RED ゲートウェイの定常状態解析
— 安定性および過渡特性 —

大崎 博之 †　米良 祐一郎 ‡　村田 正幸 †　宮原 秀夫 †

† 大阪大学大学院 基礎工学研究科
〒 560-8531 大阪府豊中市待兼山町 1-3

‡ 奈良先端科学技術大学院大学 情報科学研究科
〒 630-0101 奈良県生駒市高山町 8916-5

(Phone) +81-6-6850-6588　(Fax) +81-6-6850-6589
(E-mail) oosaki@ics.es.osaka-u.ac.jp

あらまし　近年、エンド–エンド間で動作する TCP の輻輳制御機構を補助するために、ゲートウェイにおける輻輳制御機構がいくつか提案されている。この中で、現在もっとも有望と考えられ、実際に実装されつつあるのは、ゲートウェイにおいて意図的にパケット棄却を発生させる RED (Random Early Detection) ゲートウェイである。しかし、これまで RED ゲートウェイの特性は十分には明らかにされていない。そこで本報告では、TCP によってフロー制御されたトラヒックに対する、RED ゲートウェイの定常状態特性を解析する。まず、定常状態における TCP のウィンドウサイズや、RED ゲートウェイのバッファ内パケット数を導出する。また、制御理論を適用することにより、ネットワークの安定条件および過渡特性をあらわす性能指標を導出する。さらに、数値例およびシミュレーション結果により、RED ゲートウェイの制御パラメータと定常特性との関係を明らかにする。その結果、(1) RED ゲートウェイのバッファ占有量は、ほぼ max_p (maximum packet marking probability) によって決まること、(2) TCP のコネクション数やネットワークの (帯域) × (遅延) が大きくなるにつれ、ネットワークがより安定すること、(3) 過渡特性を最適化するためには、min_{th} (minimum threshold) を慎重に決める必要があること、などが明らかになった。
和文キーワード　RED (Random Early Detection) ゲートウェイ、TCP (Transmission Control Protocol)、安定性、過渡特性

Steady State Analysis of the RED Gateway
— Stability and Transient Behavior —

Hiroyuki Ohsaki†　Yuichiro Mera‡　Masayuki Murata†　Hideo Miyahara†

†Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

‡Graduate School of Information Science, Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara 630-0101, Japan

(Phone) +81-6-6850-6588　(Fax) +81-6-6850-6589
(E-mail) oosaki@ics.es.osaka-u.ac.jp

Abstract　Several gateway-based congestion control mechanisms have been proposed to support an end-to-end congestion control mechanism of TCP (Transmission Control Protocol). One of promising gateway-based congestion control mechanisms is a RED (Random Early Detection) gateway. Although effectiveness of the RED gateway is fully dependent on a choice of control parameters, it has not been fully investigated how to configure its control parameters. In this paper, we analyze the steady state behavior of the RED gateway by explicitly modeling the congestion control mechanism of TCP. We first derive the equilibrium values of the TCP window size and the buffer occupancy of the RED gateway. Also derived are the stability condition and the transient performance index of the network by using a control theoretic approach. Numerical examples as well as simulation results are presented to clearly show relations between control parameters and the steady state behavior. Our findings are: (1) max_p (maximum packet marking probability) mostly affects the RED's buffer occupancy, (2) the network becomes more stable as the number of TCP connections or the bandwidth–delay product increases, and (3) min_{th} (minimum threshold) is a key parameter for optimizing the transient performance.
英文 key words　RED (Random Early Detection) gateway, TCP (Transmission Control Protocol), Stability, Transient behavior

1 Introduction

Several gateway-based congestion control mechanisms have been proposed to support an end-to-end congestion control mechanism of TCP [1, 2, 3]. One of promising gateway-based congestion control mechanisms is a RED (Random Early Detection) gateway [2]. The key idea of the RED gateway is to keep the average queue length (i.e., the average number of packets in the buffer) low. Basically, the RED gateway randomly discards an incoming packet with a probability that is proportional to the average queue length. The operation algorithm of the RED gateway is so simple that the RED algorithm can be easily implemented. The authors of [2] have claimed advantages of the RED gateway over a conventional drop-tail gateway as follows: (1) the average queue length is kept low, so that an end-to-end delay of a TCP connection is also kept small, (2) the RED gateway has no bias against bursty traffic as in the drop-tail gateway, and (3) a global synchronization problem of TCP connections found in the drop-tail gateway is avoided.

In this paper, we analyze steady state behavior of the RED gateway by explicitly modeling the congestion control mechanism of TCP. We first derive equilibrium values of TCP's window size and the buffer occupancy of the RED gateway. Also derived are a stability condition and a transient performance index by using a control theoretic approach. Numerical examples as well as simulation results are presented to clearly show relations between control parameters and steady state behavior. Our findings are — (1) max_p (maximum packet marking probability) mostly affects RED's buffer occupancy, (2) a network becomes more stable as the number of TCP connections or a bandwidth–delay product increases, and (3) min_{th} (minimum threshold) is a key paramter for optimizing transient performance.

2 RED Algorithm and Analytic Model

Figure 1 illustrates our analytic model that will be used throughout this paper. It consists of a single RED gateway and the number N of TCP connections. All TCP connections have an identical (round-trip) propagation delay, which is denoted by τ [ms]. We assume that the processing speed of the RED gateway, which is denoted by B [packet/ms], is the bottleneck of the network. Namely, transmission speeds of all links are assumed to be faster than the processing speed of the RED gateway.

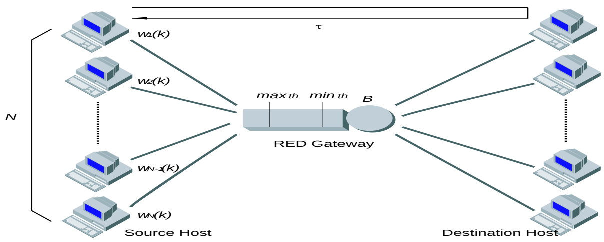

Figure 1: Analytic model.

We model the congestion control mechanism of TCP version Reno [4, 5] at all source hosts. A source host has a window size, which is controlled by the congestion control mechanism of TCP. By letting w_n be a current window size of the source host n ($1 \le n \le N$), it is allowed to send the number w_n of packets without receipt of ACK (Acknowledgement) packets. In other words, the source host n can send a bunch of w_n packets during a round-trip time. We then model the entire network as a discrete-time system, where a time slot of the system corresponds to a round-trip time of TCP connections. Note that a slot length changes at every slot since a round-trip time changes because of a queueing delay at the RED gateway. We define $w_n(k)$ [packet] as the window size of the source host n at slot k. All source hosts are assumed to have enough data to transmit; the source host n is assumed to always send the number $w_n(k)$ of packets during slot k.

The RED gateway has several control parameters. Let min_{th} and max_{th} be the *minimum threshold* and the *maximum threshold* of the RED gateway, respectively. These threshold values are used to calcurate a packet marking probability for every incoming packet. The RED gateway maintains an average queue length (i.e., the average number of packets waiting in the buffer). The RED gateway uses an exponential weighted moving average (EWMA), which is a sort of low-pass filters, to calculate the average queue length from the current queue length. More specifically, let q and $\overline{q}$ be the current and the average queue length. At every packet arrival, the RED gateway updates the average queue length $\overline{q}$ as

$$\overline{q} \quad \leftarrow \quad (1 - w_q)\,\overline{q} + w_q\, q \tag{1}$$

where w_q is a weight factor. We define $q(k)$ [packet] and $\overline{q}(k)$ [packet] be the current and the average queue lengths at slot k. We assume that both q and $\overline{q}$ will not change during a slot. As discussed in [6], this assumption is realistic for a small w_q.

Using the average queue length, the RED gateway calculates a packet marking probability p_b at every arrival of an incoming packet. Namely, the RED gateway calculates p_b as

$$p_b \quad = \quad \begin{cases} 0 & \text{if } \overline{q} < min_{th} \\ 1 & \text{if } \overline{q} \ge max_{th} \\ max_p \left(\frac{\overline{q} - min_{th}}{max_{th} - min_{th}} \right) & \text{otherwise} \end{cases} \tag{2}$$

where max_p (*maximum packet marking probability*) is another control parameter. The RED gateway randomly discards each incoming packet with probability p_b. The packet marking mechanism of the RED gateway is not per-flow basis, and the same packet marking probability is used for all TCP connections. Refer to [2] for the detailed algorithm of the RED gateway.

3 Steady State Analysis

3.1 Derivation of State Transition Equations

We first obtain the packet dropping probability at slot k. At the beginning of slot k, the source host n sends the number $w_n(k)$ of packets into the network. The RED gateway marks each arriving packet based on the average queue length. Since the average queue length $\overline{q}$ is assumed to be fixed within a slot, the packet marking probability $p_b(k)$ at slot k is also fixed. Provided that the average queue length $\overline{q}$ is between min_{th} and max_{th}, $p_b(k)$ is obtained from Eq. (2) as

$$p_b(k) \quad = \quad max_p \left(\frac{\overline{q}(k) - min_{th}}{max_{th} - min_{th}} \right) \qquad (3)$$

Every arriving packet at the RED gateway is marked with probability, $p_b(k)/(1 - count \cdot p_b(k))$, where $count$ is the number of unmarked packets that have arrived since the last marked packet [2]. The number of unmarked packets between two consecutive marked packets can be represented by an uniform random variable in $\{1, 2, \cdots, 1/p_b(k)\}$. Let $\overline{X}_k$ be the expected number of unmarked packets between two consecutive marked packets at slot k. $\overline{X}_k$ is obtained as

$$\overline{X}_k \quad = \quad \frac{1/p_b(k) + 1}{2} \qquad (4)$$

We next derive state transition equations of the network. The state transition equations that we will derive hereafter are the basis of our steady state analysis. If all packets sent from a source host are *not* marked at the RED gateway, corresponding ACK packets will be returned to the source host after a round-trip time. In this case, the congestion control mechanism of TCP increases the window size by one packet. On the contrary, if any of packets are discarded by the RED gateway, the congestion control mechanism of TCP throttles the window size to half. We assume that all packet losses are detectable by duplicate ACKs [5]. Namely, the number of discarded packets in a slot is assumed to be less than three. The probability that at least one packet is discarded from $w_n(k)$ packets is given by $\frac{w(k)}{1/p_b(k)} = w(k)\, p_b(k)$. Therefore, the window size at slot $k+1$ is given by

$$w(k+1) \quad = \quad \begin{cases} \frac{w(k)}{2} & \text{with prob. } w(k)\, p_b(k) \\ w(k) + 1 & \text{otherwise} \end{cases} \qquad (5)$$

By our definition of a slot, all packets that have sent in slot k are to be acknowledged until the beginning of slot $k+1$. The current queue length at slot $k+1$ is given by

$$q(k+1) \quad = \quad \sum_{n=1}^{N} w_n(k) - B\tau \qquad (6)$$

We then focus on a relation between $\overline{q}(k)$ and $\overline{q}(k+1)$. As explained in Section 2, the RED gateway updates the average queue length at every packet arrival according to Eq. (1). In other words, the average queue length is updated $\sum_{n=1}^{N} w_n(k)$ times during slot k. Recall that the current queue length $q(k)$ is assumed to be fixed during a slot. The average queue length in slot $k+1$ is then given by

$$\overline{q}(k+1) \quad = \quad (1 - w_q)^{\sum_{n=1}^{N} w_n(k)} \overline{q}(k) + \left\{ 1 - (1 - w_q)^{\sum_{n=1}^{N} w_n(k)} \right\} q(k) \qquad (7)$$

The network model depicted in Fig. 1 is fully described by state transition equations given by Eqs. (5)–(7), and the state vector $\mathbf{x}(k)$

$$\mathbf{x}(k) = [w_1(k) \ \cdots \ w_N(k) \ q(k) \ \overline{q}(k)]^T \qquad (8)$$

3.2 Derivation of Average State Transition Equations

The state transition equations given by Eqs. (5)–(7) contain a probability, because the RED gateway marks each arriving packet in a probabilistic way. To analyze the steady state behavior of the RED gateway, we introduce *average state transition equations* that represent a typical behavior of TCP connections and the RED gateway. We also introduce a *sequence*, which is a series of adjacent slots in which all packets from a source host have been unmarked by the RED gateway. A typical evolution of a window size within a sequence is illustrated by Fig. 2. Note that this figure shows the case where the RED gateway discards one or more packets in slot $k-1$. The window size $w_n(k)$ is changed according to Eq. (5). Namely, as long as no packets from the source host is discarded by the RED gateway, the window size is incremented by one packet at every slot. If one or more packets are discarded, the window size is halved. Such a process will be repeated indefinitely by the congestion control mechanism of TCP.

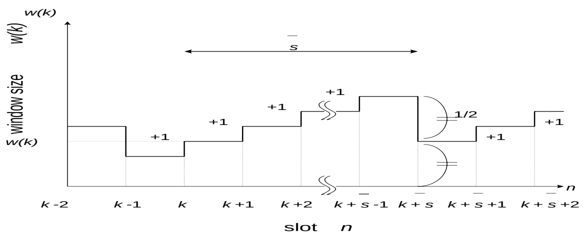

Figure 2: A typical evolution of the window size in a sequence.

The key idea in our steady state analysis is to treat the network as a discrete-time system where a time slot corresponds to a sequence, instead of a slot defined in Section 2. Let $\overline{s}_k$ be the average number of slots that consists a sequence beginning from slot k. Then, $\overline{s}_k$ satisfies the following inequality.

$$\overline{X}_k \quad \leq \quad \sum_{i=0}^{\overline{s}_k - 1} \sum_{n=1}^{N} w_n(k+i).$$

In what follows, due to space limitation, we only show the case where window sizes of all source hosts are synchronized; the window size of the source host n is equally given by $w(k)$. However, our steady state analysis can be easily applied to the case where window sizes of all source hosts are not identical. When window sizes of all source hosts are identical, solving the above equation for $\overline{s}_k$ yields

$$\overline{s}_k = \frac{1}{2} - w(k) + \frac{\sqrt{N^2(1-2w(k))^2 + 8N\overline{X}_k}}{2N} \quad (9)$$

Assuming that the number of slots in a sequence is deterministically given by $\overline{s}_k$, we derive the average state transition equations from slot k to $k+\overline{s}_k$. Since the RED gateway discards one or more packets during slot $k+\overline{s}_k-1$ (see Fig. 2), the average state transition equation from $w(k)$ to $w(k+\overline{s}_k)$ is obtained from Eq. (5) as

$$w(k+\overline{s}_k) = \frac{w(k)+\overline{s}_k-1}{2} \quad (10)$$

Similarly, the average state transition equation from $q(k)$ to $q(k+\overline{s}_k)$ is obtained from Eq. (6) as

$$q(k+\overline{s}_k) = N\,(w(k)+\overline{s}_k-1) - B\,\tau \quad (11)$$

We next derive the average state transition equation from $\overline{q}(k)$ to $\overline{q}(k+\overline{s}_k)$. Using Eq. (7), $\overline{q}(k+\overline{s}_k)$ is obtained as

$$\begin{aligned}\overline{q}(k+\overline{s}_k) &= (1-w_q)^{A_k}\overline{q}(k+\overline{s}_k-1) \\ &\quad + \left\{1-(1-w_q)^{A_k}\right\} q(k+\overline{s}_k-1) \quad (12)\end{aligned}$$

where

$$A_k = N\,w(k+\overline{s}_k-1)$$

Recall that $\overline{X}_k$ is the average number of unmarked packets between two consecutive marked packets, i.e., the average number of unmarked packets in a sequence. By assuming that the current queue length does not change excessively, $\overline{q}(k+\overline{s}_k)$ is approximated as

$$\begin{aligned}\overline{q}(k+\overline{s}_k) &\simeq (1-w_q)^{\overline{X}_k}\,\overline{q}(k) \\ &\quad + \left\{1-(1-w_q)^{\overline{X}_k}\right\} q(k) \quad (13)\end{aligned}$$

The average state transition equations given by Eqs. (10), (11), and (13) describe the average behaviors of the window size, the current queue length, and the average queue length, respectively. Our approximated analysis will be validated in Section 5.

3.3 Derivation of Average Equilibrium Value

Because of the nature of TCP's congestion control mechanism, the window size of a source host oscillates indefinitely and is never converged to a equilibrium state. In what follows, we therefore derive a *average equilibrium value*, which is defined as the expected value in steady state, to understand a typical behavior of TCP connections and the RED gateway. Let w^*, q^*, and $\overline{q}^*$ be the average equilibrium value of the window size $w(k)$, the current queue length $q(k)$, and the average queue length $\overline{q}(k)$, respectively. The average equilibrium value can be easily obtained from Eqs. (10), (11), and (13) by equating $w(k) = w(k+\overline{s}_k)$ and so on.

4 Stability and Transient Behavior

In this section, we analyze stability and transient behavior of the network by using a control theoretic approach. Since incoming traffic at the RED gateway is flow-controlled by the congestion control mechanism of TCP, the window size of a source host oscillates and never converges to a equilibrium value. The operation of the RED gateway is changed by a probability, so it is difficult to analyze its stability and transient behavior. We therefore focus on their *average behaviors* by utilizing the average state transition equations derived in Section 3. In particular, we derive the stability condition and the transient performance index of the network by considering the average behaviors of the TCP connections and the RED gateway.

In Section 2, the average state transition equations for the discrete-time system illustrated by Fig. 1 are obtained in Eqs. (10), (11), and (13). The average equilibrium values of the system states are also obtained in Section 3. Let us introduce $\delta\mathbf{x}(k)$ as the difference between the state vector $\mathbf{x}(k)$ and the average equilibrium point.

$$\delta\mathbf{x}(k) \equiv \left[w(k)-w^* \quad q(k)-q^* \quad \overline{q}(k)-\overline{q}^*\right]^T \quad (14)$$

By lineally approximating $w(k)$, $q(k)$, and $\overline{q}(k)$ around their average equilibrium values, $\delta\mathbf{x}(k+\overline{s})$ can be written as $\mathbf{A}\,\delta\mathbf{x}(k)$, where $\mathbf{A}$ is a state transition matrix. It is known that the stability and transient behavior of the system given by Eqs. (10), (11) and (13) around the equilibrium point are determined by the roots of the characteristic equation $|s\mathbf{A}-\mathbf{I}| = 0$ [7]. Let $s_i (1 \le i \le 3)$ be the roots of the above characteristic equation. The system is stable if and only if all s_i's lie in the unit circle (*stability condition*). The stability condition for a given state transition matrix can be easily computed by, for example, Jury's criterion [7].

Also known is that the transient behavior of the system given by Eqs. (10), (11) and (13) around the equilibrium point is determined by s_i's. More specifically, the transient performance of the system is mostly determined by the following value (*transient performance index*).

$$s_{max} = \max_i(|s_i|) \quad (15)$$

The smaller the transient performance index is, the faster the converges to the equilibrium point.

5 Numerical Examples and Simulation Results

5.1 Validity of Approximated Analysis

We present a simulation result to demonstrate the validity of our analysis since we have made several assumptions. We run simulation experiments for the same network model given by Fig. 1 using a network simulator *ns* [8]. We use the following network parameters in simulation: the processing speed of the RED gateway $B = 2$ [packet/ms] (i.e., about 1.5 Mbit/s for the packet size of 1,000 bytes) and the propagation delay $\tau = 1$ [ms]. For RED control parameters, the values recommended in [2] are used.

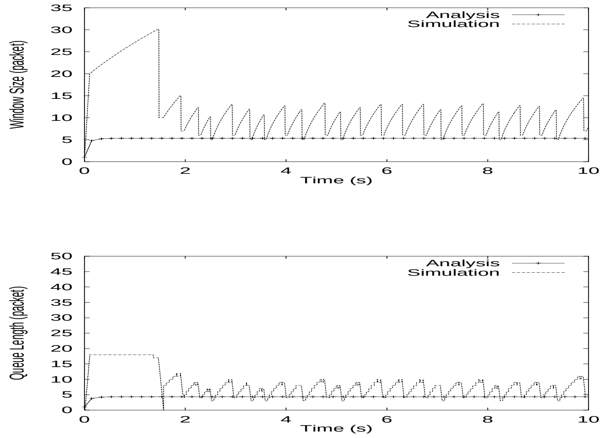

Figure 3: Comparison between analytic results and simulation results ($B = 2$ [packet/ms], $\tau = 1$ [ms], $N = 1$).

Evolutions of TCP's window size and the current queue length of the RED gateway in simulation experiments are plotted in Fig. 3 for $N = 1$. We numerically compute the average window size $w(k)$ and the average queue length $q(k)$ (i.e., the minimum values in each sequence) from Eqs. (10), (11) and (13). In the figure, those analytic results are also plotted. One can find from these figures that our analytic results match the minimum values of the window size and the current queue length, indicating a close agreement of our analytic results with simulation ones.

5.2 Discussion on Average Equilibrium Values

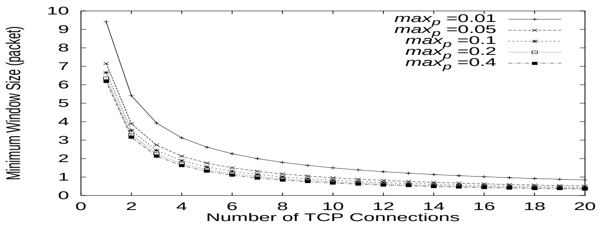

Figure 4: Average equilibrium value of the window size for different numbers of TCP connections ($B = 2$ [packet/ms], $\tau = 1$ [ms]).

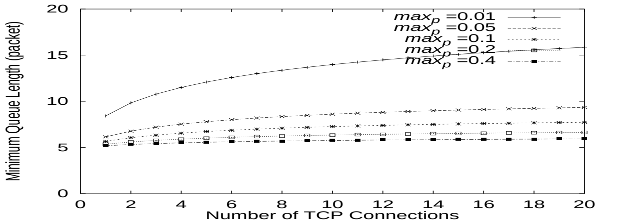

Figure 5: Average equilibrium value of the queue length for different numbers of TCP connections ($B = 2$ [packet/ms], $\tau = 1$ [ms]).

Figures 4 and 5 show the average equilibrium values of the window size w^* and the queue length q^* are plotted, respectively. Recall that w^* and q^* represent expected values of the minimum window size and the minimum queue length in steady state. We used the following network parameters: the processing speed of the RED gateway $B = 2$ [packet/ms] and the (round-trip) propagation delay $\tau = 1$ [ms]. The number of TCP connections N is varied between 1 and 20. The maximum packet marking probability max_p is also varied between 0.001 and 0.4, while other control parameters of the RED gateway are set to the values recommended in [2]. These figures indicate that the window size is heavily dependent on the number of TCP connections N, and that the queue length is mostly determined by the control parameter max_p.

5.3 Discussion on Stability and Transient Behavior

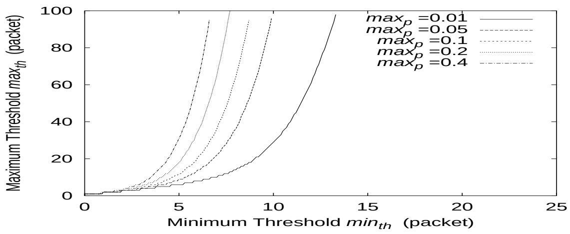

Figure 6: Stability region in the min_{th}–max_{th} plane ($B = 2$ [packet/ms], $N = 1$, $\tau = 1$ [ms]).

Figure 6 shows the stability region of the network in the max_{th}–min_{th} plane. In this figure, the number of TCP connections N is set to 1. Each line in the figure shows a boundary line of a stability conditions for different values of max_p. This figure indicate that the network is stable if the point (min_{th}, max_{th}) resides a left-side of the boundary line. For other control parameters, we used the same values used in Fig. 4: $B = 2$ [packet/ms] and $\tau = 1$ [ms]. One can find that the stability region becomes large as the maximum packet marking probability max_p decreases. This shows that the network can be stabilized with a large min_{th} if the packet marking probability max_p is set to a small value.

We next investigate how the stability region changes when the processing speed of the RED gateway B or the

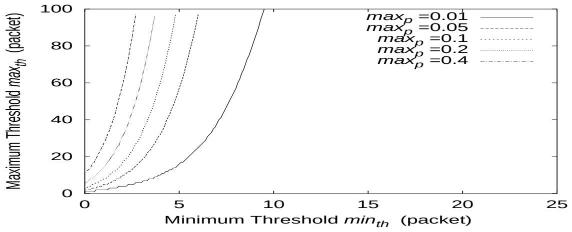

Figure 7: Stability region with a larger processing speed ($B = 10$ [packet/ms], $N = 1$, $\tau = 1$ [ms]).

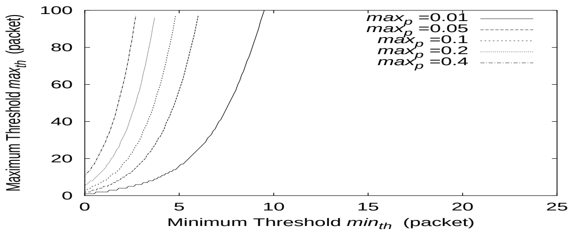

Figure 8: Stability region with a larger propgation delay ($B = 2$ [packet/ms], $N = 1$, $\tau = 5$ [ms]).

propagation delay of TCP connections τ is increased. Figure 7 is the case with a larger processing speed $B = 10$ [packet/ms]. Figure 8 is the case with a larger propagation delay $\tau = 5$ [ms]. Other parameters are identical to those used in Fig. 6. By comparing Figs. 7 and 8, one can find that the stability regions in these figures are completely identical. This indicates that for system stability, the effect of increasing the processing speed B is same with the effect of increasing the propagation delay τ. This phenomenon can be explained from Eq. (11) where B and τ are shown in the product form of $B \times \tau$.

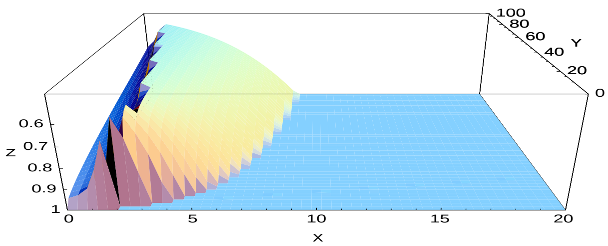

Figure 9: Relation among transient performance index and two threshould values ($B = 2$ [packet/ms], $N = 1$, $\tau = 1$ [ms], $max_p = 0.1$)

As explained in Section 4, stability and transient behavior of the network is determined by the roots of the characteristic equation. In Fig. 9, we plot the transient performance index s_{max} defined by Eq. (15) for different values of the maximum threshold max_{th} and the minimum threshold min_{th}. This figure corresponds to the case of $max_p = 0.1$ in Fig. 6. Three axes represent the minimum threshold min_{th} (x), the maximum threshold max_{th} (y), and the transient performance index s_{max} (z). These figures indicates that the network is stable if s_{max} is less than 1, and that the transient performance is better (e.g., faster convergence) with a smaller s_{max}. One can find that the optimal point (min_{th}, max_{th}) is almost independent of max_{th}. In other words, the transient performance is mostly determined by the minimum threshold min_{th}.

6 Conclusion and Future Works

In this paper, we have analyzed the steady state behavior of the RED gateway when incoming traffic is flow-controlled by the congestion control mechanism of TCP. We Of all our findings, most important results were: (1) max_p (maximum packet marking probability) mostly affects the RED's buffer occupancy, (2) the network becomes more stable as the number of TCP connections or the bandwidth–delay product increases, and (3) min_{th} (minimum threshold) is a key parameter for optimizing the transient performance.

As a future work, it would be of importance to extend our steady state analysis to relax strong assumptions. In particular, extending our steady state analysis to allow different propagation delays of TCP connections would give us much insight for better understanding of the RED gateway. More simulation experiments would be necessary to investigate the behavior of the RED gateway in complex network configurations.

References

[1] E. Hashem, "Analysis of random drop for gateway congestion control," *Technical Report MIT-LCS-TR-465*, 1989.

[2] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, August 1993.

[3] B. Barden et al., "Recommendations on queue management and congestion avoidance in the Internet," *Request for Comments (RFC) 2309*, April 1998.

[4] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. New York: Addison-Wesley, 1994.

[5] W. R. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," *Request for Comments (RFC) 2001*, January 1997.

[6] M. May, T. Bonald, and J.-C. Bolot, "Analytic evaluation of RED performance," in *Proceedings of IEEE INFOCOM 2000*, March 2000.

[7] R. Isermann, *Digital control systems, Volume 1: fundamentails, deterministic control*. Springer-Verlag Berlin Heidelberg, 1989.

[8] "LBNL network simulator (ns)." available at `http://www-nrg.ee.lbl.gov/ns/`.