# Semi-Markov Modeling for Bandwidth Sharing of TCP Connections with Asymmetric AIMD Congestion Control

Cheng Peng Fu, Chuan Heng Foh, Chiew Tong Lau, Zhihong Man, and Bu Sung Lee
School of Computer Engineering
Nanyang Technological University
Singapore
aschfoh@ntu.edu.sg

*Abstract*— **This paper presents a semi-Markov model that evaluates the performance of TCP connections with asymmetric *Additive Increase and Multiplicative Decrease* (AIMD) congestion control settings involved in sharing of a common drop-tail router. We study the fairness of the connections and their individual bandwidth utilizations as well as packet loss rates. We confirm that certain asymmetric AIMD settings may achieve fairness in bandwidth sharing. We also found that while connections with asymmetric AIMD settings operate at different bandwidth utilizations, they generally experience similar packet loss rate.**

## I. Introduction

Transmission Control Protocol (TCP) is the most widely deployed transport protocol in computer networks. TCP implements a distributed algorithm known as *Additive Increase and Multiplicative Decrease* (AIMD) mechanism to manage network congestion. Many modern TCP designs employ modified AIMD mechanisms to enhance bandwidth usage, examples of these designs include Fast TCP [1], High-Speed TCP [2], Scalable TCP [3], and TCP Veno [4].

There is a number of analytical studies of AIMD appeared in the literature. Padhye *et al.* [5] formulate the relationship between the mean window size and the packet loss rate of a TCP connection given a certain AIMD setting. With the result, given a certain packet loss rate of a TCP connection, the mean window size and bandwidth utilization can be computed. Other studies that consider the bandwidth sharing behavior among a number of TCP connections can be found in [6], [7], [10] and references therein. However, those studies evaluate only connections with a symmetric setting of a particular network congestion algorithm.

Yang *et al.* [8] and Floyd *et al.* [9] analyze the fairness of bandwidth sharing among several TCP connections with asymmetric AIMD settings. Their works assume that if two TCP connections with asymmetric AIMD settings achieve fairness in bandwidth sharing, then they packet loss rates are the same. Using this assumption with simulation justification, asymmetric AIMD setting pairs that achieve fairness are found using the result of [5].

In this paper, we present an analytical model that evaluates the bandwidth utilization and the packet loss rate of each

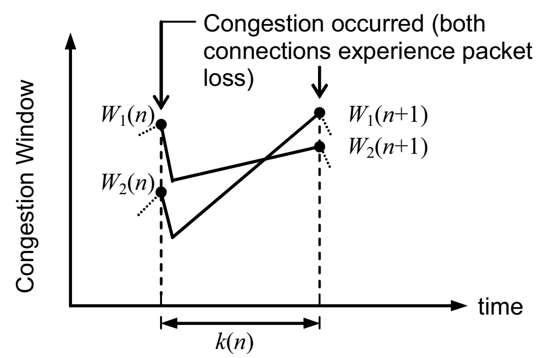


Fig. 1. Packet loss rate of Reno/AIMD$(a, b)$.

TCP connection when a number of TCP connections with different asymmetric AIMD settings sharing a common router. By applying the condition that two TCP connections with asymmetric AIMD settings utilize the same bandwidth, our model reproduces numerically the results reported in [8], [9]. Our study presented here also serves as an analytical framework to justify the assumption used in [8], [9]. Moreover, our model also determines the packet loss rate experienced by each individual TCP connection.

We briefly describe our modeling approach as follows. Let $C$ be the capacity of a drop-tail router shared among several TCP connections, and each TCP connection implements a general *Additive and Multiplicative Increase and Decrease* (AMIND) congestion control mechanism. Define $x_i(t)$ to be the load that connection $i$ places to the shared router at time $t$, employing discrete timescale, connection $i$ will adjust its load at the next *round trip time* (RTT) step by

$$x_i(t+1) = \begin{cases} a_i + c_i x_i(t), & X(t) \le C \\ d_i + b_i x_i(t), & \text{otherwise} \end{cases} \quad (1)$$

where $X(t) = \sum_i x_i(t)$. Note that the general AMIND can be described by four parameters, namely $a$, $b$, $c$ and $d$, which gives rise to several classes of congestion control algorithms. The commonly used congestion control algorithm is based on AIMD where $a > 0$, $0 \le b < 1$, $c = 1$, and $d = 0$.

Observing the congestion window at the time just before one.

To appear in IEEE GLOBECOM 2007

or more losses occur at the shared router, such an evolution of congestion window forms a stochastic process. Defining the $n$-th epoch at when the congestion window is observed, considering $m$ simultaneous TCP connections, the evolutions of congestion windows of all connections can be modeled into an $m$-dimensional process, which is $\{W_1(n), W_2(n), ..., W_m(n)\}$ where $0 \leq W_i \leq C$ and $i = 0, 1, ..., m$. It is not difficult to see that the process is a semi-Markov Process since the evolution of the congestion window of each connection as well as the one-step transition duration depend only on the current congestion windows. Fig. 1 depicts typical evolutions of congestion windows between two successive epochs assuming two TCP connections.

Viewing from a particular connection, at $n$-th epoch when the capacity of the shared router has been utilized, since all TCP connections will definitely increase their congestion windows in the next RTT step, packet loss will occur. If a connection suffers packet losses, the connection will first reduce its congestion window then increase gradually until the next epoch. Let $k(n)$ be the number of RTT steps between $n$-th and $(n+1)$-th epochs, according to (1), congestion window of a particular connection, say connection $i$, at the next epoch, $W_i(n+1)$, can be expressed as

$$W_i(n+1) = \sum_{j=0}^{k(n)-2} a_i c_i^j + c_i^{k(n)-1}(d_i + b_i(a_i + c_i W_i(n))). \quad (2)$$

Similarly, if connection $i$ did not experience any packet loss, which is possible since the shared router may drop packets of other connections, the connection will continue to increase its congestion window by another $k(n)$ RTT steps. Based on (1)

$$W_i(n+1) = \sum_{j=0}^{k(n)-1} a_i c_i^j + c_i^{k(n)} W_i(n) \quad (3)$$

and $k(n)$ can be easily solved by the fact that the sum of all the congestion windows at the next epoch must reach the capacity of the router, that is

$$\begin{cases} \text{maximize } k(n) \\ \text{subject to } \sum_i W_i(n+1) \leq C \end{cases} \quad (4)$$

where $k(n)$ is a nonzero positive integer.

In this paper, we focus on the commonly used AIMD where $c_i = 1$ and $d_i = 0$. With this setting, based on (2)-(3), $W_i(n+1)$ can be rewritten into a simpler form as

$$W_i(n+1) = (k(n) - 1)a_i + b_i(a_i + W_i(n)) \quad (5)$$

if the connection experiences a packet drop at a congestion epoch, or

$$W_i(n+1) = k(n)a_i + W_i(n) \quad (6)$$

if the connection experiences no packet drop.

In the following section, we introduce our analytical model for the study of two TCP connections implementing asymmetric AIMD settings. Section III presents the throughput performance and packet loss rate of two TCP connections with asymmetric AIMD settings as well as the friendliness study. Some important conclusions are drawn in Section IV.

## II. The Semi-Markov Model for Two Connections Sharing a Common Router

To study two TCP connections implementing AIMD sharing a common router, we adopt discretized congestion window and develop a bidimensional semi-Markov Chain describing the congestion window evolution of the two connections. Similar to [11], we assume that the probability of packet loss is proportional to the percentage of bandwidth usage of the common router during when the capacity is fully utilized.

Define $l$ to be the event of packet loss occurs when the capacity is fully utilized. For two connections, there are three possibilities, which are $l = 1$ denoting only connection 1 suffers packet losses, $l = 2$ denoting only connection 2 suffers packet losses, and $l = 3$ denoting both connections suffer packet losses. Let $q_l$ be the probability of the occurrence of event $l$. Given (5)-(6), the nonnull one-step transition probabilities[1] of the bi-dimensional semi-Markov Chain can be described by

$$\begin{aligned} &\Pr\{\hat{\omega}_1, \hat{\omega}_2 | \omega_1, \omega_2\} = q_1, \\ &\quad \hat{\omega}_1 = \lfloor (k_1(\omega_1, \omega_2) - 1) a_1 + b_1(a_1 + \omega_1) \rfloor, \\ &\quad \hat{\omega}_2 = \lfloor k_1(\omega_1, \omega_2) a_2 + \omega_2 \rfloor, \end{aligned}$$

$$\begin{aligned} &\Pr\{\hat{\omega}_1, \hat{\omega}_2 | \omega_1, \omega_2\} = q_2, \\ &\quad \hat{\omega}_1 = \lfloor k_2(\omega_1, \omega_2) a_1 + \omega_1 \rfloor, \\ &\quad \hat{\omega}_2 = \lfloor (k_2(\omega_1, \omega_2) - 1) a_2 + b_2(a_2 + \omega_2) \rfloor, \end{aligned}$$

$$\begin{aligned} &\Pr\{\hat{\omega}_1, \hat{\omega}_2 | \omega_1, \omega_2\} = q_3, \\ &\quad \hat{\omega}_1 = \lfloor (k_3(\omega_1, \omega_2) - 1) a_1 + b_1(a_1 + \omega_1) \rfloor, \\ &\quad \hat{\omega}_2 = \lfloor (k_3(\omega_1, \omega_2) - 1) a_2 + b_2(a_2 + \omega_2) \rfloor, \end{aligned}$$

where

$$q_1 = \frac{\omega_1^2}{\omega_1^2 + \omega_2^2 + \omega_1\omega_2}$$

$$q_2 = \frac{\omega_2^2}{\omega_1^2 + \omega_2^2 + \omega_1\omega_2}$$

$$q_3 = \frac{\omega_1\omega_2}{\omega_1^2 + \omega_2^2 + \omega_1\omega_2}$$

and by (4), $k_l(\cdot)$ can be conservatively estimated as

$$k_1(\omega_1, \omega_2) = \left\lfloor \frac{C - (b_1(a_1+\omega_1) - a_1) - \omega_2}{a_1 + a_2} \right\rfloor$$

$$k_2(\omega_1, \omega_2) = \left\lfloor \frac{C - \omega_1 - (b_2(a_2+\omega_2) - a_2)}{a_1 + a_2} \right\rfloor$$

$$k_3(\omega_1, \omega_2) = \left\lfloor \frac{C - (b_1(a_1+\omega_1) - a_1) - (b_2(a_2+\omega_2) - a_2)}{a_1 + a_2} \right\rfloor.$$

Let $\pi_{u,v}$ be the stationary state probability distribution of the Markov Chain, to be precise, $\pi_{u,v} = \lim_{n\to\infty} \Pr\{W_1(n) = u, W_2(n) = v\}$. $\pi_{u,v}$ can be solved numerically by the following balance equation set

$$\pi_{u,v} = \sum_{\omega_1=0}^{C} \sum_{\omega_2=0}^{C} \sum_{l=1}^{3} \left( \pi_{\omega_1,\omega_2} q_l \delta_{u, f_l(\omega_1,\omega_2)} \delta_{v, g_l(\omega_1,\omega_2)} \right)$$

[1]We adopt the short notations: $q_l = q_l(\omega_1, \omega_2)$, $\Pr\{\hat{\omega}_1, \hat{\omega}_2 | \omega_1, \omega_2\} = \Pr\{W_1(n+1) = \hat{\omega}_1, W_2(n+1) = \hat{\omega}_2 | W_1(n) = \omega_1, W_2(n) = \omega_2\}$, and $k_l(\omega_1, \omega_2) = k_l(n | W_1(n) = \omega_1, W_2(n) = \omega_2)$.

where $\delta$ denotes Kronecker delta, $u, v \in \{0, 1, \dots, C\}$, with

$$\begin{aligned}
&\pi_{0,0} = 0,\\
&f_1(\omega_1,\omega_2) = \lfloor (k_1(\omega_1,\omega_2) - 1) a_1 + b_1(a_1 + \omega_1) \rfloor,\\
&g_1(\omega_1,\omega_2) = \lfloor k_1(\omega_1,\omega_2) a_2 + \omega_2 \rfloor,\\
&f_2(\omega_1,\omega_2) = \lfloor k_2(\omega_1,\omega_2) a_1 + \omega_1 \rfloor,\\
&g_2(\omega_1,\omega_2) = \lfloor (k_2(\omega_1,\omega_2) - 1) a_2 + b_2(a_2 + \omega_2) \rfloor,\\
&f_3(\omega_1,\omega_2) = \lfloor (k_3(\omega_1,\omega_2) - 1) a_1 + b_1(a_1 + \omega_1) \rfloor,\\
&g_3(\omega_1,\omega_2) = \lfloor (k_3(\omega_1,\omega_2) - 1) a_2 + b_2(a_2 + \omega_2) \rfloor,
\end{aligned}$$

and

$$\sum_{u=0}^{C} \sum_{v=0}^{C} \pi_{u,v} = 1.$$

The mean congestion window between two successive epochs can be computed by conditioning and unconditioning the states and the transitions of the semi-Markov Chain. At a particular state, say $\{\omega_1, \omega_2\}$, the quantity $k_l(\omega_1, \omega_2)$ describes the steady state transition duration in terms of the number of RTT steps. As depicted in Fig. 1, after the first RTT during a congestion epoch, connection $i$, adjusts its congestion window to, say, $\tilde{\omega_i}$, depending on whether the connection has suffered packet losses. After then, its congestion window will continue to increase with a rate $a_i$ for another $(k_l(\omega_1, \omega_2) - 1)$ RTT steps. Hence the mean congestion window during this state can be computed by $\tilde{\omega_i} + 0.5 \cdot a_1(k_l(\omega_1, \omega_2) - 1)$. Define $W_{i|l,\omega_1,\omega_2}$ to be the congestion window conditioned upon event $l$ at state $\{\omega_1, \omega_2\}$, based on the above discussion, we obtain

$$W_{i|l,\omega_1,\omega_2} = \begin{cases} (a_i + \omega_i) + \frac{a_i(k_l(\omega_1,\omega_2)-1)}{2}, \\ \qquad \text{if } (i=1, l=2) \text{ or } (i=2, l=1) \\ b_i(a_i + \omega_i) + \frac{a_i(k_l(\omega_1,\omega_2)-1)}{2}, \\ \qquad \text{otherwise} \end{cases}.$$

Using the above results, the mean congestion window for connection $i$, denoted $\overline{W}_i$, can be computed by

$$\overline{W}_i = \sum_{\omega_1=0}^{C} \sum_{\omega_2=0}^{C} \sum_{l=1}^{3} \left( \pi_{\omega_1,\omega_2} q_l W_{i|l,\omega_1,\omega_2} \right). \tag{7}$$

Recall that $k_l(\omega_1, \omega_2)$ is the RTT steps given event $l$ at state $\{\omega_1, \omega_2\}$, the total number of transmitted packets by a connection is then $W_{i|l,\omega_1,\omega_2} k_l(\omega_1, \omega_2)$. Assuming that at most one packet is dropped for a connection during a congestion epoch, connection $i$ will suffer a packet drop when the event $l = i$ or $l = 3$ occurs. Define $p_i$ to be the average packet loss rate of connection $i$, we obtain

$$p_i = \frac{\sum_{\omega_1=0}^{C} \sum_{\omega_2=0}^{C} \pi_{\omega_1,\omega_2}(q_i + q_3)}{\sum_{\omega_1=0}^{C} \sum_{\omega_2=0}^{C} \sum_{l=1}^{3} \left( \pi_{\omega_1,\omega_2} q_l W_{i|l,\omega_1,\omega_2} k_l(\omega_1,\omega_2) \right)}. \tag{8}$$

Padhye *et al.* [5] give a formula that calculate the mean window size if a packet loss rate is known. The above result provides specifically the packet loss rate due to congestion. We may determine the mean window size using the result from [5], or directly from (7).

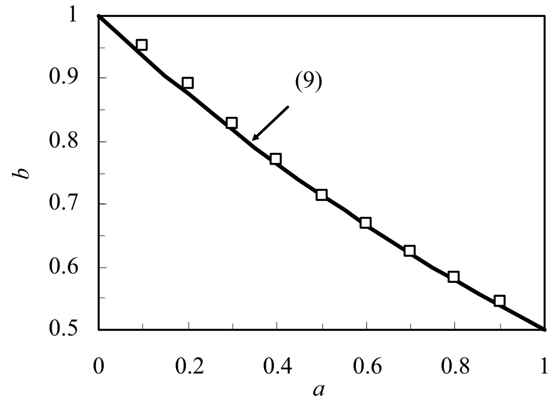


Fig. 2. TCP Fairness Curve.

## III. Numerical Results and Discussions

In this section, we presents numerical results of our model and study of the TCP throughput performance with asymmetric AIMD congestion control setting. We illustrate the AIMD friendliness of two TCP connections, then bandwidth utilizations of two TCP connections when fairness is not reached are presented. We also show the packet loss rates of two TCP connections for various asymmetric AIMD settings.

### A. AIMD Friendliness

Independent studies by Yang *et al.* [8] and Floyd *et al.* [9] have shown that TCP connections with different AIMD parameter sets of $(a, b)$ pair may utilize the same amount of bandwidth when sharing a common router. Assume that all connections experience the same packet loss rate, a TCP connection implementing AIMD, denoted AIMD$(a, b)$, may compete fairly with TCP Reno whose $a = 1$ and $b = 0.5$, when its AIMD settings satisfy the following condition

$$a = \frac{3(1-b)}{1+b}. \tag{9}$$

Simulation has been carried out to validate the above result and support the employed assumption [8], [9]. Using a different modeling approach, here we conduct a similar study that finds $(a, b)$ pair which gives the same mean window size with that of TCP Reno. In other words, we numerically find $a_1, b_1$ given $a_2 = 1, b_2 = 0.5$ that satisfy $\overline{W}_1 = \overline{W}_2$. Our numerical results (shown in symbols), along with results given in (9), are plotted in Fig. 2. It is worth noting that unlike [8], [9], our model does not require the assumption that connections experience the same packet loss rate, instead, packet dropping behavior is described in our model. In fact, the agreement between our results and that of the earlier works also suggests that connections indeed suffer similar packet loss rate under a fair bandwidth competing scenario.

### B. Throughput Performance

To further evaluate the bandwidth sharing behavior of two asymmetric TCP connections, we investigate Reno/AIMD$(0.5, b)$ combination. Precisely, a common router

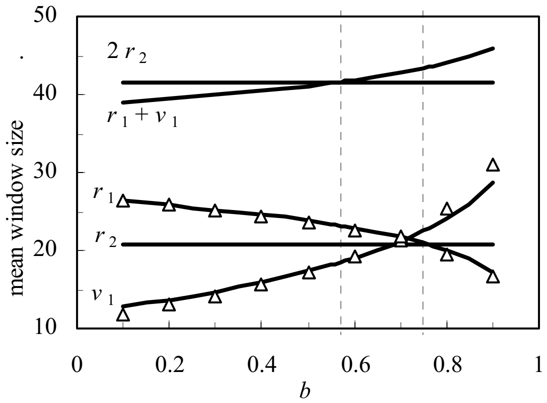


Fig. 3. Bandwidth utilization of Reno/AIMD$(0.5, b)$ and Reno/Reno.

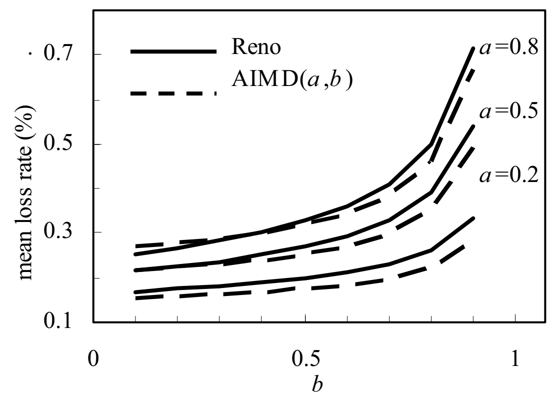


Fig. 4. Packet loss rate of Reno/AIMD$(a, b)$.

of capacity $C = 50$ is shared by a TCP Reno, that is, AIMD$(1, 0.5)$, and a TCP connection of AIMD$(0.5, b)$. Let $r_1$ and $v_1$ be the mean window sizes of the TCP Reno and AIMD$(0.5, b)$, respectively. Another considered scenario for comparison is Reno/Reno combination. Define $r_2$ to be the mean window size of the TCP Reno under Reno/Reno combination. Results of this study are presented in Fig. 3.

Fig. 3 plots $r_1$, $v_1$ and $r_2$ as a function of $b$. The aggregated throughput of Reno/AIMD$(0.5, b)$ combination, i.e. $r_1 + v_1$, as well as that of Reno/Reno combination, i.e. $2r_2$, are also depicted. We notice several findings from the numerical results, firstly, $r_1 + v_1 > 2r_2$ when $b \geq 0.57$. This suggests that Reno/AIMD$(0.5, b)$ combination may produce higher network utilization than that of Reno/Reno combination. Secondly, $r_1 > r_2$ when $b \leq 0.75$, which indicates that the performance of a TCP Reno may be improved when its competing connection implements some other AIMD$(a, b)$. This throughput gain of Reno/AIMD$(a, b)$ combination can be explained by the out-of-phase synchronization phenomenon [12].

Combining the above two conditions, we see that when $0.57 \leq b \leq 0.75$, we achieve $r_1 + v_1 > 2r_2$ and $r_1 > r_2$. Furthermore in the range where $0.7 \leq b \leq 0.75$, we notice that both $r_1, v_1 > r_2$. Our fairness study in the previous subsection suggests that Reno/AIMD$(0.5, 0.7141)$ combination results in fairness, however, here we further notice that a $(a, b)$ pair that achieves fairness may not necessarily maximize the aggregated throughput, a connection, for instance AIMD$(0.5, 0.75)$, represents a better choice since it results in a slightly higher aggregated throughput than that only maintain fairness when competing with a TCP Reno, and yet it ensures that the throughput of the TCP Reno in Reno/AIMD$(0.5, 0.75)$ combination is not degraded compared to a usual Reno/Reno combination, i.e. $r_1 > r_2$.

### C. Packet Loss Rate

The numerical results of (8) are plotted in Fig. 4 showing the packet loss rate of Reno/AIMD$(a, b)$ combination for $a = 0.2, 0.5$ and $0.8$, with a range of $b$. One interesting observation from the results is that even there is a great difference between the bandwidth utilization of the two connections implementing different AIMD settings, the packet loss rate experienced by the two connections are similar. For example, for $b = 0.1$, according to Fig. 3, $v_1$ and $r_1$ utilizes different bandwidth, however, as can be seen in Fig. 4, the packet loss rate experienced by $v_1$ is similar to that experienced by $r_1$.

The relationship between the numerical results obtained from (7) and (8) can also be confirmed by that presented in [5]. Given (8), we use the formula described in [5] to calculate the mean window size for each connection. The results for Reno/AIMD$(0.5, b)$ combination are drawn (in symbols) in Fig. 3, which match that obtained from (7). With our model, the precise packet loss rate due to the network congestion can be obtained which is impossible with that of [5].

### D. TCP Reno and TCP Veno

This subsection extends our results to study the throughput sharing between Reno and TCP Veno [4] which a modern TCP design that addresses the performance degradation of TCP Reno operating across a noisy wireless channel. The detailed design of TCP Veno is given in [4].

In brief, TCP Veno uses the number of backlogged packets at a router to estimate the network state to be either in a congestive state or a non-congestive state. TCP Veno then employs different AIMD settings to operate in the two different states. In the non-congestive state, TCP Veno uses AIMD$(1, 0.8)$, whereas in the congestive state, TCP Veno uses AIMD$(0.5, 0.5)$. Since a packet loss in the non-congestive state suggests that the loss is likely to be a random loss due to, such as, a noisy channel, a less aggressive drop of the congestion window is used, specifically, TCP Veno sets $b = 0.8$ in the non-congestive state. Moreover, when the network has reached the congestive state, the reduction in the congestion window increasing rate helps ease the network congestion. In TCP Veno, $a = 0.5$ in congestive state.

Applying TCP Veno's AIMD strategy in our model, in the congestive state and the non-congestive state, we have the bandwidth sharing of Reno/AIMD$(0.5, 0.5)$ and Reno/AIMD$(1, 0.8)$ combinations, respectively. As can be seen from Fig. 3 the throughput ratio of Reno to AIMD$(0.5, 0.5)$ is 23.4 to 17.4. In other words, Veno utilizes

0.74 of that Reno utilizes in the congestive state. This helps Veno reduce the chances of packet drop due to buffer overflow. On the other hand, in the non-congestive state, the throughput ratio of Reno to AIMD$(1, 0.8)$ is 15.7 to 29.1. That is, Reno utilizes only 0.53 of what Veno utilizes in the non-congestive state. This helps promotes the bandwidth usage of Veno during when there is no threat of buffer overflow.

Based on the above numerical results, Reno/Veno bandwidth sharing gives the ratio:

$$\frac{W_{Reno}}{W_{Veno}} = \frac{23.4T + 15.7(1-T)}{17.4T + 29.1(1-T)}$$

where $T$ is the percentage of time that Veno operates in the congestive state. This gives the total bandwidth utilization of range between 40.8 and 44.8, which is similar to the bandwidth utilization for the Reno/Reno combination reported in Fig. 3. This result is consistent with our previous finding reported in [12] that TCP Veno operates friendly with Reno in terms of bandwidth usage.

## IV. Conclusions

In this paper, we developed a Semi Markov model for the study of the bandwidth sharing of several TCP with asymmetric AIMD setting. We confirmed that Reno/AIMD$(a, b)$ combination can achieve fair sharing of bandwidth. We discovered that certain Reno/AIMD$(a, b)$ combinations, which may not result in fairness, may achieve a slightly higher aggregated throughput.

In addition, we found that the two connections in a certain Reno/AIMD$(a, b)$ combination experience similar packet loss rate even if their bandwidth utilization can be of great different. Using our developed model, we studied the bandwidth sharing behavior between Reno and Veno that implements a different AIMD scheme.

## References


[1] Cheng Jin, David X. Wei, and Steven H. Low, "TCP FAST: motivation, architecture, algorithms,performance," in *Proc. IEEE Infocom*, 2004.

[2] S. Floyd, "HighSpeed TCP for large congestion windows," RFC 3649, Dec. 2003.

[3] T. Kelly, "Scalable TCP: improving performance in highspeed wide areanetworks," *ACM Computer Communication Review*, no. 2, April 2003.

[4] C. P. Fu, S. C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks", *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, Feb 2003 pp. 216-228.

[5] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," in *Proc. ACM SIGCOMM*, 1998.

[6] F. Baccelli and D. Hong, "The AIMD Model for TCP Sessions Sharing a Common Router," in *39th Annual Allerton Conf. on Communication, Control and Computing*, October 2001.

[7] A. Akella, S. Seshan, R. Karp, S. Shenker, and C. Papadimitriou, "Selfish Behavior and stability of the Internet: A Game Theoretic Analysis of TCP," in *Proc. ACM SIGCOMM*, August 2002.

[8] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," in *Proc. ICNP*, 2000.

[9] S. Floyd, M. Handley, and J. Padhye (May 2000), "A Comparison of Equation-Based and AIMD Congestion Control," *ACIRI Technical Report*.

[10] L. Massoulie and J. Roberts, "Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet," *IEEE Transactions on Networking*, vol. 10, no. 3, June 2002, pp. 320-328.

[11] G. Hasegawa, K. Kurata, and M. Murata, "Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet," in *Proc. ICNP*, 2000.

[12] C. L. Zhang, C. P. Fu, M.-T. Yap, C. H. Foh, K. K. Wong, C. T. Lau, M. K. Lai, "Dynamics Comparision of TCP Veno and Reno," in *Proc. IEEE GLOBECOM*, November 2004.