

# Modeling TCP Window Evolution Process with both Discrete and Fluid Models

Guohan Lu, Xing Li

Dept. of Electrical Engineering

Tsinghua University, 100084

Beijing, P.R. China

[lguohan00@mails.tsinghua.edu.cn](mailto:lguohan00@mails.tsinghua.edu.cn), [xing@cernet.edu.cn](mailto:xing@cernet.edu.cn)

**Abstract**—Fluid model and discrete-packet model are two approaches to model TCP, where different loss processes are used in different models. In this paper, we present both a discrete SMP model and a fluid model to model the TCP window evolution process. Loss process is I.I.D. in the SMP model, and is a poisson process with window size dependent arrival rate in the fluid model. Analysis shows that the discrete SMP model approaches to the fluid model as  $p \rightarrow 0$ , which sets up a connection between discrete-packet models and fluid models. It allows us to understand the relationship between several existing fluid models and discrete-packet models, especially the relationship between different loss processes. In addition, we analyze the impact of maximum window size limitation ( $W_{lim}$ ) to the throughput, and a simple relation between the  $W_{lim}$  and average window size ( $W_{aver}$ ) is obtained. Finally, a set of experiment is conducted over the real Internet and in simulation to validate the model.

**Keywords**—TCP window evolution process, SMP, Fluid Model

## I. INTRODUCTION

The modeling of TCP protocol receives remarkable research attention within the research community in the last decade. Closed-form expression for the TCP throughput has been obtained. TCP is a window-based congestion control protocol. Congestion control is realized by adjusting window size in reaction to the network congestion, reducing the window when congestion and raising it on the contrary. The dynamics of TCP window evolution process reflects TCP's congestion control behavior with more details than the throughput, such as window size distribution.

The window evolution process  $W(t)$  is the outstanding packets in the network of a connection as a function of time  $t$ . A stochastic model for  $W(t)$  consist of a model for the TCP congestion control mechanism and a model for the loss process.

As for the model of TCP congestion control mechanism, there exists two approaches. First, a fluid model is used in [1],[2],[3].  $W(t)$  is assumed to increase linearly and continuously as a function of time until a loss occurs, then it is divided by two. Another approach is to model as a Semi-Markov Process(SMP).  $W(t)$  jumps up from  $W = n$  to  $W = n+1$  and stays in  $W = n+1$  for a period, then takes

another jump to  $W = n+2$  until a loss occurs, then it is jumps down to  $[W/2]$ . As TCP sends packets one by one, this discrete-packet model is more close to the real window evolution process than the fluid model.

Loss process is rather important in modeling TCP. Altman et al find in [1] the TCP throughput is varies under different loss process. As for the model of loss process, there also exist two different methods. First, usually the loss process is measured by the loss rate  $p$  as in [4], [5], [6], [7], and it is assumed as I.I.D losses. Here, the loss rate  $p$  is a packet-based measurement. Another approach is to model loss process as a random point process, where the inter-loss time interval  $\{S_n\}$  is considered. Usually, the random point process is assumed as a poisson process with arrival rate  $\lambda$ . Here, the arrival rate  $\lambda$  is a time-based measurement.

We are really concerned about the relationship between two loss processes of different measures. The I.I.D loss process is random process, it definitely does not imply a deterministic time interval between two losses, neither does it correspond to the exponential inter-loss time interval. As the TCP window size grows, if the RTT is constant, the time interval between two successive packets decreases. For I.I.D loss process, the TCP with larger window size will experience losses more frequently. The inter-loss time interval decreases as window grows. So, we believe that the arrival rate of random point process corresponding the I.I.D loss process must be dependent on the window size.

In this paper, we first propose a SMP Markov model with I.I.D loss process to model the TCP window evolution process. With this model, we obtain the TCP window distribution in steady-state. Then we propose a fluid model with window dependent poisson arrival loss process. The arrival rate  $\lambda$  of the poisson loss process is proportional to the window size,  $\lambda = W\lambda_0$ . What we find between the two TCP models is that the window distribution of the SMP model approaches that of the fluid model as  $p \rightarrow 0$ , where loss rate  $p$  has a simple relation to the arrival rate  $\lambda_0$ ,  $\lambda_0 = p/RTT$ . Thus, we have established a corresponding relationship between the discrete model and the fluid model, between the loss rate and the loss interval. The fluid model gives us a limiting behavior of the discrete SMP model.

With the help of the SMP model, we find that maximum window size limitation should be twice as large as the

average window size achieved by  $W_{lim} \rightarrow \infty$  in order not to reduce the TCP transmission rate evidently.

The paper is organized as follows. We discuss the related work in the next section. In section 3, we present the SMP model of TCP window evolution process. In section 4, we present the fluid model and discuss the relation between these two models. Section 5 is the model validation, and finally the conclusion and future work.

## II. RELATED WORK

In discrete-packet approach, TCP is usually modeled as a Markov Renewal Process as in [5], [7] to obtain the expressions of TCP throughput in relation to the RTT and loss rate  $p$ . We extend this method by modeling TCP as a Semi-Markov Process, where the window size  $W$  at time  $t$  is the state of the Markov process. SMP model gives the window distribution. Throughput is obtained from the little's formula. It is not in a closed-form expression.

Fluid model approach has been used in [1],[2],[3].  $W(t)$  is usually assumed as linearly increase and multiplicative decrease. The loss processes are different. Altman et al in [1] consider a general stationary and ergodic loss process. Vishal et al in [3] consider a poisson loss process. However, loss processes in their work are all assumed to be independent from the window process. In this paper, the loss process is dependent on the window process.

## III. SMP MODEL OF WINDOW EVOLUTION

We first define the window evolution process, then introduce the model, and finally we give the model computation method.

### A. TCP Window Evolution Process

The window evolution process  $W(t)$  is the TCP outstanding packets in the network as a function of time  $t$ . TCP congestion control mechanism decides its reaction to the network congestion signal. Loss process depicts the nature of network congestion signal sent to TCP. These two factors jointly determine the characteristic of the TCP window evolution process. A real TCP window evolution process consists of Slow Start(SS), Additive Increase(AI), Fast Recovery (FR), Timeout(TO). Its sample path is showed in fig 1.

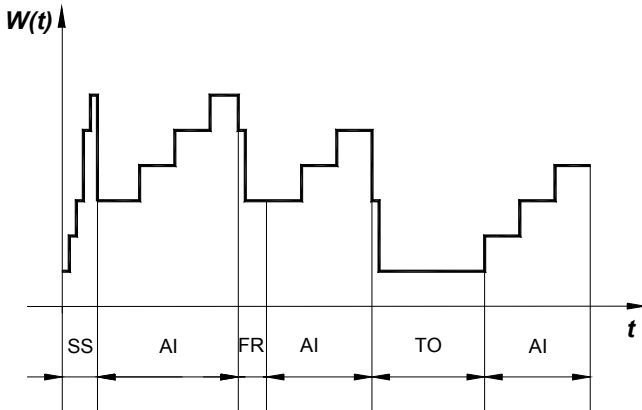


Figure 1 TCP window evolution

$W(t)$  in fig. 1 is definitely not a Markov process. Both the congestion control mechanisms and the loss process are not Markovian. It can be modeled as a MPP(Marked Point Process), but the mathematical solution of MPP is difficult. However, with appropriate assumptions,  $W(t)$  can be modeled as SMP.

### B. SMP Model

#### 1) Model Assumptions

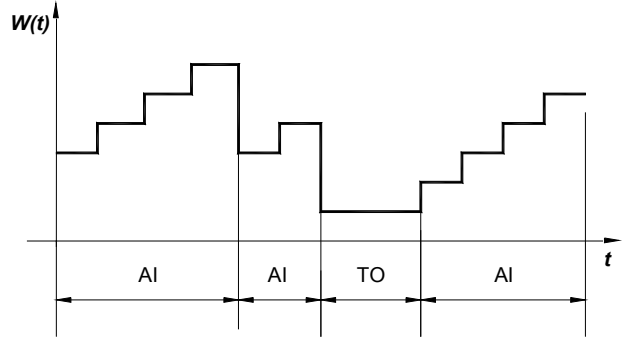


Figure 2 Window evolution without FR and TO

As usual, we ignore the fast recovery and the slow start as usual. Now, the new window evolution process is only composed of AI and TO phase. Fig. 2 shows the sample path of this new stochastic process. The loss process is assumed as I.I.D, which is the same as in [4-7]

#### 2) The SMP Model

$W(t)$  jumps from time to time, so let us first consider the embedded chain  $\{W_n\}$  of  $W(t)$ . Let  $W_0$  be the initial state and  $W_n$  be the state entered on the  $n$ th jump,  $n \geq 1$ . There are only three kinds of state transitions in the chain:

1. If no packet loss occurs, then  $W_{n+1} = W_n + 1$
2. Loss occurs but does not trigger TO, then  $W_{n+1} = W_n / 2$
3. Loss triggers TO, then  $W_{n+1} = 1$ , for any  $W_n$ .

We can see that next state  $W_{n+1}$  is dependent only on the current state  $W_n$ , so chain  $\{W_n\}$  is a Markov Chain.

Then for the time-continuous process  $W(t)$ , suppose  $\tau_n$  is the sojourn time in the state entered on the  $n$ th jump. Corresponding the 3 kinds of state transitions in EMC, we consider  $\tau_n$  in 3 situations,

1. For the 1<sup>st</sup> kind of EMC state transition, TCP is in AI phase, and  $\tau_n$  is around  $b \times RTT$  period of time for each window increase.  $b$  is 1 for non-DelAck, and 2 for DelAck. Although  $\tau_n$  may varies as RTT varies, it is indeed independent from  $\tau_{n-1}, \tau_{n-2}, \tau_{n-3} \dots$ . In the situation where RTT increases as window size  $W$  grows,  $\tau_n$  is related with its previous values. But if  $W_n$  is determined,  $\tau_n$  is independent from  $\tau_{n-1}, \tau_{n-2}, \tau_{n-3} \dots$
2. For the 2<sup>nd</sup> kind of EMC state transition,  $\tau_n$  is equally distributed in the range  $[0, b \times RTT]$ .
3. For TCP timeouts,  $\tau_n$  is  $RTO \times rtx\_backoff$ .

In all these 3 situations,  $\tau_n$  depends only on the values of  $W_n$  and  $W_{n+1}$ . According the Appendix,  $W(t)$  is SMP.

The SMP model have two window distributions,  $\{\pi_j\}$  in EMC denotes the fraction of transitions entering the state  $j$ .  $\{p_j\}$  in SMP is the fraction of time the SMP stays in state  $j$ .

### C. Model Computation

For the window evolution process  $W(t)$ , we want get the its time distribution of window size. It corresponds to  $\{p_j\}$  of SMP. Before getting  $\{p_j\}$ , we must calculate  $\{\pi_j\}$  of EMC first.

#### 1) EMC Computation

Fig. 3 shows the state transition graph of the EMC. In the figure, states in the upper row are the Timeout states, with the *rtx\_backoff* index inside the circle, below them are the AI states with window size in the circle. Dash lines represent state transitions due to losses, and real lines represent state transitions when receiving enough ACKs.

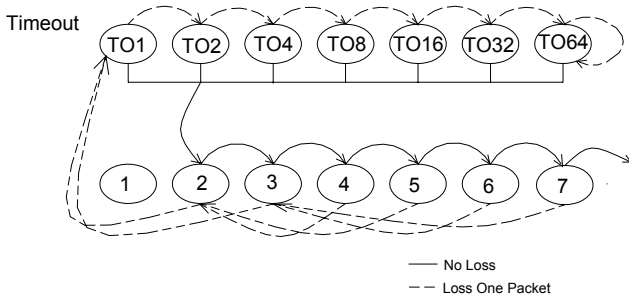


Figure 3 EMC state-transition graph

$\{\pi_j\}$  is determined by state transition probabilities matrix  $P$ . For each state, the transition probabilities are  $P(i, i+1)$ ,  $P(i, [i/2])$  and  $P(i, 1)$ , representing window increase, window decrease and timeout respectively. Obviously, for each state, the sum of these three probabilities equals to 1.  $P(i, j)$  is decided by the TCP algorithm and the loss process. Especially, for different loss processes  $P(i, j)$  can be very different. In the following, we derive the expression of  $P$  in terms of loss rate  $p$  under I.I.D loss assumption. Fig. 4 shows the matrix  $P$ .

First, timeout states. In fig. 3, they are TO1 to TO64. State transition is determined by the whether the retransmitted packet is lost or not. If the packet transmits successfully, TCP get out of TO state. The probability is  $1-p$ . If the packet is lost, TCP retransmits the packet another time and backoffs its retransmit timer. The probability is  $p$ .

Second, window increase state transitions. There is only one state transition for window increase, from  $W_n$  to  $W_n+1$ . During AI phase, TCP increases its window size from  $W_n = w$  to  $W_{n+1} = w+1$  by receiving approximately  $w$  new ACKs. If the receiver does not use DelAck algorithm, every data packet triggers a new ACK packet. So the state transition probability from  $W_n = w$  to  $W_{n+1} = w+1$  is  $(1-p)^w$ , which is the probability of successfully transmitting  $w$  data packets to the receiver. If the receiver uses DelAck, approximately every two data packets triggers a new ACK packet. Thus, the state transition probability is  $(1-p)^{2w}$ , the probability of successfully transmitting  $2w$  data packets to the receiver. So,

$$P(W_{n+1} = w+1 | W_n = w) = (1-p)^{bw},$$

Third, the window decrease state transitions. There are two state transitions for window decrease happened under different loss severities. If the loss is severe or window size is small, TCP will get into TO state, otherwise TCP cuts its window size to half. We discuss these situations according to the window size.

1. When  $W_n$  are of 1,2,3, any packet loss drives TCP into TO1 state. For  $W_n = 1, 2, 3$

$$P(W_{n+1} = TO1 | W_n) = 1 - P(W_{n+1} = W_n + 1 | W_n),$$

2. When  $W_n$  is 4, a single packet loss will not cause TCP into timeout, otherwise TCP goes into TO1 state.

$$P(W_{n+1} = 2 | W_n = 4) = 4p(1-p)^3$$

$$P(W_{n+1} = TO1 | W_n = 4) =$$

$$1 - P(W_{n+1} = 2 | W_n = 4) - P(W_{n+1} = 5 | W_n = 4)$$

3. When  $W_n > 4$ , we assume only  $W_n \rightarrow W_n/2$  transition occurs. Because, for the I.I.D loss process, the probability of losing more than 3 packets in one round is small enough to ignore, for  $W_n > 4$

$$P(W_{n+1} = [w/2] | W_n = w) = 1 - (1-p)^{bw}.$$

Finally, we consider the limitation on the maximum window size. The limitation may comes from the maximum window size advertised by the receiver at the beginning of the connection. Adding this limitation into the model is straightforward. Suppose  $m$  is the maximum congestion window size, so we have

$$P(W_{n+1} = m | W_n = m) = (1-p)^{bm},$$

and,

$$P(W_{n+1} = [m/2] | W_n = m) = 1 - (1-p)^{bm}.$$

#### 2) SMP Computation

SMP associates each state in EMC with a sojourn time  $\tau_n$ . In SMP,  $a_j$  denotes the average sojourn time in state  $j$ . For AI states,  $a_j$  equals to  $b$  multiplying  $RTT$ . If round-trip time is independent of the window size,  $a_j = b \times RTT$  for all AI state, where  $RTT$  is the average  $RTT$  of all packets. If they are not independent,  $a_j$  can be expressed as a function of window size, e.g.  $b \times RTT(W)$ .

For TO states,  $a_{TO,j}$  is  $RTO \times rtx\_backoff$ .

Now, we get the  $\{\pi_j\}$  and  $\{a_j\}$ , and we can use (A.1) to compute  $\{p_j\}$  and use (A.3) to compute  $\bar{W}$ .

#### D. Throughput and Packet lifetime

Apply little's formula to TCP, we get the throughput,

$$T = \bar{W} / l,$$

where  $l$  is the packet lifetime. Packet lifetime is the sojourn time when a packet stays in the network. A packet's lifetime begins from the time it enters the network, and ends at the time when it is acked, or at the time when the sender is convinced of its lost. When the fraction of timeout packets is small,  $l$  is very close to  $RTT$ , otherwise it is evidently larger than  $RTT$ . Later we will see that often  $RTT$  is very close to  $l$ , so we just use  $RTT$  instead, and do not need complicated method described below to compute  $l$ .

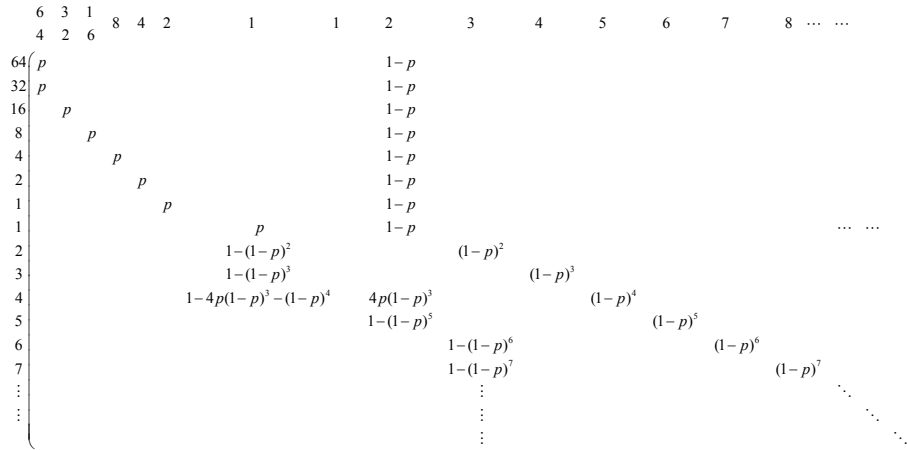


Figure 4 EMC transition probability matrix P

Now, compute the packet lifetime. We divide packets of a connection into 3 types according to how they end their lifetime. Let  $CL_j$  be the average lifetime of type  $j$  and  $CP_j$  be the proportion of  $j$  type packets to the total packets.

1. Those successfully transmitted packets. Their lifetimes end when the sender receives ACKs in time. Their lifetimes are around RTT.  $CL_1 = RTT$ ,  $CP_1 = 1 - p$ .
2. For those lost packets recovered in Fast Retransmit phase or in Fast Recovery. Their average packet lifetime<sup>1</sup> approximates  $CL_2 = RTT \times 3/2$ . However,  $CP_2$  is not easy to calculate directly. Because loss packets are either recovered in this type or from the 3<sup>rd</sup> type. Subtracting the proportion of packets recovered by Timeout mechanism from total lost packets, we get  $CP_2 = p - CP_{timeout}$ .
3. For those lost packets recovered by timeout mechanism, they should be sub-divided because their lifetimes are also determined by how many times they have been retransmitted, the  $rtx\_backoff$  index. So we get,

$$CL_{3,backoff} = \bar{RTO} \times rtx\_backoff.$$

We use following method to calculate the  $CP_{3,backoff}$ : Consider the EMC Chain, every time the state enters a TO state, 1 packets will have a lifetime RTO. For each time TCP enters a AI state  $i$ , it will send approximately  $(i+1)$  data packets. So,

$$CP_{3,backoff} = \pi_{TO,backoff} / (\sum (i+1) \times \pi_{AI,i} + \sum \pi_{TO,backoff}),$$

$$\sum \pi_{AI,i} + \sum \pi_{TO,backoff} = 1,$$

$$CP_{timeout} = \sum CP_{3,backoff}.$$

Now we use (1) to get the average packet lifetime.

$$l = \sum_j CL_j \times CP_j \quad (1)$$

#### E. Discussion

The model computation involves 3 steps. First, decide the state transition graph of EMC. Second, calculate the EMC transition probability matrix  $P$ . Last, decide the sojourn time  $a_j$ . The details in each step can be changed in different situations. If we don't consider TCP timeout, we

just eliminate the all the TO states in EMC. If TCP window grows sub-linearly, we make  $a_j$  increase as  $j$ . The matrix  $P$  can also be changed according to different TCP protocol details and loss assumptions.

## IV. FLUID MODEL

### A. Fluid Model

Consider a continuous-time continuous-state stochastic process  $W(t)$ . While no loss signal is met,  $W(t)$  increases linearly with time at a rate  $\alpha = 1/(b \times RTT)$  where  $b$  is the number of data packets covered by every one ACK. If there is loss signal,  $W(t)$  decreases 1/2. The loss process is assumed to be a poisson process, however, different from any previous approaches [2], [3], the arrival rate of the poisson process  $\lambda$  is proportional to the window size  $W(t)$ . To be different, we call this loss process the Window dependent arrival Rate Poisson Process (WRPP), in contrast to the Constant arrival Rate Poisson Process (CRPP). In the following, if not specified, we use the fluid model to refer the WRPP fluid model.  $W(t)$  is described by following equations,

If there is no loss,

$$dW(t)/dt = \alpha.$$

When loss signal arrive at time  $t$ ,

$$W(t+) = \frac{1}{2}W(t-),$$

where loss process is a poisson process with arrival rate  $\lambda = W(t)\lambda_0$ .

### B. Moments of $W$

We calculate the moments of  $W$  of steady-state. When process  $W(t)$  is in equilibrium, the probability of up-crossing at  $W(t) = x$  is

$$(1 - x\lambda_0\Delta_t) \times P\{x - \alpha\Delta_t < W \leq x\} + o(\Delta_t).$$

It equals to the probability of down-crossing

$$\sum_{\substack{x_1 = x \leq x_j \leq x_k = 2x \\ 1 \leq j \leq k, k \rightarrow \infty, \Delta_j \rightarrow 0}} x_j \lambda_0 \Delta_t \times f(x_j) \Delta x_j + o(\Delta_t)$$

<sup>1</sup> 3/2 is just a estimate. It doesn't matter much as the effect of packets recovered in FR to the average packet lifetime is rather small compared with that of those timeout packets.

When the  $\Delta_t \rightarrow 0$ , we get the steady-state Kolmogorov equation,

$$\begin{aligned}\alpha f(x) &= \lambda_0 \int_x^{2x} \tau f(\tau) d\tau \\ &= \lambda_0 \left( \int_0^{2x} \tau f(\tau) d\tau - \int_0^x \tau f(\tau) d\tau \right).\end{aligned}\quad (2)$$

The Fourier Transform of the equation (2) equals,

$$\alpha \Phi(\omega) = \lambda_0 \left( \frac{1}{2} \times \frac{\dot{\Phi}(\omega/2)}{\omega/2} - \frac{\dot{\Phi}(\omega)}{\omega} \right), \quad (3)$$

where the  $\Phi(\omega) = \mathcal{F}[f(x)]$ , which is also the characteristic function of  $f(x)$ .

The Taylor expansion of  $\Phi(\omega)$  at  $\omega = 0$ ,

$$\Phi(\omega) = \Phi(0) + \sum_{k=1}^{\infty} \frac{\Phi^{(k)}(0)}{k!} \omega^k = 1 + \sum_{k=1}^{\infty} \frac{j^k E[W^k]}{k!} \omega^k \quad (4)$$

Substitution (4) into (3),

$$\begin{aligned}\alpha \left( 1 + \sum_{k=1}^{\infty} \frac{j^k E[W^k]}{k!} \omega^k \right) &= \\ \lambda_0 \sum_{k=1}^{\infty} \left( \frac{1}{2^{k-1}} - 1 \right) \frac{j^k E[W^k]}{(k-1)!} \omega^{k-2}\end{aligned}\quad (5)$$

Equate the coefficients of equal powers of  $\omega$  in (5), we get,

$$E[W^2] = \frac{2\alpha}{\lambda_0} \quad (6)$$

$$E[W^k] = \frac{2^{k-1}(k-1)\alpha}{(2^{k-1}-1)\lambda_0} E[W^{k-2}], \text{ for } k = 3, 4, \dots \quad (7)$$

To compute  $E[W]$ , we transform  $W(t)$  into  $W_o(\tau)$  by expanding  $t \rightarrow \tau$ . However, the time expanding coefficient is not a constant, it is proportional to the current window size  $W(t)$ . On the contrary, time compresses when  $\tau \rightarrow t$ , the compressing coefficient is proportional to  $1/W_o(\tau)$ . Because the expanding coefficient is not constant, differential format is used to express the expanding relation  $dt \rightarrow d\tau$ ,

$$d\tau = W(t) \times dt, \quad dt = \frac{1}{W_o(\tau)} \times d\tau.$$

$W_o(\tau)$  is described below. If there is no loss,

$$dW_o(\tau) / d\tau = \frac{dW(t)}{dt} \times \frac{dt}{d\tau} = \frac{\alpha}{W_o(\tau)}. \quad (8)$$

When loss signal arrive at time  $t$ ,

$$W_o(\tau+) = \frac{1}{2} W_o(\tau-), \quad (9)$$

The loss process is also affected by the time expansion. The arrival rate of poisson process of  $W(t)$  is  $\lambda = W\lambda_0$  under

timescale  $t$ . Because the time expand  $W$  times from  $t \rightarrow \tau$  at window size  $W$ , the arrival rate under the timescale  $\tau$  should compress  $W$  times.  $W \times \lambda_0 / W = \lambda_0$ . The loss process in  $W_o(\tau)$  becomes a constant arrival rate poisson process.  $W_o(\tau)$  described by (8), (9) is analyzed in [8].

Suppose  $f(x)$  and  $f_o(x)$  are the probability density function (pdf) of  $W(t)$  and  $W_o(\tau)$ . They represent the time distributions of window size in steady state, where  $f(x)\Delta x$  is the probability when  $W$  is between  $[x, x+\Delta x]$ . Because of the time expansion, time spend at window size  $W$  expands  $W$  times from process  $W(t)$  to  $W_o(\tau)$ , so does the time fraction at window size  $W$ ,

$$f(x)\Delta x \propto \frac{1}{x} f_o(x)\Delta x,$$

$$\text{Thus, } f(x) = \frac{f_o(x)}{x} / \int_0^{\infty} \frac{f_o(x)}{x} dx. \quad (10)$$

$$\text{So, } E[W] = \int_0^{\infty} x f(x) dx = \int_0^{\infty} x \left( \frac{f_o(x)}{x} / \int_0^{\infty} \frac{f_o(x)}{x} dx \right) dx dx$$

$$= \frac{\int_0^{\infty} f_o(x) dx}{\int_0^{\infty} \frac{f_o(x)}{x} dx} = E[1/W_o].$$

According to [8],

$$E[W] = \frac{1}{E[1/W_o]} = \frac{2}{A} \sqrt{\frac{2\alpha}{\pi\lambda_0}}, \quad (11)$$

where  $A$  is a constant,  $A \approx 1.218229$  in [8].

Eq. (6), (7), (11) show all the moments of  $W$ . From them, we see that the moments of  $W$  can be expressed in another way,

$$E[W^k] = C_k \left( \frac{\alpha}{\lambda_0} \right)^{k/2}, \text{ for } k = 1, 2, 3, \dots \quad (12)$$

where  $C_k$  is constant.

We also use SDE described in [3] to analyze  $W(t)$  and get the same result.

### C. Window Size Distribution

Eq. (10) tells the relation between  $f(x)$  and  $f_o(x)$ . In [8], we have,

$$f_o(x) = \sum_{k=0}^{\infty} a_k(c) \frac{\lambda_0}{\alpha} x \exp\left(-\frac{c^{-k}(\lambda_0/\alpha)}{2} x^2\right).$$

Using (10),

$$f(x) = \frac{1}{E[1/W_o]} \sum_{k=0}^{\infty} a_k(c) \frac{\lambda_0}{\alpha} \exp\left(-\frac{c^{-k}(\lambda_0/\alpha)}{2} x^2\right), \quad (13)$$

where  $c=1/2$ ,  $a_k(c) = \frac{1}{L(c)} \frac{(-1)^k c^{(k-1)k/2}}{\prod_{i=1}^k (1-c^i)}$ ,

$$L(c) = \prod_{k=1}^{\infty} (1-c^k)$$

$f(x)$  have a very good property. Suppose  $f_0(x)$  and  $f_1(x)$  are pdf of  $W(t)$  under different poisson arrival rate  $\lambda_0$  and  $\lambda_1$ . With (12), it is easy to prove by using the Taylor expansion of characteristic function and the property of Fourier transformation that,

$$f_1(x) = (\lambda_1 / \lambda_0)^{1/2} f_0(x / (\lambda_0 / \lambda_1)^{1/2}). \quad (14)$$

Eq. (14) shows  $f_1(x)$  is just a time expansion of  $f_0(x)$ . The shapes of the two functions are the same. The expansion coefficient is  $(\lambda_0 / \lambda_1)^{1/2}$ .

#### D. Discussion

In the fluid model, the loss process has  $\lambda = W(t)\lambda_0$ . So, when window size grows, TCP is more likely to meet a loss signal. In SMP with I.I.D loss, the probability of window decrease is  $1-(1-p)^{bn}$ . As window grows this probability also becomes larger. Thus two models are similar.

Here we deduce a relation between the  $p$  in discrete model and the  $\lambda$  in fluid model. In discrete model, denote  $X$  to the number of packets between two successive losses. If the losses are I.I.D,  $X$  is a geometric distribution random variable (R.V.).  $E[X] = 1/p$ . When the window size is  $W$ , the average time interval between two successive packets is  $RTT/W$ , so average time interval between two losses is

$$E[S] = E[X] \times RTT / W,$$

where R.V.  $S$  denotes the inter-loss time interval. As

$$E[S] = 1/\lambda = E[X] \times RTT / W,$$

we have,

$$\lambda = W \times (p / RTT) = W \times \lambda_0,$$

where  $\lambda_0 = p / RTT$ . (15)

Later we will use (15) to compare the window distribution of the SMP model and the fluid model.

#### V. MODEL VALIDATION

Model validation consists 3 parts. First, we judge how closely the discrete SMP model fits the actual TCP window evolution process through both the real Internet experiment and the NS simulation.

Second, we compare the SMP model and the fluid model discussed in section IV.

Finally, we use the model to analyze the impact of window limitation to the throughput.

##### A. Validation of SMP Model

###### 1) Internet Measurement

We set up a FTP server to allow clients from different places of Internet to download files from our FTP server. The server (210.25.128.111) is located in the NSFCNET. It installs FreeBSD 4.5 and uses NewReno TCP.

We developed a tool that resembles with the *tcpdump* and *tcptrace* tool. It dumps TCP connections into separate files. The tool reads the dump file, calculate the number of outstanding packets ( $W$ ) in the network as a function of time  $t$ . This  $W(t)$  denotes the window evolution process discussed in this paper. The tool also delimits the loss recovery and non-loss recovery phase of a connection. For non-loss recovery phase, it finds out every window increase step, measures the sojourn time of the step, the number of ACKs received and data packets sent during the step. Then it judges whether the step is a slow start (SS) step or an additive increase (AI) step. Finally, it calculates the total number of AI step transitions into  $W = j$ , and the time fraction when the connection stays at  $W = j$ . The former corresponds to the  $\{\pi_j\}$  of EMC, and the latter corresponds to the  $\{p_j\}$  of SMP.

We studied the connection between our ftp server to client 61.240.177.54. This client kept downloading on Jun 21, 2002 from 12:00CST till 19:00CST. Table 1 lists basic information about it. Files in the ftp server are from 3MB to more than 10MB, thus every connection can be referred as a long-run connection. Data is plotted every one hour.

Table 1 Path property between server and client

Client	IP	Hops	RTT
A	61.240.177.54	15	~600ms

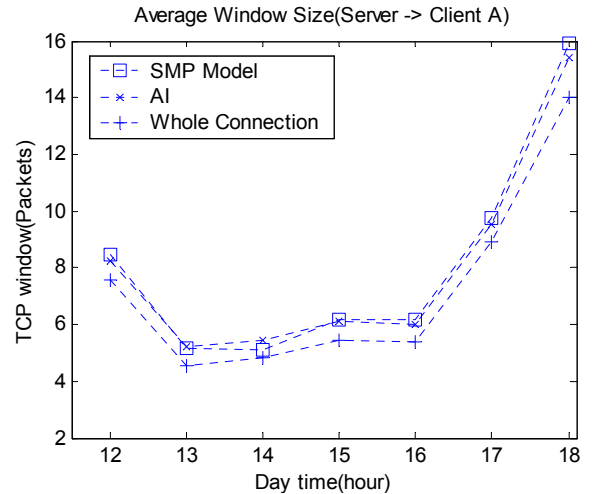


Figure 5: Client A connection

In fig 5, AI denotes the time average window of AI steps. The curve is very close to the SMP model. Because the SMP model considers only AI transitions of a TCP connection, the closeness of these two curves are actually very reasonable. The model result is larger than the average window of the whole connection. This is because existence of SS and FR phases in a actual TCP connection lowers average window size.

Next, Figure 6, 7, 8 show the window distribution at time, 16:00-17:00, 17:00-18:00 and 18:00-19:00. The loss rates are 0.0229, 0.0097, 0.0029 respectively. "EMC PDF" figures compare  $\{\pi_j\}$  of EMC with actual fractions of AI

step transitions. “SMP PDF” figures compare  $\{p_j\}$  of SMP with the actual time fractions of the whole connection.

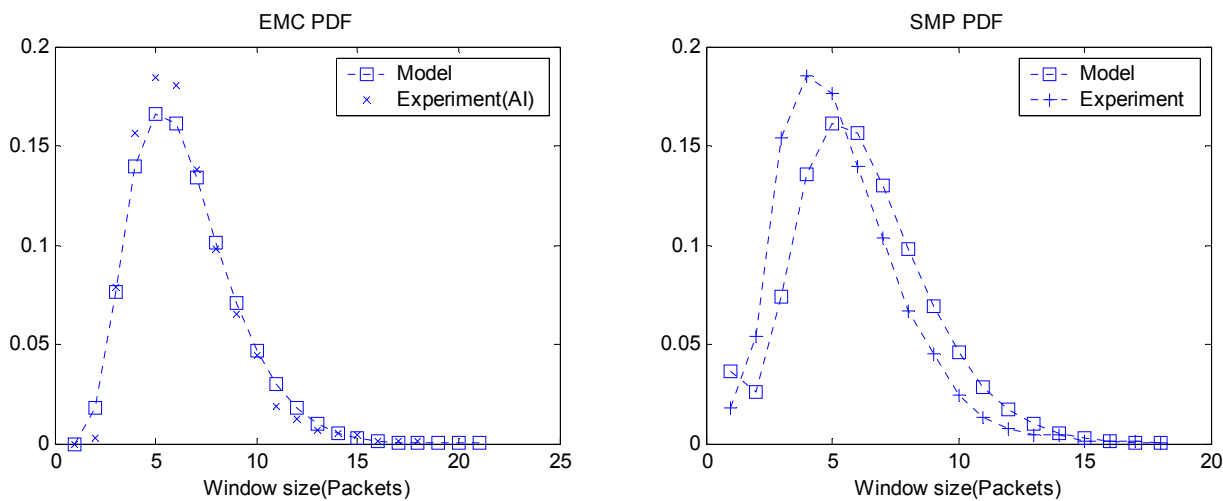


Figure 6 16:00-17:00,  $\rho = 0.0229$

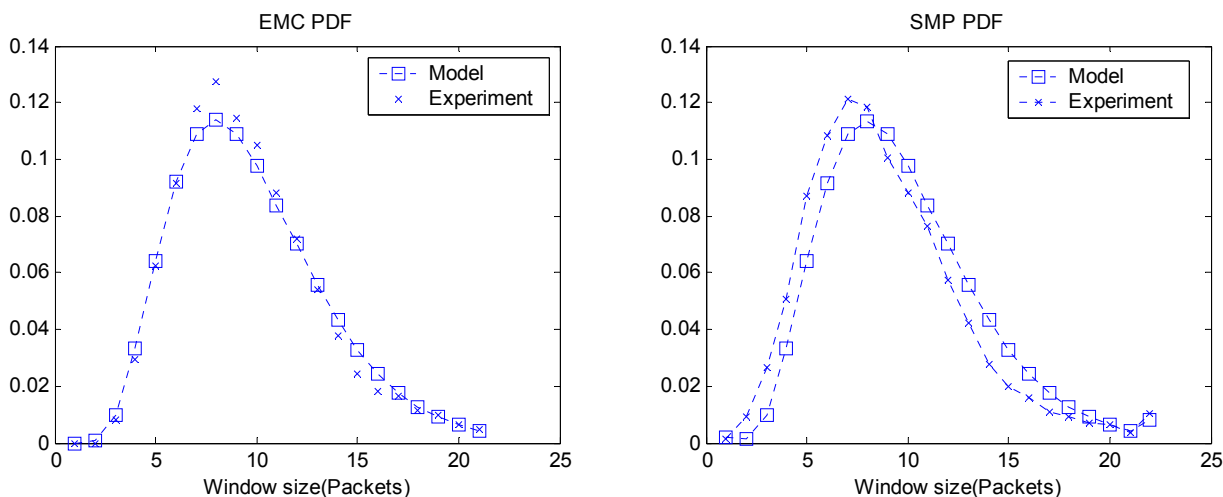


Figure 7 17:00-18:00  $\rho = 0.0097$

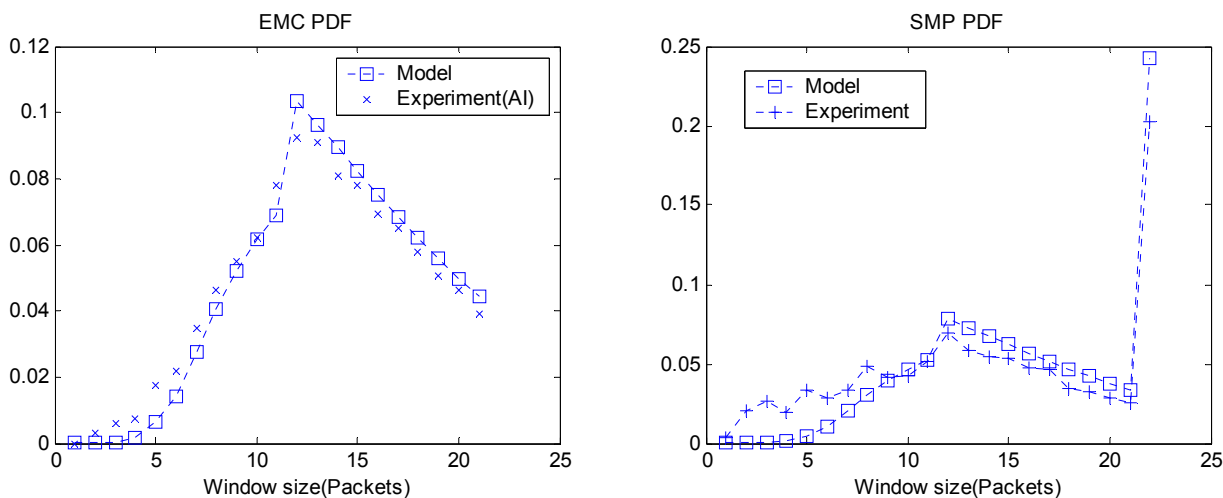


Figure 8 18:00-19:00  $\rho = 0.0029$

In the 1<sup>st</sup> period, the connection lasts 1210s, AI phase makes up to 79% of the total connection, 960s, FR takes

17%, 205s, SS takes 4%, 45s. In the 2<sup>nd</sup>, The connection lasts 780s, in which AI phase has 674s, making up to 86% of

the whole connection, while loss recovery has 78s, 10%, and SS phase has the remaining 28s, 4% of the whole connection. In the 3<sup>rd</sup> period, the connection lasts 993s, AI phase takes 86%, 855s, SS takes 8%, 81s, FR takes 6%, 57s. As the loss rate decreases, the time fraction of FR decreases. However, in the 3<sup>rd</sup> period, the time fraction of SS increases. If losses are happened when TCP reaches maximum window advertised by the receiver, during the loss recovery period, no more new packet is allow to send. So after loss recovery period, TCP's *cwnd* is 1, SS starts. In the 3<sup>rd</sup> period, lots of losses are happened in this situation, so the time fraction of SS increases. This reduces the average window of the connection obviously.

From these figures, we see that SMP model predict AI distribution quite well. Also, we see that SS and FR do lower the average window size of TCP. Right now FreeBSD 4.5 does not support SACK TCP. We believe that the SMP model will fit SACK TCP more closely, because SACK TCP is more robust to recover losses and has less time fraction of FR and SS in a connection.

2) NS Simulation

The experiment validates the model in a certain range of loss rates. In this section, simulation is taken to validate the model in a wide range of loss rates.

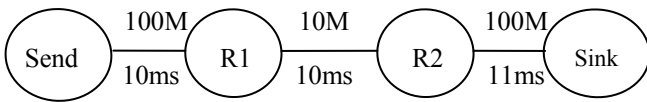


Figure 9 NS Simulation Scenario

Between Router R1 and R2, we introduce a random packet dropping module. Packets sent from TCP are dropped with a constant and independent probability  $p$  before entering the R1's input queue. We set the link bandwidth between R1 and R2 to 10Mbps in order to prevent introducing queuing delay on the link, thus makes TCP window grow linearly.

From fig. 10, we can divide entire loss range into two ranges. The first range is [0.001,0.1]. In this loss range, the curve of average window size decreases linearly, while average packet lifetime equals to RTT. In the loss range [0.1,1], the average window size decreases a little slower than before, while the average packet lifetime increases dramatically. Fig. 10 restates the basic fact in an evident way: When the loss rate is small, it is the congestion avoidance mechanism that mainly affects TCP. When loss rate becomes large, timeout mechanism becomes the main factor to affect the TCP. We also see from 10(b) that the average packet lifetime is very close to RTT for  $p < 0.1$ , so often we use RTT instead of  $l$ .

In fig. 10(c), we compare the SMP model with Padhye's equation. The figure shows that the two models are very close to each other in small loss rate. When loss rate is larger than 0.1, the two models begin to differ from each other. We think the reason may due to the slight differences between the two models' loss assumptions.

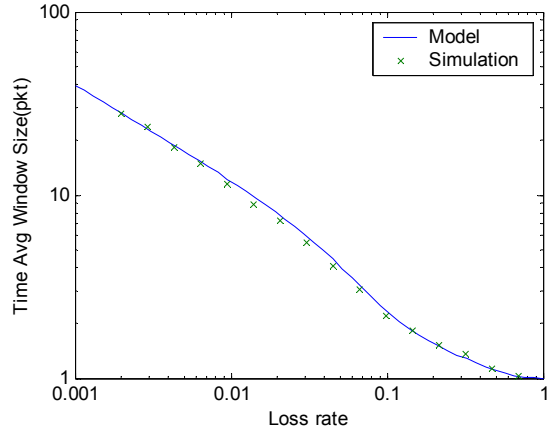


Figure 10(a)

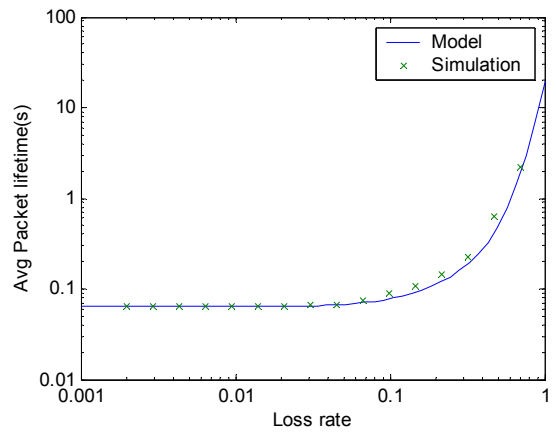


Figure 10(b)

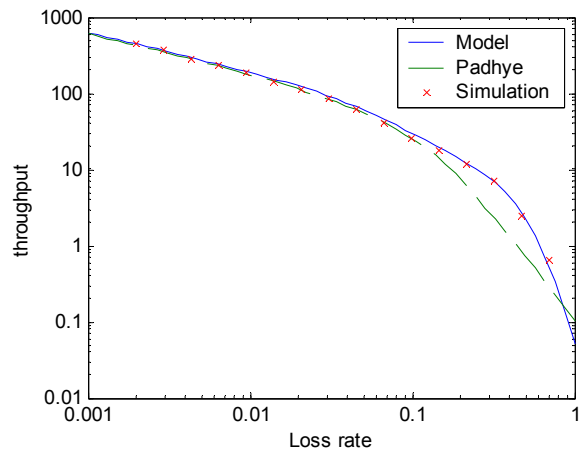


Figure 10 TCP average window size(a), average packet lifetime(b), average throughput(c)

## B. SMP Model and Fluid Model

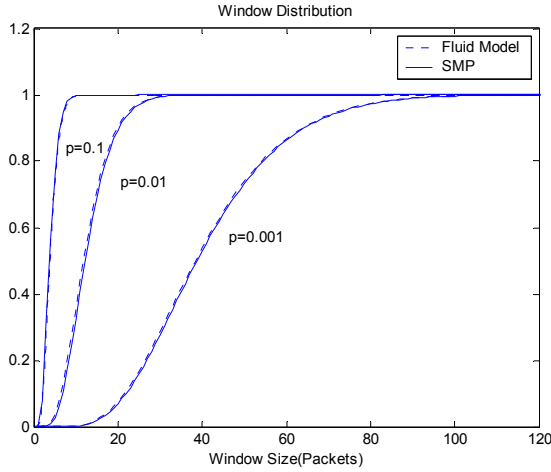


Figure 11 Window distribution of SMP and Fluid Model

In this section, TCP timeout is not considered in SMP model. The  $\lambda_0$  of the fluid model is set to  $\lambda_0 = p / RTT$ . In fig. 11, the window cumulative distributions of the SMP model and fluid model overlap. Next, we compare the moments of models' two distributions. From the characteristic function theory, if all moments are the same, the two distributions are the same.

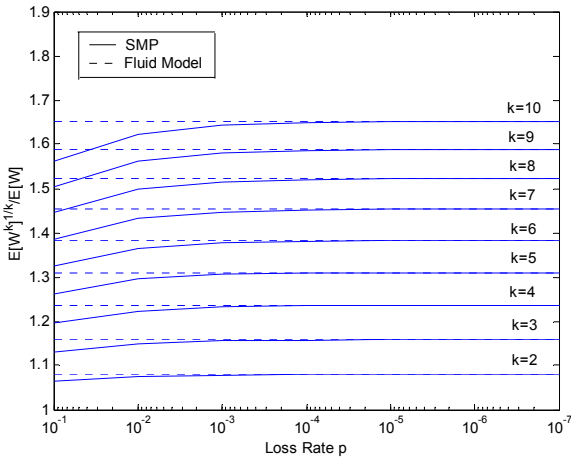


Figure 12  $\sqrt[k]{E[W^k]} / E[W]$  at different Loss Rate

In fig. 12, we show  $\sqrt[k]{E[W^k]} / E[W]$  of the SMP model and the WRPP fluid model as the loss rate  $p \rightarrow 0$ .  $\sqrt[k]{E[W^k]} / E[W]$  of the fluid model is constant  $C_k$  according to (12). For the SMP model, the value converges to  $C_k$ . Although we have only show up to 10<sup>th</sup> moment, we see the convergence trend that the SMP model approaches the WRPP fluid model as  $p \rightarrow 0$ . In fig. 13, both the WRPP and CRPP fluid model are plotted. Clearly, the limit of SMP is not CRPP fluid model. So, we conclude that the I.I.D loss process corresponds to the poisson loss process with window dependent arrival rate. It neither corresponds to the poisson loss process, nor the deterministic loss process. In [1], the square root formula is obtained when inter-loss times are exponentially distributed and deterministic. Here, the throughput in relation to loss rate also satisfies the square

root formula. All the 3 loss processes lead to the square root formula, their differences lie in the coefficient C. Table 2 shows the difference,

Table 2 The C value in  $\bar{W} = \frac{C}{\sqrt{bp}}$  of Loss Processes

Loss Process	C
Deterministic	$1.22 = \sqrt{3/2}$
I.I.D	1.31
Poisson	$1.41 = \sqrt{2}$

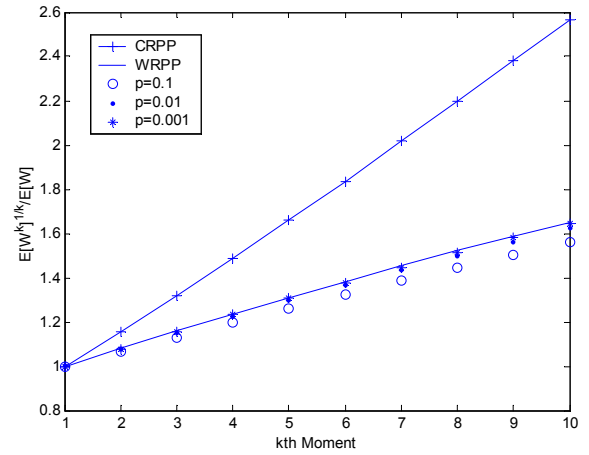


Figure 13  $\sqrt[k]{E[W^k]} / E[W]$  of different fluid models

Also fig. 13 shows that when WRPP and CRPP fluid models have a same  $E[W]$ , the variance of the latter is larger than the former. Larger variance means larger fluctuation of TCP window size.

## C. Impact of window limitation to the throughput

The receiver advertised window size may pose limitation on the maximum window size of the connection. The constraint not only limits the maximum transmission rate, but also reduces the average throughput. From little's law, the average window size is proportional to the throughput. So, instead of throughput we consider the impact to the average window size here.

Using SMP model, we plot fig. 14 showing this impact.  $W_{lim}$  is the limitation of maximum window,  $W_{aver}$  is the average window under the certain  $W_{lim}$ . Both values are normalized by dividing  $W_{aver}^\infty$ .  $W_{aver}^\infty$  is the average window size achieved without maximum window limitation, It is the largest value of all  $W_{aver}$  we can achieved under a certain loss rate. In the figure, curves under different loss rates overlap. When  $W_{lim} = W_{aver}^\infty$ , the  $W_{aver}$  is only 80% of the  $W_{aver}^\infty$  due to impact of the limitation of maximum window size, but when  $W_{lim} \geq 2 \times W_{aver}^\infty$ , the impact is negligible. So, to achieve the a same throughput as if there were no window limitation, we need to set  $W_{lim} \geq 2 \times W_{aver}^\infty$ . Curves in the figure also show a convergence behavior as  $p \rightarrow 0$ . So, we believe this basic relation holds as  $p \rightarrow 0$  and is valid for  $p < 0.1$ . Thus we can see that the impact of

$W_{lim}$  to  $W_{aver}$  is determined by  $W_{lim} / W_{aver}^\infty$ , and is independent from loss rate  $p$ . Allman note in [9], the average window advertised by the receiver in the Internet is approximately 12 packets, that is  $W_{lim} = 12$ . So when  $p > 0.1$ ,  $W_{aver}^\infty = 5 < 12/2$  packets. This do not pose a serious problem.

However, If  $p = 0.01$ , then  $W_{aver}^\infty \approx 13$  packets,  $W_{lim} / W_{aver}^\infty < 1$ . The average throughput is less then 80% of the throughput achieved when  $W_{lim} > 26$ . So we need to double the current average window advertised by the receiver.

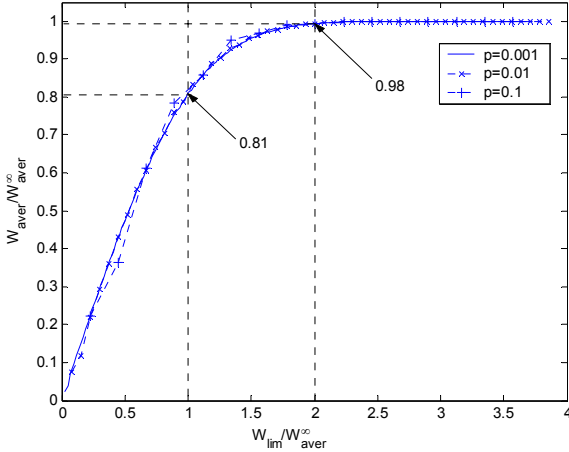


Figure 14 Impact of window limitation to the average window

## VI. CONCLUSION AND FUTURE WORK

I.I.D. packet dropping and poisson loss signal arrival are two important and widely used assumptions in modeling TCP loss process. In this paper, the relation between these two loss processes is analyzed. We find that the SMP model with I.I.D losses approaches the fluid model with WRPP loss process as  $p \rightarrow 0$ . With this, we established a connection between discrete model and fluid model, between loss rate and loss interval. The I.I.D losses with loss rate  $p$  corresponds to the poisson loss process with window dependent arrival rate  $\lambda$ , where  $\lambda = W \lambda_0$ ,  $\lambda_0 = p / RTT$ .

The SMP model gives the window distribution of a TCP connection, which agrees well with the experiments over real Internet and NS simulation. From window distribution, we found that the impact of  $W_{lim}$  to  $W_{aver}$  is determined by  $W_{lim} / W_{aver}^\infty$ , and is independent from loss rate  $p$ .

In the experiment, we found the SS and FR phases of a TCP connection do affect the window distribution. In the future, we plan to investigate more on this issue. Also, we plan to analyzed the WRPP fluid model with limitation of maximum window size.

## APPENDIX

### A. SEMI-MARKOV PROCESS

More discussions of SMP can be found in [10].

**Definition:** If  $\{W_n\}$  is a Markov Chain,  $W_n$  is the state entered on the  $n$ th jump. Let  $\tau_n$  be the sojourn time in

the state entered on the  $n$ th jump and  $t_n = \sum_{i=0}^{n-1} \tau_i$ ,  $n = 1, 2, \dots$

and for all  $t \geq 0$ , let  $x(t) = \max\{n : t_n \leq t\}$ . We define a new continuous time process,

$$W(t) = W_{x(t)}$$

If for each  $n \geq 0$ , the distribution of  $\tau_n$  depends only on the values of  $W_n$  and  $W_{n+1}$ , then the continuous time process  $W(t)$  is called a *semi-Markov process (SMP)*. SMP differs from EMC simply by associating a real-time sojourn time with each jump.

**Theorem:** Suppose  $p_j$  is the fraction of time that the SMP spends in state  $j$ , and  $\pi_j$  is the fraction of transitions that are visits to state  $j$ . The relation of these two distributions is,

$$p_j = (\pi_j a_j) / \sum \pi_j a_j, \quad (A.1)$$

where  $a_j$  is the average sojourn time in state  $j$ , and

$$\pi = \pi P. \quad (A.2)$$

As  $p_j$  is the fraction of time the SMP spends in state  $j$ ,  $\bar{W} = \lim_{t \rightarrow \infty} \int_0^t W(t) dt / t$  can also be written as,

$$\bar{W} = \sum_j j p_j \quad (A.3)$$

First, we obtain EMC stationary distribution  $\{\pi_j\}$  from (A.2), then we use (A.1) to get the  $\{p_j\}$  in SMP, at last use (A.3) to obtain  $\bar{W}$ .

## REFERENCE

- [1] Altman, E., K. Avrachenkov, and C. Barakat. *A stochastic model of TCP/IP with stationary random losses*. in *ACM SIGCOMM*. 2000.
- [2] Altman, E., et al. *Sate-dependent M/G/1 type queueing analysis for congestion control in data networks*. in *INFOCOM*. 2001.
- [3] Misra, V., W.-B. Gong, and D. Towsley. *Stochastic differential equation modeling and analysis of TCP-window size behaviour*. in *Performance*. 1999.
- [4] Mathis, M., et al., *The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm*. *Comp. Commun. Rev.*, 1997. **27**(3).
- [5] Padhye, J., et al. *Modeling TCP throughput: A simple model and its empirical validation*. in *ACM SIGCOMM*. 1998.
- [6] Lakshman, T. and U. Madhow, *The performance of TCP/IP for networks with high bandwidth-delay products*. *IEEE/ACM Trans. Networking*, 1997. **5**: p. 336-350.
- [7] Kumar, A., *Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link*. *IEEE/ACM Trans. Networking*, 1998. **6**(4).
- [8] Ott, T.J., J.H.B. Kemperman, and M. Mathis, *The Stationary Behavior of Idealized Congestion Avoidance*. 1996.
- [9] Allman, M., *A Web Server's View of the Transport Layer*. *Comp. Commun. Rev.*, 2000. **30**(5).
- [10] Wolff, R.W., *Stochastic Modeling and the Theory of Queues*. 1990: Englewood Cliffs, NJ: Prentice-Hall.