Analyzing the Impact of TCP Connections Variation on Transient Behavior of RED Gateway

Motohisa Kisimoto[1], Hiroyuki Ohsaki[2], and Masayuki Murata[2]

[1] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560–8531, Japan
kisimoto@ics.es.osaka-u.ac.jp

[2] Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka 567-0043, Japan
{oosaki, murata}@cmc.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. In this paper, we analyze the transient behavior of the RED gateway when the number of TCP connections is changed. We model both TCP connections and the RED gateway as a single feedback system, and analyze the dynamics of the number of packets in the RED gateway's buffer when the number of TCP connections is increased or decreased. Through numerical examples, we quantitatively show that the transient performance of the RED gateway is quite sensitive to system parameters such as the total number of TCP connections, the processing speed of the RED gateway. We also show that control parameters of the RED gateway have little impact on the transient behavior of the RED gateway.

1 Introduction

Several gateway-based congestion control mechanisms have been recently proposed to support an end-to-end congestion control mechanism of TCP [1, 2]. One of promising gateway-based congestion control mechanisms is a RED (Random Early Detection) gateway that randomly drops an incoming packet at the buffer [1]. A number of studies on the steady state performance of the RED gateway using simulation experiments have been performed [1, 3, 4]. Although effectiveness of the RED gateway is fully dependent on a choice of control parameters, it is difficult to configure them appropriately. For example, the authors of [1] have proposed a set of control parameters for the RED gateway, but this is only a guideline acquired empirically using simulation experiments. On the other hand, there are a few studies analyzing the characteristics of the RED gateway. Stability and transient behavior of the RED gateway in the steady state have been analyzed in [5-8] by assuming that the number of TCP connections is constant. It has not been cleared how the variation of the number of TCP connections affects the transient behavior of the RED gateway. In an actual network, the number of TCP connections changes frequently. When the number of TCP connections is increased or decreased, either buffer overflow or buffer underflow may occur, resulting in the performance degradation of the RED gateway. It is therefore important to evaluate the transient behavior of the RED gateway by taking account of the variation of the number of TCP connections.

In this paper, we analyze the transient behavior of the RED gateway by extending the analytic results obtained in [5]. More specifically, we analyze the dynamics of the number of packets in the RED gateway's buffer (i.e., the queue length) when one or more TCP connections newly start or terminate their data transmissions. Showing numerical results, we reveal how control parameters of the RED gateway affect its transient behavior.

This paper is organized as follows. In Section 2, we explain the algorithm of the RED gateway in short. In Section 3, the analytic model of the RED gateway is explained, which is used throughout this paper. In Section 4, we briefly present the derivation of the average state transition equations, which describe the dynamics of the RED gateway. In Section 5, using the average state transition equations, we analyze the transient behavior of the RED gateway when the number of TCP connections is changed. In Section 6, several numerical examples are presented to clearly show how control parameters of the RED gateway or system parameters affect the transient performance. In Section 7, we finally conclude this paper and discuss future works.

2 RED Algorithm

The RED gateway has four control parameters: min_{th}, max_{th}, max_p, and q_w. min_{th} is the minimum threshold and max_{th} is the maximum threshold. These thresholds are used to calculate a packet marking probability for every incoming packet. The RED gateway maintains its average queue length $\overline{q}$, which is calculated from the current queue length using EWMA (Exponential Weighted Moving Average) with a weight factor of q_w. The RED gateway calculates the packet marking probability p_b for every incoming packet from the average queue length. Namely, the RED gateway determines the packet marking probability p_b using the function shown in Fig. 1. In this figure, max_p is a control parameter that determines the maximum packet marking probability. The packet dropping mechanism of the RED gateway is not per-flow basis, so the same packet marking probability p_b is used for all the incoming packets.

3 Analytic Model

In this paper, we analyze the transient behavior of the RED gateway using the analytic results obtained in [5]. We show our analytic model in Fig. 2. The analytic model consists of a single RED gateway and multiple TCP connections. We assume that all TCP connections have an identical (round-trip) propagation delay (denoted by τ). We also assume that the processing speed of the RED gateway (denoted by B) is the bottleneck in the network. Namely, transmission

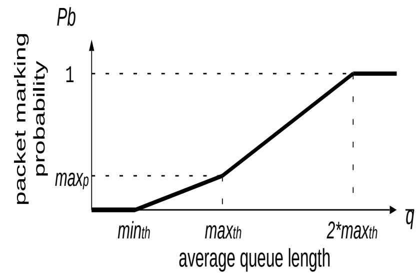

Fig. 1. Calculation of packet marking probability p_b.

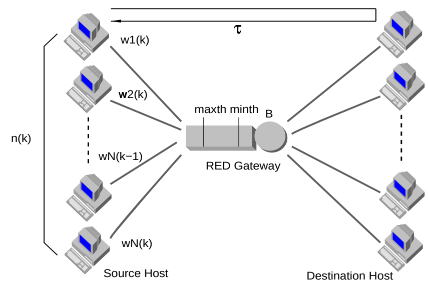

Fig. 2. Analytic model.

speeds of all links are assumed to be sufficiently faster than the processing speed of the RED gateway.

We model the congestion control mechanism of TCP version Reno [9] at all source hosts. We further assume that all TCP connections change their window sizes (denoted by w) synchronously. Source hosts are allowed to send w packets without receipt of an ACK (ACKnowledgement) packet. Thus, the source host can send w packets during its RTT (Round Trip Time). In our analysis, we model the entire network as a discrete-time system, where a time slot of the system corresponds to an RTT of TCP connections. We define $w(k)$ as the window size of the source host at slot k. All source hosts are assumed to have enough data to transmit; that is, the source host is assumed to always send the number $w(k)$ of packets during slot k. We define $q(k)$ and $\overline{q}(k)$ as the current and the average queue lengths (i.e., the current and the average number of packets in the buffer of the RED gateway). We assume that both $q(k)$ and $\overline{q}(k)$ will not change during a slot [5]. For taking account of a TCP connections variation, the number of TCP connections at slot k is denoted by $n(k)$.

4 Derivation of Average State Transition Equations

In this section, we present the derivation of average state transition equations, which describe the dynamics of the RED gateway [5]. Refer to [5] for the detail of the analysis.

4.1 Derivation of State Transition Equations

Provided that the average queue length $\overline{q}(k)$ lies between min_{th} and max_{th}, and that the number $n(k)$ of TCP connections is constant, $p_b(k)$ is given by

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

The RED gateway discards an incoming packet with a probability $p_a(k)$:

$$p_a(k) = \frac{p_b(k)}{1 - count \cdot p_b(k)}$$

where *count* is the number of unmarked packets that have arrived since the last marked packet. The number of unmarked packets between two consecutive marked packets, X, can be represented by an uniform random variable in $\{1, 2, \cdots, 1/p_b(k)\}$. Namely,

$$P_k[X = n] = \begin{cases} p_b(k) & 1 \le n \le 1/p_b(k) \\ 0 & \text{otherwise} \end{cases}$$

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

$$\overline{X}_k = \sum_{n=1}^{\infty} n \, P_k[X = n] = \frac{1/p_b(k) + 1}{2}$$

The probability that at least one packet is discarded from $w(k)$ packets, $\overline{p}$, is given by

$$\overline{p} = \min\left(\frac{w(k)}{1/p_b(k)}, 1\right)$$

Therefore, by assuming that all TCP connections are in the congestion avoidance phase, the window size at slot $k + 1$ is given by

$$w(k+1) = \begin{cases} \frac{w(k)}{2} & \text{with probability } \overline{p} \\ w(k) + 1 & \text{otherwise} \end{cases} \qquad (1)$$

Note that in the above equation, it is assumed that all packet losses can be detected by duplicate ACKs [5]. The current queue length at slot $k + 1$ is given by

$$\begin{aligned} q(k+1) &= q(k) + n(k+1)\, w(k+1) - B\left(\tau + \frac{q(k)}{B}\right) \\ &= n(k+1)\, w(k+1) - B\,\tau \qquad (2) \end{aligned}$$

The average queue length at slot $k + 1$ is given by

$$\overline{q}(k+1) = (1 - q_w)^{n(k)\, w(k)} \overline{q}(k) + \frac{q_w\{1 - (1 - q_w)^{n(k)\, w(k)}\}}{1 - (1 - q_w)} \, q(k)$$

4.2 Derivation of Average State Transition Equations

We derive average state transition equations that represent a typical behavior of TCP connections and the RED gateway [5]. We introduce a *sequence*, which

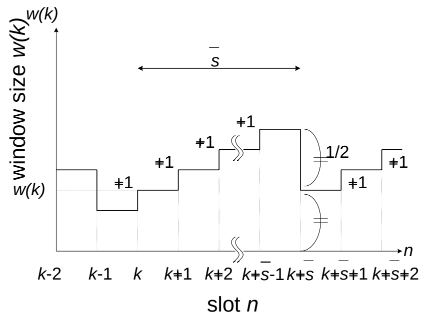

Fig. 3. Relationship between slot and sequence.

is a series of adjacent slots in which all packets from a source host have been unmarked by the RED gateway (Fig. 3). We then treat the entire network as a discrete-time system where a time slot corresponds to a sequence, instead of a slot. Let $\overline{s}(k)$ be the average number of slots that consists of a sequence that begins at slot k.

The average state transition equation from $w(k)$ to $w(k+\overline{s}(k))$ is obtained from Eq. (1) as

$$w(k+\overline{s}(k)) = \frac{w(k)+\overline{s}(k)-1}{2} \tag{3}$$

Note that $w(k)$ represents the expected value of the *minimum* window size. Similarly, the average state transition equations from $q(k)$ to $q(k+\overline{s}_k)$ is obtained from Eq. (2) as

$$q(k+\overline{s}(k)) = n(k+\overline{s}(k))\, w(k+\overline{s}(k)) - B\,\tau \tag{4}$$

The average state transition equation from $\overline{q}(k)$ to $\overline{q}(k+\overline{s}_k)$ is obtained as

$$\overline{q}(k+\overline{s}(k)) \simeq (1-q_w)^{\overline{X(k)}}\overline{q}(k) + \{1-(1-q_w)^{\overline{X(k)}}\}q(k) \tag{5}$$

Average state transition equations given by Eqs. (3), (4), and (5) describe the average behaviors of the window size, the current queue length, and the average queue length, respectively. An *average equilibrium value* is defined as the expected value in steady state. Let w^*, q^*, $\overline{q}^*$, and n^* be the average equilibrium values of the window size $w(k)$, the current queue length $q(k)$, the average queue length $\overline{q}(k)$, and the average number of TCP connections $n(k)$, respectively. Let us introduce $\delta\mathbf{x}(k)$ as the difference between the state vector $\mathbf{x}(k)$ and the average equilibrium point.

$$\delta\mathbf{x}(k) \equiv \begin{bmatrix} w(k)-w^* \\ q(k)-q^* \\ \overline{q}(k)-\overline{q}^* \\ n(k)-n^* \end{bmatrix}$$

By lineally approximating $w(k)$, $q(k)$, $\overline{q}(k)$, and $n(k)$ around their average equilibrium values, $\delta\mathbf{x}(k+\overline{s})$ can be written as

$$\delta\mathbf{x}(k+\overline{s}(k)) \simeq \mathbf{A}\delta\mathbf{x}(k) \tag{6}$$

where $\mathbf{A}$ is a state transition matrix.

5 Analysis of Transient Behavior

5.1 Types of TCP Connections Variation

We assume that N TCP connections exist in steady state. We also assume that all TCP connections are in the congestion avoidance phase. In this case, there are four types of changes in the number of TCP connections.

The first case is that ΔN ($\Delta N < N$) TCP connections of N TCP connections end their data transmissions (C1). In this case, $N - \Delta N$ TCP connections are in the congestion avoidance phase and will reach the steady state again. The second and the third cases (C2 and C3) are that ΔN TCP connections resume their data transmissions after an idle period. In these cases, the behavior of these ΔN TCP connections depends on the length of the idle period. When the idle period is short (C2), ΔN TCP connections operate in the congestion avoidance phase with using their previous window sizes. In this case, there exist totally $N + \Delta N$ TCP connections in the congestion avoidance phase.

On the other hand, when the idle period is long (in general, longer than the TCP's retransmission timer) (C3), ΔN TCP connections operate in the slow start phase with the initial window size. Moreover, the fourth case is that ΔN TCP connections newly start their data transmissions (C4). In this case, similar to the third case, there exist N TCP connections in the congestion avoidance phase and ΔN TCP connections in the slow start phase. In this paper, we analyze the transient behavior of the RED gateway in each case. We use two different approaches for the cases that all TCP connections are in the congestion avoidance phase (C1 and C2) and for the cases that some TCP connections are in the slow start phase (C3 and C4).

5.2 Cases C1 and C2: Congestion Avoidance Phase Only

We consider the cases that all TCP connections are in the congestion avoidance phase (C1 or C2). Let $u(k)$ ($\equiv n(k) - n(k-1)$) be the difference of the number of TCP connections from slot $k-1$ to slot k. For instance, when the number of TCP connections is increased by ΔN at slot i, $u(k)$ is given by

$$u(k) = \begin{cases} \Delta N & \text{if } k = i \\ 0 & \text{otherwise} \end{cases}$$

We analyze the effect of the TCP connections variation on the dynamics of the current queue length of the RED gateway. The main idea is to treat the TCP connections variation, $u(k)$, and the current queue length, $q(k)$, as the input to, and the output from the system formulated by Eq. (6), respectively. Namely, by adding both the input $u(k)$ and the output $q(k)$ to Eq. (6), we have

$$\delta\mathbf{x}(k+\overline{s}(k)) = \mathbf{A}\delta\mathbf{x}(k) + \mathbf{B}u(k)$$
$$q(k) = \mathbf{C}\delta\mathbf{x}(k)$$

where $\mathbf{B}$ and $\mathbf{C}$ are defined by the following equations.

$$\mathbf{B} = [\ 0\ 0\ 0\ 1\ ]^T$$
$$\mathbf{C} = [\ 0\ 1\ 0\ 0\ ]$$

Namely, the variation of the number of TCP connections, $u(k)$, is added to the number of active TCP connections, $n(k)$, by $\mathbf{B}$. And the current queue length of the RED gateway, $q(k)$, is extracted from the state vector by $\mathbf{C}$.

Using such a SISO (Single-Input Single-Output) model given by Eq. (7), the dynamics of the current queue length of the RED gateway can be precisely analyzed. For example, the evolution of the current queue length, $q(k)$, for a given TCP connections variation, $u(k)$, can be calculated by

$$q(k) = \sum_{i=0}^{k} u(i)\,\delta\mathbf{x}(k-i) \tag{7}$$

The great advantage of this approach is that various analytic techniques used in the control theory can be directly applied. For example, if the number of TCP connections is increased by ΔN at slot k, the input $u(k)$ becomes the impulse function [10]. Therefore, it is easy to analyze the dynamics of the current queue length, $q(k)$, by investigating the impulse response of the system. We can investigate the dynamics of the current queue length not only for an instantaneous TCP connections variation but also for an arbitrary TCP connections variation.

5.3 Cases C3 and C4: Congestion Avoidance Phase and Slow Start Phase

We next focus on the other two cases: when TCP connections resume their data transmissions (C3) and when TCP connections newly start their data transmissions (C4). Let $u'(k)$ be the difference in the total number of packets from slot $k-1$ to slot k, which are sent from all TCP connections in the slow start phase. More specifically, $u'(k)$ is defined by

$$u'(k) = \sum_{i=1}^{\Delta N} (w_i(k) - w_i(k-1))$$

where ΔN is the number of TCP connections operating in the slow start phase, and $w_i(k)$ is the window size of ith TCP connection. In the slow start phase, the window size is first initialized and then doubled every RTT. Thus, when

ΔN TCP connections newly start their data transmissions at slot i, $u'(k)$ is approximately given by

$$u'(k) \simeq \begin{cases} \frac{\Delta N}{n(k)} \times 2^{\overline{s}(k)(k-s_i-1)} & \text{if } k > s_i \\ 0 & \text{otherwise} \end{cases}$$

Similarly to the previous subsection, the dynamics of the current queue length of the RED gateway can be analyzed. Namely, the difference in the total number of packets, $u'(k)$, and the current queue length, $q(k)$, are added to the system given by Eq. (6) as the input and the output, respectively.

$$\begin{aligned} \delta\mathbf{x}(k+\overline{s}(k)) &= \mathbf{A}\delta\mathbf{x}(k) + \mathbf{B}'u'(k) \\ q(k) &= \mathbf{C}\delta\mathbf{x}(k) \end{aligned}$$

where

$$\mathbf{B}' = \begin{bmatrix} 1 \ 0 \ 0 \ 0 \end{bmatrix}^T$$

In the above equations, the window size of a TCP connection, $w(k)$, is increased by $u'(k)$ by $\mathbf{B}$.

6 Numerical Examples and Discussions

6.1 Performance Measures for Transient Behavior

Three performance measures called *overshoot*, *rise time* and *settling time* are widely used for evaluating the transient behavior of dynamic systems (Fig. 4) [11]. These are criteria for the damping performance (the overshoot), the response performance (the rise time), and both the response and the damping performance (the settling time). In this paper, we define the overshoot as the difference between the maximum and the equilibrium queue lengths. The rise time is defined as the time taken for the current queue length to reach the 90 % of the equilibrium queue length. The settling time is the time taken for the current queue length to converge within 5% of the equilibrium queue length. In general, all of these performance measures should be small for achieving better transient behavior. However, there is a tradeoff among the overshoot, the rise time, and the settling time. It is therefore important to balance these three performance measures according to the desired transient behavior.

These performance measures have the following implications to the RED gateway. A large overshoot means that the current queue length of the RED gateway grows excessively when the number of TCP connections is changed. Since the current queue length is limited by the buffer size, a large overshoot sometimes causes buffer overflow at the RED gateway. Otherwise, it results in a long queueing delay in the buffer. Hence, a small overshoot is desirable for preventing buffer overflow and minimizing the queueing delay. In addition, the

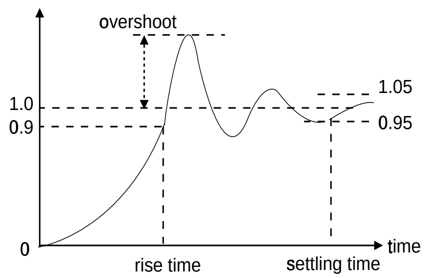

Fig. 4. Performance measures for transient behavior (overshoot, rise time, and settling time).

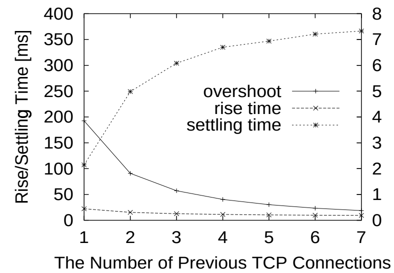

Fig. 5. Performance measures for transient behavior (the number of previous TCP connections $N = 1$–7).

rise time represents the convergence speed of the current queue length after a change of the number of TCP connections. As can be seen from Eq. (2), the current queue length, $q(k)$, directly reflects the window sizes $w(k)$. So it is possible to estimate the convergence speed of TCP connections from the rise time. The settling time implies the convergence speed of the current queue length to its equilibrium value after the number of TCP connections is changed.

6.2 Case C2: Congestion Avoidance Phase Only

Due to space limitation, we only show numerical examples for the case (C2): when ΔN TCP connections resume their data transmissions in the congestion avoidance phase after a short idle period. We use the equilibrium values, w^*, q^*, and $\overline{q}^*$, as the initial values for $w(k)$, $q(k)$, and $\overline{q}(k)$. We calculate the dynamics of the queue length $q(k)$ of the RED gateway using Eq. (7) when ΔN TCP connections resume at slot 0; i.e.,

$$n(k) = \begin{cases} N & \text{if } k < 0 \\ N + \Delta N & \text{if } k \geq 0 \end{cases}$$

Figure 5 shows performance measures for the transient behavior (the overshoot, the rise time, and the settling time) for different number N of TCP connections in steady state, n^*. In the following figures, unless explicitly stated, we use a set of control parameters of the RED gateway recommended by the authors of [1]. We also use the following system parameters: the processing speed of the RED gateway $B = 2$ [packet /ms], the propagation delay $\tau = 1$ [ms], and the number of resumed TCP connections $\Delta N = 1$. Figure 5 shows that the current queue length of the RED gateway changes more dynamically (i.e., a larger overshoot) when N is smaller. It is because when the number of TCP connections in steady state, N, is smaller, the impact of the resumed TCP connection becomes larger. The figure also shows that the overshoot is smaller than 1 [packet] when the number of TCP connections, N, is greater than 4. It suggests that the buffer

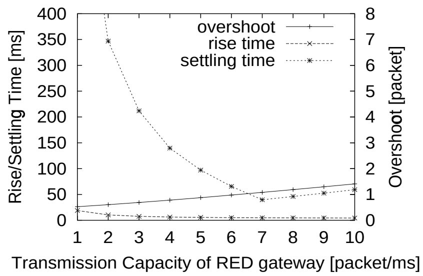

Fig. 6. Performance measures for transient behavior (the processing speed of the RED gateway $B =$ 1–10 [packet/ms])

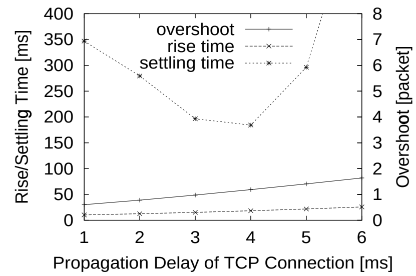

Fig. 7. Performance measures for transient behavior (the propagation delay of the TCP connection $\tau = $ 1–5 [ms])

overflow at the RED gateway is not likely to happen when the number of TCP connections is sufficiently large.

Shown in Fig. 6 is the case that the processing speed of the RED gateway, B, is changed from 1 to 10 [packet/ms]. One can find from this figure that as the processing speed of the RED gateway decreases, the overshoot and the settling time becomes small and long, respectively. This implies that the effect of TCP connections variation on the current queue length of the RED gateway sustains for a long period if the processing speed of the RED gateway is small.

Figure 7 illustrates the effect of the (round-trip) propagation delay of the TCP connection on the transient behavior of the RED gateway. In this figure, the propagation delay of the TCP connection, τ, is changed from 1 to 6 [ms]. This figure clearly shows that the transient behavior of the RED gateway is degraded when the propagation delay of the TCP connection is large. For example, as the propagation delay increases, both the overshoot and the rise time increase. This phenomenon can be understood by the fact that when both TCP connections and the RED gateway are considered as a single feedback system, a longer propagation delay corresponds to a longer feedback delay. In general, both the stability and the transient performance of a feedback system are degraded by a long feedback delay.

Figure 7 also shows that the settling time is minimized when the propagation delay of the TCP connection, τ, is about 4 [ms]. It can be conjectured from this phenomenon that the current queue length of the RED gateway will change slowly when the propagation delay of the TCP connection is short, and that the current queue length changes oscillatory when the propagation delay is long. From this observation, it is expected that the operation of the RED gateway becomes unstable if the propagation delay of the TCP connection is very long. In most feedback-based systems, a small feedback delay improves both the stability and the transient performance. However, in the congestion avoidance phase of TCP, the window size of the source host is increased every its RTT. In other

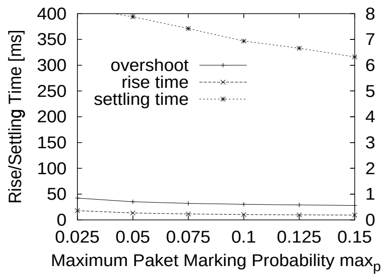

Fig. 8. Performance measures for transient behavior (the maximum packet marking probability max_p =0.025–0.15)

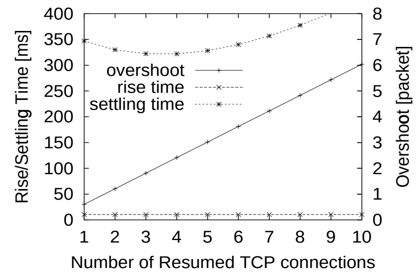

Fig. 9. Performance measures for transient behavior (the number of resumed TCP connections ΔN = 1–10)

words, the congestion avoidance phase of TCP has a feedback gain, which is dependent on the feedback delay.

We then investigate the effect of the maximum packet marking probability, max_p, on the transient behavior of the RED gateway. Figure 8 suggests that three performance measures — the overshoot, the rise time, and the settling time — are slightly increased as max_p increases. Namely, the maximum packet marking probability, max_p, has little impact on the transient behavior of the RED gateway. The maximum packet marking probability, max_p, should therefore be configured by taking account of the steady state performance of the RED gateway (e.g., the average throughput and the average queue length). Although we do not include results due to space limitation, we found that two threshold values, min_{th} and max_{th}, also have little impact on the transient behavior of the RED gateway.

We finally show the dynamics of the current queue length of the RED gateway for a different number of resumed TCP connections ΔN. In this figure, ΔN is changed from 1 to 10. It can be found from this figure that the current queue length of the RED gateway changes more excessively with a larger number of resumed TCP connections, ΔN. This phenomenon can be intuitively understood. Namely, when the number of resumed TCP connections is large, more packets arrive at the RED gateway. It gives a larger impact on the transient behavior of the RED gateway.

7 Conclusion and Future Work

In this paper, we have analyzed the impact of TCP connections variation on the transient behavior of the RED gateway by utilizing the average state transition equations obtained in [5]. We have modeled the entire network including both TCP connections and the RED gateway as a feedback system. We have investigated the transient behavior (in particular, the dynamics of the current queue length) of the RED gateway when the number of TCP connections is changed. We have quantitatively shown that the transient behavior of the RED gateway is sensitive to system parameters such as the number of TCP connections in the steady state, the capacity of the RED gateway, and the propagation delay of the TCP connection. We have also shown that the control parameters of the RED gateway have little influence on the transient behavior of the RED gateway.

As a future work, it is important to analyze the transient behavior of the RED gateway in various situations since our analytic approach enables us to investigate the transient behavior of the RED gateway for realistic TCP connection variation.

Acknowledgement

This work was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems" (JSPS-RFTF97R16301).

References

1. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.
2. B. Barden et al., "Recommendations on queue management and congestion avoidance in the Internet," *Request for Comments (RFC) 2309*, Apr. 1998.
3. D. Lin and R. Morris, "Dynamics of random early detection," in *Proceedings of ACM SIGCOMM '97*, pp. 127–137, Sep 1997.
4. M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED," in *Proceedings of IWQoS '99*, pp. 260–262, Mar. 1999.
5. H. Ohsaki, M. Murata, and H. Miyahara, "Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting," to appear in *IEICE Transactions on Communications*, Jan. 2002.
6. M. May, T. Bonald, and J.-C. Bolot, "Analytic evaluation of RED performance," in *Proceedings of IEEE INFOCOM 2000*, pp. 1415–1424, Mar. 2000.
7. V. Sharma, J. Virtamo, and P. Lassila, "Performance analysis of the random early detection algorithm," available at `http://keskus.tct.hut.fi/tutkimus/com2/publ/redanalysis.ps`, Sept. 1999.
8. H. M. Alazemi, A. Mokhtar, and M. Azizoglu, "Stochastic approach for modeling random early detection gateways in TCP/IP networks," in *Proceedings of IEEE International Conference on Communications 2001*, pp. 2385–2390, June 2001.
9. W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols.* New York: Addison-Wesley, 1994.
10. R. Isermann, *Digital control systems, Volume 1: fundamentails, deterministic control.* Springer-Verlag Berlin Heidelberg, 1989.
11. N. S. Nise, *Control Systems Engineering.* New York: John Wiley & Sons, third ed., 2000.