0. \end{cases}$$

The long term fairness index, F_∞, can be computed numerically using (2) and (5). In Table II we give the fairness index for different value of β.

TABLE II

FAIRNESS INDEX FOR DIFFERENT VALUES OF β.

β	0.95	0.875	0.75	0.6	0.5	0.1
F_∞	0.987	0.97	0.942	0.91	0.88	0.777

E. Convergence to steady state distribution

The second largest eigenvalue of a matrix gives the rate of convergence to the steady state distribution. Therefore, we can get an indication of the rate of convergence of the Markov chain y by looking at the eigenvalues of its transition probability matrix, P. P can be written as

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & \cdots \end{array} \\ \begin{pmatrix} 1-\epsilon & \epsilon & 0 & \cdots \\ \epsilon(1-q_1) & 1-\epsilon & \epsilon q_1 & \cdots \\ 0 & \epsilon(1-q_2) & 1-\epsilon & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{array}. \tag{7}$$

Let ζ_i denote the i^{th} eigenvalue of P such that $\zeta_i \geq \zeta_j$ for $i < j$, and $\zeta = 1$. We can obtain the following lower bound on ζ_i.

Proposition 3.3: For $i > 0$, $1 - \epsilon \leq \zeta_i < 1$.

Proof: Since y_n is irreducible, the multiplicity of eigenvalues at 1 is 1. Therefore, $\zeta_i < 1$ for $i > 0$.

We can rewrite P as

$$P = (1-\epsilon)I + \epsilon A. \tag{8}$$

where I is the identity matrix and A is a transition matrix of a pure birth-death process with up transition probability q_i and down transition probability $1-q_i$. A is a stochastic matrix and, therefore, all its eigenvalues belong to the interval $[0, 1]$. Let μ_i be the i^{th} eigenvalue of A and let v_i be the corresponding left eigenvector. Then, from (8) we get

$$v_i P = (1-\epsilon)v_i + \epsilon v_i A = ((1-\epsilon) + \epsilon\mu_i)v_i.$$

Therefore, v_i is also the left eigenvector of P, and the corresponding eigenvalue is $(1-\epsilon) + \epsilon\mu_i$. Since $\mu_i \geq 0$, we get the inequality $\zeta_i \geq 1 - \epsilon$. ■

Therefore, $1-\epsilon$ gives a lower bound on the rate of convergence of the Markov chain to the steady state.

F. Mean first passage time

In this section we compute the mean first passage time to the state 0 starting from a random state. This gives us an estimate of the first time the rate vector reaches the fairness line starting from a given initial random state. The Markov chain s is a birth-death process which is symmetric about the state 0. If the initial state is positive, the Markov chain will stay in the set of positive states before visiting state 0. Similarly, if the initial state is negative, the Markov chain will stay in the set of negative states before visiting state 0. Therefore, we can obtain the mean first passage time to state 0 for s by obtaining the mean first passage time to state 0 for y.

Let $p = (p_0 \; p_1 \; \ldots)$ be the steady state probability vector of P, the transition probability matrix of y, as given by (7). Its i^{th}

component, p_i, is given by Eqn. (5). Let $m = (m_1\ m_2\ \ldots)$ denote the mean first passage time vector with m_j, $j \geq 1$ denoting the mean first passage time to the state 0 starting from state j.

Proposition 3.4: $m_i, i \geq 2$ can be obtained from the following recursion: $m_0 = 0$,

$$m_1 = \frac{1}{\epsilon}\frac{1-p_0}{p_0}, \tag{9}$$

$$m_{i+1} = \frac{\epsilon m_i - \epsilon \frac{1}{1+\beta^i} m_{i-1} - 1}{\epsilon \frac{\beta^i}{1+\beta^i}}. \tag{10}$$

Proof: Let P_1 be the transition probability matrix conditioned on y not being in state 0. We can rewrite P as

$$P = \begin{array}{c} \\ 0 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cc} 0 & \ldots \end{array} \\ \begin{pmatrix} 1-\epsilon & a \\ b & P_1 \end{pmatrix} \end{array}, \tag{11}$$

where the vector a is given as $a = \begin{pmatrix} \epsilon & 0 & \ldots \end{pmatrix}$ and the vector b is defined as $b = \begin{pmatrix} 1-\beta & 0 & \ldots \end{pmatrix}'$.

The vector m satisfies the equation [13]

$$(I - P_1)m' = \underline{1}', \tag{12}$$

where $\underline{1}$ is vector of ones. Since P_1 is a tridiagonal matrix, we can rewrite (12) as

$$m_{i+1} = \frac{\epsilon m_i - \epsilon \frac{1}{1+\beta^i} m_{i-1} - 1}{\epsilon \frac{\beta^i}{1+\beta^i}}, \ i \geq 2, \tag{13}$$

with the definition $m_0 = 0$.

Let the vector r be defined as $r = \begin{pmatrix} p_1 & p_2 & \ldots \end{pmatrix}$. Since p is the steady state probability vector of P, we have $pP = p$. Equivalently, using (11),

$$(p_0 \quad r) \begin{pmatrix} 1-\epsilon & a \\ b & P_1 \end{pmatrix} = (p_0 \quad r). \tag{14}$$

Solving for r, we obtain

$$r(I - P_1) = p_0 \cdot a.$$

Multiplying the above equation by m', and substituting for $(I - P_1) \cdot m'$ from (12), we obtain

$$r \cdot \underline{1}' = p_0 \cdot a \cdot m'.$$

Substituting $r \cdot \underline{1}' = 1 - p_0$ in the above equation, we obtain

$$\frac{1-p_0}{p_0} = a \cdot m'.$$

The vector a has ϵ in its first column and 0 elsewhere. Therefore, the above equation reduces to (9). From (13) and (9) we can now obtain the recursion in (13). ■

We note that the steady state probabilities are independent of ϵ whereas the mean first passage times are inversely proportional to ϵ.

IV. Fairness in MIMD sessions (unequal RTTs)

In this section we assume that the two sessions have different time constants. Let τ_1 and τ_2 be the time constants of session 1 and of session 2, respectively. The rate evolution for session i in the absence of control signals can be written as

$$\mathsf{x_i}(t+\tau) = \mathsf{x_i}(t)\alpha^{\tau/\tau_i}, i = 1, 2.$$

We now make the following transformation

$$\mathsf{z}(t) = \log\left[\frac{\mathsf{x_2}(t)}{\mathsf{x_1}(t)}\right]. \tag{15}$$

We consider again both synchronous losses as well as asynchronous ones. In the absence of asynchronous losses, the evolution of $\mathsf{z}(t)$ becomes

$$\mathsf{z}(t+\tau) = \mathsf{z}(t) + \gamma\tau, \tag{16}$$

where $\gamma = \log[\alpha]\left(\frac{1}{\tau_2} - \frac{1}{\tau_1}\right)$. If a control signal arrives at t, then $\mathsf{z}(t+)$ can be written as

$$\mathsf{z}(t+) = \begin{cases} \mathsf{z}(t), & \text{signal is either } (0,0) \text{ or } (1,1), \\ \mathsf{z}(t) + \log[\beta], & \text{signal is (0,1)}, \\ \mathsf{z}(t) - \log[\beta], & \text{signal is (1,0)}. \end{cases} \tag{17}$$

The evolution in time of z is shown in Figure 3. Here we

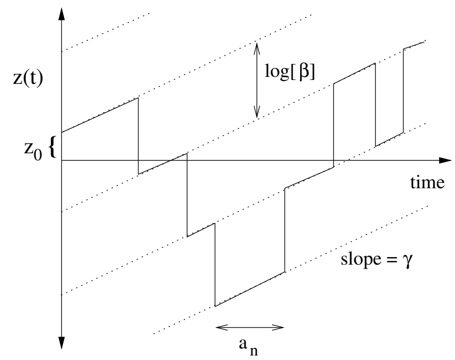


Fig. 3. Evolution in time of $z(t)$. $\tau_2 < \tau_1$.

assume that γ is positive, i.e., $\tau_2 < \tau_1$. From (16) and (17), it can be seen that, if there were only synchronous signals (i.e., only (0,0) and (1,1) signals) then there would be a drift in time towards $+\infty$. This suggests that the rate for session 1 would approach 0. Similarly, if γ were to be negative, i.e., $\tau_2 > \tau_1$, there would be a drift towards $-\infty$ suggesting that the rate for session 2 would approach 0. It can also be seen that if the slope were to be 0, i.e., τ_1 were to be equal to τ_2, the dotted lines would be parallel to the time axis. In this case $\mathsf{z}(t)$ would remain constant in the absence of asynchronous losses which was also observed in the previous section. We conclude that some buffer management scheme that creates rate dependent losses is necessary in order to have some fairness.

We therefore assume in the sequel that some buffer management scheme occasionally creates rate dependent losses. Let $\{a_n, n \geq 0\}$ denote the time between the n^{th} and the $(n+1)^{th}$ asynchronous congestion signals. The process a_n is assumed to be i.i.d. Let $z_n = \mathsf{z}(t_n)$ be the process embedded just before the arrival of an asynchronous control signal. Using an argument similar to one in section III, the probability that session i receives the n^{th} such signal can be written as $\frac{1}{\exp(z_n)+1}$ for session 1, and $\frac{\exp(z_n)}{\exp(z_n)+1}$ for session 2.

Proposition 4.1: The process $\{z_n, n \geq 0\}$ is a Markov chain with state space $\mathbb{R}$, and it follows the recursive equation

$$z_{n+1} = z_n + \gamma a_n + c_n, \tag{18}$$

where c_n is defined as

$$c_n = \begin{cases} -\log[\beta] & \text{w.p. } \frac{1}{\exp(z_n)+1}, \\ +\log[\beta] & \text{w.p. } \frac{\exp(z_n)}{\exp(z_n)+1}. \end{cases}$$

A. Stability

We now analyze the Markov chain z_n and obtain a sufficient condition for it to be positive recurrent, i.e., a condition under which the process $\mathsf{z}(t)$ does not have a drift towards ∞ as $t \to \infty$. Let $u_n = \gamma a_n$, and let $U(\cdot)$ and υ be the distribution function and the mean, respectively, of u_n. We assume that the density function, $\frac{dU(\cdot)}{du}$, of u is a non-increasing function. Let $b = -\log[\beta]$. The number b is positive since β is less than unity.

Proposition 4.2: The Markov chain z defined by (18) is positive recurrent if

$$\upsilon < b. \tag{19}$$

Proof: To show the positive recurrence of the Markov chain we use the following theorem from [14].

Theorem 4.1 ([14]): For some "small"[3] set $W \in \mathbb{B}(\mathbb{R})$, some constant $h < \infty$, $\mu > 0$, and an extended real-valued function $V : \mathbb{R} \to [0, \infty]$, z is stable if

$$\Delta V(x) := \int_{\mathbb{R}} P(x, dy)V(y) - V(x) \leq -\mu + h\mathbf{1}_W(x), \tag{20}$$

where $P(x, \cdot)$ is the one-step transition function of z.

To check for the drift condition of this theorem, we consider $V(y) = |y|$, and the set $W = [-b, b]$. The LHS of (20) becomes

$$\begin{aligned} \Delta V(x) + |x| &= \int_{y\in\mathbb{R}} \frac{1}{1+e^x}|y|dU(y-(x+b)) \\ &\quad + \int_{y\in\mathbb{R}} \frac{e^x}{1+e^x}|y|dU(y-(x-b)) \\ &= \int_{y=(x+b)}^{\infty} \frac{1}{1+e^x}|y|dU(y-(x+b)) \\ &\quad + \int_{y=(x-b)}^{\infty} \frac{e^x}{1+e^x}|y|dU(y-(x-b)) \end{aligned}$$

[3] A set W is called a "small" set if there exists a measure ϕ, $\phi(\mathbb{R}) > 0$, such that

$$P(x, A) \geq \phi(A), \quad x \in C, A \in \mathbb{B}(\mathbb{R}).$$

$$\begin{aligned} &\leq \int_0^{\infty} \frac{1}{1+e^x}(|y| + |x+b|)dU(y)) \\ &\quad + \int_0^{\infty} \frac{e^x}{1+e^x}(|y| + |x-b|)dU(y) \\ \Delta V(x) + |x| &\leq \upsilon + \int_0^{\infty} \frac{1}{1+e^x}|x+b|dU(y) \\ &\quad + \int_0^{\infty} \frac{e^x}{1+e^x}|x-b|dU(y). \end{aligned} \tag{21}$$

For $x \in W$, (21) can be rewritten as

$$\begin{aligned} \Delta V(x) + |x| &\leq \upsilon + \int_0^{\infty} \frac{1}{1+e^x}(|x| + |b|)dU(y) \\ &\quad + \int_0^{\infty} \frac{e^x}{1+e^x}(|x| + |b|)dU(y) \\ \Delta V(x) &\leq \upsilon + b < \infty. \end{aligned}$$

For $x \in W^c$, (21) can be rewritten as

$$\begin{aligned} \Delta V(x) + |x| &\leq \upsilon + \int_0^{\infty} \frac{1}{1+e^{|x|}}(|x| + b)dU(y) \\ &\quad + \int_0^{\infty} \frac{e^{|x|}}{1+e^{|x|}}(|x| - b)dU(y) \\ \Delta V(x) &\leq \upsilon + \frac{1-e^{|x|}}{1+e^{|x|}}b. \end{aligned}$$

Let x^* be the value of x for which

$$\Delta V(x) \leq \upsilon + \frac{1-e^x}{1+e^x}b = -\mu.$$

Then, $x^* = \log\frac{b+\upsilon+\mu}{b-(\upsilon+\mu)}$. For $b > \upsilon + \mu$, there exits a $x^* \in \mathbb{R}$ for which the $\Delta V(x) \leq -\mu$. If x^* is less than b then the drift condition is satisfied for the $W = [-b, b]$. However, if x^* is greater than b then we can consider the set $W = [-b, x^*]$. Hence, for $W = [-b, \max(b, x^*)]$, the drift condition (20) is satisfied. It follows from Lemma 4.1 that W is indeed a small set.

Lemma 4.1: For any d such that $-b \leq d < \infty$, the set $W = [-b, d]$ is a "small" set.

Proof: For $x \in W$,

$$\begin{aligned} P(x, A) &= \frac{1}{1+e^x}\int_{y\in A} dU(y-(x+b)) \\ &\quad + \frac{e^x}{1+e^x}\int_{y\in A} dU(y-(x-b)) \\ &\geq \frac{e^x}{1+e^x}\int_{y\in A} dU(y+b-x). \end{aligned} \tag{22}$$

Since $\frac{dU(u)}{du}$ was assumed to be a non increasing function in u, $\int_{y\in A} dU(y+b-x)$ is non decreasing function in x. Also, $\frac{\exp(x)}{1+\exp(x)}$ is an increasing function in x. Since $x \geq -b$, we can rewrite (22) as

$$\begin{aligned} P(x, A) &\geq \frac{e^{-b}}{1+e^{-b}}\int_{y\in A} dU(y+2b) \\ &\geq \phi(A), \end{aligned} \tag{23}$$

where $\phi(A) := \frac{\exp(-b)}{1+\exp(-b)} \int_{y \in A} dU(y+2b)$. Since there exists a measure ϕ such that $P(x, A) \geq \phi(A), \quad x \in W, A \in \mathbb{B}(\mathbb{R})$, a closed and bounded set $W = [-b, d]$ is a small set. ■

Since W is a "small set" when $b > \upsilon$, from Theorem 4.1 we can conclude that, for $b > \upsilon$, the Markov chain z is positive recurrent. ■

We note that $b > \upsilon$ is a sufficient condition for positive recurrence. Let $\frac{1}{\omega}$ be the mean time between losses. Then $\frac{1}{\omega} = \frac{\upsilon}{\gamma}$. From Prop. 4.2, for z to be positive recurrent

$$\omega > \frac{\gamma}{b}. \tag{24}$$

Therefore, to achieve some fairness, the arrival rate of the losses process has to be greater than $\frac{\gamma}{b}$.

B. Simulation results

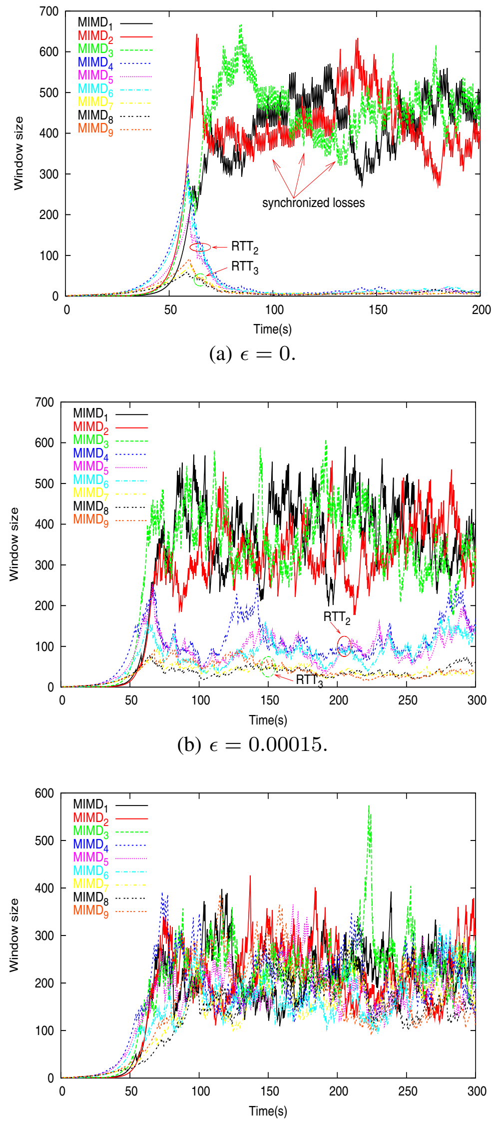
In this subsection we present the results of simulations. Our objective is to verify the analytical result obtained in Prop. 4.1 which noted that sufficient number of asynchronous losses are required so that sessions with different RTTs can share the capacity fairly. In the simulation scenario, nine Scalable TCP sessions shared a link of 200Mbps. Sessions 1,2 and 3 had a RTT of 50ms. Sessions 4,5 and 6 had a RTT of 90ms, and sessions 7,8 and 9 had a RTT of 140ms. The simulations were performed using *ns-2*(version 2.26)[15]. In Fig. 4, the window size is plotted as a function of time for different values of ϵ (i.e., packet drop probability). We note that $\epsilon = 0$ corresponds to only congestion losses which are seen to be not always synchronous. Therefore, in Fig. 4(a) there are asynchronous as well as synchronous losses even though $\epsilon = 0$. However, during periods of synchronous losses (which have been pointed out in the figure) there is short-term unfairness. Even though there are asynchrounous losses due to congestion, the window sizes of the sessions with larger RTTs go to 0. We now induce further asynchronous losses by dropping each packet with probability $\epsilon \neq 0$. In Figs. 4(b) and 4(c) there is a marked improvement in the throughput obtained by sessions with larger RTTs as the loss probability is increased. For small loss probability, there is still some unfairness between sessions with different RTTs. However, for sessions with RTT of 50ms, there is no short-term unfairness as was observed when there were no induced asynchronous losses. For a larger loss probability (i.e., $\epsilon = 0.0003$), sessions share the capacity fairly. This confirms the analytical result which stated that the fairness in MIMD sessions with different RTTs can be achieved by introducing sufficient asynchronous losses. Let η_1, η_2 and η_3 be the total throughput of sessions

TABLE III

THROUGHPUT FOR EACH RTT CLASS AND OVERALL EFFICIENCY

ϵ	η_1(Mbps)	η_2(Mbps)	η_3(Mbps)	$(\eta_1+\eta_2+\eta_3)/C$
0	178	2.8	1	0.91
0.00015	148	25.5	7.14	0.905
0.0003	101	48	28.4	0.89

with RTT of $50ms$, 90ms and 140ms, respectively. In Table III,



(a) $\epsilon = 0$.

(b) $\epsilon = 0.00015$.

(c) $\epsilon = 0.0003$.

Fig. 4. Window evolution MIMD sessions. $\text{RTT}_1 = 50ms$. $\text{RTT}_2 = 90ms$. $\text{RTT}_3 = 140ms$.

the values of throughput and the overall efficiency are given. We note that, for $\epsilon = 0.0003$, the ratio of the throughputs of two different classes, η_i/η_j, are almost in proportion of the respective RTTs. Therefore, we can say that a certain degree of fairness has been acheived at the cost of marginal decrease in efficiency.

V. INTER PROTOCOL FAIRNESS (SAME RTT)

In the second part of this paper, we study the fairness issue when sessions using two different congestion control algorithms share a common link, and the losses are synchronous. Recently Scalable TCP, which uses the MIMD algorithm, has been proposed as an enhancement for TCP in high-speed networks. Situations may, therefore, arise in which a user with Scalable TCP shares a link with a user with standard TCP. Specifically, we study the equilibrium behaviour of the window size, and the throughput obtained by a session of each algorithm at equilibrium in the presence of synchronous losses only. We also look at conditions under which a user of one algorithm can obtain a better throughput than a user of the other algorithm. Previous work (e.g., [6], [8]) mainly studied the behaviour of sessions using the same algorithm.

In this section, we assume that each session has the same RTT, τ. As mentioned in Section II, window-based notation is equivalent to rate-based notation. In the rest of this paper, we use the window-based notation since we are interested in obtaining the equilibrium window sizes for the sessions.

A. System Model

Consider l sessions which share a link of capacity $C\ bits/s$. Each session transmits data using packets of size M bits. Let Λ be the bandwidth-delay product (BDP) of the network. We assume that each session has the same RTT, τ, and that the RTT is mainly determined by the propagation delay and, hence, can be considered to be a constant.

Let $\mathsf{x}(t) = (\mathsf{x}_1(t)\ \mathsf{x}_2(t)\ \ldots\ \mathsf{x}_l(t))$ denote the vector of window sizes of the k sessions at time t. A synchronous loss (i.e., a loss for each session) is assumed to occur at time t if

$$\sum_{i=1}^{l} \mathsf{x}_i(t) > \Lambda. \tag{25}$$

The above condition is equivalent to saying that a synchronous loss occurs when the total number of outstanding packets in the network exceeds the total number of packets that the network can handle.

Without loss of generality, let sessions $1, 2, ..., k$ use the MIMD congestion control algorithm and the rest of the $l-k$ sessions use the AIMD congestion control algorithm. In the absence of losses, the two algorithms increase the window in the following way

$$\mathsf{x}_i(t+\Delta) = \begin{cases} \mathsf{x}(t)\alpha_m^{\Delta/\tau}, & 1 \le i \le k, \\ \mathsf{x}(t) + \alpha_a \frac{\Delta}{\tau}, & k+1 \le i \le l, \end{cases} \tag{26}$$

where α_m and α_a are the increase parameters of the MIMD and the AIMD algorithm, respectively. For example, $\alpha_m = 1.01$ for Scalable TCP, and $\alpha_a = 1$ for standard TCP. Let t_n denote the time instant when the n^{th} congestion signal is received. We note that a congestion signal is generated when a synchronous loss occurs. In response to a congestion signal the two algorithms decrease the window in the following way.

$$\mathsf{x}_i(t_n^+) = \begin{cases} \beta_m \mathsf{x}(t_n), & 1 \le i \le k, \\ \beta_a \mathsf{x}(t_n), & k+1 \le i \le l, \end{cases}$$

where β_m and β_a are the decrease parameters of the MIMD and the AIMD algorithm, respectively. For example, $\beta_m = 0.875$ for Scalable TCP, and $\beta_a = 0.5$ for standard TCP.

Let $x(n)$ denote the window-size vector embedded just after the n^{th} congestion signal is received, i.e, $x(n) = \mathsf{x}(t_n+)$. Let δ_n denote the time between two congestion signals. Since all the sessions are assumed to receive congestion signals at the same instant, we can write the following recursive equation for $x(n)$.

$$x_i(n+1) = \begin{cases} \beta_m x_i(n)\alpha_m^{\delta_n/\tau}, & 1 \le i \le k, \\ \beta_a \left(x_i(n) + \alpha_a \frac{\delta_n}{\tau}\right), & k+1 \le i \le l. \end{cases} \tag{27}$$

B. Bandwidth Sharing

The transient behaviour of the window sizes can be obtained by solving (27) alongwith (25). Given the initial window vector $x(0)$, the time to the first loss t_1 and, hence, $x(1)$ can be computed. This way we can recursively compute $x(n)$. This allows us to obtain the behaviour of the window-size vector and the loss instants before the equilibrium is reached. At equilibrium, δ_n and $x(n)$ will converge to their steady state values denoted by δ^* and ψ, respectively. We are interested in finding the window size, ψ_i, of each session at equilibrium. Then, ψ_i together with δ^* will allow us to obtain the throughput for session i. At equilibrium $x(n)$ would be identical to $x(n+1)$, $x(n+2)$, and so on. Therefore, for each session i, we can obtain ψ_i from (27) as follows.

$$\psi_i = \begin{cases} \beta_m \psi_i \alpha_m^{\delta^*/\tau}, & 1 \le i \le k, \\ \beta_a(\psi_i + \alpha_a \delta^*/\tau), & k+1 \le i \le l. \end{cases} \tag{28}$$

The l equations in (28) are fixed point solutions of the corresponding equations in (27). We now have $n+1$ variables in $\psi_i, 1 \le i \le n$ and δ^*, and n equations. The final equation can be obtained by noting that a loss occurs when condition (25) is satisfied. Therefore, the $(n+1)^{th}$ equation is given by

$$\sum_{i=1}^{k} \frac{\psi_i}{\beta_m} + \sum_{i=k+1}^{l} \frac{\psi_i}{\beta_a} = \Lambda. \tag{29}$$

We note that from any one of the first k equations in (28) we can obtain the value of δ^*. The variables $\psi_i, 1 \le i \le k$ cannot be uniquely determined from these k equations. They will depend on the window vector just after the first synchronous loss. This result is equivalent to the result obtained in [4] where the rate vector of symmetric MIMD sessions was dependent on the initial rate vector. However, the equilibrium window size of the AIMD sessions can be uniquely determined from (28). Therefore,

$$\delta^* = \tau \frac{\log[1/\beta_m]}{\log \alpha_m}, \tag{30}$$

$$\psi_i = \alpha_a \frac{\beta_a}{1-\beta_a} \frac{\log[1/\beta_m]}{\log \alpha_m}, \quad k+1 \le i \le l. \tag{31}$$

From (29) we can, however, obtain the sum of the equilibrium window sizes of the MIMD sessions.

$$\sum_{i=1}^{k} \psi_i = \beta_m \Lambda - \frac{\beta_m}{\beta_a} \sum_{j=k+1}^{l} \psi_j. \tag{32}$$

In order to compute the throughput, η_i, for session i, we divide the time interval δ^* in slots of length τ. We note that, just after a loss instant, the window size of session i is ψ_i. In between two loss instants, the window size of each session increases using the algorithm given in (26). Also, in every RTT (i.e., in every slot), session i transfers packets equivalent to its present window size. Therefore, in between two loss instants, the total number of packets that are transferred by session i can be obtained by summing the window sizes during the δ^*/τ RTTs. As before, we can obtain the throughput η_i for each AIMD session whereas we can obtain the total throughput, η_m, for all the MIMD sessions.

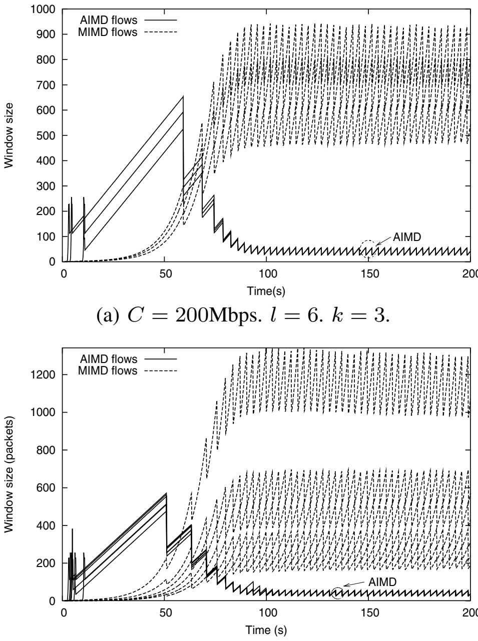
$$\eta_m = \frac{M}{\delta^*}\sum_{i=1}^{k}\psi_i\frac{\alpha_m^{\lfloor\frac{\delta^*}{\tau}\rfloor+1}-1}{\alpha_m-1}, \quad (33)$$

$$\eta_i = \frac{M\alpha_a}{\tau}\left(\psi_i+\alpha_a\frac{\lfloor\frac{\delta^*}{\tau}\rfloor+1}{2}\right),\ k+1\le i\le l. \quad (34)$$

We note that the throughput expressions are approximate since the number of packets transferred in an RTT is an integer whereas ψ_i can take non-integer values. Also, the number of packets transferred in the RTT in which a loss occurs may not be equal to $\lfloor\psi_i\rfloor$.

We can make the following observations from equations (30)-(34). The equilibrium value of the time between two loss instants, δ^*, is independent of the parameters of the AIMD algorithm. It is determined by the RTT, τ, and the parameters of the MIMD algorithm only. The equilibrium window size of the AIMD sessions depends only on the increase and decrease parameters of the two algorithms. Also, the AIMD sessions have the same equilibrium window behaviour and, hence, obtain the same throughput. The rest of the capacity is utilized by the MIMD sessions.

Simulations: We now compare these observations with simulations performed using *ns-2* (version 2.26). Unless stated otherwise, the simulation had the same set of parameters. The MIMD sessions used Scalable TCP, and the AIMD sessions used TCP New Reno. The packet size, M, for each session was set to 1040 bytes (1000 bytes of data + 40 bytes of header). The propagation delay, σ, was taken to be $100ms$. The increase and decrease parameters for the two algorithms were set to $\alpha_m = 1.01$, $\alpha_a = 1.0$, $\beta_m = 0.75$, and $\beta_a = 0.5$. Since the ψ_i for AIMD increases with decrease in β_m, we set β_m to a value smaller than its recommended value so that the AIMD sessions also obtain a certain throughput. From Figs. 5(a) (3 MIMD sessions and 3 AIMD sessions) and 5(b)(6 MIMD sessions and 6 AIMD sessions), we note that the AIMD sessions indeed converge to the same equilibrium window size whereas the equilibrium window size of an MIMD session depends on its window just before the first synchronous loss. The ψ_i for AIMD sessions remains the same even though the link capacity is increased from 200Mbps to 300Mbps and the total number of sessions is increased from six to twelve. Let η_a and ψ_a denote the throughput and the equilibrium window size, respectively, of any one of the AIMD sessions. In Table IV, the analytical and simulation values of δ^*, η_m, η_a, and ψ_a are given. The simulation values are given within parentheses. As predicted in the analysis, the equilibrium window size and the throughput of the AIMD sessions remains unchanged even when the capacity is increase from 200Mbps to 300Mbps, and the total number of sessions is increased from six to twelve.



(a) $C = 200$Mbps. $l = 6$. $k = 3$.

(b) $C = 300$Mbps. $l = 12$. $k = 6$.

Fig. 5. Window evolution of sessions.

TABLE IV
SEVERAL MIMD AND SEVERAL AIMD SESSIONS.

	Link Speed (Mbps)	δ (s)	η_m (Mbps)	η_a (Mbps)	ψ_a (packets)
$l=6$ $k=3$	200	2.83 (3)	164(151)	3.5(3.4)	29 (27)
$l=12$ $k=6$	300	2.83 (3)	238.6(218)	3.5(3.4)	29 (27)

C. Throughput Comparison

We now study the scenario where one MIMD user and AIMD user share the same link. We note that each user can initiate several sessions of the same algorithm. We obtain the condition under which the AIMD user can a obtain better throughput than the MIMD user.

First, we consider the case in which each user initiates only one session. In such a scenario, the window size and the throughput of each session is obtained from (31)-(33) with $l = 2$ and $k = 1$. From (33) and (34), as $\Lambda \to \infty$ (i.e., $C \to \infty$), the ratio of the throughputs, η_2/η_1, goes to 0. This suggests that in high-speed networks, the MIMD user will get most of the capacity. On the other hand, if the BDP of the network is small, the MIMD user will obtain a lower throughput compared to the AIMD session.

Proposition 5.1: Let Λ_l denote the threshold BDP below which an AIMD session will get a better throughput compared to an MIMD session. The threshold value, Λ_l, is given by

$$\Lambda_l = \psi_2\left(\frac{\alpha_a\delta^*}{2\tau}\left(\frac{\delta^*}{\tau}+1\right)\kappa+\frac{1}{\beta_a}\right), \quad (35)$$

where $\kappa = (\alpha_m - 1)/(\alpha_m - \beta_m)$.

Proof: The above relation can be obtained using the inequality $\eta_1 \le \eta_2$ together with (32). ■

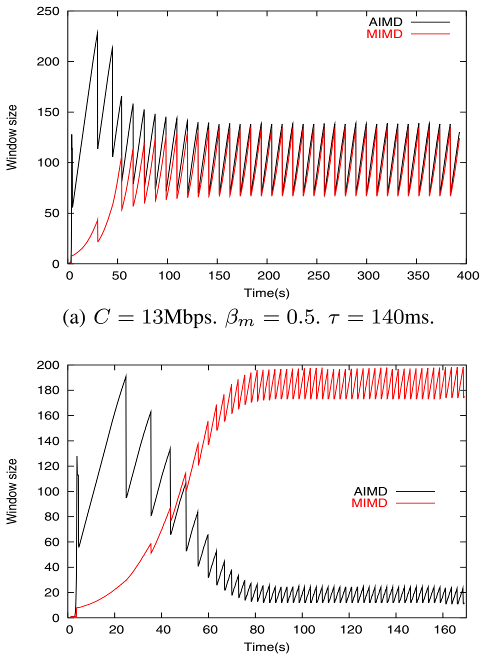
The value of Λ_l depends only on the increase and decrease parameters of the two algorithms. Table V gives the values of Λ_l for different β_m with $\alpha_m = 1.01$ and $\alpha_a = 1$.

TABLE V

Λ_l FOR DIFFERENT VALUES OF β_m.

β_m	0.875	0.75	0.5
Λ_l	47.34	106.6	282.73

In Fig. 6(a), the window evolution is plotted for the two sessions for C = 13Mbps and β_m = 0.5. The BDP, Λ, is less than the Λ_l. The AIMD algorithm obtains a better throughput in this case. In the next set of simulations, we set β_m to its recommended value of 0.875. In Fig. 6(b), the corresponding window evolution is plotted. The effect of increasing β_m is to reduce the share of the AIMD session. In Table VI, a comparison of the values obtained from analysis and simulations is presented. A good match is observed between the analysis and simulations.



(a) C = 13Mbps. $\beta_m = 0.5$. τ = 140ms.

(b) C = 10Mbps. $\beta_m = 0.875$. τ = 140ms.

Fig. 6. Window evolution for one MIMD session and one AIMD session.

From (34), it was observed that the throughput obtained by each AIMD session remains constant whereas the total throughput of the MIMD sessions increases with increase in capacity. An AIMD user may want to obtain throughputs similar to a MIMD user. In this case, the AIMD user may open several sessions in order to improve its observed throughput. Since each AIMD session gets the same throughput independent of the number of AIMD sessions (assuming there is sufficient capacity), an AIMD user can improve its observed throughput by opening multiple sessions.

Proposition 5.2: The smallest number of sessions, ν, with which an AIMD user will obtain a better global throughput compared to single MIMD user is given by

$$\nu = \left\lceil \frac{\beta_m}{h \cdot \kappa} \left(\Lambda - \frac{\psi_a}{\beta_a} \right) \right\rceil, \qquad (36)$$

where $h = \psi_a \alpha_a \sum_{j=0}^{\lfloor \frac{\delta^*}{\tau} \rfloor} j$, ψ_a is the equilibrium window of any one of the AIMD sessions and is defined in (31), and κ is as defined in (35).

The number of sessions, ν, depends only on Λ and the increase and decrease parameters of the two algorithms. Table VII gives the value of ν for different values of Λ for $\beta_m = 0.875$.

TABLE VI

ONE MIMD AND ONE AIMD SESSION. τ = 140MS.

β	Capacity (Mbps)	η_1 (Mbps)	η_2 (Mbps)	ψ_1 (packets)	ψ_2 (packets)
0.875	3	2.22 (2.36)	0.6 (0.52)	66.2 (70.7)	13.41 (11)
	5	4.05 (4.1)	0.75 (0.71)	96.82 (98)	13.41 (12)
	10	8.86 (8.74)	0.92 (0.86)	173.39 (173)	13.41 (12)
0.5	13	4.73 (5.19)	4.92 (5.64)	69.09 (67)	69.6 (69)
	15	6.08 (6.62)	5.04 (5.65)	86.59 (84)	69.6 (68)
	30	16.64 (16.95)	5.47 (5.86)	217.83 (211)	69.6 (69)

TABLE VII

ν FOR DIFFERENT VALUES OF Λ.

$\beta_m = 0.875$	Λ	100	500	1000	10000	50000
	ν	3	18	37	372	1863
$\beta_m = 0.5$	Λ	100	500	1000	10000	50000
	ν	1	3	7	71	358

Similar to the AIMD user, a MIMD user may also try to improve its observed throughput by opening several sessions. Since, from (34), the AIMD user will get a throughput independent of the number of MIMD sessions, the observed throughput of an MIMD user will not improve by opening several sessions. This result is in contrast to the result obtained in (36) where we noted that an AIMD user can improve its observed throughput by opening several sessions.

VI. INTER PROTOCOL FAIRNESS (DIFFERENT RTTS)

In this section we study the effect of different RTT for each session on the equilibrium window behaviour. The notation used and the scenario is the same as in Sec. V. We assume that there exists a BDP, Λ, such that there is a synchronous loss when condition (25) is satisfied. Let τ_i be the RTT of session i. Then, we can rewrite (28) as follows.

$$\psi_i = \beta_m \psi_i \alpha_m^{\delta^*/\tau_i}, \; 1 \le i \le k, \qquad (37)$$
$$\psi_i = \beta_a (\psi_i + \alpha_a \delta^*/\tau_i), \; k+1 \le i \le l. \qquad (38)$$

The expressions for throughput are as follows.

$$\eta_i = \begin{cases} \frac{M}{\delta^*} \psi_i \sum_{j=0}^{\lfloor \frac{\delta^*}{\tau_i} \rfloor} \alpha_m^j, & 1 \le i \le k, \\ \frac{M}{\delta^*} \psi_i \alpha_a \sum_{j=0}^{\lfloor \frac{\delta^*}{\tau_i} \rfloor} j, & k+1 \le i \le l. \end{cases} \qquad (39)$$

For (37) to be consistent δ^* has to be equal to $\frac{\log[1/\beta_m]}{\log[\alpha_m]} \min_{1 \le i \le k} \tau_i$. Therefore, among the MIMD sessions, only the session with the least RTT will have an equilibrium window size different from 0. The equilibrium window of the other MIMD sessions will go to 0. We can, therefore, consider the case where there is only one MIMD session and several AIMD sessions.

For $k = 1$, from (37) and (38), we obtain

$$\begin{aligned}\delta^* &= \tau_1 \frac{\log[1/\beta_m]}{\log \alpha_m},\\ \psi_i &= \alpha_a \frac{\beta_a}{1-\beta_a} \frac{\log[1/\beta_a]}{\log \alpha_a} \frac{\tau_1}{\tau_i}, \quad 2 \le i \le l,\\ \psi_1 &= \beta_m \Lambda - \frac{\beta_m}{\beta_a} \sum_{i=2} l\psi_i.\end{aligned}$$

The inter-loss time depends entirely on the parameters and the RTT of the MIMD session. The effect of different RTTs for the AIMD sessions is to scale ψ_i by a factor of τ_1/τ_i. Therefore, an AIMD session with lower RTT can obtain a better throughput.

A. Several MIMD

If there are l MIMD sessions with different RTTs sharing a link, then the session with the smallest RTT will get all the capacity and the windows for other sessions will go to 0. For l equal to 2, this result was also mentioned in [8]. However, if the sessions have the liberty to choose their increase and decrease parameters then each session can obtain some share of the capacity. Let α_{mi} and β_{mi} be the increase parameter and the decrease parameter, respectively, of the i^{th} MIMD session.

Proposition 6.1: l sessions with different RTTs will have a behaviour similar to l sessions with the same RTT if

$$\tau_i \frac{\log[1/\beta_{mi}]}{\log[\alpha_{mi}]} = \sigma \text{ (a constant)}. \tag{40}$$

The inter-loss time, δ^*, will then be equal to σ.

Proof: For this value of δ^*, (37) is consistent. Therefore, an equilibrium solution exists. Let $x(0)$ is the initial window vector. The time to the first synchronous loss, t_1, can be computed using the condition $\sum_{i=0}^{l} x_i(0)\alpha_{mi}^{t_1/\tau_i} = \Lambda$. We can now compute $x_i(1) = \beta_{mi} x_i(0)\alpha_{mi}^{t_1/\tau_i}$. The next loss will occur after a time δ^*. This can be verified by noting that $\sum_{i=0}^{l} \frac{x_i(1)}{\beta_{mi}} = \Lambda$, and $t_2 = \delta^*$ given by (40) satisfies $\sum_{i=0}^{l} x_i(1)\alpha_m^{t_2/\tau} = \Lambda$. From this we obtain $t_2 = \delta^*$. Since t_n is the equilibrium value of the inter-loss time, $x_i(1)$ will also be the equilibrium value of ψ_i. Now, the system will be similar to the same RTT case where the equilibrium window vector is the same as the window vector just after the first synchronous loss. ■

Proposition 6.1 gives a condition on setting the increase and decrease parameters of the MIMD algorithm as a function the RTT in order not to be extremely unfair.

VII. Conclusions

In the first part of the paper, we studied the fairness in sessions using MIMD congestion control algorithm. For sessions with the same RTT, it was observed that there was extreme unfairness when the asynchronous losses were rate independent. It was shown that fair sharing could be achieved by introducing a stream of rate dependent losses. For sessions with different RTTs, it was observed that the arrival rate of these rate dependent losses had to be greater than a certain minimum rate in order to achieve fairness. Therefore, in networks with sessions using MIMD algorithms, a stream of rate dependent losses, using, for example, some buffer management scheme, would be necessary to ensure fair sharing. In the second part of the paper, we studied capacity sharing between MIMD sessions and AIMD sessions. For a given set of parameters, it was noted that the throughput of an AIMD session was independent of the BDP, and that the rest of the capacity was utilized by the MIMD sessions. In networks with BDP less than a threshold value, it was observed that one AIMD session obtained a better throughput than one MIMD session. It was also observed that an AIMD user could open multiple sessions in order to improve its observed throughput whereas for the MIMD user the throughput was invariant to the number of sessions it opened.

Ackowledgements

This work was partially supported by the EURO NGI Network of Excellence, and by the *Indo–French Center for Promotion of Advanced Research (IFCPAR/CEFIPRA)* under research contract number 2900–IT.

References

[1] S. Floyd. HighSpeed TCP for Large Congestion Windows. RFC 3649, Experimental, December 2003.

[2] T. Kelly. Scalable TCP: Improving Performance in Highspeed Wide Area Networks. *Computer Comm. Review*, 33(2):83–91, April 2003.

[3] G. Vinnicombe. On the Stability of Networks Operating TCP-like Congestion Control. In *Proceedings of the IFAC World Congress*, 2002.

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

[5] S. Gorinsky. Feedback Modeling in Internet Congestion Control. In *Proceedings of the NEW2AN*, February 2004. Also see a techincal report at http://www.arl.wustl.edu/~gorinsky/ TR2002-39.ps.

[6] G. Hasegawa, M. Murata, and H. Miyahara. Fairness and Stability of congestion control mechanisms of TCP. *Telecommunication Systems*, November 2000.

[7] D. Loguinov and H. Radha. End-to-End Rate-Based Congestion Control: Convergence Properties and Scalability Analysis. *IEEE/ACM Transactions on Networking*, 11(4), August 2003.

[8] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control (BIC) for Fast Long-Distance Network. In *Proceedings of the IEEE INFOCOM*, March 2004.

[9] A. Budhiraja, F. Hernndez-Campos, V.G. Kulkarni, and F. D. Smith. Stochastic Differential Equation for TCP Window Size: Analysis and Experimental Validation. *Prob. in the Engg. and Informational Sciences*, 18:111–140, 2004.

[10] O. Ait-Hellal, E. Altman, D. Elouadghiri, M. Erramdani, and N. Mikou. Performance of TCP/IP: the case of two Controlled Sources. In *Proceedings of the ICCC*, pages 469–477, 1997.

[11] G. Fayolle, V.A. Malyshev, and M.V. Menshikov. *Topics in the Constructive Theory of Countable Markov Chains*. Cambridge University Press, 1995.

[12] L. Kleinrock. *Queueing Systems Volume I: Theory*. Wiley & sons, 1975.

[13] J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains*. Springer-Verlag, New York, 2nd edition, 1976.

[14] S.P. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*. Springer, London, 1993.

[15] S. McCanne and S. Floyd. ns: Network Simulator. Available at http://www.isi.edu/nsnam/ns/.