Analysis of Dynamic Behaviors of Many TCP Connections Sharing Tail–Drop/RED Routers

Go Hasegawa and Masayuki Murata

Cybermedia Center, Osaka University
1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan

E-mail: {hasegawa,murata}@cmc.osaka-u.ac.jp

Abstract—

Appropriate control parameters are important for the successful deployment of RED (Random Early Detection) routers, especially when many TCP connections share the bottleneck link. In this paper, we first describe a new simple analysis method for determining the window size distribution of many TCP connections sharing a bottleneck router. We consider two kinds of buffering disciplines: TD (Tail Drop) and RED. We model the window size evolution of TCP connections by using a Markov process whose state is represented by a set of the current window size and the ssthreth value. The state transition matrix is then calculated by considering the characteristics of TD and RED routers. We show numerical results demonstrating the accuracy of our analysis and we discuss the fairness of TD and RED. We confirm that RED does not help improve the router's throughput even when appropriate control parameters are chosen but that it is still useful to provide the fairness among many competing TCP connections.

Keywords— **TCP (Transmission Control Protocol), Window Size, TD (Tail Drop), RED (Random Early Detection), Fairness**

I. Introduction

The future development of the Internet requires a better understanding of the behavior of TCP (Transmission Control Protocol) [1] widely used in the current Internet, and many research efforts have been devoted to revealing the characteristics of the TCP connection. For example, equations for calculating the throughput of the TCP connection from several parameters (packet loss probability, round trip time, maximum window size, etc.) have been shown in [2, 3]. Those equations, however, estimate only the long-term throughput of the TCP connection, and cannot be used to examine instantaneous behavior. It is reported in [4] that the average size of Web documents at several Web servers is about 10 [KBytes], and the instantaneous TCP throughput is important to the estimation of the transfer time of such small documents. More important, the equations estimating throughput are based on the long–term averages of packet loss probabilities and RTT (Round Trip Time) values, which implies that the interaction among active TCP connections at the router cannot be investigated.

An approach for examining the instantaneous throughput of the TCP connection can be found in [5-7], where the distribution of the congestion window size of the TCP connection is obtained. Then the TCP window size is directly related to the short–term throughput of the TCP connection. This approach is based on a stochastic modeling of the TCP window size behavior. Those works, however, assume the probabilistic packet loss model, in which each packet sent from the TCP sender is dropped with a constant probability p. In a sense, it can be considered that RED (Random Early Detection) [8] is used at the bottleneck router. When this is assumed, however, it is impossible to examine the effects of misconfigured control parameters of the RED routers because the packets tend to be lost in bursts by such misconfigured RED.

Because we wanted to investigate the effect of the RED routers when many TCP connections share the bottleneck link, we developed a simple method for evaluating the TCP window size distribution by using Markov analysis. In the analysis, we allow two kinds of the packet loss models: the probabilistic loss model and the bursty loss model, which respectively correspond to RED routers and Tail-Drop (TD) routers. In our approach, the Markov model is used to explain the evolution of the TCP window size, which is done by explicitly considering packet queuing at the router's buffer. Steady state probabilities are then calculated and used to derive the distribution of TCP window size. One of our main contributions in this paper is the derivation of the TCP window size evolution in the case where packet loss occurs in a bursty fashion at RED routers. This allows to analyze the case in which the RED parameters are inappropriately configured to be considered. Such a case cannot be examined when using the analytic approaches assuming a constant packet loss probability adopted in, since the existing approaches [5-7] implicitly assume that RED routers always work effectively. Another contribution is the analysis of the window size behavior of TCP connections under conditions in which bursty packet losses occur at TD routers. The traditional TD routers tend to drop the incoming packets in a bursty fashion [8], and our approach allows to evaluate the effect of those routers. It then becomes possible to compare TD and RED routers in a unified way.

Our analysis allows us to investigate the routers shared by many active TCP connections. When the number of active TCP connections is large, as it is in the current backbone routers, the packet buffer that each TCP connection can utilize becomes small and throughput degradation becomes obvious [9]. Our analysis can treat such a case and determine the packet buffer size sufficient for, say, thousands of active TCP connections. Furthermore, we use the analysis results to evaluate the fairness of the TD and RED routers and that the fluctuation of the window size is much smaller for RED routers. We provide numerical examples showing that RED routers can only provide as high throughput as TD routers can even when the configuration parameters of RED are determined appropriately [10, 11] but that they can greatly improve the fairness among many TCP connections. Our analysis can be used to determine, for a given buffer size and given number of active TCP connections, the appropriate control parameter set for the RED routers.

The rest of this paper is organized as follows. We briefly explain TD and RED disciplines in Section II. The analysis model is also introduced. Then we show the analysis of the window size distribution in Section III. In Section IV, we show some numerical examples to validate the analysis, and discuss the fairness property of the TD/RED routers. Finally, we conclude this paper and show some future work in Section V.

II. Model Description

A. TD and RED Routers

Historically, Internet routers have used a TD (Tail Drop) discipline as a buffer management mechanism: the TD router serves incoming packets in order of their arrival and simply discards newly arriving packets when the buffer is full. The problem with this mechanism is that routers tend to discard packets in bursts [12], which results in packets from the same connection being likely to be discarded. As a result, the fast retransmit algorithm does not help avoid timeout expirations, and this leads to the global synchronization problem [13]. Furthermore, since the duration of bursty packet losses depends on many factors (network configurations, the number of active connections, and so on), it is difficult to determine the packet loss rate of the TD router. As will be shown in Section III, however, we can analyze TD router if we make some reasonable assumptions about the network.

The RED (Random Early Detection) algorithm [8] is designed to cooperate with the congestion control mechanism of the TCP. It detects the beginning of congestion by monitoring the average queue size at the router (the average number of packets in the router buffer) and notifies TCP senders that congestion has occurred by intentionally dropping packets at a certain probability. The RED algorithm sets the packet dropping probability as a function of the average queue size. By keeping the average queue size low, burst packet dropping can be avoided even when packets from the same connection arrive continuously. That is, the algorithm has no bias against bursty traffic. More specifically, it uses a low–pass filter with an exponentially weighted moving average when calculating the average queue size avg, which is maintained and compared with two thresholds: a minimum threshold (min_{th}) and a maximum threshold (max_{th}). The packet dropping probability is determined in different ways according to the queue size avg:

- If $avg < min_{th}$, all arriving packets are accepted.
- If $min_{th} < avg < max_{th}$, arriving packets are dropped with probability $p_{\text{red}}(avg)$, which is defined as follows:

$$p_{\text{red}}(avg) = \frac{avg - max_{th}}{max_{th} - min_{th}} max_p \qquad (1)$$

- If $max_{th} < avg$, all arriving packets are dropped.

The RED router helps prevent TCP's retransmission timeouts, and most lost packets are thus retransmitted by the fast retransmit algorithm. It also helps avoid the phase effect [12] causing all connections to exhibit the similar window changes.

In recent works [10, 11], however, the authors have pointed out that it is difficult to choose the control parameters of RED ($max_{th}, min_{th}, max_p$) to work well, and even when those are appropriately configured, the RED routers cannot provide good performance compared with the TD routers. As opposed to these results, we present a new observation in this paper that RED is still useful, especially from a viewpoint of the fairness among many TCP connections.

B. Network Model

Fig. 1 depicts a network model used in the following analysis and simulation. It consists of sender hosts, a receiver host, a router, and links interconnecting the hosts and the router. N sender hosts share a bottleneck link of ρ [packets/sec], and sends data packets to the receiver host by TCP Reno. The propagation delay between the sender hosts and the receiver host is τ [msec]. The intermediate router has a buffer of the TD or RED discipline. The buffer size is represented by B [packets].

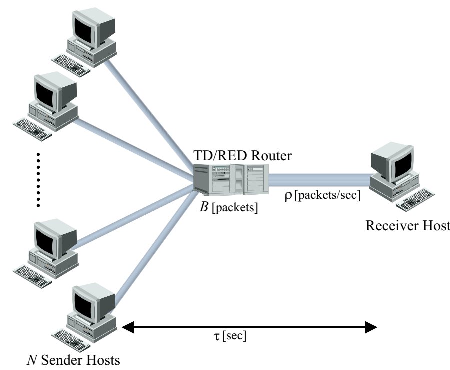


Fig. 1. Network model.

III. Fairness Analysis and Evaluation

A. Notations and Settings

We assume that a TCP connection changes its state at every RTT (Round Trip Time) and we call the interval between state changes a *round*. To describe the Markov process, we define the state of the TCP connection i by the window size w_i [packets] and a ssthreth value t_i [packets] of the current round. That is, (w_i, t_i) represents the state of the current round of connection i. By assuming that the maximum window size of the TCP connection is W_{max} [packets], w_i is ranged from 1 to W_{max}, and t_i is from 1 to $W_{max}/2$, since t_i is reset to half of w_i when packet loss is detected [1]. Then, the number of the states of this Markov process becomes $W_{max}^2/2$.

We define $P_{(w_i,t_i)(w'_i,t'_i)}$ as the state transition probability that the state of the connection i changes from (w_i, t_i) to (w'_i, t'_i). The goal of the analysis in this section is to derive the state transition matrix of $P_{(w_i,t_i)(w'_i,t'_i)}$ with consideration of the congestion control algorithm of TCP and the characteristics of TD and RED routers. The TCP connection changes its state by increasing the window size when no packet loss occurs, or by decreasing the window size when packet loss is detected [1]. That is, we must consider the following cases about the state transitions from (w_i, t_i).

1. When no packet loss occurs:
   1.a During a slow start phase, the window size is increased and the state is changed to $(\max(2w_i, t_i), t_i)$.
   1.b During a congestion avoidance phase, the window size is increased and the state is changed to $(\max(w_i + 1, W_{max}), t_i)$.
2. When packet loss occurs:
   2.a If the packet loss is detected by timeout, the state is changed to $(1, \lfloor w_i/2 \rfloor)$.
   2.b If the packet loss is detected by fast retransmit, the state is changed to $(\lfloor w_i/2 \rfloor, \lfloor w_i/2 \rfloor)$.

To derive the state transition matrix, we have to obtain the probability that each case in the above takes place. The packet loss probability in each state is affected by the number of connections and the queue size. For a meanwhile, we assume that the packet loss probability of the state (w_i, t_i), which is denoted by $p(w_i, t_i)$, is known. $p(w_i, t_i)$ in the TD and RED disciplines are derived in the following subsections in turn.

The probability that no packet loss occurs in the current state (w_i, t_i) is obtained by collecting the probabilities that w_i packets of connections are not lost. It is given by

$$p_{\text{noloss}} = (1 - p(w_i, t_i))^{w_i} \qquad (2)$$

Since the window size is simply increased when the no packet loss occurs, the cases 1.a and 1.b takes place with the following probabilities:

$$P_{(w_i,t_i)(\max(2w_i,t_i),t_i)} = p_{\text{noloss}}, \text{ if } w_i < t_i \quad (3)$$
$$P_{(w_i,t_i)(\max(w_i+1,W_{max}),t_i)} = p_{\text{noloss}}, \text{ if } w_i \geq t_i \quad (4)$$

When packet loss occurs, on the other hand, we must consider whether the lost packet is retransmitted by timeout or fast retransmit. We as denote $p_{(w_i,t_i),j,\text{TO}}$ as the probability that j packets are lost in state (w_i, t_i), and the timeout takes place to retransmit the lost packets. When one of w_i packets is lost and the window size is too small (i.e., smaller than three), the packet loss is detected by timeout [14]. That is, $p_{(w_i,t_i),j,\text{TO}}$ for j = 1 is given by

$$p_{(w_i,t_i),1,\text{TO}} = \begin{cases} \begin{pmatrix} w_i \\ 1 \end{pmatrix} p(w_i,t_i)(1-p(w_i,t_i))^{w_i-1}, & \text{if } w_i \leq 3 \\ 0, & \text{if } w_i > 3 \end{cases}$$

When more than one packets are lost – specifically, when j of w_i packets are lost – the first lost packet is retransmitted by timeout or fast retransmit with probabilities of $p_{(w_i,t_i),1,\text{TO}}$ or $1 - p_{(w_i,t_i),1,\text{TO}}$, respectively. When timeout occurs, all of lost packets are retransmitted and the window size becomes one. When fast retransmit occurs, however, the window size is halved. If the halved window size is larger than three, the next lost packet is again detected by the fast retransmit algorithm, and the window size is further halved. Otherwise, the timeout occurs. That is, all of the j packet losses can be detected by the fast retransmits if the window size is large enough to be kept larger than three when it is halved j times. We therefore have

$$p_{(w_i,t_i),j,\text{TO}} = \begin{cases} \begin{pmatrix} w_i \\ j \end{pmatrix} p(w_i,t_i)^j(1-p(w_i,t_i))^{w_i-j}, & \text{if } w_i/2^j \leq 3 \\ 0, & \text{if } w_i/2^j > 3 \end{cases}$$

From the above two equations, we can derive $p_{(w_i,t_i),\text{TO}}$, the probability that the timeout occurs in state (w_i, t_i), as:

$$p_{(w_i,t_i),\text{TO}} = \sum_{k=\lfloor \log_2(w_i/4) \rfloor + 1}^{w_i} p_{(w_i,t_i),k,\text{TO}}$$

Then, we can obtain the probabilities that cases of 2.a and 2.b take place:

$$\begin{aligned} & P_{(w_i,t_i)(1,\lfloor w_i/2 \rfloor)} = p_{(w_i,t_i),\text{TO}} \\ & P_{(w_i,t_i)(\lfloor w_i/2^j \rfloor, \lfloor w_i/2^j \rfloor)} \\ & \quad = \begin{pmatrix} w_i \\ j \end{pmatrix} p(w_i,t_i)^j(1-p(w_i,t_i))^{w_i-j}, \\ & \qquad 1 \leq j \leq \lfloor \log_2(w_i/4) \rfloor \qquad (5) \end{aligned}$$

We have now to determine, for the TD and RED disciplines, the value of $p(w_i, t_i)$ at the bottleneck router.

B. Analysis for TD router

In the TD router, packet loss occurs in bursts when the router buffer is fully occupied. To calculate $p(w_i, t_i)$ we need to take into account the following factors:

- the total number of lost packets as a result of buffer overflow
- the frequency of buffer overflow
- the number of lost packets from each TCP connection

B.1 How many packets are lost as a result of buffer overflow?

Here we denote by W the total window size of N TCP connections. A buffer overflows when W exceeds the sum the buffer size B and the bandwidth-delay product of the bottleneck link. That is, when the buffer is fully occupied, W reaches $W_f = 2\tau\rho + B$. Suppose that all of the TCP connections are in the congestion avoidance phase, W is increased by N [packets] in every RTT, since each TCP connection increases its window size by one packet [1]. Therefore, when the total window size reaches W_f, it will be increased to $(W_f + N)$ [packets] in the next round. That is, N [packets] are discarded at the TD router buffer when buffer overflow occurs.

B.2 How frequently do buffer overflows occur?

When buffer overflow occurs, some of N connections experience packet losses and decrease their window sizes. As shown in Subsection III-A, the TCP connection with window size w_i decreases its window size to $w_i/2^j$ when j packets belonging to the same connection are lost by the fast retransmit algorithm [1]. When no packet loss occurs, on the other hand, the window size increases by one. Assume that the mean window size of each connection is $\overline{w} = W_f/N$ when the buffer is fully occupied. Then, the mean window size of each connection after the buffer overflow occurs, denoted by $\overline{w'}$, is given by

$$\begin{aligned} \overline{w'} &= \sum_{j=1}^{\overline{w}} \begin{pmatrix} \overline{w} \\ j \end{pmatrix} p_f^j (1-p_f)^{\overline{w}-j} \lfloor \frac{1}{2^j} \overline{w} \rfloor \\ & \quad + (1-p_f)^{\overline{w}} (\overline{w}+1) \qquad (6) \end{aligned}$$

where $p_f = N/W_f$, which is the packet loss probability when the buffer overflow occurs. Note that the above equation includes the case where the TCP connection experiences the timeout, but in that case, $\lfloor \frac{1}{2^j} \overline{w} \rfloor$ is zero. Then, the mean of the total window size just after the buffer overflow, $\overline{W'}$, becomes $N \cdot \overline{w'}$. Since the total window size is increased by N packets per RTT, the probability that the buffer overflows occur in the current round (denoted by p_{overflow}) can be calculated as follows:

$$p_{\text{overflow}} = \frac{1}{N(W - N \cdot \overline{w'})} \qquad (7)$$

B.3 How many packets are lost from each TCP connection?

We have shown in Subsection III-B.1 that the total number of lost packets in the event of buffer overflow is given by N. We assume that the number of lost packets belonging to each connection is proportional to the size of that connection's window. Then l_i, the number of lost packets of connection i, is given by

$$l_i = \frac{w_i}{W} \cdot N \qquad (8)$$

B.4 Derivation of $p(w_i, t_i)$

Using Eqs. (7) and (8), we can obtain the following equation for $p(w_i, t_i)$:

$$p(w_i, t_i) = \min\left(1, p_{\text{overflow}} \cdot \frac{l_i}{w_i}\right) \qquad (9)$$

C. Analysis for RED router case

In RED, the packet loss probability is determined from the average queue size and control parameters by Eq. (1). But because our analysis is based on the Markov process model, we use instead of Eq. (1) the following function for the packet discarding probability of RED. That is, we use the instantaneous queue length (q) instead of the average queue length (avg). This makes little difference with regard to the packet discarding probability of RED because q always fluctuates around avg.

$$p'_{\text{red}}(q) = \begin{cases} 0, & \text{if } q \le min_{th} \\ \dfrac{q - max_{th}}{max_{th} - min_{th}} max_p, & \text{if } min_{th} \le q \le max_{th} \\ \frac{q \cdot max_p + (q - max_{th})}{q}, & \text{if } q \ge max_{th} \end{cases}$$

The third form of this equation corresponds to the case where the queue length equals or exceeds max_{th} and all of incoming packets are dropped. Note that the previous work [5-7] did not consider this condition. Although the above equation is an approximation for the behavior of the RED router, in Section IV it will be shown to a good estimate of the packet discarding probability of RED.

To calculate $p(w_i, t_i)$ for the RED router, we first derive the average packet loss probability of the RED router by using a steady–state analysis. Note that the appropriateness of using the average queue length of a RED router to derive the window size distribution of a TCP connection has been already validated in [5]. Here we use W^* and q^* to respectively represent the steady-state values of the total window size of N TCP connections and the queue size. i.e.,

$$W^* = 2\tau\rho + q^* \tag{10}$$

The mean window size of each connection becomes $\overline{w}^* = W^*/N$. Since RED discards the incoming packets with probability $p_{\text{red}}(q^*)$, the following equation should – from Eq. (6) and considerations similar to those that in Subsection III-B.2 – hold in the steady state:

$$W^* = N \left\{ \sum_{j=1}^{\overline{w^*}} \binom{\overline{w^*}}{j} p'_{\text{red}}(q^*)^j (1 - p'_{\text{red}}(q^*))^{\overline{w}-j} \frac{1}{2^j} \overline{w^*} + (1 - p'_{\text{red}}(q^*))^{\overline{w^*}} (\overline{w^*} + 1) \right\} \tag{11}$$

We obtain q^* and W^* by solving Eqs. (10) and (11). If the window sizes of the other connections except connection i are equal to W^*/N, the current queue size (q) is given by

$$q = \frac{N-1}{N} W^* + w_i - 2\tau\rho$$

where w_i is the current window size of connection i. Then $p(w_i, t_i)$ is given by

$$p(w_i, t_i) = p'_{\text{red}}(q) \tag{12}$$

IV. Numerical Results and Discussions

In this subsection, we examine the accuracy of our analysis approach by comparing its results with the results of simulation experiments, and we discuss the fairness of TD and RED routers. The simulation results were obtained by using ns-2 simulator [15].

We set as network parameters $\rho = 187.5$ [packets/sec] ($\simeq 1.5$ [Mbps]), $\tau = 2$ [msec] and $N = 1,000$ (refer to Fig. 1). The analysis and simulation results for TD and RED router cases are shown in Fig. 2, where probability density is plotted as a function of the window size of the TCP connection. The buffer size B for TD and the max_{th} for RED are both set to 8,000 [packets] (Fig. 2(a)), to 5,000 [packets] (Fig. 2(b)), and to 3,000 [packets] (Fig. 2(c)). These results show that the analysis gives a good estimate of the window size distribution regardless of the buffer size. Even for a buffer size of 3,000 [packets], which is too small for 1,000 TCP connections in the TD router (only three for each connection) and many buffer overflows take place at the router buffer[9], the analysis results are very close to the simulation results and our analysis well follows the behavior of TD router.

TABLE I

Effect of max_{th} on the throughput and fairness.

Router Discipline	Throughput	Fairness
TD	1.38 [Mbps]	0.80
RED ($max_{th} = 8,000$)	1.41 [Mbps]	0.94
RED ($max_{th} = 5,000$)	1.41 [Mbps]	0.95
RED ($max_{th} = 3,000$)	1.42 [Mbps]	0.81

Furthermore, comparing the results of TD and RED routers in Fig. 2, we can make the following observations about their fairness. When the TD's buffer size is equal to the RED's max_{th}, the window size varies more widely for TCP connections at the TD router. This can be understood more clearly by looking at Fig. 3, which shows the analysis results of 99.99% and 99.9999% values of the window size distribution. We can see that TCP connections at the TD router tend to have much larger window sizes than those at the RED router. That is, the RED router results in a smaller dispersion of the window sizes, which implies that the RED router improves fairness among many TCP connections.

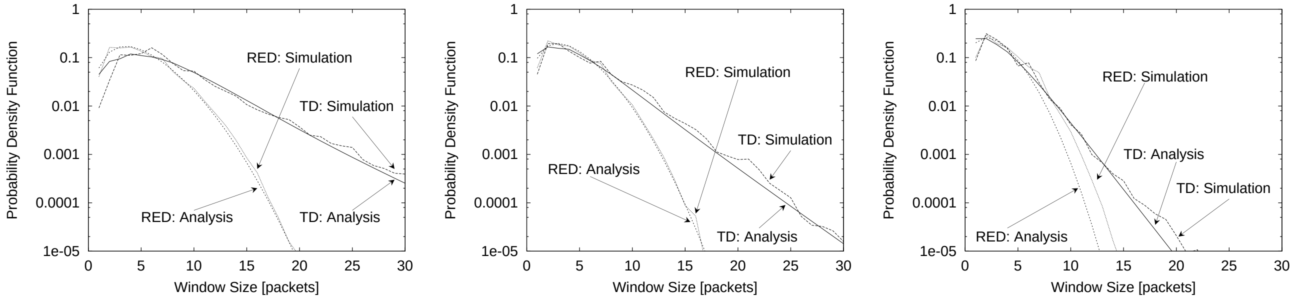
When the max_{th} of the RED router is set too small, however, the fairness of the RED router is degraded. This can be seen in Fig. 2(c), where the probability of a very small window size (< 5) is much higher than in Figs. 2(a) and 2(b). When max_{th} is too small, RED's probabilistic packet dropping does not work well, and the incoming packets are discarded in bursts. This situation corresponds to the third form of Eq. (10): when the queue length exceeds max_{th} all incoming packets are dropped.

The total throughput of the bottleneck link and the fairness values through simulation experiments are shown in Table I. We use a *fairness index* defined in [16] to evaluate fairness. When that the throughput of the TCP connection i is denoted x_i and the number of TCP connections is denoted n, the fairness index f is defined as follows:

$$f = \frac{(\sum_{i=1}^{n} x = i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{13}$$

Note that the value of f ranges from 0 to 1, and a larger value shows a better fairness.

Table I clearly shows that while the throughput of RED remains almost same as that of TD regardless of the max_{th} value, RED can provide better fairness than TD if max_{th} is set appropriately. However, it does not help improve the fairness when a too small value of max_{th} is used. Therefore, max_{th} for the RED router should be large enough for the number of connections accommodated at the router. In [10, 11], the authors have concluded that there is little reason to deploy RED to Internet routers since RED routers cannot provide not so much throughput improvement of the bottleneck router and that it is sometimes lower than that of TD routers. We make the opposite conclusion. RED is still valuable to apply to the router since



(a) $B = max_{th} = 8,000$ (b) $B = max_{th} = 5,000$ (c) $B = max_{th} = 3,000$

Fig. 2. Analysis and simulation results.

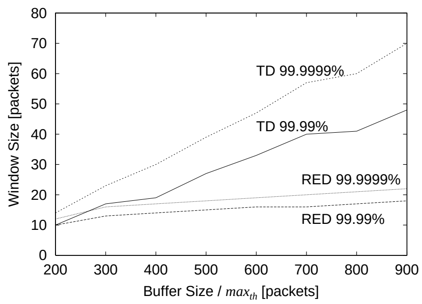


Fig. 3. 99.99/99.9999 percentile values of the window size.

it can improve the fairness among connections under the condition that max_{th} is appropriately chosen.

V. Conclusion

This paper has described a new analysis method using the Markov process model to determine the window size distribution of many TCP connections sharing the bottleneck router. We considered the RED router and the TD router, which drop incoming packets probabilistically and in bursts, respectively. Through numerical examples, we have shown that our analysis can give good estimates of the window size distributions at TD and RED routers. We have also discussed the fairness of TD and RED routers by using the analysis results and have concluded that RED is very effective in improving the fairness among many TCP connections.

In the past literature on stochastic/Markov modeling of TCP behavior, as well as in this paper, the homogeneous network model, where all TCP connections have the same propagation delays, has been used. The fundamental characteristics of the TCP has been revealed through these studies, but in the future we will extend the analysis in this paper and treat the network model in which each TCP connection has a different propagation delay and bandwidth. This will make clear the effect of the propagation delay on the window size distribution of each TCP connection. We also plan to use the analysis in this paper to estimate the throughput of TCP connections.

Acknowledgements

This work was partly supported by the Research for the Future Program of the Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems," by a Grant-in-Aid for Scientific Research (B) 13450158 from the Japanese Ministry of Education, Culture, Sports and Science and Technology of Japan, and by the "Research on High-performance WWW server for the Next-Generation Internet" program of from the Telecommunications Advancement Foundation.

References

[1] W. Richard Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, Reading, Massachusetts, 1994.

[2] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 67–82, July 1997.

[3] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proceedings of ACM SIGCOMM'98*, Aug. 1998, pp. 303–314.

[4] Masahiko Nabe, Masayuki Murata, and Hideo Miyahara, "Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines," *Performance Evaluation*, vol. 34, no. 4, pp. 249–271, Dec. 1999.

[5] Archan Misra, Teunis Ott, and John Baras, "The window distribution of multiple TCPs with random loss queues," in *Proceedings of IEEE GLOBECOM '99*, Nov. 1999.

[6] Jitendra Padhye, Victor Firoiu, and Don Towsley, "A stochastic model for TCP reno congestion avoidance and control," Tech. Rep. 99-02, CMPSCI Technical Report, Oct. 1999.

[7] Claudio Casetti and Michela Meo, "A new approach to model the stationary behavior of TCP connections," in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000.

[8] Sally Floyd and Van Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[9] Robert Morris, "TCP behavior with many flows," in *Proceedings of IEEE International Conference on Network Protocols*, October 1997.

[10] Martin May, Jean Bolot, Christophe Diot, and Bryan Lyles, "Reasons not to deploy RED," in *Proceedings of IWQoS'99*, June 1999.

[11] Mikkel Christiansen, Kevin Jeffay, David Ott, F. Donelson Smith, "Tuning RED for web traffic," in *Proceedings of ACM SIGCOMM 2000*, August 2000.

[12] Sally Floyd and Van Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Experience*, vol. 3, no. 3, pp. 397–413, Aug. 1992.

[13] Lixia Zhang and David D. Clark, "Oscillating behavior of network traffic: A case study simulation," *Internetwoking: Research and Experience*, vol. 1, pp. 101–112, 1990.

[14] Kevin Fall and Sally Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 2, pp. 5–21, July 1996.

[15] "Network simulator - `ns` (version 2).," available from `http://www.isi.edu/nsnam/ns/`.

[16] Raj Jain, "Throughput fairness index: An explanation," *ATM Forum Contribution 99-0045*, February 1999.